INTERFACING MICRO-CONTROLLER AND RFID TOWARDS A SECURE EMBEDDED CAR PARKING MODEL

JUSTINE OBWOGI OGUTA

A Thesis Submitted to the Institute of Postgraduate Studies of Kabarak University in Partial Fulfillment of the Requirements for the Award of Doctor of Philosophy in Information Technology

KABARAK UNIVERSITY

NOVEMBER, 2023

DECLARATION

1. I do by declare that:

   i. This thesis is my own work and to the best of my knowledge it has not been presented for the award of a degree in any university or college.

   ii. That the work has not incorporated material from other works or a paraphrase of such material without due and appropriate acknowledgement

   iii. That the work has been subjected to processes of anti-plagiarism and has met Kabarak University 15% similarity index threshold

2. I do understand that issues of academic integrity are paramount and therefore I may be suspended or expelled from the University or my degree may be recalled for academic dishonesty or any other related academic malpractices.

Signature:.. Date:.....................................

Justine Obwogi Oguta

GDS/M/0230/01/17

RECOMMENDATION

To the Institute of Postgraduate Studies and Research:

The research thesis entitled - **Interfacing Micro-Controller and Rfid towards a Secure Embedded Car Parking Model.** Written by **Justine Obwogi Oguta** is presented to the Institute of Postgraduate Studies of Kabarak University. We have reviewed the thesis and recommend it to be accepted in partial fulfillment of the requirement for the Doctor of Philosophy in Information Technology (Security and Audit).

Signature ……………………………. Date …………………………

Prof. Simon Maina Karume
Department of Computing and Informatics
Kabarak University

Signature ……………………………. Date …………………………

Dr. Peter Rugiri
Department of Computing and Informatics
Kabarak University

Signature ……………………………. Date …………………………

Dr. Nelson Masese
Department of Computer Science and IT
Kabarak University

COPYRIGHT

ACKNOWLEDGMENTS

I praise God for the gift of life, family, friends and provision without which I would not have made it this far in my academic pursuits. Indeed, the very ability to read and comprehend is a gift from God. I also express thanks to Kabarak University for giving me chance to pursue my Doctorate studies.

Many thanks to my supervisors Prof. Simon Maina Karume, Dr. Nelson Masese and Dr. Peter G. Rugiri for their guidance and patience throughout my studies. Your wise counsel and input during my research have made this possible and to all my friends who directly and indirectly supported me during my studies I say thank you.

DEDICATION

I dedicate this research project to God Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout my studies. I also dedicate this work to my family for they encouraged me all the way and for whom their support made sure that I gave my all to complete my studies.

ABSTRACT

This research focused on an integrated car parking solution that guides drivers on the exact parking lots at a specified period of time. In addition, it is a simplified monitoring and intelligence gathering of parking occupancy. It is based on the Radio frequency identification (RFID), micro controller and sensors which guide entry and outgoing of the vehicles. In this technique the RFID card is swiped, with the permission of vehicle’s parking owner. Using the Half Bridge concept, DC motors were used for the operation of entry and exit boom. The DC motors operate clock wise and antilock wise as per the program. When the vehicles enter the parking space, the space available in the parking system reduce and vice versa. It operated on a standard power supply of 5 volt and a LCD display showing all the activities. The microcontroller was interfaced to a car parking software with hashed database encryption and an RFID card authenticator recording incoming and outgoing vehicles in real-time. The use of cameras, RFID tags, readers and antennas made it easier to automate the 'in and out' privileges of parking subscribers.

Keywords: *Model, RFID, Micro-controller, Parking*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ABIPNGS	Agent Based Intelligent Parking Negotiation and Guidance Systems
AM	Amplitude Modulation
APGIS	Active Parking Guidance Information System
ART	Additional Reference (Carrier) Transmission
CCTV	Closed-Circuit Television
CMOS	Complementary Metal Oxide Semiconductor
COINS	Car-Park Occupancy Information System.
CPU	Central Processing Unit
DOD	Department of Defense
DSRC	Dedicated Short Range Communications
EAN	European Article Number
EAS	Electronic Article Surveillance
EEPROM	Electrically Erasable Programmable Read-Only Memory
EPC	Electronic Product Code
EPC	Engineering, Procurement, and Construction
FLC	Fuzzy Logic Controller.
FPGA	Field-Programmable Gate Array
GPS	Global Positioning Systems
GRAI	Global Returnable Asset Identifier
HF	High Frequency
HKIA	Hong Kong International Airport
I2V	Infrastructure-To-Vehicle
IC	Integrated Circuit
ID	Identification/Identity/Identifier
IE	Interrupt Enable
IoT	Internet of Things
IP	Interrupt Priority
ISM	Industrial, Scientific and Medical
LF	Low Frequency
M2M	Machine-To-Machine
MCS	Missile Control System

NAPA	National Automotive Parts Association
NMOS	Negative Metal Oxide Semiconductor
PARC	Parking Access and Revenue Control
PCB	Printed Circuit Board
PCON	Power Control
PIS	Parking Information Service Center
PMS	Parking management system
PWM	Pulse Width Modulation
RAM	Read Only Memory
RFID	Radio Frequency Identification
RF	Radio Frequency
ROM	Random Access Memory
RTU	Remote Terminal Unit
SCM	Source Code Management
SMS	Short Message Services
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
UCC	Uniform Code Council.
UHF	Ultra High Frequency
UPC	Universal Product Code
VANET	Universal Serial Bus
VIP	Very Important People
WMS	Warehouse Management System
EPM	Electronic Parking Management
USB	Universal Serial Bus

CHAPTER ONE

INTRODUCTION

1.1 Overview

This Section exposes the background of the concepts and problems to be addressed like causes of inefficiencies in car parking centers and efforts in form of technologies for car parking. It further proceeds to make the statement of the research problem, it also outlines the research objectives, lists the research questions, and defines the scope, assumptions, significance as well as the expected outcomes of the study.

1.2 Background of the Study

Automatic identification is the broad term given to a host of technologies that are used to help machines identify objects. Auto identification is often coupled with automatic data capture (koh *et al*, 2003). That is, companies want to identify items, capture information about them and somehow get the data into a computer without having employees type it in. The aim of most auto-ID systems is to increase efficiency, reduce data ingress errors, and free up staff to perform more value- integrated functions. There are a host of technologies that fall under the auto-ID umbrella. These include bar codes, astute cards, voice apperception, some biometric technologies (retinal scans, for instance), optical character apperception, radio frequency identification (RFID) is a means of identifying an item based on radio frequency transmission (Zumsteg and Qu., 2018). This technology can be utilized to identify, track and detect a wide variety of objects. Communication takes place between a reader and a transponder (derived from Transmitter/responder-Silicon Chip connected to an antenna), customarily called―tag‖. Tags come in many forms, such as perspicacious labels that are stuck on boxes, keenly intellective cards and a box that you stick on your windshield to enable you to pay tolls without ceasing. Tags can either be passive (powered by the reader field), semi-passive or active (powered by battery) (Somervell, et al., 2019). Active RFID tags are powered by an onboard powering source and incline to be more extravagant than passive tags that harvest power from the Radio Frequency Identification energy of the reader. On board power sanctions the active tags to have more preponderant communication distance and more expeditious replication time. These tags are more multifarious and customarily have more sizably voluminous recollection capacity. Passive RFID tags have no internal power source and use external power to operate. These tags are powered by the electromagnetic signal received from a reader. The received electromagnetic signal charges an internal capacitor on the tags, which in turn, acts as a puissance source and supplies the potency to the chip (Geist and Borras. 2018).

RFID systems differentiation criteria depend on operating reader frequency, physical coupling method and communication distance (read range) (Mbacke, Mitton and Rivano, 2018). The communication frequency used ranges from 135 KHz long wave to 5.8 GHz in the microwave range and are classified into four basic Ranges: LF (low frequency, 30 - 300 kHz), HF (high frequency, 3 - 30 MHz), UHF (ultra high frequency, 300 MHz – 2 GHz) and Microwave (> 2 GHz). The physical coupling uses magnetic and electromagnetic fields. The communication distance varies from few millimeters to above 35 meters (close coupling: 0 - 1 cm, remote coupling: 0 - 1 m, long range: > 1 m) (Pichorim, Gomes and Batchelor. 2018).

In the typical configuration, tags are placed on the objects to be identified (Li et al., 2018). Each tag is provided with an internal memory, which is partially ―read-only and, optionally, rewritable, where the information (unique ID serial number, manufacture date and product composition) about the object is stored. When these tags pass through the field generated by a reader, they transmit this information back to the reader, thus allowing the object identification (Dorsey et al., 2018). The communication process between the reader and tag is managed and controlled by one of several protocols, such as the ISO 15693 and ISO 18000-3 for HF or the ISO 18000-6, and EPC for UHF. The identification process begins when the reader is switched on; it starts emitting a signal at the selected frequency band (typically 860 – 915 MHz for UHF or 13.56 MHz for HF); the tags reached by the reader's field will ―wake up (supplied by the field itself, if passive). Once the Tag has decoded the signal, it replies to the reader, by modulating the reader field (backscattering modulation). If many tags are present, then they will all reply at the same time. If this occurs, the reader detects a signal collision and an indication of multiple tags (Sundstrom & Iyer, 2018). In this case the reader uses an anti-collision algorithm designed to allow tags to be sorted and individually selected. The number of tags that can be identified depends on the frequency and protocol used, and typically ranges from 50 tags/s to 200 tags/s. Once one tag is selected, the reader can perform all the allowed operations such as read the tags identifier number, or also write data in it (in case of a read/write tag). After operations on the first tag are finished, the reader starts processing the second one and so on until the last one.

In order to receive a signal and communicate with a reader, passive tags use either one of the two following methods. These are near field, which employs inductive coupling of the tag to the magnetic field circulating around the reader antenna (like a transformer), and far field, which uses similar techniques to radar (backscatter reflection) by coupling with the electric field (Pooley, 2019). The near field is generally used by RFID systems operating in the LF and HF frequency bands, and the far field for longer read range UHF and microwave RFID systems (Duroc and Tedjini, 2018).

The reason is that in the near field, the field energy decreases, as a first approximation, proportionally to 1/R3 (where R is the distance from the antenna), while in the far field the energy decreases proportionally to 1/R; the borderline between near and far field is; $R = \lambda/\pi$ as result in the far field the energy of lower frequencies waves will turn out to be much more reduced than that of higher ones, whose use is thus mandatory in that zone. Passive technology is most widely used for RFID applications. Passive technology operates in a range of frequency bands, of which 860 – 956 MHz (ISM) band is most popular. Passive tags operating at UHF communicate with the reader through Amplitude Modulation (AM), and receive their power from the reader field, with energy transfer based on the far field properties. Communication from tag to reader is achieved by altering the antenna input impedance in time with the data stream to be transmitted: in this way the power reflected back to the reader is 3modulated in time with the data (VanBladel, 2002).

The use of far field backscatter modulation introduces problems that are not present in HF and lower frequency systems (Lopez and Conley, 2019). One of the most consequential of such undesired effects is due to the fact that the field emitted by the reader is not only reflected by the tag antenna, but additionally by any objects with dimensions in the order of the wavelength used: these reflected fields could damp the reader‘s and the back scattered field thus reducing the systems efficiency;

It is better to utilize more than just one antenna per reader. ISO defines the Air interface communication between Reader->Tag and Tag->Reader, and includes parameters like Communication protocol, Signal Modulation types, Data coding and frames, Data Transmission rates and Anti-collision (detection and sorting of many tags in the Reader field at the same time) (Ihlenburg, Steigerwald and Schöppner., 2019).

From the various development and considerations for RFID integration with microcontroller embedded systems they have not specifically considered an inclusive model that is interfaced with various components. Therefore, my study will seek to be a harmonizing technology that will integrate all the desperate solutions that have not individually provided a secure, more focused and a reliable solution that is efficient, effective, reliable, scalable, and accurate for real time data.

1.3 Statement of the Problem

The lack of a model for automatically scheduling cars in the parking centers by automatically auditing available empty spaces in the parking and ushering in the next car while synchronously taking real time images and video for forensic audit and tracking of criminal activities has proved difficult to control parking space. There is continued use of parking guides who have to be available at all times to direct cars while moving around to check empty lots. The auditing of cars going through the gates is inefficient as it lacks supporting real time evidence in form of media captor report. Therefore, the recording system in place is cumbersome and inefficient more often lack up to date information in case a person engages in a criminal activity. In addition, there has not been a mechanism in form of a secure model based on RFID integrating embedded system for car parking. The development of a parking model with real-time audit and forensic tracking solution aims to address the challenges faced in managing parking spaces effectively. By integrating these technologies, the model can provide accurate and up-to-date information about the status of each parking space in real-time. This real-time audit and tracking solution eliminates the need for manual checks by parking guides, as the system can automatically detect and monitor empty parking spaces. Furthermore, the model incorporates image-processing techniques to capture real-time images and videos for forensic audit and tracking of criminal activities. The implementation of this model will greatly improve the efficiency and effectiveness of parking management

1.4 Objective of the Study

The main objective of the study is to develop a model that would aid in automatically scheduling cars in parking centers while integrating media streaming at real time. The design of the said model is guided by the following specific objectives;

1.4.1 Specific Objectives of the Study

i. To investigate security components in existing technological solutions for managing public parking lots

i. To design an architectural model based on RFID microcontroller integrating real time media auto-streaming for vehicle packing.

ii. To develop the prototype of an RFID microcontroller based real time media auto-streaming model for vehicle parking.

iii. To validate and verify the RFID microcontroller based real time media auto-streaming model for vehicle parking.

1.5 Research Questions

The research seeks to give answers to the following questions: -

i. What are the security components in existing technological solution for managing public parking lots?

ii. How can an architectural model based on RFID microcontroller integrating real time media auto-streaming for vehicle packing be designed?

iii. How can a prototype of a RFID microcontroller based real time media auto-streaming model for vehicle parking be developed?

iv. How can a model of RFID microcontroller on real time media auto-streaming model for vehicle parking be validated and verified?

1.6 Significance of the Study

The use of an RFID microcontroller-based technology integrated for real-time data captor in a car parking center has not been done before. Municpalities, town dwellers, mall management and universities with large parking lots will greatly benefit from an efficient solution relying on RFID integration technology. This will be a significant addition to the body of knowledge in the area of minimizing inefficiencies through proper guiding on empty slots and information auditing in car parking centers. Such an embedded model will be instrumental in the development of affordable and easy to use RFID based car entrance and parking control with microcontroller integration which will have the potential to revolutionize the way many stakeholders, and especially the law enforcement, conduct their activities.

A lot of effort by stakeholders and partners has done little in controlling vehicular scheduling in parking centers for efficiency and forensic audit of cars in developing countries. In Kenya alone where there’s upsurge in the number of cars with the upward trajectory in the increase of population it becomes difficult to access social places and malls because of lack guidance in parking lots. Also, it has been difficult to identify parking lot owners in case of a criminal ensue posing a risk to a large populace of people and property in a certain specific area which warrants an urgency for data captor at real time to facilitate forensic audit.

1.7 Research Contributions

The output of this research is to come up with the following deliverables;

i. A report on causes of inefficiencies in vehicle parking centers and also technological efforts that have been put in place.

ii. An architectural model using RFID Microcontroller integration for real time embedded model of vehicular data in a parking center.

iii. An Integrated RFID architecture model with the RFID hardware based assembled microcontroller.

iv. A prototype implementation of an RFID based real time media auto-streaming parking system.

Therefore, a feasible novel idea would be the only remedy to the situation at hand and in this case to reduce packing inefficiencies and security pilferage in specific places then coming with an RFID microcontroller based real time embedded model is the only remedy.

1.8 Scope of the Study

The study came up with a secure RFID microcontroller based integrated model for car Parking. This study research does not overlap already existing remedies towards controlling entry to vehicle parking centers. A lot of effort has been implemented so far in many organizations with the vision towards parking efficiency in places strategies that are often theoretical in nature as opposed to this technological RFID microcontroller embedded Model. The study however did not concentrate on concerns of payments towards entry to a particular car parking center nor logical security implementation rather it centered on the development of the prototype of an RFID Based Embedded Microcontroller to improve efficiency through scheduling, finding empty spaces and forensic audit within parking centers using Cameras.

1.9 Limitations of the Study

Despite attaining an RFID microcontroller based integrated model for car parking in this study. Their existed limitations.

i. Test Organization: Organizations to test the model by implementing within their gate facility was a problem. They felt that the system could not work in tandem with their existing system and therefore interfere with their existing infrastructure. Because of the challenge I had to undertake an experimental study and therefore csame up with a mimick of a gate system and realized on a chip board with RFID cards to be swipped before the dummy cars can go through the gates.

ii. Model test Data: The test data to be used for purposes of validation and verification of this study became a challenge owing to the fact that it was an experiment. To be able to get test data had to capture in and out movement of cars using a packing management management system PMS.

iii. Achieving data security: To be able to realize the security of the user car details captured. The data relayed was first hashed before securely storing it in the database.

1.10 Assumption of the Study

This study assumption is based on the understanding that technologies for micro controller embedded car parking can improve efficiency and mitigate problems such as congestion and limited parking availability. Furthermore, it is assumed that these technological solutions, such as smart devices and surveillance cameras, can assist drivers in finding available parking spots and locating their parked vehicles more efficiently this assumption is further supported by previous studies that have shown the effectiveness of smart car parking solutions in improving efficiency and reducing the time spent looking for parking.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In this chapter we discuss, Security components in existing technological solution for managing public parking lots, existing Smart Parking Systems and their technologies, design components required for RFID 8087 Microcontroller, research gaps, thesis framework and data flow logic.

2.2 Security Components in Existing Technological Solution for Managing Public Parking Lots

The use of Microcontrollers has been in existence with the rise in chip technology. The technologies have been utilized in different devices to offer solutions especially in car parking. The next section discusses various technologies that use microcontroller for car parking.

2.2.1 Radio Frequency Identification (RFID)

As mainstream media in the last few years has increasingly begun to cover radio frequency identification (RFID) technology, it may seem as if this technology was invented recently. This is definitely not the case, on the contrary: one would be surprised to learn that it has been around for more than 60 years.

According to Landt (Malagi, 2019) RFID was actually invented in 1948; it is assumed that one of the first to explore RFID was Harry Stockman in 1948, who wrote the paper entitled: Communication by Means of Reflected Power. Furthermore, Landt (Zhang et al., 2018) states that the first actual use of RFID related technology is considered to be during World War II, when the British used radar to distinguish allied planes from enemy planes, so-called: identification, friend or foe (IFF). He further mentions that electronic article surveillance (EAS) is the―first and most widespread commercial use of RFID‖ which was developed by several companies in the 1960s to stop theft. Only a few decades ago, to be precise in 1983, Charles Walton received a patent that contained, for the very first time, the acronym RFID. Obviously, the technology as it is available today could only be possible due to the contributions of many different scientist, companies and governments, for more detailed information about the history of RFID we refer to Landt (Smith *et al.*, 2019).

Even though RFID has been in use for a while now, only recently it is getting more and more attention in different applications. The reasons for many companies not to implement or consider RFID technology earlier, was because of the high costs of RFID hardware, software and services, the immaturity of the technology and the lack of common standards; as RFID components costs continue to decrease and the technology matures, it is assumed that more companies will become interested in RFID technology.

Among the leading players in utilization of RFID include the Wal-Mart (Jones, 2018) and the United States Department of Defence (DOD) also have contributed to an increased interest in RFID. Both are considered by many to be the biggest pushers of RFID technology. The main reasons being that Wal-Mart asked their top 100 suppliers to tag their cases and pallets starting from January 2005 (Lele, 2019) and all their suppliers by the end of 2006, and the DOD wanted all their suppliers to put RFID tags on all their shipments (Lele, 2019).

Bar code technology is similar in theory to RFID. With RFID, the electromagnetic or electrostatic coupling in the Radio Frequecy (RF) portion of the electromagnetic spectrum is used to transmit signals. RFID is a technology that belongs to the Automatic Identification (AUTO-ID) technologies which as stated by also includes (amongst others): Bar Code, Optical Character Recognition and Magnetic Stripe.

With AUTO-ID technologies it is not necessary for humans to both read data and enter it manuallyinto a computer system, because this all happens automatically and thus data entry is done efficiently and errors are minimized in a RFID system an object or person can be assigned a unique serial number (for instance, the identity)and this number is sent out wirelessly by means of radio waves (Kuang and Xu, 2018).

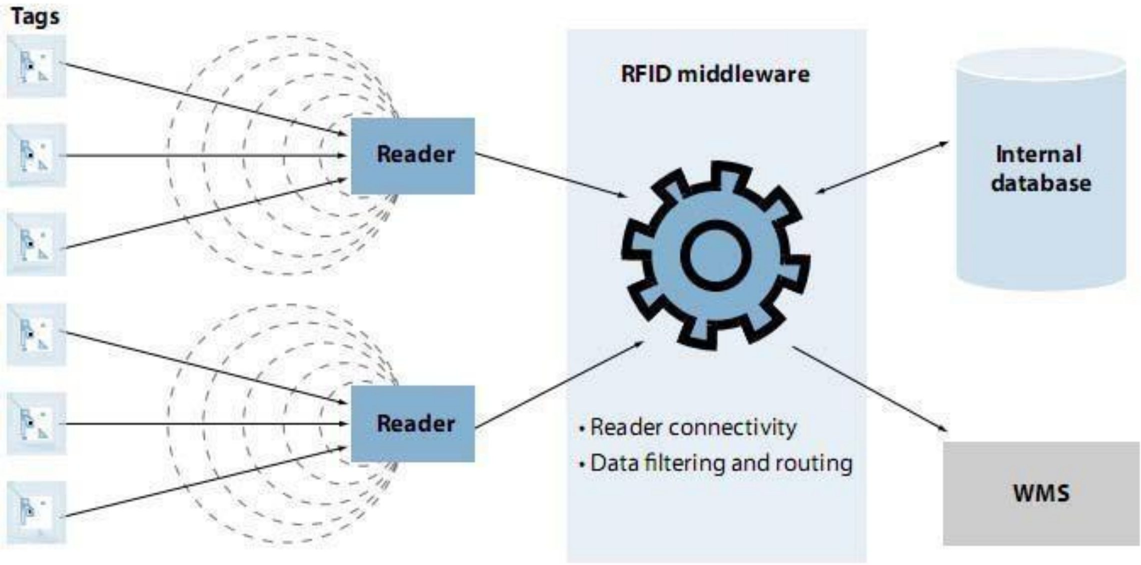
i. Components of RFID and How it Works

An RFID system is comprised of several components, they are: tag, reader, antenna, and middleware and enterprise applications. The working of RFID is described in the next paragraph.

For example, a pallet is transported from location A (the pool manager) to B (the retailer). The pool manager can identify this particular pallet as follows: a reader will send out radio frequency signals via its antenna and then wait for a response from the tag. When the tag is in the neighborhood of this reader it is activated (for example this only applies to passive tags (for example a type of tag) and then sends back its data (via its own antenna) which is collected by the same antenna/reader that send out those signals. This data is then transported to software called middleware which filters data, and then (usually) sends it to an enterprise application (for example Warehouse Management System (WMS) or a database. This basically describes how RFID technology works. This process is summarized in Figure 1. As shown in below, the RFID system is made up of different components and each serves a specific purpose in the system, hence they are only useful when used collectively.

Figure 1

General RFID system overview



Source: Leaver et al. 2004, p. 10

The following section describes the various components of RID

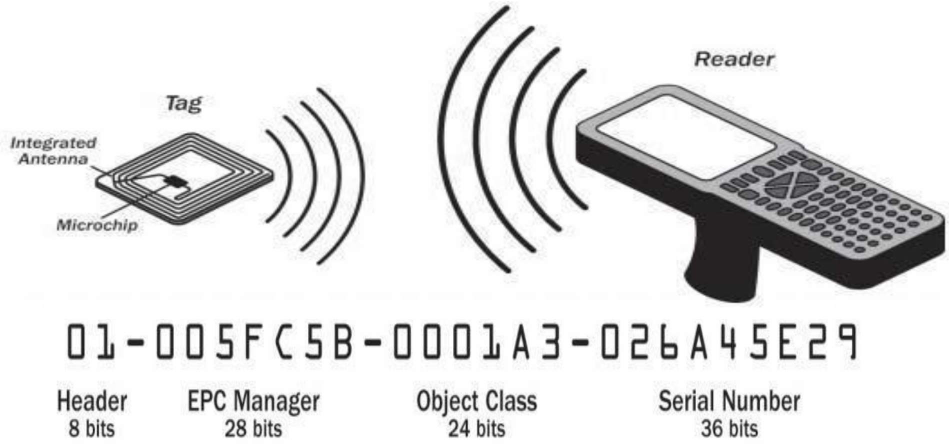
i. Tag

The tag (or transponder) could either be attached to a physical object for example a pallet or with screws or an adhesive) or carried by a person (i.e. embedded in a card.

Or animal (i.e. implanted in body) and it contains data that uniquely identifies or provides information about the carrier. This identification is possible due to the Electronic Product Code (EPC) that is stored on a microchip which can be found on every tag, the integrated antenna is used to send out the tag's data (see figure 2.2) (Trangkanukulkij, 2018).

Figure 2

Wireless communication between tag and reader



Source: Karygiannis et al. 2007, p. 21

The EPC is made up of four parts (see figure 2.2), they are: Header, EPC Manager, Object Class and Serial Number; each part is used for identification (Karygiannis et al. 2007). Specifically, for returnable assets a Global Returnable Asset Identifier (GRAI) was developed that enables unique identification of an asset all over the world and it can be embedded into EPC (Aftab et al., 2019).

There are three different types of RFID tags: passive, semi-passive and active tags. The difference between them is the source from which they get their power and how communication takes place. The passive tag is unable to send data on its own and needs radio waves from the reader's antenna as energy in order to communicate. This dependency means that it has a short-read range for example the distance from which a tag and reader can communicate with each other (RFID Journal glossary of terms). On the other hand, the active tag has a battery which is used for communication and is able to send data independently from a further distance. The semi-passive tag has a battery, but still needs radio waves from the reader's antenna to operate (Fahmy *et al.*, 2019).

There are various frequencies on which a RFID system can operate, namely: low frequency (LF), high frequency (HF), ultrahigh frequency (UHF) and microwave. The frequency determines the read range (i.e. the maximum distance between tag and reader) and the data transfer rate (i.e. the number of characters that are transmitted from tag to reader in a certain time (RFID Journal glossary of terms). This means that one frequency is better suited for certain applications (RFID Journal the basics of RFID technology) as is shown in table 2.1 (*Sun et al.*, 2018).

Table 1

Overview of Different Frequencies

Band	Estimated Read Range	Common Applications
Low Frequency (LF)	less than 0,5 meter	animal ID, beer kegs, auto key&lock, library books
High Frequency (HF)	up to 1,5 meters	item level tracking, airline baggage, building access
Ultra HighFrequency (UHF)	up to 100 meters or 0,5 to 5meters (depending on frequency band)	case, pallet and container tracking, truck and trailer tracking
Microwave	up to 10 meters	access control (vehicles)

Source: Ward & Van Kranenburg, 2006)

i. Reader and antenna

Both the reader and antenna will be used to communicate with tags and to capture data. The function of the antenna, which can be placed at diverse locations for example on the ceiling, near the door, in the ground, is to send out radio frequency signals and to receive data from the tag. However, actual communication with a tag is done by the reader (Weston, 2018).

There are a variety of readers available, namely: agile, multi-frequency, dumb and intelligent. The main difference between them is what they are capable of: the agile, multi-frequency and intelligent readers are able to perform more operations than a dumb reader (RFID Journal RFID system components). A reader can be placed either into the mobile or fixed category. The mobile reader can be brought along by an individual with a held reader or attached to a vehicle to read tags in an environment where fixed readers are not present. In contrast, the fixed reader is attached to fixed structures for instance, wall) so to read tags passing it (RFID Journal glossary of terms) (Sadr, 2018).

ii. Middleware

Before the captured data can be made available to the end user, it needs to undergo a quality check. The component that will perform this check is called middleware. It is software that operates between readers and enterprise applications. The function of middleware is to filter raw data received from readers, because this data is generally polluted (for example multiple reads). Once the filtering is complete, the clean data will be transported to enterprise applications. Besides filtering, some middleware can perform extra tasks, for example, automatically send and receive shipping information between business partners (Rehman, 2019).

iii. Enterprise applications

Filtered RFID data on its own has no value at all, because it is merely an amount of gathered data that has not yet been subject to analysis by the end user. Though, to actually unleash the value of this data, enterprise applications like Warehouse Management System (WMS) software are required which can help the end user to analyze RFID data by providing the necessary analytical tools. This data, after thorough analysis, will then evolve into valuable information which will form the basis for decisions and relevant actions (Mahdavinejad et al., 2018).

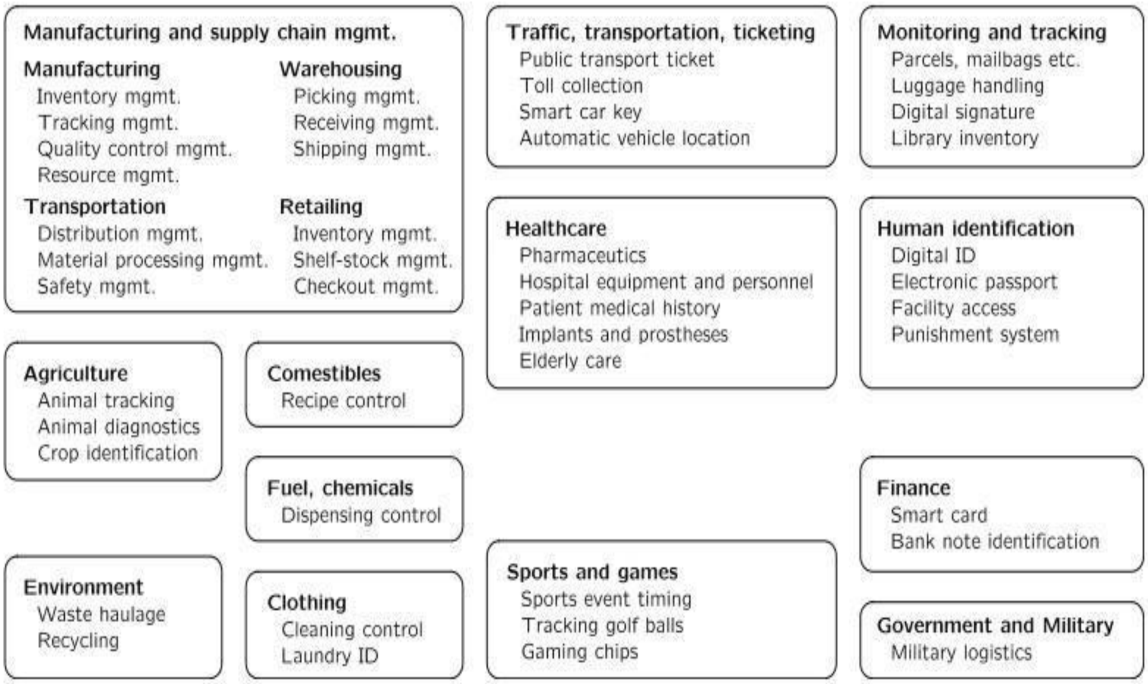
2.2.1.1 RFID Application Areas

Currently there is much interest in RFID technology from many practitioners. While some rather wait-and-see how this technology further develops, others in different sectors have already opted for implementation. As is apparent from the overview below (figure 2.3), use of RFID is not limited to a particular business, but it already is, or soon will be, applied in various sectors for different applications. Only a few applications like luggage handling, hospital equipment, public transport ticket and passport) will be highlighted to give an idea of this diversity by giving examples from practice.

Luggage handling (Swedberg, 2009): at Hong Kong International Airport (HKIA) RFID tags are used for all 40.000 pieces of luggage that are handled daily at the airport. HKIA has automated the transportation of luggage by having readers at strategic locations read the tags and route luggage from start (check-in) to finish (designated plane). Medical equipment (Swedberg, 2010): Health First located in Florida USA is using RFID to track all its infusion pumps and other medical equipment. Besides tracking, italso uses RFID to check the temperatures of coolers at their facilities to see if the infusion pumps are active or not and to determine if the pumps are properly cleaned and repaired.

Figure 2

Examples of current and expected RFID applications



Source: Ilie- Zudor et al. 2006

Public Transport: to pay (without coins or bank notes) for all public transport (i.e. metro, bus, tram and train) in the Netherlands, people can either purchase an anonymous, personal or disposable OV-chipkaart. As of 16th March 2011, the Ov-chipkaart is accepted everywhere (ANP, 2011) in Holland. The personal OV-chipkaart contains the following data: balance, automatic upgrade of balance (optional), last 10 transactions and birthdate of cardholder (ov-chipkaart.nl, 2012). The anonymous card lacks the birthdate as well as the automatic upgrade of balance (ov-chipkaart.nl, 2012). The disposable card strictly contains a preset travel time. Government: Dutch citizens, as of August 2006 (paspoortinformatie.nl, 2009), receive a passport that has a RFID chip which contains: personal data (e.g. name, date of birth, gender etc.) a full colour photo and, as of 21 September 2009, it also includes two fingerprints (De Jonge, 2010) RFID security issues.

2.2.1.2 RFID Security Mechanism

i. **RFID Security**

A Physical Unclonable Function (PUF) is based on the physical system and easy to evaluate (using the physical system). Its output looks like random function unpredictable for an attacker with physical access due to process variations no two identical circuits are the same. Identical circuits with the same layouts placed on different Field Programmable Gate Arrays (FPGA) shows that path delays will vary enough across ICs to use them for identification (Baker, 2019).

ii. **PUFs and Security**

PUF Security is based on wire delays, gate delays, and quantum mechanical fluctuations. PUF characteristics include uniqueness, reliability and unpredictability. PUF assumptions include the ability to infeasible to accurately model PUF. Physical tampering will modify PUF and reduce security since removing the PUF will lead to destruction. Attacker cannot tamper with the communication between the PUF and RFID Tag. Furthermore PUFs provide a secure, robust, low cost mechanism to authenticate chips (Liang, 2019).

2.2.1.3 RFID Frequency Types

There are three main types of RFID systems: low frequency (LF), high frequency (HF) and ultra-high frequency (UHF). Microwave RFID is also available. Frequencies vary greatly by country and region. Low-frequency RFID systems range from 30 KHz to 300 KHz, though the typical frequency is 125 KHz. LF RFID has short transmission ranges, generally anywhere from a few inches to less than six feet (Qureshi, 2018). High-frequency RFID systems range from 3 MHz to 30 MHz, with the typical HF frequency being 13.56 MHz. The standard range is anywhere from a few inches to several feet. UHF RFID systems range from 300 MHz to 960 MHz, with the typical frequency of 433 MHz and can generally be read from 25-plus feet away. Microwave RFID systems run at 2.45 GHzand can be read from more than 30-plus feet away (Doan, 2018)

The frequency used will depend on the RFID application, with actual obtained distances sometimes varying considerably from what might be expected. For example, when the U.S. State Department announced it was to issue electronic passports enabled with an RFID chip, it said the chips would only be able to be read from approximately four inches away (Rhoads, *et al*, 2018). However, evidence was later presented to the State Department indicating that RFID readers could skim information from the RFID tags from much farther than 4 inches, some claiming upward of 33 feet away, proving the difference between advertised and actual range can vary immensely (Musey & Keener, 2018).

2.2.1.4 RFID Reader System Architecture

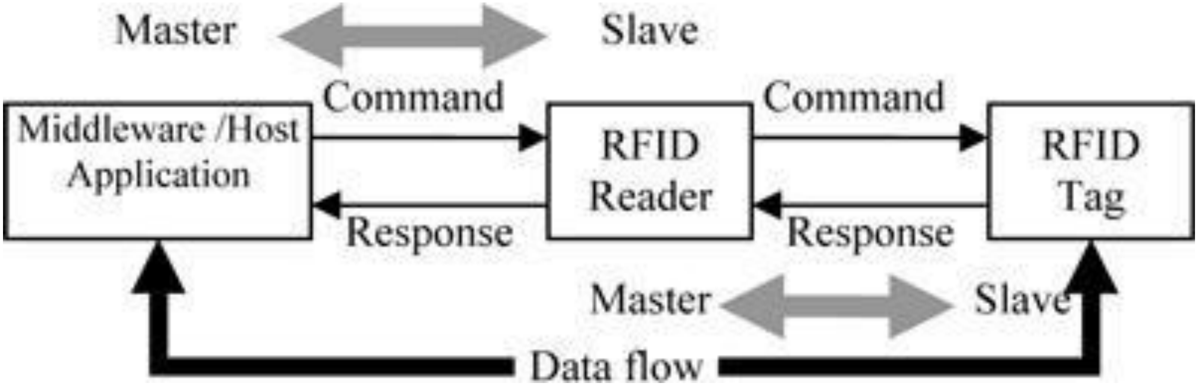
RFID readers are devices that perform the interrogation of RFID transponders. In a passive RFID system, the RFID reader supplies the tag with essential power in order for it to perform modulation of the reader‘s interrogation signal. Therefore, the reader and transponders are in a master-slave relationship where the reader acts as a master and the transponders as slaves. Nevertheless, RFID readers themselves are in a slave position as well. A software application, also called middleware, processes data from the RFID reader, acts as the master unit and sends commands to the reader (Martínez-Sala, 2009).

This means that all activities of the reader and transponder are initiated by the application software. In a hierarchical system structure the application software represents the master while the reader is the slave, which is only activated when it receives a command from the application software, and therefore, the reader perform read or write operations of RFID transponders that are in its interrogation area.

The following Figure 4 presents how data flows in the RFID system architecture (Finkenzeller, 2010).

Figure 3

RFID Reader Systems

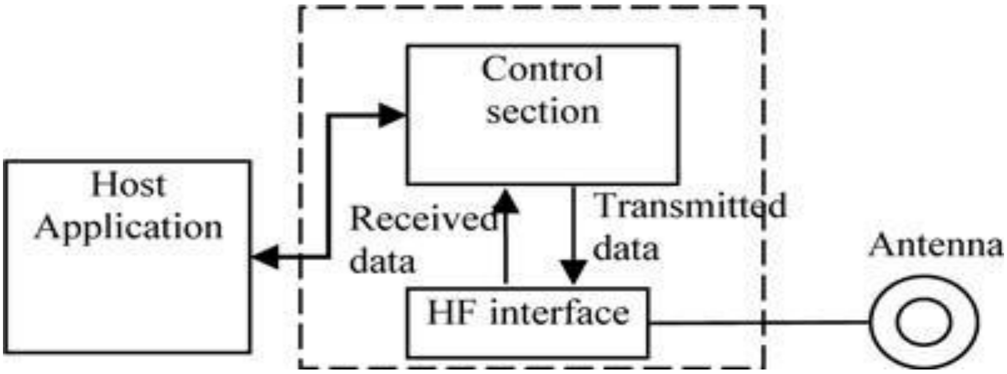


Source: Jones et al, 2018

RFID systems can be classified as passive and active tag systems. Powered by batteries active tags have on-board signal amplification and modulation facilities. Passive tags are battery less and rely on the reader‘s radiated energy to do some minor tasks of modulation and transmission of data. Therefore, active tags do not rely on the reader as much as passive tags do. As a result, active tags enjoy much superior performances in interrogation and data transmission due to their independence of power supply on a reader. In a passive tag system, a reader needs to have a powerful radiation field so that the passive RFID transponder can effectively cull sufficient power to perform some minor signal processing functions (Recouly et al. 2019) at the user end, the reader is connected to the host application such as enterprise software.

Figure 4

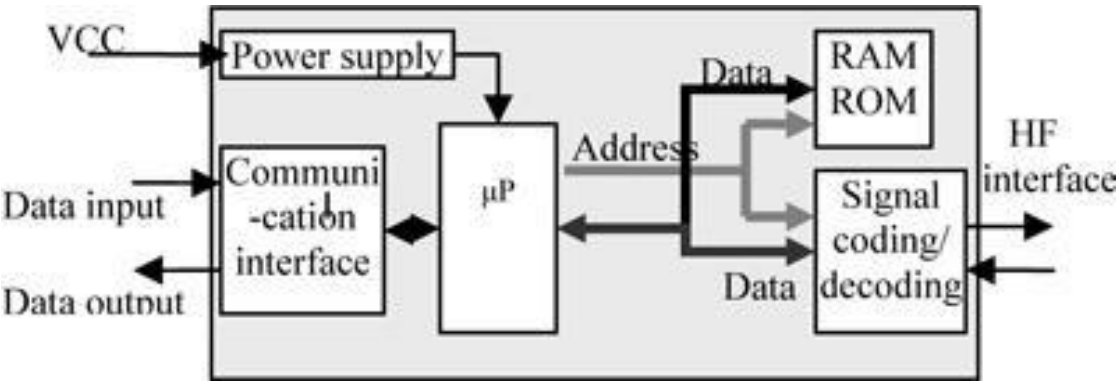
Parts of the RFID reader (Ung et al, 2019)



The control section of the RFID reader performs digital signal processing and procedures over the received data from the RFID transponder. Also, the control section enables the reader to communicate with the transponders wirelessly by performing modulation, anti collision procedures and decoding the received data from the transponders. This data is usually used to interrogate tags (read) or to reprogram the tag (write). This section usually consists of a microprocessor, a memory block, a few analog-to digital converters and a communication block for the software application (Recouly, 2019) as shown in Figure 6 below.

Figure 5

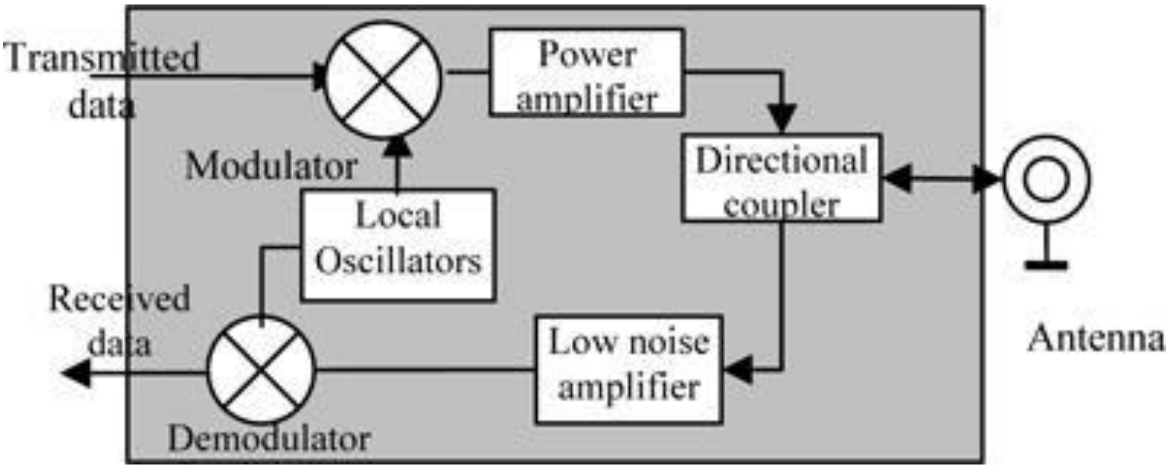
A block diagram for the reader control section (Eppler et al, 2019)



The high frequency interface of the reader is used for RF signal transmission and reception. HF interfaces are consisted of two separate signal paths to correspond with the two directional data flows from and to the transponder as shown in Figure 7 below.

Figure 6

HF interface of a RFID reader operating in the 2.4 GHz ISM band (Ahmed, 2018).

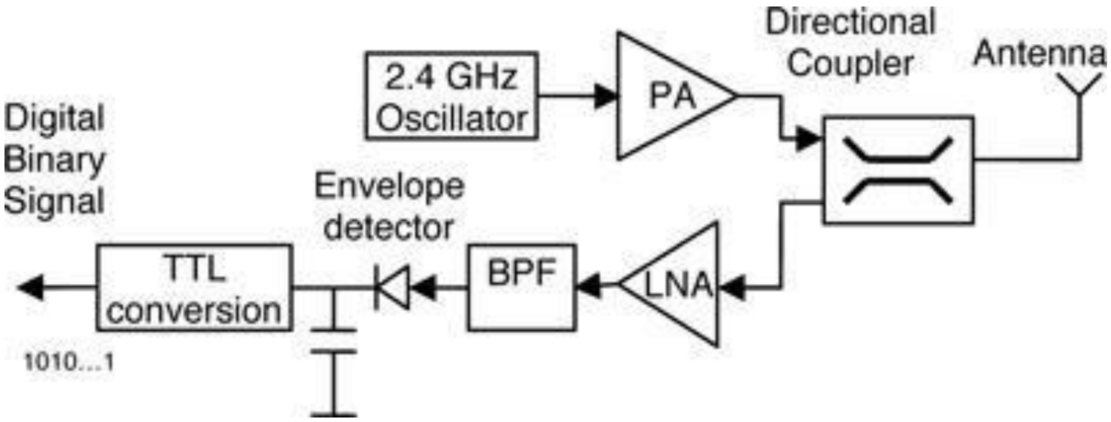


The local oscillator generates the RF carrier signal, a modulator modulates the signal, the modulated signal is amplified by the power amplifier, and the amplified signal is transmitted through the antenna. A directional coupler separates the system‘s transmitted signal and the received weak back-scattered signal from the tag. A directional coupler consists of two continuously coupled homogenous wires, and if all ports are matched, the power of the incoming and outgoing signal is divided in the coupler. The received backscattered signal is weak and the low noise amplifiers increase the signal‘s amplitude before and after the signal is decoded in the demodulator.

The HF interface is one of the most complex sections of the reader. Most HF interfaces are protected from EM interference using metal cages. Different demodulation techniques are used when decoding the data received from the transponder (Bernhart & Leitgeb, 2018).Most RFID systems operate using Binary Phase Shift Keying (BPSK) and amplitude shift keying (ASK). A simple HF interface for a Continuous Wave (CW) RFID reader operating in the 2.45 GHz Industrial, Scientific and medical (ISM) band, using ASK demodulation is shown in the below figure 2.8. The reader transmits a CW signal at 2.4 GHz ISM band. Therefore, the transmit section needs not to be connected with the control section as shown in Fig. 2.8 The CW signal is amplified with a PA and the amplified continuous wave signal is transmitted (Cairó, 2018) via the antenna to the tag.

Figure 7

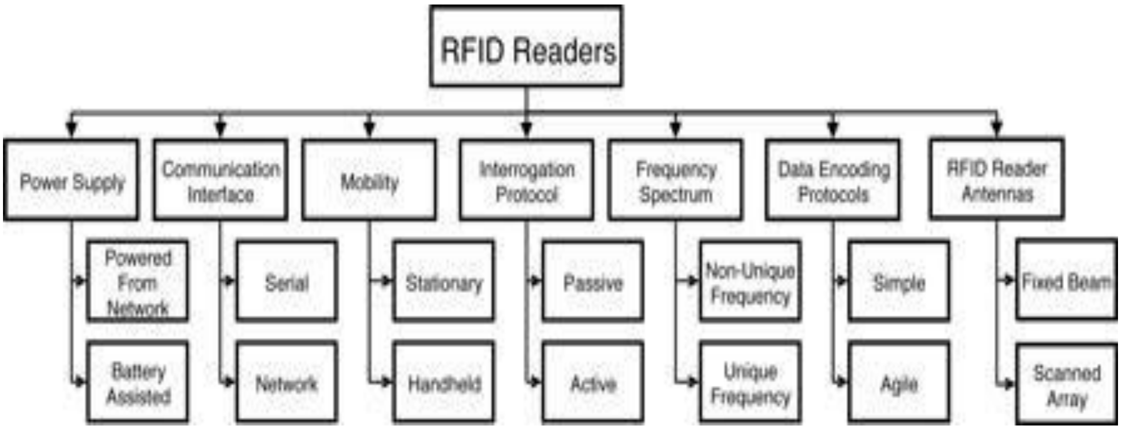
Block diagram of a RFID reader operating at 2.45 GHz with ASK demodulation technique (Ye et al., 2019)



The radiation intensity of the reader antenna determines the interrogation range and zone. Depending on the RFID system's applications the RFID reader can be designed in different ways where the antenna's resonating frequency, gain, directivity and radiation pattern can vary. Antennas are spatial filters (Rui, 2019). Adaptive antennas are a promising technique for implementing this spatial diversity into RFID readers. This antenna is a 5-element rectangular patch antenna array with an intelligent beam forming network at 2.45 GHz. A number of different reader antennas have been developed during the years based on microstrip patch antennas. The diagram below shows the distribution.

Figure 8

RFID Readers



The diagram above is the representation of discussion of various RFID reader systems available in the open literatures (Elfergani, 2018).

2.2.1.5 RFID Challenges

RFID is prone to two main issues: reader collision and tag collision. Reader collision, when a signal from one RFID reader interferes with a second reader, can be prevented by using an anti-collision protocol to make RFID tags take turns transmitting to their appropriate reader.

Tag collision occurs when too many tags confuse an RFID reader by transmitting data at the same time. Choosing a reader that gathers tag info one at a time will prevent this issue. A common RFID security or privacy concern is that RFID tag data can be read by anyone with a compatible reader. Additionally, tags can often be read after the item leaves the store or supply chain. Tags can also be read without the user's knowledge, and if the tag has a unique serial number, it can be associated to a consumer. While a privacy concerns for individuals, in military or medical settings this can be a national security concern or life-or-death matter. Technically two specific events, cloning and spoofing are usually done back to back. Cloning is duplicating data from a pre-existing tag, and poofing is then using the cloned tag to gain access to a secured area or item (Harbi, 2018).

A Denial of Service attack is the broad concept of an RFID system failure that is associated with an attack. These attacks are usually physical attacks like jamming the system with noise interference, blocking radio signals, or even removing or disabling RFID tags. A man-in-the-middle attack can happen during the transmission of a signal. Like eavesdropping, one listens for communication between a tag and reader and then intercepts and manipulates the information. In order to divert the original signal and then sends false data while pretending to be a normal component in the RFID system (Zheng, 2018).

Because RFID tags do not have a lot of compute power, they are unable to accommodate encryption, such as might be used in a challenge-response authentication system. One exception to this, however, is specific to the RFID tags used in passports, Basic Access Control (BAC). Here, the chip has sufficient compute power to decode an encrypted token from the reader, thus proving the validity of the reader (Von Mueller, 2018). At the reader, in turn, information printed on the passport is machine-scanned and used to derive a key for the passport (Robinette & Gernandt, 2016). There are three pieces of information used the passport number, the birth date of the passport holder and the passport's expiration date along with a checksum digit for each of the three. Researchers have pointed out that this means passports are protected by a password with considerably less entropy than is normally used in e-commerce, and further that the key is static for the life of the passport, so that once an entity has had one-time access to the printed key information, the passport is readable with or without the consent of the passport bearer until the passport expires (Ramya, Sandhya and Gayathri, 2018).

2.3 IR Sensors

Infrared technology addresses a wide variety of wireless applications. The main areas are sensing and remote controls. In the electromagnetic spectrum, the infrared portion is divided into three regions: near infrared region, mid infrared region and far infrared region. The wavelengths of these regions and their applications are shown below.

i) Near infrared region-700 nm to 1400 nm - IR sensors, fiber optic

ii) Mid infrared region - 1400 nm to 3000 nm - Heat sensing

iii) Far infrared region - 3000 nm to 1 mm - Thermal imaging

The frequency range of infrared is higher than microwave and lesser than visible light. For optical sensing and optical communication, photo optics technologies are used in the near infrared region as the light is less complex than RF when implemented as a source of signal. Optical wireless communication is done with IR data transmission for short range applications.

An infrared sensor emits and/or detects infrared radiation to sense its surroundings. The working of any Infrared sensor is governed by three laws:

i. Planck's Radiation law, Stephen – Boltzmann law and Wien's Displacement law.

ii. Planck's law states that ―every object emits radiation at a temperature not equal to 0^0K‖.

iii. Stephen – Boltzmann law states that ―at all wavelengths, the total energy emitted by a black body is proportional to the fourth power of the absolute temperature‖.

According to Displacement law, ―the radiation curve of a black body for different temperatures will reach its peak at a wavelength inversely proportional to the temperature‖ (John, 2019).

The basic concept of an Infrared Sensor which is used as Obstacle detector is to transmit an infrared signal, this infrared signal bounces from the surface of an object and the signal is received at the infrared receiver (Waradkar *et al*, 2016).

There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LED of specific wavelength can be used as infrared sources (Ghassemlooy, 2019). The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibers. Optical components are used to focus the infrared radiation or to limit the spectral response (Karaksina *et al*, 2016).

Optical lenses made of Quartz, Germanium and Silicon are used to focus the infrared radiation. Infrared receivers can be photodiodes, phototransistors etc. some important specifications of infrared receivers are photosensitivity, detectivity and noise equivalent power. Signal processing is done by amplifiers as the output of infrared detector is very small (Raj *et al*, 2018).

2.3.1 Types of IR Sensors

Infrared sensors can be passive or active. Passive infrared sensors are basically Infrared detectors. Passive infrared sensors do not use any infrared source and detects energy emitted by obstacles in the field of view.

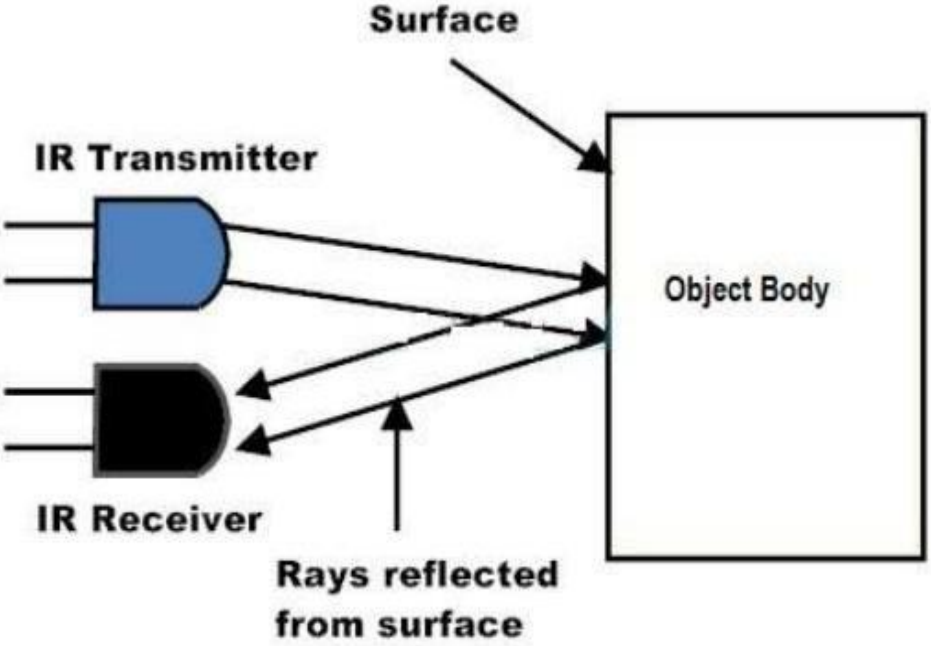
They are of two types: quantum and thermal. Thermal infrared sensors use infrared energy as the source of heat and are independent of wavelength. Thermocouples, pyroelectric detectors and bolometers are the common types of thermal infrared detectors.

Quantum type infrared detectors offer higher detection performance and are faster than thermal type infrared detectors. The photosensitivity of quantum type detectors is wavelength dependent. Quantum type detectors are further classified into two types: intrinsic and extrinsic types. Intrinsic type quantum detectors are photoconductive cells and photovoltaic cells.

Active infrared sensors consist of two elements: infrared source and infrared detector. Infrared sources include a LED or infrared laser diode. Infrared detectors include photodiodes or phototransistors. The energy emitted by the infrared source is reflected by an object and falls on the infrared detector.

Figure 9

Infra red Transmitter and Transiever



Source: Druml, 2019

***i.* IR Transmitter**

Infrared Transmitter is a light emitting diode (LED) which emits infrared radiations. Hence, they are called IR LED‘s. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

The picture of a typical Infrared LED is shown below.

Figure 10

A typical Infrared LED

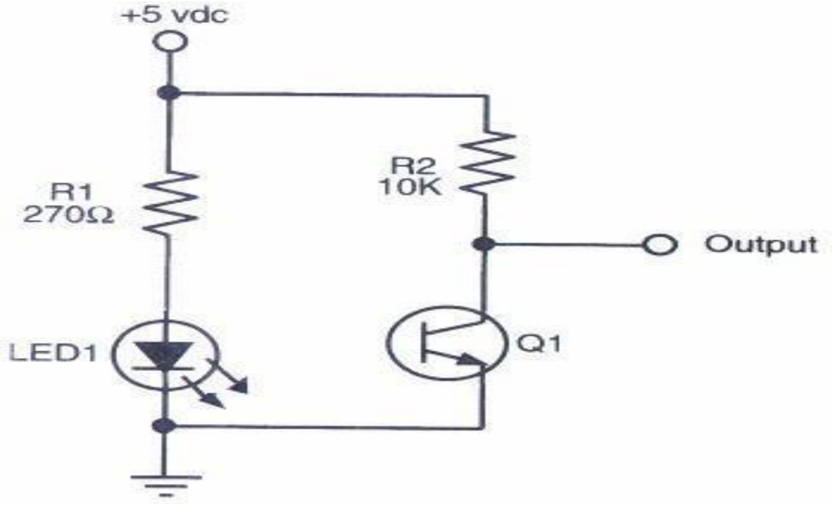
Source: Guri & Bykhovsky, 2019

There are different types of infrared transmitters depending on their wavelengths, output power and response time.

A simple infrared transmitter can be constructed using an infrared LED, a current limiting resistor and a power supply. The schematic of a typical IR transmitter is shown below.

Figure 11

Schematic of a typical IR transmitter



When operated at a supply of 5V, the IR transmitter consumes about 3 to 5 mA of current. Infrared transmitters can be modulated to produce a particular frequency of infrared light (Raj, 2018). The most commonly used modulation is OOK (ON – OFF.

– KEYING) modulation.

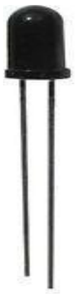
IR transmitters can be found in several applications. Some applications require infrared heat and the best infrared source is infrared transmitter. When infrared emitters are used with Quartz, solar cells can be made.

ii. IR Receiver

Infrared receivers are also called as infrared sensors as they detect the radiation from an IR transmitter. IR receivers come in the form of photodiodes and phototransistors. Infrared Photodiodes are different from normal photo diodes as they detect only infrared radiation. The picture of a typical IR receiver or a photodiode is shown below (Fereyre, 2018).

Figure 12

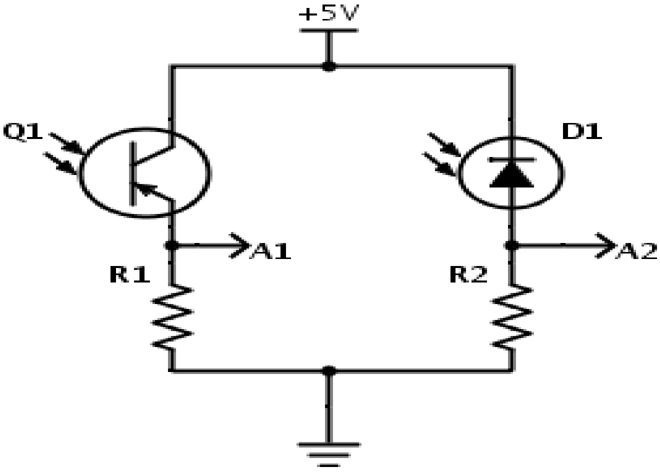
A typical IR receiver or a photodiode



Different types of IR receivers exist based on the wavelength, voltage and package. When used in an infrared transmitter – receiver combination, the wavelength of the receiver should match with that of the transmitter. A typical infrared receiver circuit using a phototransistor is shown below.

Figure 13

A typical infrared receiver circuit using a phototransistor



Source: Rezaei, 2019

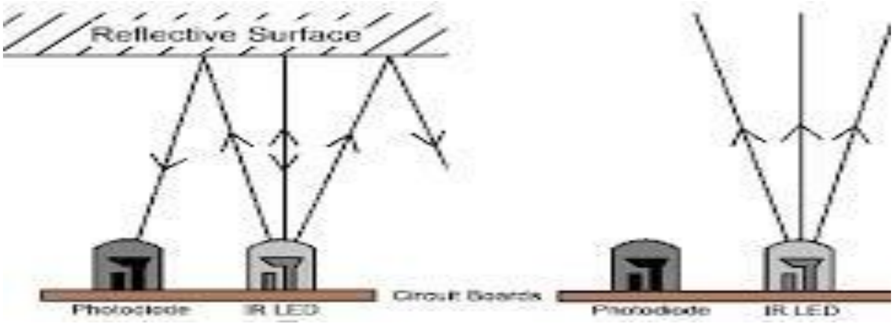
It consists of an IR phototransistor, a diode, a MOSFET, a potentiometer and an LED. When the phototransistor receives any infrared radiation, current flows through it and MOSFET turns on. This in turn lights up the LED which acts as a load. The potentiometer is used to control the sensitivity of the phototransistor (Baichtal, 2018).

2.3.2 Principle of Working

The principle of an IR sensor working as an Object Detection Sensor can be explained using the following figure 2.15. An IR sensor consists of an IR LED and an IR Photodiode; together they are called as Photo – Coupler or Opto – Coupler (Krishnan, 2018).

Figure 14

IR Transmiter Radiation



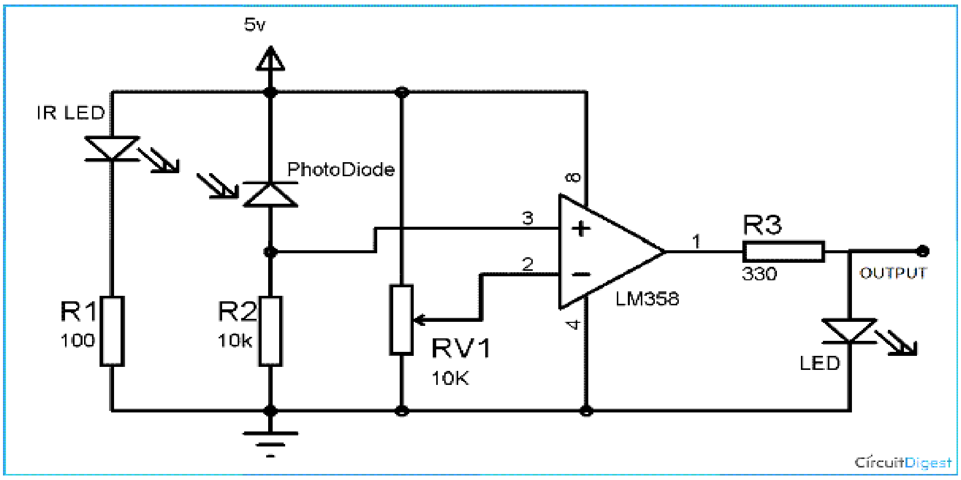
When the IR transmitter emits radiation, it reaches the object and some of the radiation reflects back to the IR receiver. Based on the intensity of the reception by the IR receiver, the output of the sensor is defined.

2.3.3 Obstacle Sensing Circuit or IR Sensor Circuit

A typical IR sensing circuit is shown below.

Figure 15

IR Sensing Circuit



It consists of an IR LED, a photodiode, a potentiometer, an IC Operational amplifier and an LED. IR LED emits infrared light. The Photodiode detects the infrared light. An IC Op – Amp is used as a voltage comparator. The potentiometer is used to calibrate the output of the sensor according to the requirement (Rai, 2019). When the light emitted by the IR LED is incident on the photodiode after hitting an object, the resistance of the photodiode falls down from a huge value. One of the inputs of the op – amp is at threshold value set by the potentiometer.

The other input to the op-amp is from the photodiode's series resistor. When the incident radiation is more on the photodiode, the voltage drop across the series resistor will be high. In the IC, both the threshold voltage and the voltage across the series resistor are compared. If the voltage across the resistor series to photodiode is greater than that of the threshold voltage, the output of the IC Op – Amp is high. As the output of the IC is connected to an LED, it lightens up. The threshold voltage can be adjusted by adjusting the potentiometer depending on the environmental conditions (McKeon, 2018).

The positioning of the IR LED and the IR Receiver is an important factor. When the IR LED is held directly in front of the IR receiver, this setup is called Direct Incidence. In this case, almost the entire radiation from the IR LED will fall on the IR receiver. Hence there is a line of sight communication between the infrared transmitter and the receiver. If an object falls in this line, it obstructs the radiation from reaching the receiver either by reflecting the radiation or absorbing the radiation (Bensky, 2019).

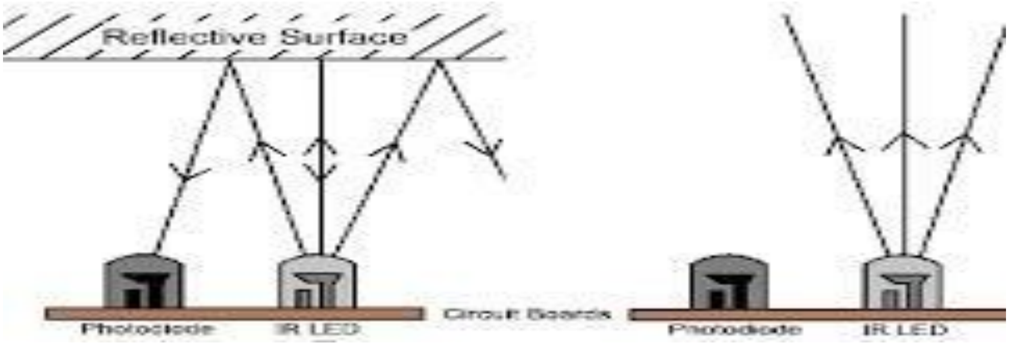
2.3.4 Distinguishing Between Black and White Colors

It is universal that black color absorbs the entire radiation incident on it and white color reflects the entire radiation incident on it. Based on this principle, the second positioning of the sensor couple can be made. The IR LED and the photodiode are placed side by side. When the IR transmitter emits infrared radiation, since there is no direct line of contact between the transmitter and receiver, the emitted radiation must reflect back to the photodiode after hitting any object (Vollmer & Möllmann, 2017). The surface of the object can be divided into two types: reflective surface and non-reflective surface. If the surface of the object is reflective in nature i.e. it is white or other light color, most of the radiation incident on it will get reflected back and reaches the photodiode. Depending on the intensity of the radiation reflected back, current flows in the photodiode (Petrie & Toth, 2018).

If the surface of the object is non-reflective in nature i.e. it is black or other dark color, it absorbs almost all the radiation incident on it. As there is no reflected radiation, there is no radiation incident on the photodiode and the resistance of the photodiode remains higher allowing no current to flow. This situation is similar to there being no object at all. The pictorial representation of the above scenarios is shown below.

Figure 16

IR Trnsmiter Angle of Detection



The positioning and enclosing of the IR transmitter and Receiver are very important. Both the transmitter and the receiver must be placed at a certain angle, so that the detection of an object happens properly. This angle is the directivity of the sensor which is +/- 45 degrees.

In order to avoid reflections from surrounding objects other than the object, both the IR transmitter and the IR receiver must be enclosed properly. Generally, the enclosure is made of plastic and is painted with black color. It is affected by electromagnetic interference.

2.4 Barcodes

Barcodes are printed horizontal strips of vertical bars used for identifying specific items‖. A scanning device reads the barcode by moving a beam across the symbol‖. The first barcode system was developed around the 1940s and 1950s, since then people have become very accustomed to their use, through common applications such as in retail and grocery markets (Ako, 2019).

There is a large array of works, such as Behind Bars, that delve into the history and development of barcodes. Similarly, Collins and Whipple in Using Bar Code: Why it’s taking over, offer a concise history of barcodes along with examining the typical components that make up a barcode system. Unfortunately, Collins & Whipple only provide a very short section on RFID and omit the inclusion of a comparison section. Again, a work by Palmer also identifies the typical elements of a barcode system and related concepts. Although this work provides a sound review of barcodes, it too, fails to critically compare barcodes with other technologies (Vargas *et a,l* 2017).

2.4.1 Barcode Standards

As with the history of barcodes, most works provide, to some degree, background information regarding the development of barcode standards. While most sources examine a number of standards, there is a common facet among all works in that they discuss the importance of the UPC (Universal Product Code) standard. One work in particular provides a thorough insight into the development of the UPC standard, including its formation and in-depth historical aspects. Regrettably, this survey does not explore the use of barcodes or their application to SCM, as it solely covers the UPC standard for barcodes and related issues (Cai, 2019). Most works on barcodes include some background information regarding the importance and development of the UPC and the Uniform Code Council (UCC).

2.4.2 Barcode Applications

Barcode technology has been used for decades and during this time has proved itself as a reliable performer when supporting SCM. Testament to this is the versatility in which the technology can be adapted to suit specific applications. There is a plethora of case studies which examine the role of barcodes within a particular SCM scenario such as Auto Group hits high notes with new inventory management system and RF cuts processing time for Lexus car and parts deliveries. The articles provide a snapshot of specific barcode implementations and, as with most works of this time, both case studies provide an analysis of how barcode technology is helping the company improve their SCM processes. The articles review most aspects and processes of the new system and outline the benefits realized post implementation. RFID technology has had a number of substantial technology advancements of late and as both of these articles were written in 1998, alternative RFID technology has not been compared in the case studies, which would have made the case studies much more applicable to this thesis (Selvini, 2018).

The majority of works which provide information about barcodes offer some examples of their applications within a vertical sector. There is significant level of similar findings among works which offer examples of barcode applications. Technologies all present similar example applications in cases like hospitals and health care, retail, library and distribution. In addition to this, most of these examples do not address the importance of barcodes in SCM (Hasan, 2019).

2.4.3 The Future of Barcodes

The work of Cohen in Automatic Identification and Data Collection Systems explores current trends in a range of automatic identification and data collection technologies (Michael, 2003). No formal methodology is used; instead, Cohen uses an informal case study approach when identifying these technologies. Cohen goes on to explore future trends and developments in the industry. Like a number of the reviewed works, Cohen has not given all technologies equal weight in his work, opting to focus on barcode technology.

An article by Yu (2008), explores new developments in barcode applications and emphasizes that further developments in barcode technology are continually being made. Although the article is simply descriptive and does not make use of any formal research methods, it does offer some modern barcode concepts. It notes that there is ―no shortage of innovative new applications for barcodes and that ― useful developments in the technology barcodes continue to be made. It goes on to discuss specific new barcode developments such as 2D barcodes. ―Unlike standard barcodes, which depend on links to a larger database, 2D barcodes contain a kind of mini- database themselves, which includes information on the product and can be encrypted‖. Another useful feature of this technology is that even if the barcode is partially damaged, information can still be gathered. A small section notes that many systems now allow for barcode and RFID technology to be combined, alluding to the concept of the integration and convergence of the two technologies (Başyazicioğlu, 2018).

In conclusion, there lies a common theme and structure to most of these works, which is that they all cover elementary ground and follow a similar structure. Typically, they cover barcode definitions and provide a brief history, followed by describing parts that make up a barcode system such as printers and readers. Finally, they cover common past business applications. Many of these sources neglect to delve into exploring what the future of barcode technology may yield. The few articles that attempt to explore the future of barcodes, typically, do not utilize a rigorous methodology. Several works explore alternative technologies such as RFID, however rarely is RFID given an equal weighting to that of barcodes. In addition to this, there is a tendency to avoid the critical analysis of barcodes and their use within SCM.

2.4.4 RFID versus Barcode

RFID technology is regularly compared with barcode (for example Universal Product Code (UPC) or European Article Number (EAN) by practitioners when discussing the disadvantages of RFID. This is quite logical since the barcode is most commonly used in retail (the first UPC barcode was scanned on June 26, 1974 (Harris, 2009) and logistics as it is inexpensive to use (i.e. a barcode costs ―half a cent each‖ (Shih, 2009) and can be easily integrated into packaging; to this day barcodes are still in use in many different areas. Moreover, both technologies function as data carriers and are attached to an object which eventually will be read. Currently, there is a debate among practitioners about whether RFID will be the successor of the barcode or not, or that it will simply co-exist. For clarity purposes in Table 2 a comparison is made, so that one is able to exactly tell the difference between the two.

Table 2

RFID and barcode a comparison

	RFID	Barcode
Read Rate	A large amount of tags can be read at the same time from a (certain) distance, within a second: humans are not required to read tags as it is done automatically.	Barcodes have to be read individually by hand from a very small distance; reading a single barcode takes a few seconds, mistakes are easily made, and are labour intensive.
Line of Sight	Not compulsory for a reader to see the tag to read it; tags do have to be in read range in order to communicate.	A barcode must be visible for it to be read by a scanner.
Read/write	A tag can have read /write capabilities.	A barcode is strictly limited to reading.
Placement	A tag can be placed inside an object. This means it is less likely to get damaged and thus it can be reused numerous times. It can be used in environments that are not suited for barcodes.	Due to line of sight requirements, barcodes must be visible. They can get filthy or damaged due to (rough) handling resulting
Security	A tag is not easy to copy since it contains unique data which is stored in a secure manner.	Barcodes lack this protection and thus are simple to copy or forge.
Costs	At the moment RFID, compared to barcodes, is more expensive to implement.	Barcodes are inexpensive to use and ubiquitous.

Sources: Atlas RFID solutions 2010

2.5 Existing Smart Parking Systems and their technologies

Over the years, the number of parking - system-related technologies has increased. In any era where there has been a significant amount of traffic, there have been car-parking systems (Melsen 2013). Car parking systems were first developed in the early 20th century in response to the need for storage space for vehicles. (Melsen 2013) Indeed, the using of e-smart parking systems dated back to the 1920s, as automated parking systems appeared in U.S. cities such as Los Angeles, Chicago, New York, and Cincinnati. In addition, one of the Kent automatic parking garages in New York is an art deco landmark that was converted into luxury condominiums in 1983. A system that is prevalent all over Japan is the Ferris-wheel or ―paternoster system, which ― was created by the Westinghouse Corporation in 1923 and built in 1932 on Chicago‘s Monroe Street.

In the past two decades, the concept of intelligence in terms of smart parking systems became more popular in the most vibrant cities, especially in malls and shopping centres. (Melsen 2013) In the mid-80s, the systems used for parking relied mainly on the traditional method of pushing a button in the device next to the checkpoint to get a parking ticket and on exiting, the driver must pay before inserting their ticket in order for the barrier to rise. This was the method used to determine how many cars came in and out the system each day, and it was used to count the number of vacant spaces available. It began by utilizing different methods such as sensors or barriers to be able to know the status of parking lots. All these methods developed dramatically further until recently the term smart city vision ‘emerged (Silva, 2018).

2.5.1 Smart Parking Technologies

This section discusses the existing methodologies used for smart parking. Also in this section, all related studies are gathered into groups based on the techniques used specifically in the academic domain. It is very clear from all the references discussed in chapters below that category and classifications of smart parking vary from source to source. Some rely on the technology used while others rely on data processing to get information about the parking statues. For example, in the centralized assisted parking search, the information processing will be stored on the central processor (server). The non-assisted parking search does not have a server, and no information will be provided to a user. Numerous technological methods are grouped into the following classification (Al-Turjman and Malekloo, 2019).

2.5.2 Smart Parking Systems Based on Agent Model

The systems based on this technique have been proposed in academia (Mateo et al. 2009; Yang *et al*. 2009; Chou *et al*. 2008; Longfei *et al*. 2009; Adler *et al*. 2005; Khoukhi et al. 2010; Li *et al*. 2004; Balbo; *et al*. 2005). These types of system can be any entity capable of observing facts via sensors, as the system is acting upon the changes of the environment through exchanging information and interaction upon that act. It has useful characteristics, such as autonomy, reactivity, and adaptability (Mahmud, 2013). Essentially, a multi-agent system is a modeling method developed to represent systems with entities, autonomy, and interaction. Agent -based intelligent parking systems are a form of mobile agent technology with a multi agent system. The following section considers a few selected works, chosen according to the aforementioned criteria.

A multi-phase navigation method based on a two-layer traffic map, which is employed for parking route negotiation and guidance was proposed by Li et al. (2004). The distribution approach involves building an active Parking Guidance Information System (APGIS). The APGIS is composed of cars, car parks, and a parking information service center (PISC), which has four functions: parking space searching, parking price negotiation, parking space booking, and parking route negotiation and guidance. Khoukhi *et al.* (2010) proposed a multi- agent system; this contains a main control kernel agent, a learning navigation agent, a localization agent and, finally, a communication agent. The system worked well in a simulation environment and the results were promising and encouraging. (Longfei *et al.* 2009) developed a multi agent system called a multi agent based intelligent parking negotiation and guidance systems (ABIPNGS) parking system. They proposed a negotiation algorithm based on the human bargaining process. The system operation mechanism was outlined in Longfie‘s paper. However, it can be noted that the ideation of this concept is at conceptualization level and not a real working experiment. Therefore, all still lacking to give a working model that can be used in practice.

2.5.3 Smart Parking Systems -Based on Fuzzy Logic

Since 1965, when fuzzy logic was introduced by Professor Zadeh (1996), it has played an outstanding role in design and production in industry. Actually, fuzzy control systems are control systems based on the fuzzy logic system, which analyses analog input values in terms of logical variables that take on continuous values between 0 and 1, while digital logics operate on discrete values of either 1 or 0. Nowadays, fuzzy logic has become a standard technology, which is applied in data and sensor signal analysis. Fuzzy code is designed to control something, usually something mechanical.

This type of system is proposed in (Sharafi *et al*. 2010; Mohammadi *et al*. 2011, Song *et al*. 2006, and Zhao *et al*. 2005). The following proposals as examples are chosen in order to clarify the principles of this technique. (Song *et al*, 2006) proposed a system that depends on an FPGA-based fuzzy logic controller (FLC). The benefit of using an FPGA-based FLC compared to software FLC is that takes less time to process the information. First, a Fuzzy Control System is chosen. Then, the implementation of the fuzzy rule-based system takes place upon the neural network architecture. It is the main reason for learning and adapting from the training data: ―The neuro-fuzzy system has the ability to reason like human beings as well as it has expert knowledge.

Benson *et al*. (2006) proposed that an RF transceiver and antenna with an ATMega 128 L micro-controller system could operate by monitoring the availability of car-parking spaces and send this information to customers and facility administrators. (Sharafi *et al*. 2010) presented a fuzzy approach for the control of the backward movement of trucks and trailers in a dynamic environment. This method was then expanded to circumstances in which there are obstacles in the truck's pathway. In the first scenario, it is assumed the obstacles are constant. The second scenario assumed by the authors is that there are moving obstacles which can mean the truck must be directed to the parking facility. The parking process is completed due to the intelligence of fuzzy logic. The proposed ultrasonic sensor identifies objects and obstacles longitudinally. (Zhao *et al*. 2005) developed and demonstrated a robust automatic parking algorithm for parking, using a genetic algorithm's learning ability for space detection errors by employing a Kinematic model for a skid steering autonomous ground vehicle.

2.5.4 Wireless Sensor Networks-Based Systems

These types of systems have generated increased interest in researchers since 2005. They are the most popular technique in the last decade with researchers, as wireless sensor networks have various advantages, such as flexibility, intelligence, reasonable cost, rapid deployment, and sensing, as it usually consists of sensor nodes. The following papers discuss WSN-based parking systems (Yan and Olariu 2008; Yan *et al*. 2009; Miura *et al*. 2006; O'flynn *et al*. 2005; Kumar *et al*. 2007; Lee *et al*. 2008; Park *et al*. 2008; Tubaishat, *et al*. 2009; Reve *et al*. 2012; Sharma *et al*. 2011; Boda *et al*. 2007; Haranguing, *et al*. 2007; Cheung *et al*. 2005; Wenzhi *et al*. 2006; and Agarwal, et al. 2009). This type of system, which utilizes sensors to monitor environmental conditions, is widely used, especially in academia, due to the ease of installation and configuration, and the reasonable price. (Zheng *et al*. 2006) developed a system using crossbow products, which have a low unit cost. This system enables a car to detect entry to the car park, and it efficiently guides the driver to an empty parking space through signs displayed to the driver.

A new smart parking system using an ultrasonic detector was also presented by (Kianpisheh *et al*. 2012). For each individual car park, one sensor is fixed on the ceiling above each parking space. Ultrasonic Sensors operate based on echolocation. The sensor transmits a sound, which hits a solid object (car or ground) and is reflected back to the sensor. (Mathur *et al*. 2009) discussed the research challenges relating to parking technology and proposed some possible solutions. In the centralized solution, some cars are equipped with ultrasonic sensors as well, which drive past the parking spaces to collect occupancy data and upload the data to the centralized database. The cars that need to park simply query the centralized database. (Lee *et al*. 2008) proposed the use of a combination of magnetic and ultrasonic sensors for the accurate and reliable detection of vehicles in a parking lot, and described a modified version of the min max algorithm for the detection of vehicles using magnetometers.

2.5.5 Smart Parking Systems Based on Vehicular to Infrastructure Communication (V2I)

According to Stibor *et al*. 2007; Holfelder *et al*. 2004; Yousefi *et al*. 2006; Bilstrup *et al*. 2008; Panayappan et al 2007; Geng *et al*. 2012; Lu *et al* 2010), proposed using the term (CVT) to refer to Connected Vehicle Technology which depends on wireless data transmission between vehicle and infrastructure (V2I). This promising technology emerged recently. It proposes a new smart parking technique that depends on developing a new VANET-based smart parking to be used for smart steering and smart parking. It refers to Vehicular Communication Systems, in which vehicles and roadside units are the communicating nodes, that is, they communicate and exchange information with each other, such as safety warnings or supplying the traffic congestion information and even for finding vacant parking spaces.

Basically, vehicular networks are considered to contain two types of nodes: vehicles and roadside stations (Geng *et al*. 2012). Both are categorized under the term Dedicated Short Range Communications '(DSRC) devices. DSRC works in 5.9 GHz bands with a bandwidth of 75 MHz and a range of about 1000 m. This is a two-way method of communication including Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V) communication. In the smart parking‖ system, usually, V2I communication includes drivers sending their parking requests, providing driver information, and confirming that reservation to the system. I2V communication involves the DRPC sending allocation results, driving directions, payment details, and more, back to vehicles. It is worth mentioning that cellular networks are usually applied in V2I and I2V solutions. (Stibor *el al*. 2007) proposed a novel parking system called SPARK, which consists of four parts: system setting, real-time parking navigation, intelligent anti-theft protection, and friendly parking information dissemination.

In (Lu et al. 2010), the SPARK scheme is characterised by employing a parking lot's RSUs to provide surveillance and manage the whole parking lot using VANET communication technology. The system uses light sensors, and in the proposed SPARK scheme, the whole parking lot is under the surveillance of the three parking lot RSUs. (Panayappan et al. 2007) described a parking system in VANET which locates any available parking spaces. This system depends on roadside units deployed to relay parking messages and GPS to locate parking positions.

2.5.6 Smart Parking Systems Based on Global Positioning Systems GPS

Global Positioning Systems (GPS) technology (also known as Satnav) is used to determine and track a vehicle's precise location. In this domain, it is used to offer information about the location and availability of parking spaces at the destination. This technique proposed in (Pullola et al. 2007; Chon et al 2002; Hanif et al. 2010). Chon et al presented a location-based system called NAPA. The server in the system associates buildings on the campus with parking lots in the order of distances to the building. After locating the nearest available parking lot, the user sends the NAPA server a message that he/she has parked. Then the server updates the information about the lot accordingly. When the user leaves the parking lot, the NAPA server can automatically charge the appropriate parking fee if necessary. (Hanif et al. 2010) proposed a new smart parking system using SMS services. This system is capable of finding parking spaces in specific car park areas. A parking reservation system is developed in such a way that users can book their parking spots over short message services (SMS) using the GPS. The SMS is processed by a wireless communication instrumentation device called a micro- RTU (Remote Terminal Unit). The proposed porotype have the following specification; the circuit has a simple design, the reliability level is high, and the system accuracy is excellent.

2.5.7 Smart Parking Systems Based on Computer Vision

Recently, many researchers have focused on these methods (Takizawa et al. 2004; Xu et al. 2000; Zhu et al. 2007; Funck et al. 2004; Fabian et al 2008; Tan et al. 2009; Jung and Choi el al. 2007; Tanaka et al. 2006; Jung et al. 2010; Banerjee et al. 2011). This field of study includes methods for acquiring, processing, and analysing images. It uses computers to emulate human vision, including learning and being able to make inferences and take actions based on visual inputs, also called computer vision. The goal of computer vision is to make computers efficiently perceive and process visual data, such as images and videos, and act upon changes in these images.

Usually, the technique involves analysing a few frames per second and then sending the data to a central database, after which, the user can retrieve information about the changes at the parking lot. In (Takizawa et al. 2004), their system utilized CCTV in a vehicle detection stream to detect the presence of a car or vehicle in a particular parking lot. Pixel detection is used to detect the presence of a vehicle in each parking lot. A certain number of pixels in the grayscale are used as the threshold to differentiate pixels from the vehicle and from the unoccupied lot. Another parking system, called CCTV, uses images to detect parking spaces. CCTV cameras are fitted in car parks to automatically detect car parking spaces. However, these methods may incorrectly detect parked vehicles. The system is targeted on cases where occupancy values are required. The reliability is high and the system is very accurate; however, all other parameters are unclear in the paper.

(Funck *et al*. 2004) proposed a system that uses CCTV cameras that are fitted in car-parks to automatically detect car parking spaces. However, these methods are not always accurate in cases where occupancy values are required. (Bong at el 2008) proposed a research project which was developed to acquire car- park occupancy information using an integrated approach of image processing algorithms. Motivation for developing this system came from the fact that minimum cost is involved because image processing techniques are used rather than sensor-based techniques. This project is called the Car-Park Occupancy Information System (COINS).

2.5.8 Smart Parking Systems Based on RFID Technology

This technology (RFID) is proposed in the following selected systems (Pala and Inanc 2007; W Gueaieb and M s Miah 2008; Jian et al. 2008; Cervantes et al. 2007; Hsieh et al. 2008; Liu et al. 2010; Zhang et al. 2006; Šolić, et al. 2015). In many academic papers, smart parking's RFID solutions make it possible to manage permit parking easily, especially in the prototype stages. The main mechanism of RFID technology depends on an electromagnetic field to identify and track tags attached to objects automatically. (Pala and Nihat 2007) used RFID technology in automation. Their system uses a software program for controlling and reporting changes in the status of the parking space, and for the operation of tasks such as choosing the closest vacant parking space, and it then sends the report to the driver. Meanwhile, (Jian et al. 2008) proposed a system containing Gate-PC Controller and Embedded Gate Hardware, an RFID System, and a Modular Parking Management Platform: ―Most systems in the Modular RFID Parking Management System are modulated and can be substituted for any other similar system or hardware‖.

2.5.9 Other Hybrid, M2M, IoT Systems

(Yeh *et al*. 2016; Pham *et al*. 2015; Alhammad *et al*. 2012; Foschini *et al*. 2011) and others use certain kinds of hybrid techniques and mixed methods. As mentioned earlier, it is difficult to classify each type with a certain group, but it is useful to do so for the sake of clarity. (Lee *et al*. 2016) proposed a smart system that detects and finds the parked location though systems based on IoT, smart sensors, and actuators, with the middleware connecting clients with terminal devices. The system is dependent on Bluetooth communication between the smartphone and wireless sensor motes. (Fraifer and Fernström 2014) proposed a smart car parking prototype using camera nodes and an OpenCV algorithm to detect the vacant parking space to facilitate the parking service to the users. (Foschini *et al.* 2011) presented and discussed the design and implementation of an M2M application in the field of road traffic management.

This study used an integrated IoT retractable bollard management system to allow vehicular access to restricted city areas, based on standard infrastructure and software components; the authors have invented an intelligent parallel technique which involves using RFID technology with fuzzy logic controllers and two ultrasound range sensors. This system contains a Gate-PC Controller and Embedded Gate.

Hardware, an RFID System, and a Modular Parking Management Platform. As mentioned previously, most systems in a Modular RFID Parking Management System are modulated and can be substituted for any other similar system or hardware. Finally, check-ins and checkouts of the parking lots are under the control of RFID, reader, labels, and barriers. (Klappenecker et al. 2014) modelled a parking lot as a continuous-time Markov chain. The parking area is modelled as a grid, and schemes for information aggregation and dissemination over the grid are proposed. Moreover, in (Foschini *et al.* 2011), M2M system technology has recently emerged as a promising enabler for the development of new solutions in a plethora of IoT application domains including transportation, healthcare, smart energy, smart utility metering, supply and provisioning, city automation, manufacturing, and others (Foschini *et al.* 2011). M2M enables highly scalable direct communications among wireless enabled heterogeneous terminals, called M2M devices.

Basically, the principal of M2M applications that realize M2M communication involves four stages: i) data collection, ii) transmission of specific data over a communication network, iii) assessment of the data, and iv) response to the available information. All these specifications make the involvement of the M2M desirable in smart parking systems. The M2M networking architecture for IoT connectivity uses aggregator devices to serve multiple end nodes. A gateway connects to a cellular network for eventual Internet attachment. Specifically, the end M2M nodes contain one or more sensors that report physical conditions to a remote site or that are used with local embedded intelligence.

Table 3

Overview of Smart Parking System Based on different Technique

Technique	Reliability	Communication	Circuit	Detection
Based		Method	Complexity	Accuracy
RFID	High	Wi-Fi	Complex	Accurate
CCTV coins	High	Wi-Fi a, g	N/A	False detection may occur
light sensor	High	Zig-bee	Complex	Accurate at day time Cannot b at night
Acoustic senso	High	RF	Complex	Seriously influenced environmental noise
Optical sensor	High	Blue-tooth	Complex	Very accurate
Ultra-sound	High	Switch and LAN	Simple	Accurate
SMS	High	GPS	Simple	Accurate
Magnetic sens	High	WIFI /RF	Simple	Accurate
Infrared	High	RF/Wi-Fi	Simple	Too sensitive Maximum accu day time

Source: Al-Turjman (2019).

In relation to the points above, the review has identified several techniques of modified versions, especially sensor-based smart parking technology. Numerous different smart parking systems have been chosen to fulfil the same purpose with a different technique and the data have been analyzed according to different factors, such as technique’s cost, reliability, scalability, accuracy, communication type, circuit complexity, method of operation and ease of installation, as illustrated in Table 3.

2.5.10 Vehicle Detection and Current Parking Occupancy

Sensors are one of the prominent tools used in vehicle detection for parking management. Research by Guru Prasad et al. (2012) utilizes Fiber Braggs Grating sensors for vehicle detection. This type of sensor is embedded into the ground. As the pressure exerted on the sensor changes, the wavelength of light reflected at the sensor changes as well. With of help of this change, the presence of a vehicle can be detected. In the study, the sensors were placed approximately where the rear wheels of the vehicles would be after parking the spot.

The research showed successful implementation of such technology for a prototype of two parking spaces. There are several other research studies which use combination of sensors in detecting vehicle movement and parking spot occupancy. A research by Haoui et al (2008) at Sensys Networks Inc., uses a combination of underground and overhead sensors in estimating parking occupancy. This system consists of several underground sensor nodes (covering an area of 3‘x 3‘), which use Earth’s magnetic field to estimate a vehicle passing over them. For a group of such sensors, an overhead access point is also included in this vehicle detection system. Such an access point will analyze the signals collected at its group of nodes and applies several algorithms to detect the possibility of vehicle passing over the sensor. Essentially, each parking spot would have such a sensor under it. Data from these access points is further accumulated over larger network and used in determining characteristics of the system (Jacobus, 2018).

These systems typically end up using a whole lot of sensors which makes the vehicle detection for parking a costly affair. Research by Caicedo on systems such as PARC

(Parking Access and Revenue Control) takes an alternate approach at vehicle detection (Caicedo, 2009). Rather than using the sensors to detect each parking space, this system uses them at key locations so as to divide the overall parking area into smaller zones and calculates the parking occupancy over such zones. This does result in loss of information, as the occupancy for each parking spot is no longer available. However, in many cases such level of accuracy is not particularly required. Such alternatives do provide a decent alternative which provides useful information at lesser costs.

2.5.11 Cell Phones Based for Parking Occupancy Detection

A research by Grazioli et al discusses the complete framework of having a smart phone based user interface for parking management systems and the advantages resulting from it. These researchers made an attempt to implement a complete system consisting of web applications for parking management staff and web or smart phone based applications for users. It also provides a way for parking controllers to manage the parked vehicles and help them in revenue collection and enforcement. Functions such as enabling users to monitor their parking activity by keeping them informed about the parking duration, utilization of algorithms such as Ray casting algorithm along with GPS precision, integration with apps such as google maps & Foursquare and also enabling users to participate in crowd sourcing activities are some of its strong suits (Meurer, 2018). Examples of existing solutions:

i. Parking-Mate: Smartphone based application. It displays the nearby parking facilities and their rates. It does not show any information about availability of spaces.

ii. Voice-Park: It simplifies the process to find a parking space by adding the voice search feature. This application guides you to the parking space with voice directions, removing the need to look at the phone and in turn get distracted while driving.

iii. Park-Mobile: This mobile based app displays the data from parking meters and enables the user to enter the parking id or parking meter number and add time on the meter.

iv. Secure-Parking: This app shows the availability of parking spaces in addition to enabling users to add time one those.

v. SF-Park: This application was developed as a prototype for San Francisco in order to improve the parking utilization in the city. Other than displaying the available spaces and being able to pay online, this app enables to have a dynamic pricing module to improve parking utilization.

vi. Merge (Xerox): The next two solutions on this list are more sophisticated. Merge is an effort by Xerox to provide an integrated parking solution which can manage people, parking infrastructure and the revenue. It can provide real time parking (collected with help of sensors) availability data to the customer. Additionally, it records this data to determine dynamic pricing of the parking spaces in order to manage the demand better and increase the revenue.

vii. Cisco and Street-line: This solution was developed in collaboration of Cisco with street-line. It combines street-line's sensor based solutions with Cisco's Camera based solution in order to collect data. As a part of this effort a City

Information System is also being proposed to expand the use of the system for other uses such as smart lighting, video surveillance.

Android, where is my car? (App Inventor): Developed at MIT, this is yet another example of simple but highly useful mobile application pertaining to parking. It enables the user to store the location of parking so that at a later point of time, it would be easier for the user to find their car easily. Use of this application comes to fore during events such as concerts, games or other such events where the parking is spread over large areas. Basically, when we are visiting some place for the first time and are not accustomed with the parking arrangements, this app will help in tracing back the location of the car. It uses a link with Google maps to calculate the path between the current location and the remembered location of the car.

2.5.12 Problems and Challenges facing today's Parking Systems

As society increasingly relies on computer-based information systems to improve efficiency, smart parking systems are rapidly increasingly in number. Smart parking systems have become a necessity, especially in dense population centers. While designing with a user-center approach, two main issues emerged, regarding, first, operational issues and second, specific technical security challenges. These challenges must be addressed from the very beginning to ensure that the system will work efficiently. All the available studies related to traditional smart parking systems in the last decade indicate that they do not satisfy drivers' requirements.

2.5.13 Operational and Technical Challenges, Issues, and Problems

From all the selected prototypes in academia, it is clear that some systems only guarantee a vacant parking space virtually, but do not indicate where it is exactly; these types of systems depend on counting how many cars have entered the parking space and how many cars have left, thus continually calculating the number of parking spaces available. These systems rely on the driver then finding the available parking space by driving around until they see it. One of the greatest challenges for smart parking systems when users employ these systems is how to engineer trust in the systems (Gupte & Younis, 2018).

In the City of Westminster, London, there is a smart parking system that uses a wireless sensor network with no barrier, but the parking area is still constantly monitored by CCTVs so that when cars park in the wrong places, they can beclamped. However, this system is still not smart enough to identify and then match specific cars to their pre-booked parking spots, and so relies on customer complaints after the fact. To sum up, most of the recent proposed smart parking systems that have been investigated depend on wireless sensor networks which, in reality, need to be placed in small holes in each parking spot, so applying this system involves changing the infrastructure in addition to many complex requirements regarding circuits, etc. In addition, system operation is inconsistent, as environmental factors such as snow or dust (or anything that might cover the sensors, including the units being covered deliberately) could result in the failure of the whole system (Nguyen, Salcic and Zhang, 2018). Despite comprehensive research, systems that depend on image processing are still not accurate or reliable in practice, especially at night or when cars are parked incorrectly.

2.6 The Design Component Required for Microcontrollers

This project adopted an Intelligent Parking Controller (IPC) Microcontroller module to communicate with personal computers (PCs) and a switch. The IPC module device of model number TCP-300 The conceptual design of the intelligent parking system (IPS) involves two identical IPC devices – an entry controller and an exit controller – connected to a power supply, a server PC, a charge PC, and a switch. Each controller was programmed as needed. The IPS records the number of vehicles entering and leaving a facility, calculates empty parking slots, and displays the available number of parking slots. The IPC has the following primary features:

i. Embedded TCP/IP network design
ii. Two Wiegand reader interfaces
iii. Two RS232 reader interfaces
iv. RS485 display interface
v. DC12V/24V working voltage

2.6.1 Types of Microcontrollers

The different types of micro-controllers have been expounded based on characterization as indicated by Number of Bits. The available micro-controllers bit ranges include 8-bits, 16-bits and 32-bits microcontroller. In 8-bit microcontroller, the moment that the insides transport is 8-bit then the ALU plays out the number-combination and rationalizes tasks. The cases of 8-bit microcontrollers are Intel 8031/8051, PIC1x and Motorola MC68HC11 families (Düll, M., Haase, B. 2015).

The 16-bit microcontroller performs more prominent exactness and execution when contrasted with 8-bit. For instance, 8-bit microcontrollers can just utilize 8 bits, bringing about a last scope of 0×00 – 0xFF (0-255) for each cycle. Conversely, 16-bit microcontrollers with its 16-bit information width has a scope of 0×0000 – 0xFFFF (0-65535) for each cycle. A more drawn out clock most outrageous worth can almost certainly turn out to be helpful in specific applications and circuits. It can consequently work on two 16-bit numbers (Mathur& Panda, 2016). A few cases of 16-bit microcontroller are 16-bit MCUs are broadened 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families.

The 32-bit microcontroller utilizes the 32-bit guidelines to play out the number arrangement and rationalise tasks. These are utilized as a part of naturally controlled gadgets including implantable therapeutic gadgets, motor control frameworks, office machines, apparatuses and different sorts of inserted frameworks. A few cases are 8080, 8087 family, Intel/Atmel 251 family, PIC3x.

2.6.3 Grouping as indicated by Memory Gadgets

The memory gadgets are separated into two sorts, they are; Embedded memory microcontroller and External memory microcontroller. Inserted memory microcontroller: When an implanted framework has a microcontroller unit that has all the useful squares accessible on a chip is called an installed microcontroller (Singh, Harkishen. 2015). For instance, 8051 having program and information memory, I/O ports, serial correspondence, counters and clocks and hinders on the chip is an inserted microcontroller.

Outer Memory Microcontroller: When an implanted framework has a microcontroller unit that has not all the useful squares accessible on a chip is called an outside memory microcontroller. For instance, 8031 has no program memory on the chip is an outside memory microcontroller (Kant, 2014).

2.6.4 What is a 32-bit Microcontroller?

A 32-bit microcontroller is a modest PC on a solitary coordinated circuit that contains the memory, a processor center, and programmable information/output peripherals. Program memory, for example, OTP ROM or NOR streak and a little measure of Smash can likewise be found on the chip (Kant, 2014). The performance micro-controller that possesses the mentioned desired characteristics is 8087. With high efficiency and real time data transfer.

2.6.5 Kinds of 32-bit Microcontrollers

Microcontrollers can contain a differing number of broadly useful pins. The pins can be designed by programming to either information or an output state. At the point when these pins are arranged to an information state, they can be utilized to peruse sensors or outside signs (Kant, 2014). When they are arranged to output express, these pins can drive outside gadgets like engines or LEDs.

2.6.6 The 32-bit Microcontrollers from Future Gadgets

Future Gadgets has a full choice of 32-bit microcontrollers. The 32-bit microcontroller specialized properties underneath will rapidly be limited to coordinate particular microcontroller embedded application needs (Alaba, E. S., &Abiodun, R. O. 2017).

2.6.7 Applications for 32-bit Microcontrollers:

Microcontrollers are intended to be utilized in embedded systems applications, in opposition to the microchips found in PCs. Microcontrollers are utilized as a part of naturally controlled gadgets including implantable therapeutic gadgets, motor control frameworks, remote controls, control devices, toys, office machines, apparatuses and different sorts of installed frameworks. And all are categorized as Embedded Programming improvement underpins on 8087 (Singh, Harkishen. 2015).

2.6.7.1 The 8087 Microcontroller

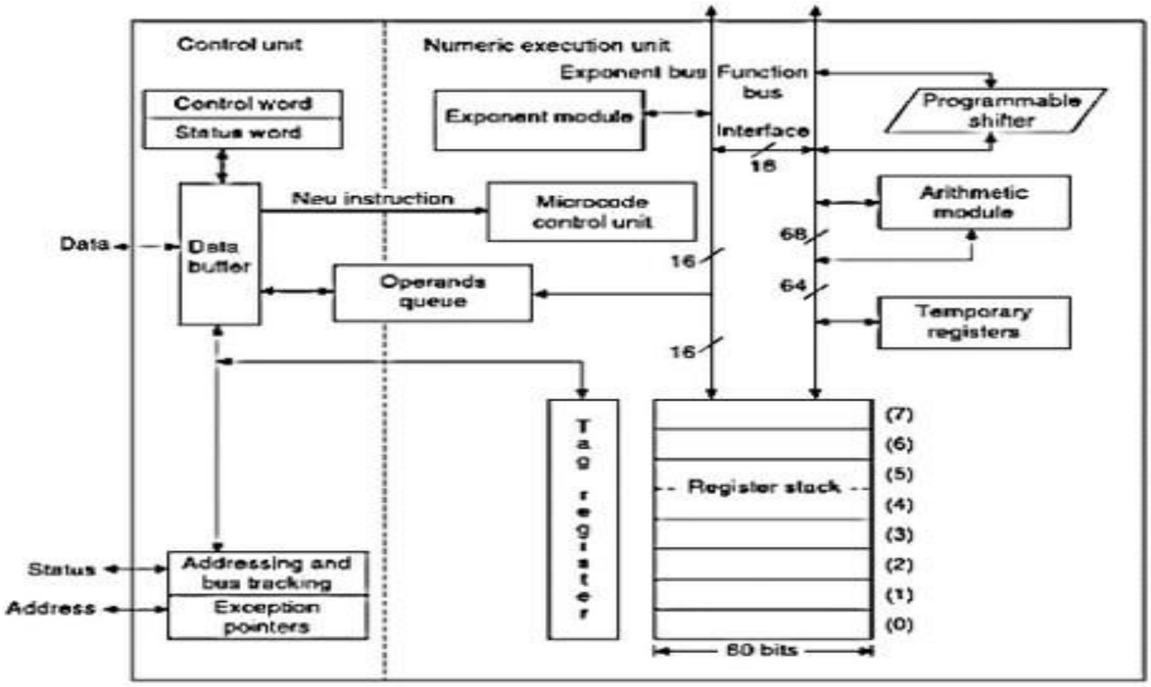
This section gives a brief overview of the hardware that will be utilized. It first reviews the 8087 microcontroller (uC) block diagram, the memory organization and elaborates a little bit on the special function registers, especially those used in interrupts and the serial communication. This is because their relevant peripherals are used in the process of emulation.

2.6.7.2 8087 Microcontroller review

The 8087 is an 8-bit microcontroller originally developed by Intel. There are currently hundreds of derivatives produced by dozens of chip manufacturers. The 8087 family is widely used today for embedded control applications (Drozd, J., 2017). The 8087 is an 8-bit micro-controller. The basic 8087 chip includes a number of peripheral I/O devices including two (or three) Timer / Counters, 8-bit I/O ports, and a 11 ART as shown in Figure 18.

Figure 17

8087 Block Diagrams



2.6.7.3 Why Embedded Systems

Embedded microcontrollers tire is used extensively in robotics and industry. In this application, many specific tasks might be distributed among a large number of controllers in one system. Communications between each controller and a central would enable in format ion to be processed by the central computer, or to be passed around to other controllers in the system. A special application that microcontrollers arc well suited for is data logging. Stick one of these chips out in the middle of a corn Held or up in a balloon, and monitor and record environmental parameters (temperature, humidity and rain) Small size, low power consumption, and flexibility make these devices ideal for unattended data monitoring and recording. The automotive market is the most important driving force in the microcontroller market. Several microcontroller families were developed specifically for automotive applications and were subsequently modified to serve other embedded applications (Guerra, R. (2013).

2.6.7.4 Why 8087 Microcontrollers

The Intel 8087 is Harvard architecture, single chip microcontroller (pC) which was developed by Intel in 1980 for use in embedded systems (Zhai, 2014). Intel's original versions were popular in the 1980s and early 1990s, but has today largely been superseded by a vast range of faster and/or functionally enhanced 8087-compatible devices manufactured by more than 20 independent manufacturers including Atmel, Infineon Technologies (formerly Siemens AG), Maxim Integrated Products (via its Dallas Semiconductor subsidiary), NXP (formerly Philips Semiconductor). Intel's official designation for the 8087 family of uCs is MCS. Intel's original 8087 family was developed using NMOS technology, but later versions, identified by a letter "C" in their name, e.g. 80C5I. Used CMOS technology and were less power-hungry than their NMOS predecessors - this made them eminently more suitable for battery-powered devices (Tao & Dubrova, 2017).

2.6.7.5 8087 Important Features

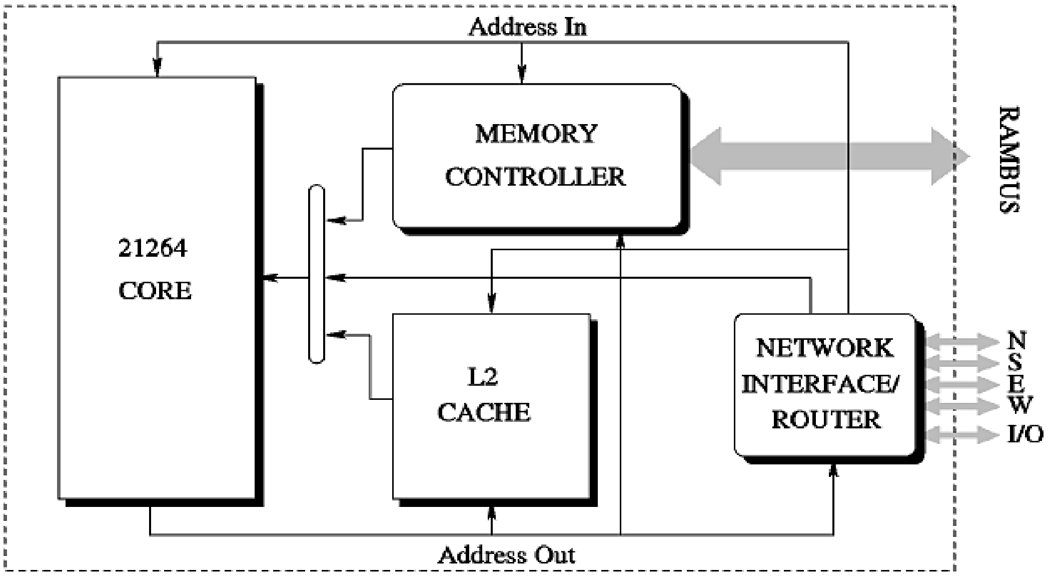
It provides many Functions (CPU, RAM, ROM, l/O, interrupt logic, timer, etc.) in a single package. 8-bit data bus - It can access 8 bits of data in one operation (hence it is an 8-bit microcontroller).16-bit address bus - It can access 2 I (> memory locations - 64 kB each of RAM and ROM. On-chip RAM - 128 bytes ("Data Memory"). On-chip ROM - -I kB ("Program Memory"). Four-byte bi-directional input/output port. UART (serial port), two 16-bit Counter/timers. I two-level interrupt priority Power saving mode (Fernández Caballero, M. A., 2017).

A particularly useful feature of the 8087 core is the inclusion of a Boolean processing engine which allows bit-level Boolean logic operations to be carried out directly and efficiently on internal registers and RAM (Taraate, V., 2017). This feature helped to cement the 8087's popularity in industrial control applications. Another valued feature is that it has four separate register sets, which can be used to greatly reduce interrupt latency compared to the more common method of storing interrupt context on a slack. The 8087 UARTs make it simple to use the chip as a serial communications interface. External pins can be configured to connect to internal shift registers in a variety of ways, and the internal timers can also be used, allowing serial communications in a number of modes, both synchronous and asynchronous. Some modes allow communications with no external components. Once a UART - and a timer, if necessary, have been configured, the programmer needs only to write a simple interrupt routine to refill the 'send' shift register whenever the last bit is shifted out by the UART and/or empty the full 'receive' shift register (Pedersen, M. H. R., 2017).

The main program then performs serial reads and writes simply by reading and writing 8-bit data to slacks. 8087 based microcontrollers typically include one or two UARTs, two or three times, 128 or 256 bytes of internal data RAM (16 bytes of which are bit-addressable), up to 128 bytes of I/O, 512 bytes to 64 kB of internal program memory, and sometimes a quantity of extended data RAM (F.RAM) located in the external data space (White, B., et al, 2017).

Figure 18

A 8087 Bus Architecture



The original 8087 core ran at 12 clock cycles per machine cycle, with most instructions executing in one or two machine cycles. With a 12 Mil/, clock frequency, the 8087 could thus execute one million one-cycle instructions per second or 500,000 two-cycle instructions per second. Enhanced 8087 cores are now commonly used which run at six, four, two, or even one clock per machine cycle, and have clock frequencies of up to 100 MHz, and are thus capable of an even greater number of instructions per second. Some Dallas and a few Atmel devices have single cycle cores. Common features included in modern 8087 based microcontrollers include built-in reset timers with brown-out detection, on-chip oscillators, self-programmable flash ROM program memory, boot-loader code in ROM, EEPROM non-volatile data storage, l-C, SPI. and USB host interfaces, PWM generators, analog comparators, A/D and D/A converters, R'I'Cs, extra counters and timers, in-circuit debugging facilities, more interrupt sources, and extra power saving modes (Shah, S., 2017).

2.6.7.6 The 8087 Programming

Several compilers are available for the 8087, most of which feature extensions that allow the programmer to specify where each variable should be stored in its six types of memory, and provide access to 8087 specific hardware features such as the multiple register banks and bit manipulation instructions. Other high-level languages such as forth, BASIC, Pascal/Object Pascal, PL/M and Modula 2 are available for the 8087, but they are less widely used than C and assembly (Chen, F. C., &Jahanshahi, M. R. et al, 2017).

2.6.7.7 The 8087 Related Processors

The 8031 was a cut down version of the original Intel 8087 that did not contain any internal program memory (ROM). To use this chip external ROM is to be added that will contain the program that the 8031 will fetch and execute. On the other hand, the 8052 was an enhanced version of the original Intel 8087 that featured 256 bytes of internal RAM instead of 128 bytes, 8 kB of ROM instead of 4 kB, and a third 16-bit tinier. The 8032 had these same features except for the internal ROM program memory. The 8052 and 8032 are largely considered to be obsolete because these features and more are included in nearly all modern 8087 based microcontrollers (Hakim, C. U., Surjati, I., &Nurwijayanti, K. N., 2017).

2.6.7.8 Memory Organization

The 8087 device has separate address spaces for program and data memory. There are separate memories for program storage, internal memory, registers, I/O functions, and external data memory. They each hold up to 64 kilobytes of instructions and data respectively. Its memory is organized in Intes and practically all its instruction deals with byte quantities. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit data memory addresses can also be generated through the Data Pointer (DPI R) register, figure 6 below shows the external code memory and data memory connected to the 8087 chips (Kamath, 2019). The 8087 has three very general types of memory. To effectively program the 8087, it is necessary to have a basic understanding of these memory types. They are: External Code Memory, External RAM and On-Chip Memory (internal memory) (Hockney & Jesshope, 2019).

The executable program code is stored in external code memory. This is often in the external EPROM. The code memory size is limited to 6-IKBytcs in a standard 8087. The code memory is read-only in normal operation and is programmed under special conditions e.g. it is a PROM or a flash RAM type of memory (Zhu, Y., Goel, M., & Bittlestone, C., 2017).

i) External RAM Data Memory

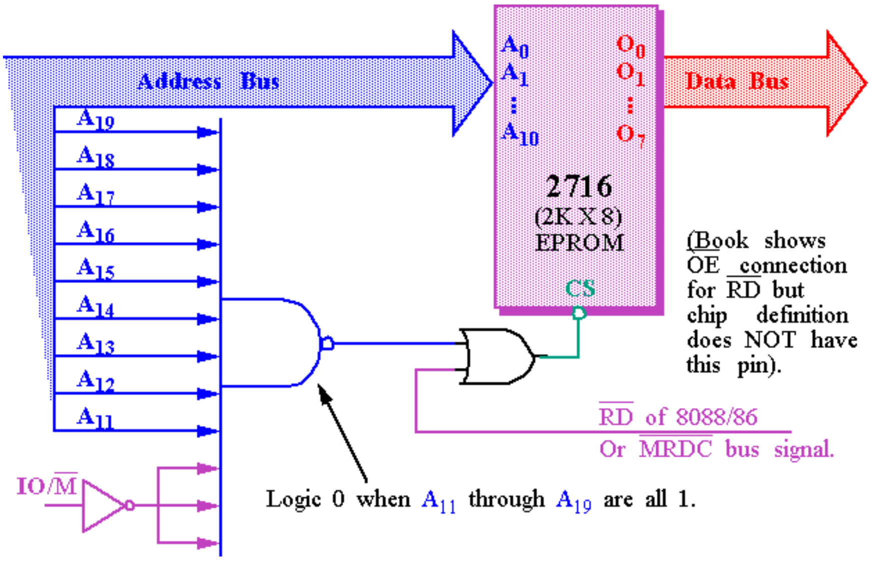
External RAM is memory that resides off-chip and often in the form of static RAM or Hash RAM. This is read-write memory and is available for storage of data. Up to 64KBytes of external RAM data memory is supported (in a standard 8087). The External Data Memory space contains all the variables, buffers and data structures that can't lit on-chip. It is principally addressed by the 16-bit Data Pointer (DPI R), although the first two general purpose register (R0. Rl) of the currently selected register bank can access a 256-byte bank of External Data Memory. The maximum size of the External Data Memory space is 64Kbytes. External data memory can only be accessed using the indirect addressing mode with the R0 or Rl (Kim, Y., 2017).

ii) Ink-null Memory

It refers to any memory that physically exists on the microcontroller itself. The 8087's on-chip memory is mapped in Figure 19 below.

Figure 19

8087's on-chip memory mapping



Address Bus
A_{19}
A_{18}
A_{17}
A_{16}
A_{15}
A_{14}
A_{13}
A_{12}
A_{11}
IO/$\overline{M}$
A_0
A_1
A_{10}
O_0
O_1
O_7
2716
(2K X 8)
EPROM
$\overline{CS}$
Data Bus
(Book shows OE connection for RD but chip definition does NOT have this pin).
$\overline{RD}$ of 8088/86
Or $\overline{MRDC}$ bus signal.
Logic 0 when A_{11} through A_{19} are all 1.

Source: Zhu, B., 2017

The internal memory is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR. The maximum size of the Internal Data Memory space is 256-byles. However, different 8087 family members integrate different amounts of on-chip memory as shown in fable 2.1 below. The register implicit, indirect and direct addressing modes can be used in different parts of the Internal Data Memory space. The Internal Data Memory space is functionally the most important data memory space. In it resides up to four banks of general-purpose registers.

iii) Register Banks (OOli to I Mi)

The lowest 32 bytes are grouped into 4 banks of 8 registers. The 8087 uses 8 general-purpose registers R0 through R7. When the instruction is executed, one of the eight registers in the selected bank is accessed. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing. All of the bytes in the Lower 128 bytes can be accessed by either direct or indirect addressing. The Upper 128 bytes can only be accessed by indirect addressing (Kilzer, et al, 2018).

The 8087 supports a special feature which allows access to bit variables. This is where individual memory bits in Internal RAM can be set or cleared. In all there are 128 bits numbered OOh to 7Ph. being bit variables any one variable can have a valuer I. A bit variable can be set with a command such as SH IB and cleared with a command such as CLR. For instance: Sli I'B I5h: sets the bit I5h (becomes I). (7/v'/5//; clears bit 15h (becomes 0). the bit addressing can also be performed on some of the SIR registers, which will be discussed later on. The Bit Memory space is used for storing bit variables and Hags. Hits can only be accessed using the bit instructions and the direct addressing model (Xekalakis et al, 2018).

iv) General Purpose RAM (30h lo 7Fh)

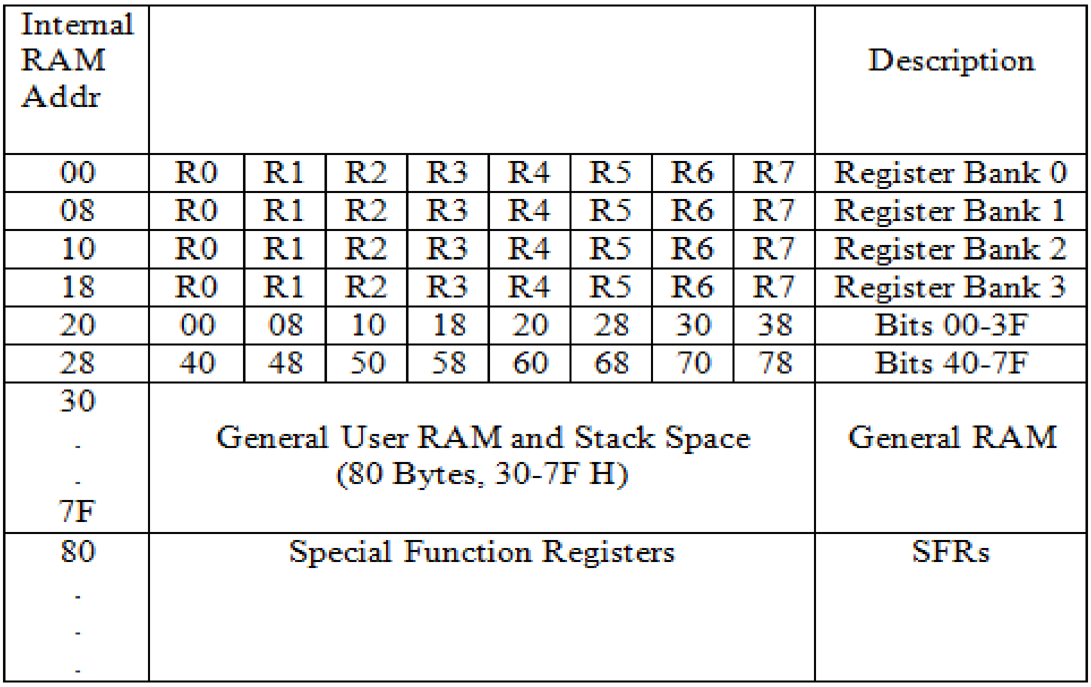
The 80 Intesoi'lntemal RAM memory are available for general-purpose data storage. Access to this area of memory is fast compared to access to the main memory and special instructions with single byte operands are used. The general-purpose RAM can be accessed using by direct or indirect addressing modes.

2.6.7.9 SFR Registers

The SI R registers are located within the Internal Memory in the address range 80h to I'l'h, as shown in figure 28 below. Not all locations within this range are defined. Each SIR has a very specific function. SIRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. Sixteen addresses in SFR space are both byte- and bit-addressable. The Special Function Register space contains all the on-chip peripheral I/O registers as well as particular registers that need program access. These registers include the Slack Pointer, the PSW and the Accumulator.

Figure 20

Special Function Register Layout



Internal RAM Addr									Description
00	R0	R1	R2	R3	R4	R5	R6	R7	Register Bank 0
08	R0	R1	R2	R3	R4	R5	R6	R7	Register Bank 1
10	R0	R1	R2	R3	R4	R5	R6	R7	Register Bank 2
18	R0	R1	R2	R3	R4	R5	R6	R7	Register Bank 3
20	00	08	10	18	20	28	30	38	Bits 00-3F
28	40	48	50	58	60	68	70	78	Bits 40-7F
30 - - 7F	General User RAM and Stack Space (80 Bytes, 30-7F H)								General RAM
80 - - -	Special Function Registers								SFRs

Source: Kalyan, 2018

The standard SOS I has four 8 bit I/O ports: PO, PI, P2 and P3. All four ports in the 80C5I are bidirectional. Each consists of a latch (Special Function Registers PO through P3), an output driver, and an input buffer. The output drivers of Ports 0 and 2, and the input buffers of Port I), are used in accesses to external memory. In such application Port 0 outputs the low byte of the external memory address, lime-multiplexed with the hue being written or read, while Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the Port 2 pins continue to emit the P2 SFR content. All the Port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features. List of all special function registers and their addresses Symbols and addresses The

Program Status (PSW) register configures register banks and user defined tag. Table Below shows bits distribution in the bit addressable register PSW 11.

The first line initializes paw nil variable by zero while (ho second line configures PSW by zero making all Hags disabled, and selecting register bank zero by making register bank selectors (RSO and RSI) zero. Here PSW.5 (10 llagt) available lor user) is used to signal previous letter display to continue displaying small characters. It has to be stated that PSW.5 (user defined (lag) is used several times for different tasks.

i. Power ('Control (PCON) Register

The Power Control (PCON) register configures power consumption mode of the chip whether it is IDLE or in POWER DOWN mode, PCON is clear initial power saving mode is not enabled and only SMOD is enabled which implements baud rate doubling. Power saving is enabled when battery powered electronic circuits is implemented for example in data loggers when used in remote areas.

ii. Timer/Counter Mode Control (TMOD) Register

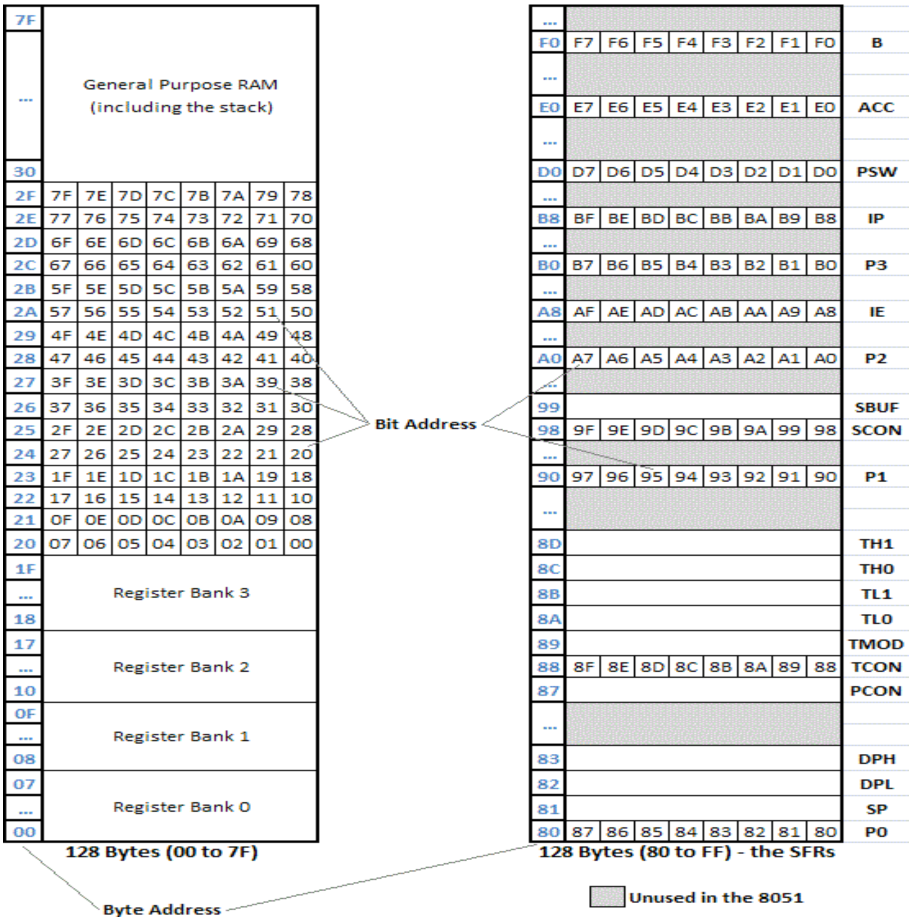
The 80C51 has two 16-bit Timer/Counter registers. Timer and timer I. Doth can be configured to operate either as timers or event counters. This is defined by Timer/Counter Mode Control TMOI.) Special function registers. In this program timer and timer, I am used lo measure the baud-rate of the host PC to start communication with the PC. After measuring the baud-rate and calculating the baud-constant using the following equation Band const256 - (crystal (12 * 16 * baud- rate) Timer/Counter Low Byte (II.) and Timer/Counter High Byte (I'll) are loaded with the baud-constant calculated to establish automatic reloading.

iii. Serial Control (SCON) Register

The Serial Control (SCON) register configures the serial port. The figure 29 below shows the different bits distribution in the bit addressable register. It has to be stated that the program developed has very limited use of special function register and only when very necessary so as to spare them for the user program developed by the developer. Also, the register used should not be modified in the user program to avoid eon diet. This means the baud rate specified should he kept. The resources used for example like the serial port should not be used by the developer as it is specified for communication with the host PC at a certain baud rate.

Figure 21

Bit Addressable Register



7F	General Purpose RAM (including the stack)							
...								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Register Bank 3							
...								
18								
17	Register Bank 2							
...								
10								
0F	Register Bank 1							
...								
08								
07	Register Bank 0							
...								
00								

128 Bytes (00 to 7F)

...	▨								
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
...	▨								
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
...	▨								
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW
...	▨								
B8	BF	BE	BD	BC	BB	BA	B9	B8	IP
...	▨								
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
...	▨								
A8	AF	AE	AD	AC	AB	AA	A9	A8	IE
...	▨								
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
...	▨								
99									SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
...	▨								
90	97	96	95	94	93	92	91	90	P1
...	▨								
8D									TH1
8C									TH0
8B									TL1
8A									TL0
89									TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87									PCON
...	▨								
83									DPH
82									DPL
81									SP
80	87	86	85	84	83	82	81	80	P0

128 Bytes (80 to FF) - the SFRs

Source: Priyanka, S. S., 2017

i) Interrupt Enable (IE) Registers

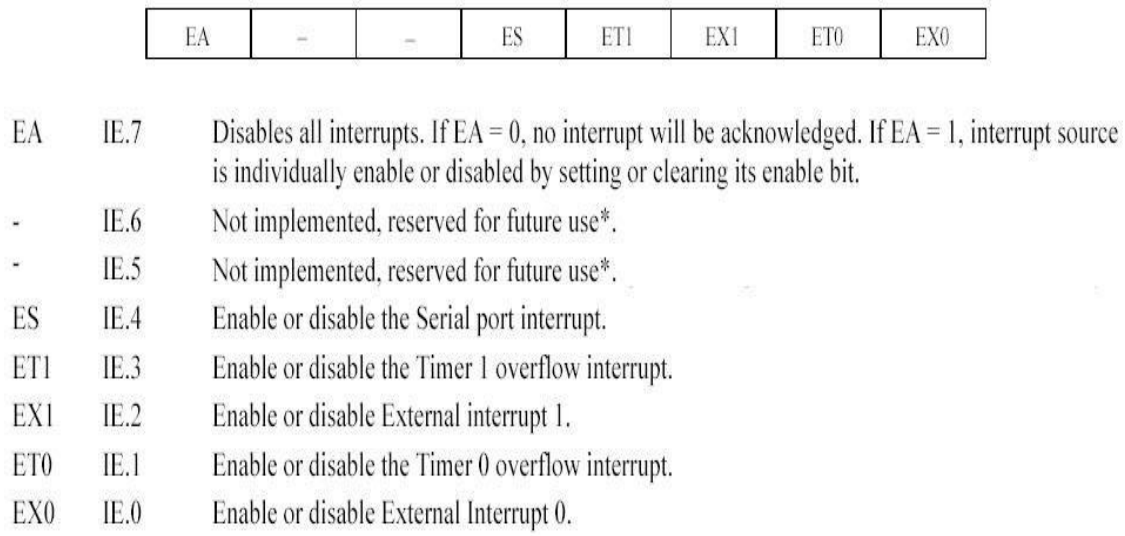
This register is responsible for enabling and disabling the interrupt. It is a bit addressable register in which EA must be set to one for enabling interrupts. The corresponding bit in this register enables particular interrupt like timer, external and serial inputs. In the below IE register, bit corresponding to 1 activates the interrupt and 0 disables the interrupt (Praveen, B. L. N., 2017).

ii) Interrupt Priority (IP) Register

All interrupts are internally prioritized as follows INTO. TEO, IN'I'I, TF1 and lastly the serial port interrupt (Rl or TI). But any interrupt source can be individually programmed to one of two priority levels to achieve some kind of flexibility. These interrupts are prioritized by the bit addressable Interrupt Priority (IP) register (Busaba, F. Y., Gschwind, M. K., & Schwarz, E. M., 2017).

Figure 22

Enable (IE) Registers



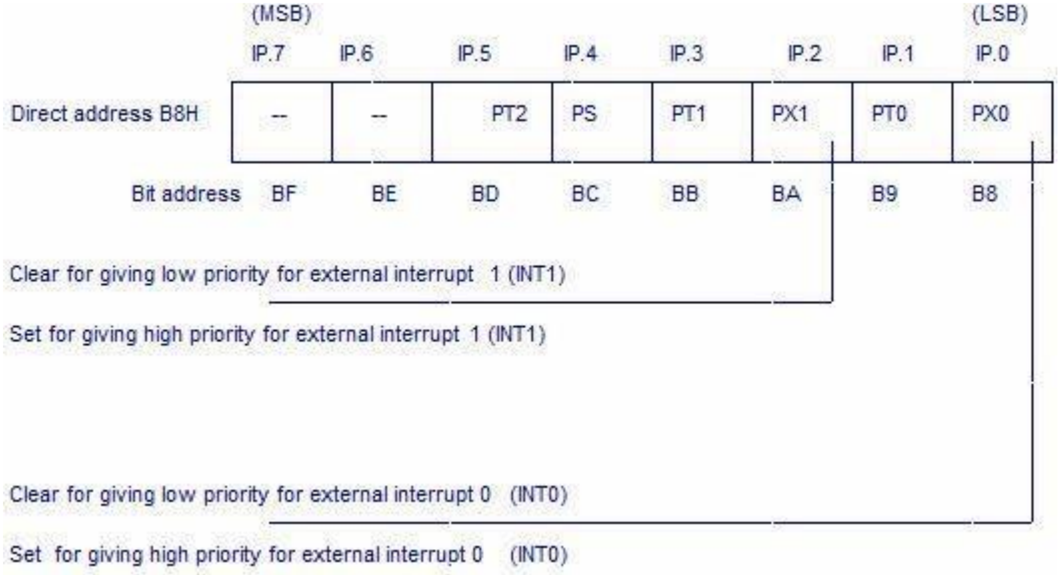
EA	–	–	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, interrupt source is individually enable or disabled by setting or clearing its enable bit.
-	IE.6	Not implemented, reserved for future use*.
-	IE.5	Not implemented, reserved for future use*.
ES	IE.4	Enable or disable the Serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

Source: Praveen, B. L. N., 2017

Figure 23

Interrupt Priority (IP) register


(MSB)
(LSB)
IP.7
IP.6
IP.5
IP.4
IP.3
IP.2
IP.1
IP.0
Direct address B8H
--
--
PT2
PS
PT1
PX1
PT0
PX0
Bit address
BF
BE
BD
BC
BB
BA
B9
B8
Clear for giving low priority for external interrupt 1 (INT1)
Set for giving high priority for external interrupt 1 (INT1)
Clear for giving low priority for external interrupt 0 (INT0)
Set for giving high priority for external interrupt 0 (INT0)

Source: Praveen, B. L. N., (2017)

Since all interrupts are already prioritized by hardware, there might be no need for making some priority changes by software. clra clear Aniov //>, are\cl interrupt priority register the above two lines enable the hardware priority as they clear all IP bits.

2.6.7.10 8087 Addressing Modes

All members of the 8087 family execute the same instruction set which is optimized for 8- bit. It provides a variety of addressing modes for different memory types. The definitions of these modes are as follows: -

v) **Direct Addressing**

The address of the operand is specified by an 8-bil Held in the instruction (only for internal RAM and SERs).

vi) Indirect Addressing

The address of the operand is specified by a register. Used for both internal and external memories. RO or Rl arc used for addressing.

vii) Indexed Addressing

The address of the operand is specified by DPTR or PC. It is used only for program area for reading look-up table.

2.6.7.11 Register Instruction:

Contents of RO through R7 in the current register bank can be accessed by these instructions.

i) Register-specific instruction:

Contents of certain register like Ace and DPTR are accessed by these instructions.

ii) Immediate constants:

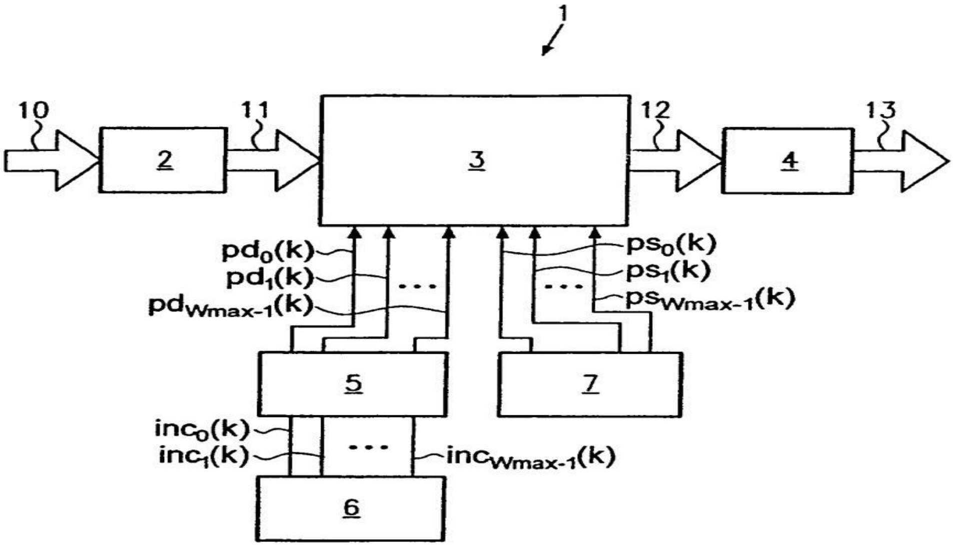
The value following the opcode (//constant) is operated with the Acc or DPTR. The programming instructions deals with arithmetic operations, logical operations, and data transfer operations (Mano, M. M., 2017). Also, jumping and calling operations. Details of the information areas available in instruction set.

2.6.7.12 Circuit Hardware

The hardware to be used is a simple circuit consisting of an 8087 microcontroller beside some necessary items as shown in Figure 12 below. The different items are as follows:- The 8087 microcontroller, dell tied by UI,7411('TOO dual input NAND gates, defined by U2 (for making RAM like, R( )M containing the code as the PSEN is active when fetching the opcode), 7411(I 573 octal D latch, defined by 11 is for latching the lower address byte to the memories as lower address byte and data are multiplexed on porlO, 27C512 normal or 301 512 Hash ROM. defined by U4 (pin compatible ICs, even different si/e are also possible, depending on user program size), 6264 static RAM (64Kbit), defined by D5 (pin compatible ICs, even different size are also possible, depending on user data size), 24COI IIC protocol Ef PROM, defined by U6, PCI 8583 real lime clock, defined by 117**,** 74 I IC'f 10 triple NAND gate, defined byU8 (for decoding and Hash writing), 7805 voltage regulator, defined by 110**,** 11.006MHz. XI Al for the microcontroller, 32.768 KHz for real time clock, BC857and BC847 transistors for buffering TXD and RXD signal to the serial port, defined Ql and Q2, Some resistors, capacitors, diodes and connectors (Hussein, 2009). The circuit will be built on a double side Printed Circuit Board (PCB). The schematic drawing of the circuit is shown in Figure 24 below.

Figure 24

Circuit Hardware



Source: Awad, et al, (2017)

2.7 Research Gaps

From this review, it is clear that there is a pressing need for a smart parking solution that fully addresses efficiency and security, with the existing problem being exacerbated as time passes. The review found that there were no previous studies published in the literature related to the car packing systems that took into account the views of stakeholders. There is a general trend among scholars to use WSNs due to the aforementioned reasons and, in the last five years, there has been a trend to utilize the IoT principle. This review has identified a number of gaps in the literature. It appears that there is no consensus on the best technological security solution package, as each possible solution has deficiencies in one area or another.

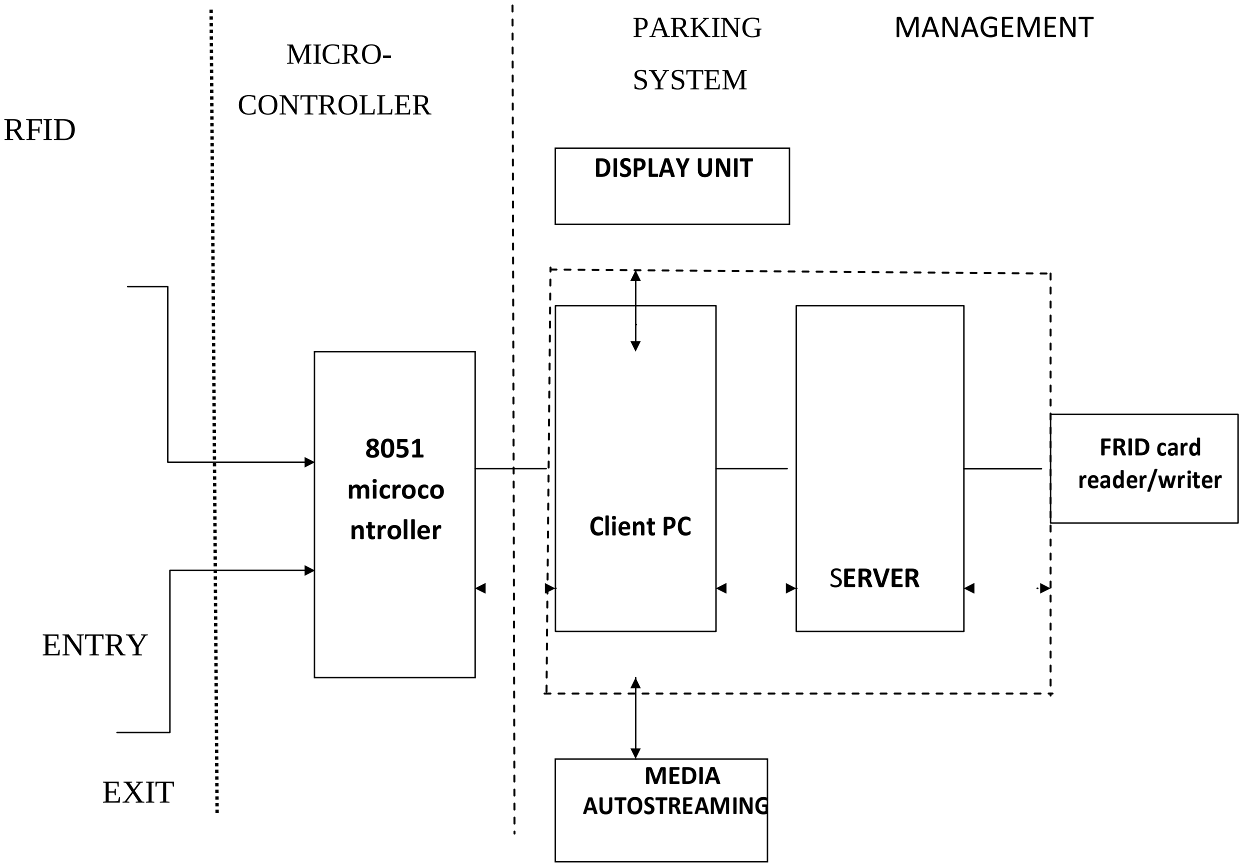
It can be noted specifically from survey that each solution has different technological requirements and utilize different methods to facilitate parking services for the user. Each solution has advantages and disadvantages in terms of the following criteria: Security, reliability, scalability, accuracy, communication type, circuit complexity, method of operation, and ease of installation/usability of the Model. Many types of research have influenced the way existing systems have been designed, measuring their performance and quality from the perspective of the designer, engineer, or developer, and not the stakeholders. From a smart parking security perspective, there is an inadequate research base to inform fully how to design and evaluate efficient parking systems; therefore, this issue requires an all integrated package which clusters all independent solution into a single working model through interfacing. Therefore, my study will seek to be an ammonizing technology that will integrate all the disparate solutions that have not individually provided a secure, more focused and a reliable solution that is efficient, effective, reliable, scalable, and accurate.

2.8 Conceptual Framework

The framework for efficiency in car packing while integrating media streaming is as shown below. The layout of the framework is as shown below in Figure 25 below.

Figure 25

RFID with Media Integration Framework



Source: Author (2023)

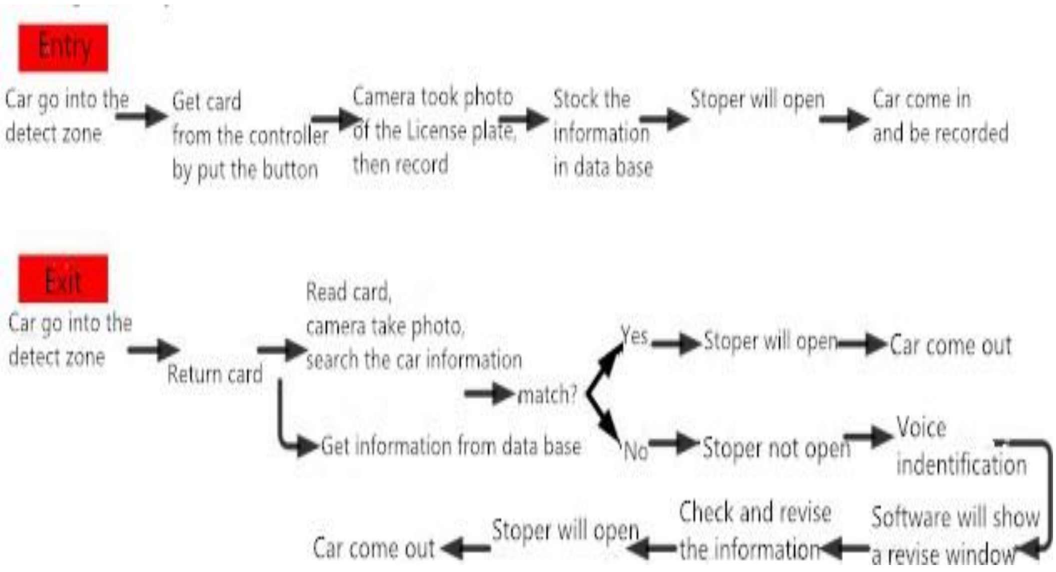
The process starts from the gate when a car entrant takes the RFID card and swipes through the reader which in turn relays the signal to the microcontroller and once the microcontroller processes the signal it uses a system of relay to trigger the stopper to let the vehicle enter the parking center. The micro-controller simultaneously based on the instructions pre-programmed will convey signals to the server which directs the media auto streaming to begin. When the taking of photos is being done the vehicle is led to the parking center by being shown on the screen interface the parking space allocated as per the time and schedule. At the time of entry the cameras are focused on the vehicle registration number which are send to the database of the parking management system for retrieval in case of any audit.

Data flow logic

The logic of the process flow from request of information by huffing the RFID tag over the reader and subsequently relay of information to the opening of the gate and letter exit of the vehicle from the parking center is a shown in Figure 26 below.

Figure 26

RFID with Media Integration Framework data flow


Entry
Car go into the detect zone
Get card from the controller by put the button
Camera took photo of the License plate, then record
Stock the information in data base
Stoper will open
Car come in and be recorded
Exit
Car go into the detect zone
Return card
Read card, camera take photo, search the car information
match?
Get information from data base
Yes
Stoper will open
Car come out
No
Stoper not open
Voice indentification
Software will show a revise window
Check and revise the information
Stoper will open
Car come out

Source: Author (2023)

CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

This chapter explains the research methodologies that the study will concentrate on in order to realize the micro-controller based online integrated media auto stream model for car parking. The study used experimental science approach. This was used to come up with an experimental integrated smart parking facility that is an abstract of existing legacy approaches. It will be an ammonizing platform, cheap, scalable and realistic. The simulation of the real system is meant to be cheaper and less complex than the system it models, therefore this will allow the researcher to better understand and utilize the model to perform experiments that might not be better carried out in a real environment because of cost and accessibility.

The experimental science approach will also be used to come up with a build of a working software system that will come up with a Parking management system (PMS) software that will be used to allocate and de-allocate parking lots. The software system so envisioned will be new and will be an innovational approach towards coming up with an all integrated system that will enable smart parking thereby achieving the proposed concept.

3.2 Research Design

This study will adopt a proof of concept research methodology. Proof of concept is used to test various aspects of the intended experiment without attempting to exactly recreate the engineering process. Thus, the prototype can be used to prove viability of a car parking model with embedded 8087 micro-controller. These types of models are often used to identify which design options will not work as well as where further development and testing is needed will be addressed.

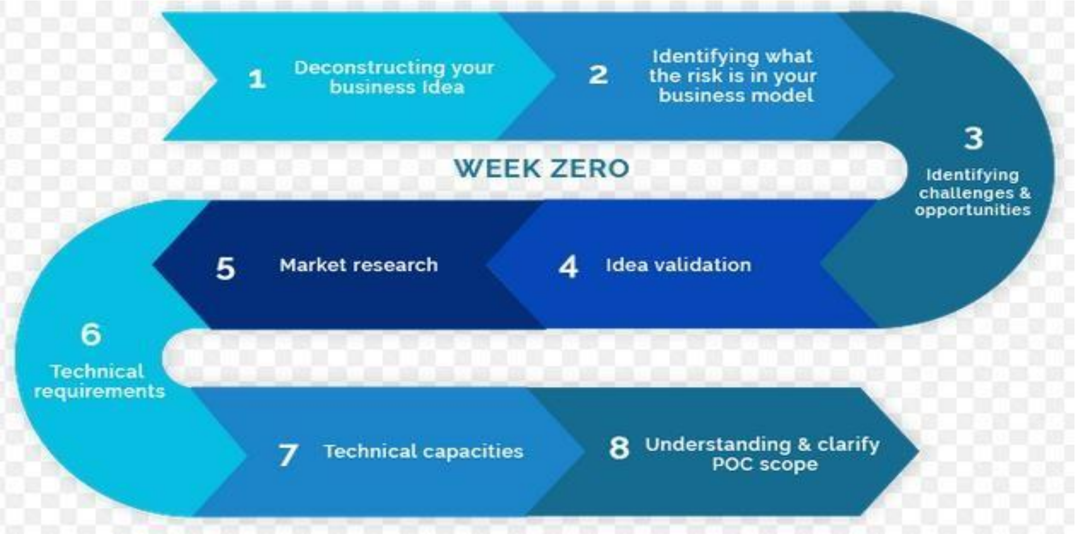
A proof of concept is generally applied early in the system development cycle. It is used to validate technical feasibility, helps identify potential stumbling blocks, identifies what a platform can or can't provide, and helps determine the scope and level of customization necessary to complete the project. It can also help identify performance issues. Here, we assemble many of our components and solutions in a "composite" fashion. We are re-using for example services, and functions from other components. This re-use requires integration points. It is these interfacing points in our overall "context" that we are vetting with the experiment. Using the proof of concept methodology, we are able to examine some RFID and micro-controller based platform implementation success factors along with factors that impact overall scope and estimates of effort for the proposed platform. Based on the following; Design of an architectural model and the Implementation of the prototype.

3.2.1 Proof of Concept

This proof of concept was designed and applied early in the system development cycle. It is used to validate technical feasibility, identify potential stumbling blocks, identify what the RFID based microcontroller interfaced model for car packing can or can't provide, and help determine the scope and level of customization necessary to complete deployment of the proposed model. As captured in chapter 5.In prototype.

Figure 28

Proof of Concept Phase


1 Deconstructing your business Idea
2 Identifying what the risk is in your business model
3 Identifying challenges & opportunities
WEEK ZERO
4 Idea validation
5 Market research
6 Technical requirements
7 Technical capacities
8 Understanding & clarify POC scope

Source: Nintech (2013)

This step will involve assembling of components, services, functions for the proposed model. Several functions can also be re-used via various integration points. Using the proof of concept methodology, we are able to examine some micro-controller based platform implementation success factors along with factors that impact overall scope and estimates of effort for the proposed platform.

3.2.2 Data Collection

This study utilized secondary research sources such as existing parking technologies, standard documents, conference journals and published academic papers to investigate the research questions outlined above. In order to summarize the current state of knowledge in the area to serve as a background for the study, a literature review of the development and implementation of various micro-controller architecture has been carried out.

3.3 Model Development

The development of the parking model with real-time media auto stream involved extensive research and testing. Through proof of concept and experimental design this focused on leveraging existing technologies, such as image-processing and integrating microcontrollers with sensors, to design a model that could effectively manage parking spaces in real-time. This is extensively covered in chapter 4 and 5 the model development process consisted of the following steps:

Identifying the problem: The first step in developing the model was identifying the problem of inefficient parking management due to manual checks and lack of real-time audit and tracking.

Conducting literature review: Researchers conducted a thorough literature review to understand existing solutions and technologies related to parking management and real-time tracking.

Designing the model: Based on the literature review and the identified problem, the model was designed that integrated various technologies, which was and embedded system using microcontrollers, sensors and image-processing techniques. Through experimental design.

Implementing the model: Once the design was finalized, the experiment was implemented with the the model by integrating the necessary sensors and software components.

The model is such that it is apportioned sections as per activity laid out. The activity functional sections are as listed below;-

i. Entrance Control Part-Automatic car, vehicle detector, ARM controller, remote reader, automatic card machine.

ii. Exit Control Part-Automatic car, vehicle detector, ARM controller, remote reader, hard, time card reader.

iii. Image contrast section-Camera, video capture card, image contrast module

iv. Guard-Charge management computer, RS485 communication card, parking management software, time card reader, switches, etc.

v. Management system- Central server, RS485 communication card, card management software

vi. The model development was addressing the type of technology used reliability, communication, efficiency, and circuit complexity.

3.4 Ethical Consideration

All ethical concerns pertaining user data protection were considered. All the data collected was treated with outmost protection and care. The information obtained and participants participating in the experiment were respected and data obtained was only used for the purpose of this study. In addition to, the researcher sought formal authorization to carry out research by obtaining a letter of introduction from Kabarak University Institute of Postgraduate Studies and a research permit from the National Commission for Science and Technology (NACOSTI).

CHAPTER FOUR

MODEL DEVELOPMENT AND RESULTS

4.1 Introduction

The chapter discusses the model design, specifically the hardware components considered for model development, the logical and architectural design, simulation using Quemu and the overall model configuration.

4.1.1 Design of the Media Autostream Model for Car Parking

The outlay of the model was considered using a high level logical diagrammatic representation of how the actual model operated with the RFID, camera and sensors working together. On the other hand, the architectural model simulation showed the high level view of how each of the components interrelated and interfaced each other into one whole running model using Qemu. Also, Qemu model simulation followed were the simulation was done through configuration of the paramenters needed for the overall model operation.

4.1.2 Hardware Requirements

The following hardware devices were used in the experiment.

ii. Raspberry Pi Camera V2 8MP – this was used for taking pictures for number plate processing for both incoming and outgoing vehicles.

iii. RFID Card Reader/Detector Module Kit – This was used to authenticate if a user is allowed to the car parking garage.

iv. LCD 16x2 Blue backlight /w I2C – This was used to output the garage status.

v. Micro Servo motor 180 Degree – This was used to open and close the gate as well move the camera to the desired position.

vi. 9V Battery Snap Power Cable – This was used to supply power to the bread board module.

vii. Breadboard Power Supply Module – This was used to supply power to the whole car parking system.

viii. IR Infrared sensor module - Used to detect presence of cars, whether incoming, outgoing and parked cars.

ix. Micro Servo bracket Pan/Tilt Kit - Used to move the camera to the desired position with help from the cameras

x. RFID Card 13.56Mhz -Used as an authentication identification tool.

xi. Raspberry Pi 3 Model B+ - Its the brain of the system.

   a. Contains OS
   b. Co-ordinates the system working with the Arduino.
   c. As well as storing the SQLITE database for the recorded number plate as well as storing the vehicle pictures for later analysis as well as entrance and depature time on which day.

xii. Jumper Wires - Used to transfer power to the system components.

xiii. 9V 3A Raspberry Pi 3 Power Charger Adapter - Used to power the raspberry pi

xiv. Arduino - Used to control the system with hand with the raspberry pi through serial communications.

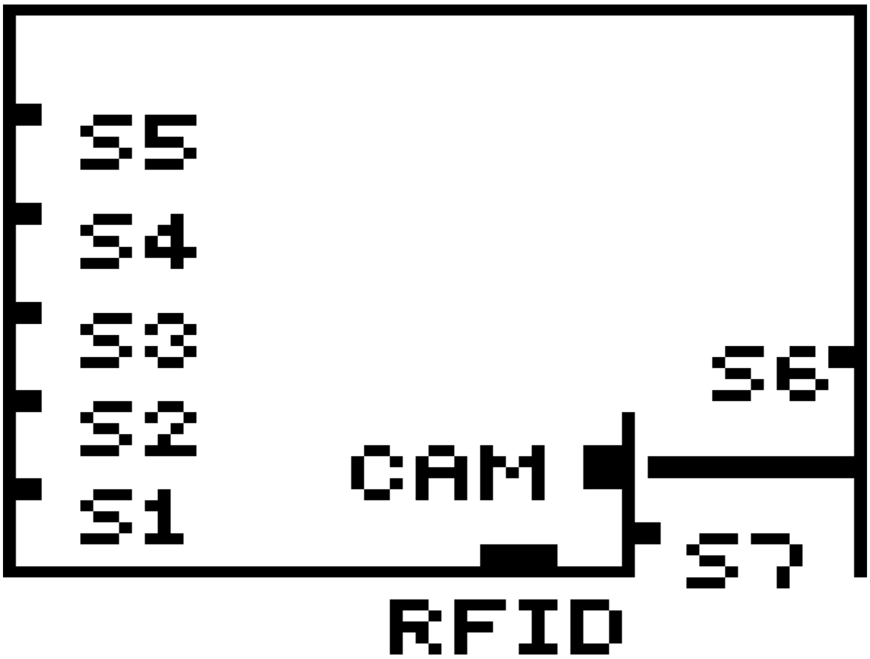
xv. Micro SD card. - Used to store the raspberry pi operating system.

x. Car Toys - Used to simulate real cars.

4.1.3 Logical Model Design of RFID, Sensors and Camera

Figure 29

Logical Design of the RFID, Sensors and Camera



Source: Author (2023)

i. S1 - sensor1
ii. S2 - sensor2
iii. S3 - sensor3
iv. S4 - sensor4
v. S5 - sensor5
vi. S6 - sensor6
vii. S7 - sensor7
viii. G1 - gate
ix. CAM - Camera
x. RFID - RFID tag

When a car approaches the gate, it turns sensor 7 on alerting the system on an incoming car. The system then checks if there are available slots by checking through sensor one to sensor five of any available cars. This is done by checking if any of the sensors if high which indicates a car presence or low which indicating the empty parking slots. If any parking slots available, the camera turns towards the incoming car, takes the picture of the incoming vehicle, processes the number plate as well as storing the encrypted car plate numbers. Then the system prompts the user to produce a RFID card for authentication.

i) The RFID Card Tag number is checked and if authenticated the gate opens and upon entrance the car will be guided towards the vacant slot by the system. The LCD will tell which slot is vacant.

ii) If the parking garage is full the systems outputs their parking slot are not available to the LCD and does not allow the vehicle owner to enter the parking garage.

iii) On arrival to its parking location and on sensor activation a time count is done which stops when the sensor turns low indication the car is on its exit way.

iv) On reaching the gate, sensor six is turned on and indicates to the system the car is on exit. This turns the camera to capture the exiting vehicle and stores it encrypted on the sqlite database, prompts the user for his or her RFID card for authentication.

v) If authenticated the gate opens for the outgoing car and a free slot is advertised on the LCD.

4.1.4 Architectural Model Design for the RFID Based Microcontroller Car Park

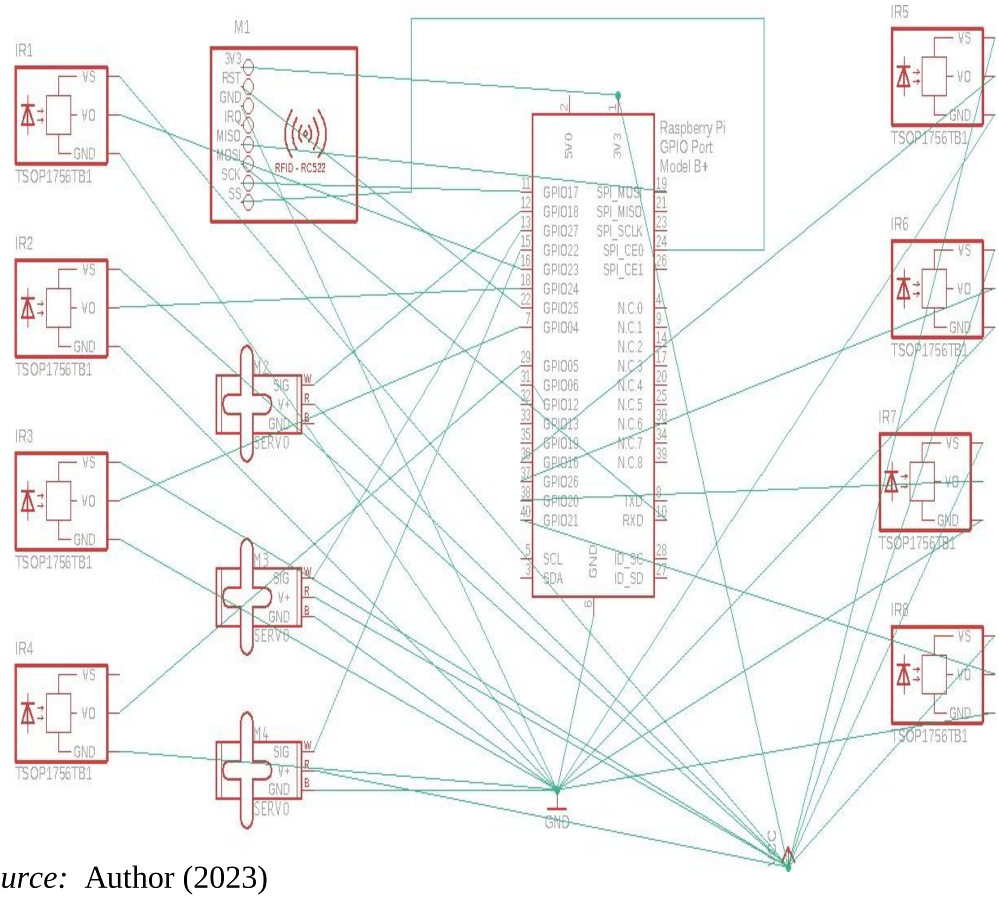
This research followed a proof of concept research design. This phase involved validation of user needs, technical feasibility, identifying potential stumbling blocks, identifying what the RFID based microcontroller interfaced model for car packing would or would not provide. This helped determine the scope and level of customization necessary so as to complete deploy men of the proposed model.

The design was created using a Qemu system running Raspbian on Windows 10. The source code provided for in this simulation deviated insignificantly with results even as much as there were various input sources and from real data.

The figure below shows design of the carpark structure that was implemented on the Qemu system.

Figure 30

Design of the Car Park system later implemented on Simuli



Source: Author (2023)

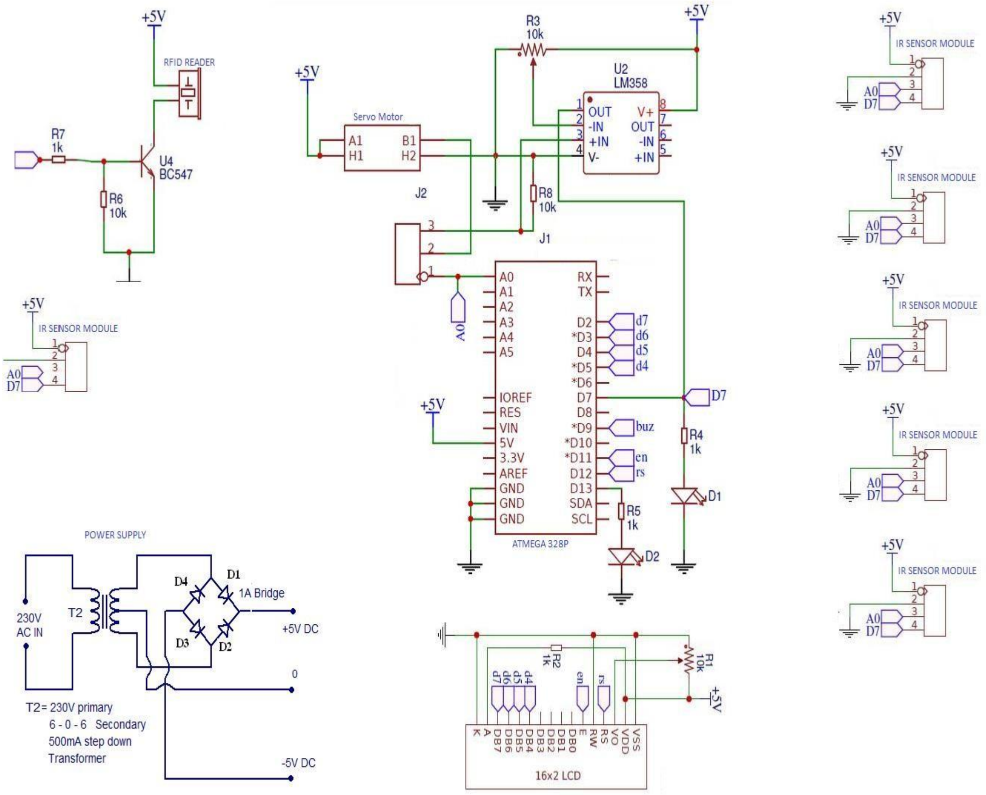
The Figure 30 shows the simuli simulation for bus interconnection of car park structure and components. It illustrates the general overview on how the different components that make up the model interface with each other.

4.1.4.1 Design of the Simulation Parameters on Simuli

The simulation flow diagram involved a precise description of functionalities and data flow from inception of car entry into the car park area, entry/exit outputs, checking of vacant car spaces, capturing of multimedia into the data repository, RFID tagging and recording, below in figure 31 indicates the circuit components and the sumulated circuit of the experiment.

Figure 31

Simuli Circuit Diagram

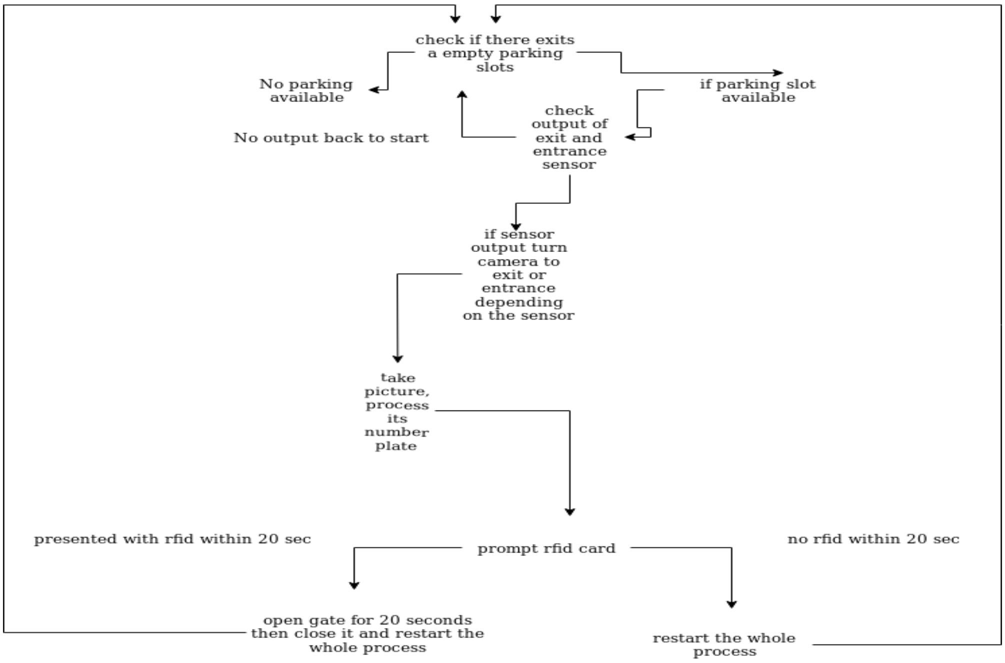


Source: Author (2023)

The figure below shows design of the simulation flow which the experiment followed on the Qemu system.

Figure 32

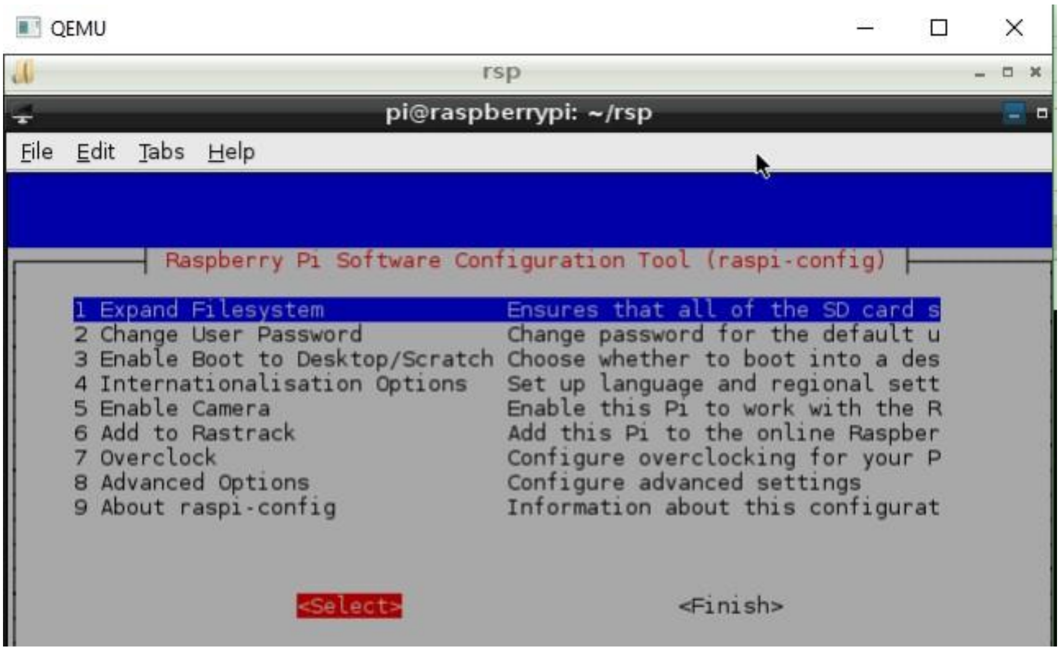
The Simulation Flow Diagram


check if there exits a empty parking slots
No parking available
if parking slot available
No output back to start
check output of exit and entrance sensor
if sensor output turn camera to exit or entrance depending on the sensor
take picture, process its number plate
presented with rfid within 20 sec
prompt rfid card
no rfid within 20 sec
open gate for 20 seconds then close it and restart the whole process
restart the whole process

This involved selection of the simulation parameters on the Qemu system. The figure below shows the selection process of the parameters for this research.

Figure 32

Simulation using Qemu System

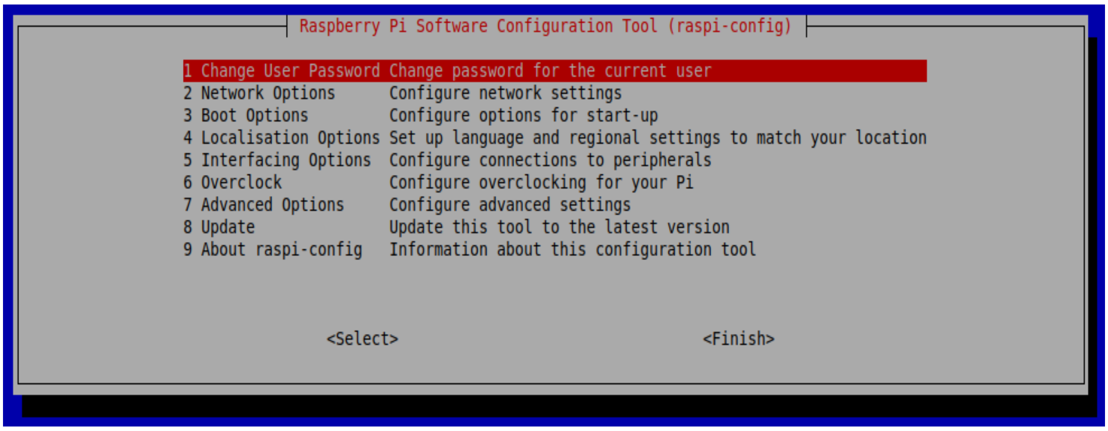


```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem                Ensures that all of the SD card s
2 Change User Password             Change password for the default u
3 Enable Boot to Desktop/Scratch   Choose whether to boot into a des
4 Internationalisation Options     Set up language and regional sett
5 Enable Camera                    Enable this Pi to work with the R
6 Add to Rastrack                  Add this Pi to the online Raspber
7 Overclock                        Configure overclocking for your P
8 Advanced Options                 Configure advanced settings
9 About raspi-config               Information about this configurat
```

Further, to enable negligible results deviation, there was need to align this simulation within real-time events that are experienced in a car-parking lot system. Thus in comparison and to be in tandem to the real raspi-config which could be used to configure the LCD, localization and interfacing options were configured to align the camera and the RFID as to be used in this project. The figure below shows configuration and alignment of various options within the Raspberry Pi.

Figure 33

Simulation configuration and alignment on the Qemu system



```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool


                <Select>                                  <Finish>
```

4.1.4.2 Readjustment of Parameters on Qemu

The following re-adjustments were done on the simulator since it failed to install RPi.GPIO pins as the module is only for real raspberry pi. It was also unable to connect sensors for example camera, RFID module and ir modules. There was also lack of open source free raspberry simulators. The following were the steps followed;

i) Scripts were modified so as to run on the simulator.

ii) Picamera, RPi.GPIO, MFRC522 modules were re-written to run on our simulated environment

iii) Qemu simulator was used running Raspbian

4.1.5 The RFID Microcontroller Based Model Simulation

The following modified Scripts were run on the Qemu simulator and the block diagam of Figure 34 indicating all the embedded components.

Figure 34

Simuli System Block Diagram

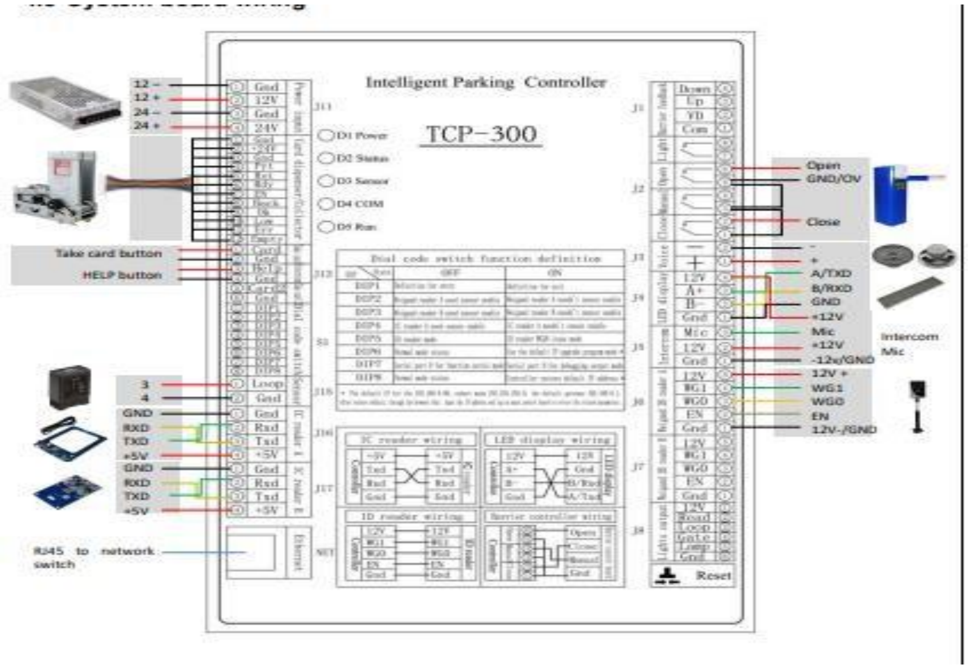


Table 4

Modified Scripts - Picamera.py

def capture(a): print 'taking pic'

When picamera is invoked, this function to take a pic is run which just prints taking picture.

MFRC522.py def MFRC522():

print 'intilize MFRC522'

Just made to invoke the initialization of MFRC522. GPIO.py

BCM=1 OUT=1 IN=1

defsetmode(a):

print 'setting gpio mode' def PWM(a,b):

pass def start(): pass

def input(a):

print 'input invoked' def setup(a,b):

#print a,' ',b pass def start():

print 'starting PWM' def cleanup():

print 'cleaning up'

4.1.5.1 Function to simulate packed sensor output

A function to simulate if a car is parked sensor output. The function returns a random value with 0 being no car and 1 being a car present. The function averages 5 input values for more certain results. If the input value of a car present averages more tha 0.4 of 1 is considered a steady input of a car present else if it falls below the car is assumed not to be there.

Table 5

Simulate Packed Sensor - sensorPin.py

```
defcar_sensors(sensorPin): status_sensor
for i in range (5): status_sensor. append (random. choice ([1,0]))
av = reduce (lambda x, y: x + y, status_sensor) / len(status_sensor) if av> 0.4:
return 1 else: return 0
```

The display value takes 2 args with the row1 value being the upper row of the i2c 16x2 lcd and row2 being the bottom row. It then sleeps for 3 seconds and clears the output before proceeding.

```
def display(row1='-> ‘, row2='->'):
try: while True:
print ("Writing to display") print row1,' ‘, row1
display.lcd_display_string (row1, 1)
display.lcd_display_string (row2, 2) time. sleep (3)
display.lcd_clear ( ) except:
Pass
```

This function takes a servo GPIO pin number and the desired angel to move the servo position. It can take several

```
defmove_camera (servoPin, desired_angle): pwm= GPIO.PWM(servoPin,50)
dc=1./18.*(desired_angle)+2
```

The check_for_outgoing_cars, check_for_incoming_cars and returns a random value either 1, 0 with 1 a car present and 0 no car.

```
defcheck_for_outgoing_cars ( ):
incoming_car=car_sensors (carsensor6) return random. Choice ([1,0])
defcheck_for_incoming_cars ( ): outgoing_car=car_sensors(carsensor7) return
random.choice([1,0])
```

The functions space_checker check if there are any empty parking slots and returns 1 if their is any and the empty parking slot else 0 and a empty unpacked_spaces list.

```
defspace_checker (): unpacked_spaces= [] cs0=car_sensors (carsensor0)
cs1=car_sensors(carsensor1) cs2=car_sensors(carsensor2)
```

4.1.5.3 Main script for emulating license plates

This is the main script used in emulating the automatic license plate recognition as well as automatic parking system with the changes made in the above script. The random module was used to emulate sensor values. This does not present the actual input from the hardware sensors such as infrared sensors.

Import all modules needed for simulation included ours modified scripts. The scripts must be in the same folder as the main Project.Py

4.1.5.4 Emulating license plate - Project.Py

Import Time Import Time Import Signal

From Pil Import Image Import Sqlite3

From Datetime Import Datetime import lcddriver import GPIO

from GPIO import * import MFRC522 import random import picamera

Define what happens during a keyboard interrupt. defend_read (signal, frame):

global continue_reading

print "Ctrl+C captured, ending read." continue_reading = False GPIO.cleanup()

Initialize the modified display module as well as GPIO, database and the RFID reader.

display = lcddriver.lcd() GPIO.setmode(GPIO.BCM) continue_reading = True signal.signal(signal.SIGINT, end_read) MIFAREReader = MFRC522.MFRC522()

dbconn = sqlite3.connect('plates.sqlite') dbcur = dbconn.cursor()

Initialize GPIO sensor pins to be used as IR infrared sensor and servo pins. carsensor0=17

carsensor1=40 carsensor2=38 carsensor3=26

carsensor4=18 carsensor5=7 carsensor6=12 carsensor7=16 servo1=20 servo2=21 servo_gate=26

```
GPIO.setup(servo1,GPIO.OUT)GPIO.setup(servo2,GPIO.OUT)
GPIO.setup(servo_gate,GPIO.OUT) pwmgate=GPIO.PWM(servo_gate,50) while
True: full_packed,empty_slots=space_checker() if full_packed==1:
display('Parking Available',str(len(empty_slots))+' slots available') else:
display('Parking Fully Parked','No available slots') if check_for_outgoing_cars():
display('Outgoing Car','Processing...') move_camera_right( )
take_pic ( )
display('Place your Card ','for processing...') if rfidreader( ):
gate_opener() else:
pass display('Restaring services ',' in a second...') time.sleep(1)
if check_for_incoming_cars(): display('Incoming Car','Processing...')
move_camera_left() take_pic()
display('Place your Card ','for processing...') if rfidreader():
gate_opener() else:
pass display('Restaring services ',' in a second...') time.sleep(1)
```

While the program is running, it checks first checks if their exists empty parking slots and displays to the lcd necessary information,then checks the inputs of the check_for_outgoing_cars followed by check_for_incoming_cars input to check if there are cars at the entrance or exit of the parking system.

Then depending on the output of the the exit and entrance sensors, the camera is adjusted accordingly to either position with the vehicle, takes a picture for the number plate to be processed.

Then the picture number plate text is processed and then stored at the database as well as the date time.

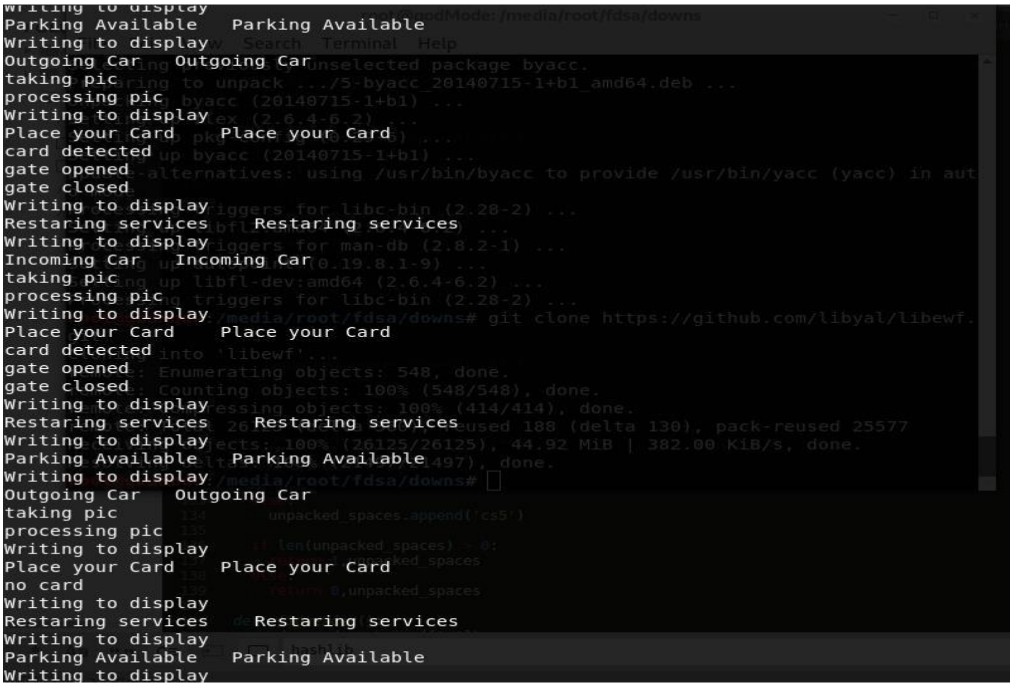
After automatic car number place licence recognition, the user is the prompted to place their RFID card, on detection off the card the gate is opened for 20 seconds to allow the car to pass and then jumps right back to the beginning and starts all the process again.

4.1.5.5 Simulation Output Project.Py

The simulation was successfully implemented. The Figure 35 below shows a sample output reading of Project.Py

Figure 35

Simulation Output Results



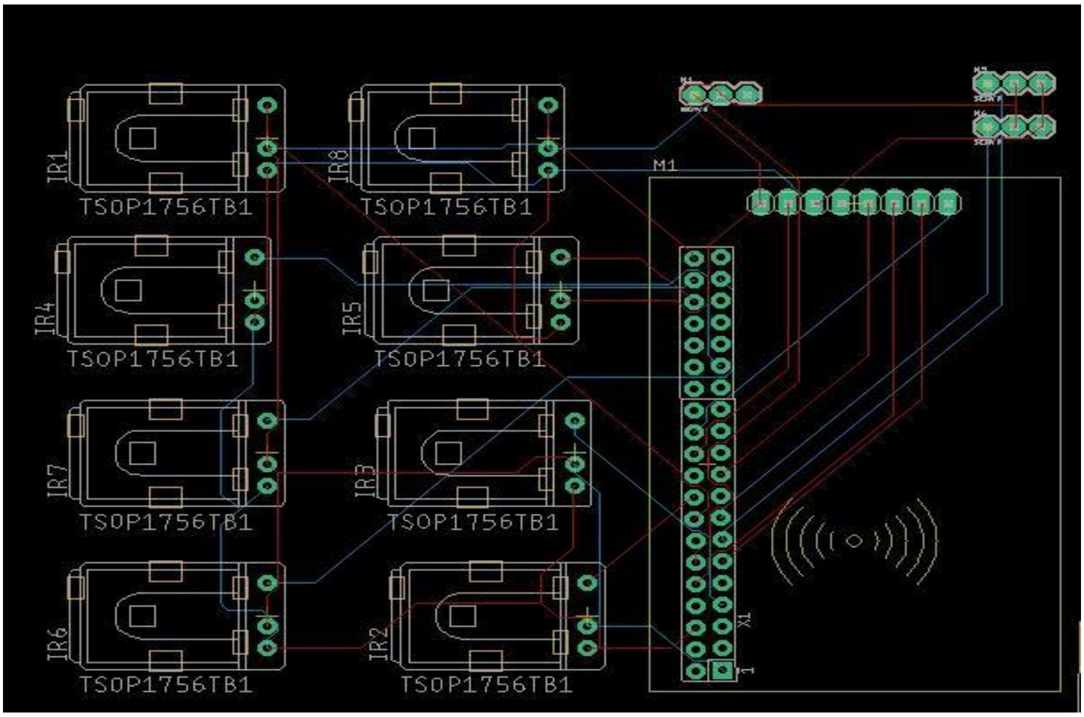
```
Writing to display
Parking Available   Parking Available
Writing to display
Outgoing Car   Outgoing Car
taking pic
processing pic
Writing to display
Place your Card    Place your Card
card detected
gate opened
gate closed
Writing to display
Restaring services    Restaring services
Writing to display
Incoming Car   Incoming Car
taking pic
processing pic
Writing to display
Place your Card    Place your Card
card detected
gate opened
gate closed
Writing to display
Restaring services    Restaring services
Writing to display
Parking Available   Parking Available
Writing to display
Outgoing Car   Outgoing Car
taking pic
processing pic
Writing to display
Place your Card    Place your Card
no card
Writing to display
Restaring services    Restaring services
Writing to display
Parking Available   Parking Available
Writing to display
```

4.1.5.6 Pin Configuration

The diagram below was produced by eagle fritzer showing the connection between different hardware parts and sensors. Figure one shows a simplified version of the pin configuration.

Figure 36

Pin Configuration



The simplified pin configuration layout is as shown above. The representation shows the different hardware arrangement and the associated sensors. The specific pin configuration for the different pins to be considered for emulating hardware sensors is a shown in the diagram below. The pin configuration scripts are shown in Table 6, Figure 37 Pin configuration.

Figure 37

Pin Configuration

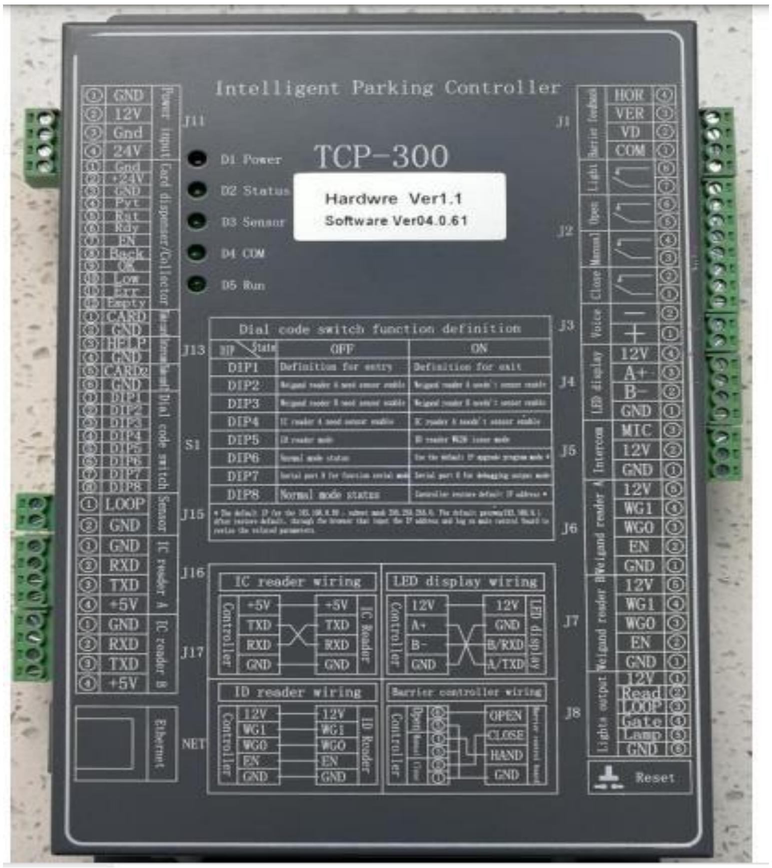


Table 6

Pin Configuration Script for Emulating Hardware Sensors

RST	gpio25
MISO	gpio9
MOSI	gpio10
SCK	gpio11
SDA	gpio8

IR Infrared sensor module 1 = gpio23 IR Infrared sensor module 2 = gpio24 IR Infrared sensor module 3 = gpio4 IR Infrared sensor module 4 = gpio29 IR Infrared sensor module 5 = gpio16 IR Infrared sensor module 6 = gpio26 IR Infrared sensor module 7 = gpio20 IR Infrared sensor module 8 = gpio21 Servo config Servo1= gpio18 Servo2= gpio27 Servo3= gpio22

4.2 Model Implementation

This area discusses the stages followed in realising the Media Auto-stream prototype of an RFID microcontroller based real time media auto-streaming model for vehicle parking. The proof of concept approach was used in guiding the entire process goal realization. This step involved assembling of components, services and functions for the model. Several functions were re-used in various integration points. Using the proof-of-concept methodology, it realizes micro-controller based platform implementation success factors along with factors that impact overall scope and estimates of effort for the platform.

It also describes the 4 steps functional decomposition process that was followed in model development as described in section 4.1.1 to 4.1.6.

4.3 Implementation of the Media Auto-stream prototype of an RFID microcontroller based real time media auto-streaming model for vehicle parking

At this stage Optical Character Recognition (OCR) algorithm was used to extract important information from images, specifically focusing on fonts and text. OCR involves several stages of compression to convert an image into editable text. Different AI algorithms, such as artificial neural networks, machine learning, deep learning, and pattern recognition, have been explored for OCR analysis Convolutional Neural Networks (CNN) have shown significant improvements in recognizing characters of different languages, making OCR more robust to variations in image quality, contrast, font style, and font size OCR systems utilize algorithms for text detection, identification, localization, segmentation, and categorization of extracted features Recent advancements in deep learning have accelerated improvements in OCR, but challenges remain for languages with limited annotated datasets, Pre-processing and segmentation are crucial steps in OCR to ensure accuracy and consistent results in converting printed images into editable texts In the field of image processing, one common technique that was used is image "binarization" or "thresholding," which aims to convert a grayscale image into a binary image by separating foreground objects from the background based on their pixel intensities. This process involves setting a specific threshold value, where pixel intensities below the threshold are assigned one value (e.g., 0) to represent the background, and pixel intensities above the threshold are assigned a different value (e.g., 255) to represent the foreground objects. This technique has been widely used in various applications, such as human activity recognition systems real-time space object tracklet extraction automatic data process line identification, and construction site

Otsu's method is a widely used technique in image segmentation that determines a global threshold for segmenting an image into foreground and background categories based on histogram shape analysis and the twin modal distribution of the histogram. This method is preferred because it offers promising performance in accurately segmenting images, making it useful for various applications such as object detection, image recognition, and medical image analysis Additionally, Otsu's method is adaptive, meaning it can adaptively adjust the threshold based on the characteristics of the image being segmented

Otsu's method is an adaptive threshold determination method that determines a global threshold for segmenting an image into foreground and background categories based on the histogram shape analysis and the twin modal distribution of the histogram. Otsu's method is widely used in image segmentation due to its ability to determine a global threshold for segmenting an image into foreground and background categories based on the histogram shape analysis. Otsu's method is preferred for image segmentation because it offers several advantages Otsu's method was used in image pre processing and was preferred for image segmentation because it offers several advantages including simplicity in calculation, stability, and its ability to handle bimodal or multimodal distributions of histogram. Otsu's method is a popular choice for image segmentation due to its simplicity, adaptability, and effectiveness.

4.3.1 Stages of threshholding in the implementation of Media Auto-stream of the prototype

The diagrams below in image figure depicts the original image or the real image of the car before undergoing processing and eventually recognition of the registration number.

i) **Original Picture**

Figure 38

Original Picture

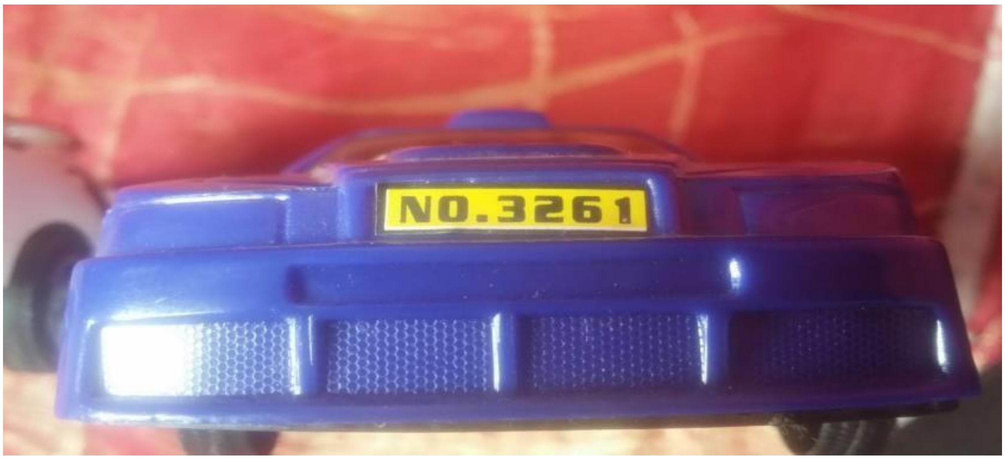
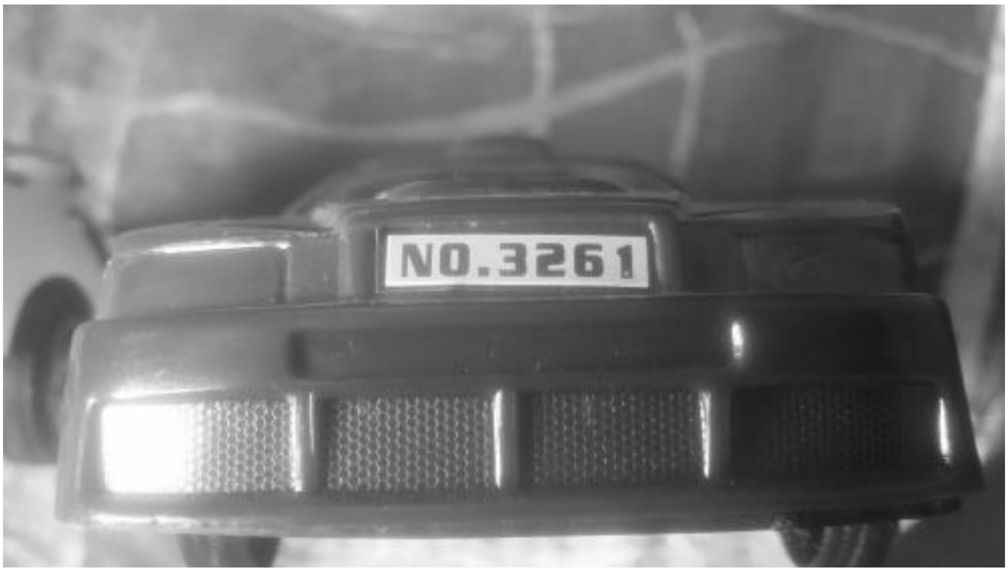


Figure 39 is the output after undergoing thresholding so has to reveal the important font and text.

Figure 39

Thresh-Holding



ii) Image Binarization

This method create bitonal (black and white) versions of images in which black pixels are considered to be the foreground (characters or ink) and white pixels are the document background. The simplest form of binarization is global thresholding, in which a grayscale intensity threshold is selected and then each pixel is set to either black or white depending on whether it is darker or lighter than the threshold, respectively. Since the brightness and contrast of document images can vary widely, it is often not possible to select a single threshold that is suitable for an entire collection of images.

The Otsu method is commonly used to automatically determine thresholds on a per-image basis (Kittaneh, 2023). The method assumes two classes of pixels (foreground and background) and uses the histogram of grayscale values in the image to choose the threshold that maximizes between-class variance and minimizes within-class variance. This statistically optimal solution may or may not be the best threshold for OCR, but often works well for clean documents

Kittler-etal is also used to determine thresholds, Findling, et, al (2016). For some images, no global (image- wide) threshold exists that results in good binarization. Background noise, stray marks, or ink bleed-through from the back side of a picture may be darker than some of the desired text. Stains, uneven brightness, picture degradation, or faded picture can mean that some parts of the picture are too light for a given threshold while other parts are too dark for the same threshold.

Adaptive thresholding methods attempt to compensate for inconsistent brightness and contrast in images by selecting a threshold for each pixel based on the properties of a small portion of the image (window) surrounding that pixel, instead of the whole image.

The Sauvola method is a well-known adaptive thresholding method used when the backround is not uniform (Michalak and Okarma, 2023). Sauvola performs better than the Otsu method in some cases; however, neither is better in all cases, and in some cases adaptive thresholding methods even accentuate noise more than global thresholding. In addition, the results of the Sauvola method on any given document are dependent on user-tunable parameters. Like global thresholds, a specific parameter setting may not be sufficient for good results across an entire set of documents.

4.3.2 Image binarization framework in the implementation of Media Auto-stream.

In the past, vehicle parking management was fraught with challenges such as inefficiency, security concerns, and a lack of real-time monitoring capabilities. These challenges prompted the development of innovative solutions to revolutionize parking management. The Media Auto-Stream system, which integrates Radio-Frequency Identification (RFID) technology with real-time media streaming, emerged as a game-changer. Within the Media Auto-Stream system, image binarization serves as a pivotal component that enhances security and real-time monitoring. This research provides a historical perspective on the image binarization framework, detailing its importance, techniques employed in the past, integration into the system, and its impact on parking management.

i. Context of Image Binarization

Image binarization is not a new concept. It has a rich historical context in image processing and computer vision. In the past, it was employed to address various challenges, including image enhancement, feature extraction, and document processing. With the integration of image binarization into the Media Auto-Stream system, it has evolved to meet the specific needs of modern vehicle parking management and this research enhanced the process to achieve the required threshold.

ii. Techniques Employed.

Image binarization techniques was adopted and refined for contemporary applications. Several techniques were employed, and their historical significance is worth noting:

iii. Global Thresholding

Global thresholding was one of the earliest and simplest binarization techniques. In the past, it involved selecting a single threshold value to divide the image into black and white regions. This technique was suitable when the lighting conditions across the image were relatively uniform.

iv. Adaptive Thresholding

Adaptive thresholding, in the past, was a revolutionary technique that addressed local variations in image lighting. Unlike global thresholding, adaptive thresholding divided the image into multiple regions and assigned different thresholds to each. This historical technique was employed to handle images with varying lighting conditions effectively.

v. Otsu's Method

Otsu's method, was significant, was an optimal global thresholding technique used. It automatically calculated the threshold value to minimize the intra-class variance between black and white pixels. It was particularly useful when dealing with images with substantial variation in pixel intensities.

vi. Integration into the System

The integration of image binarization into the Media Auto-Streaming system was a significant development. This integration was crucial in shaping the system's capabilities and functionality.

vii. Image Capture and Processing in the Past

In the past, the image capture process involved cameras placed strategically within the parking facility. These cameras captured visual data that was subsequently processed in real-time. This processing in the past encompassed image enhancement, noise reduction, and image binarization using techniques relevant to that era.

viii. Integration into Media Auto-Streaming

The integration of binarized images into the Media Auto-Streaming system was a landmark achievement these binarized images were seamlessly integrated into the system, ensuring real-time accessibility for users, especially vehicle owners. Data transmission methods and technology, while less advanced and , facilitated real-time monitoring in the past.

ix. Role in Ensuring Real-Time Security

Image binarization's role in enhancing security and real-time monitoring within the Media Auto-Streaming system has realtime significance. Severalmechanisms were employed to ensure security and real-time alerting:

x. Intrusion Detection

Image binarization played a crucial role in intrusion detection within parking facilities. Changes in the binary image that indicated unauthorized entry or suspicious activities triggered real-time alerts to security personnel.

xi. Anomaly Detection

In the past, the system employed image binarization to identify anomalies within the binary images. Historical anomalies, such as unexpected objects or behaviors, were detected, leading to alerts and further investigation.

xii. Motion Detection

Changes in the binary image that signaled motion were historically used to detect potential security breaches. The past saw the development of historical methods to identify unauthorized vehicle movements and trigger alerts.

xiii. Time Stamped Alerts

Historically, all security-related alerts, including intrusions, anomalies, and motion detections, were time stamped and logged. This historical data provided a comprehensive record of security events within the parking facility, facilitating post-incident analysis and investigation.

xiv. Privacy Measures

In the past, as with contemporary systems, privacy concerns were paramount in the use of image binarization. Historical privacy measures were in place to protect the rights and interests of all parties involved:

xv. Data Encryption

In the past, data encryption was employed to secure historical binarized images before transmission. This encryption, while less sophisticated than modern encryption

xvi. Historical Access Control

Access control was historically rigorous. Only authorized users, such as vehicle owners and security personnel, were granted access to the visual data in the past.

xvii. Data Retention Policies

Systems implemented well-defined data retention policies to ensure that images were not stored for longer than necessary. This historical practice reduced the risk of data misuse or breaches of privacy.

xviii. User Experience and System Efficiency

The implementation of image binarization within the Media Auto-Streaming system had a significant impact on the user experience and system efficiency:

xix. User Experience Enhancement

In the past, the availability of real-time binarized images greatly enhanced the user experience for vehicle owners. It provided users with easy access to visual information about their parked vehicles, ensuring peace of mind and a sense of control.

xx. System Efficiency

Historical systems were efficient in handling a large volume of visual data in real-time. Binarized images, being smaller in size, facilitated efficient transmission and storage in historical contexts. This efficiency made historical systems suitable for use in busy parking facilities.

xxi. Conclusion

The image binarization framework within the historical context of the Media Auto-Streaming system has a rich history that has evolved to meet contemporary needs. Image binarization techniques employed in the past have shaped the system's capabilities, enabling real-time security and monitoring. privacy measures, access control, and data retention policies have contributed to the protection of user rights and interests. The integration of image binarization and its role in enhancing security, user experience, and system efficiency underscores its significance in the evolution of parking management systems.

Image binarization process

Figure 40

Image Binarization Process

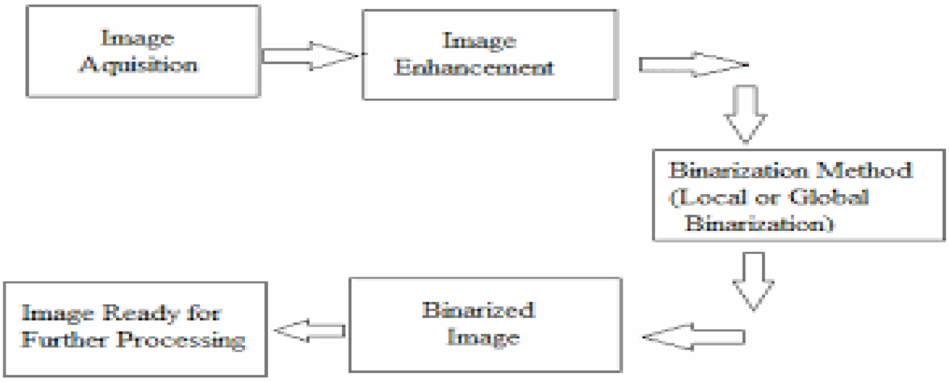
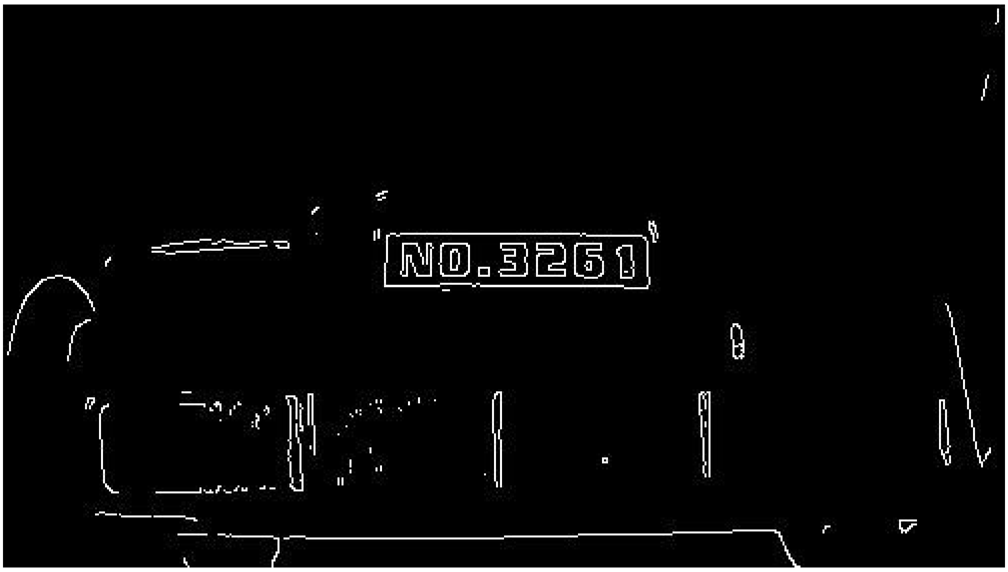


Image Aquisition
Image Enhancement
Binarization Method (Local or Global Binarization)
Binarized Image
Image Ready for Further Processing

With global binarization image being the Otsu and local binarization riming and enhancing image functions. To show the particular image of the number plate as shown in Figure 41.

Figure 41

Image Binarization



4.3.2 Location and Segmentation

In order to ensure the representation of an image into something that is more meaningful and easier to analyze. Segmentation is a process that determines the constituents of an image. It is necessary to locate the regions of the document where data have been printed and distinguish them from figures and graphics.

Applied to text, segmentation is the isolation of characters or words. The majority of optical character recognition algorithms segment the words into isolated characters which are recognized individually. Usually this segmentation is performed by isolating each connected component that is each connected black area. This technique is easy to implement, but problems occur if characters touch or if characters are fragmented and consist of several parts. The main problems in segmentation may be divided into four groups:

i. Extraction of touching and fragmented characters. Such distortions may lead to several joint characters being interpreted as one single character, or that a piece of a character is believed to be an entire symbol. Joints will occur if the document is a dark photocopy or if it is scanned at a low threshold. Also joints are common if the fonts are segmented. The characters may be split if the document stems from a light photocopy or is scanned at a high threshold.

ii. Distinguishing noise from text Dots and accents may be mistaken for noise, and vice versa

iii. Mistaking graphics or geometry for text. This leads to non-text being sent to recognition.

iv. Mistaking text for graphics or geometry. In this case the text will not be passed to the recognition stage. This often happens if characters are connected to graphics.

i) Preprocessing

The image resulting from the scanning process may contain a certain amount of noise. Depending on the resolution on the scanner and the success of the applied technique for thresholding, the characters may be smeared or broken. Some of these defects, which may later cause poor recognition rates, can be eliminated by using a preprocessor to smooth the digitized characters.

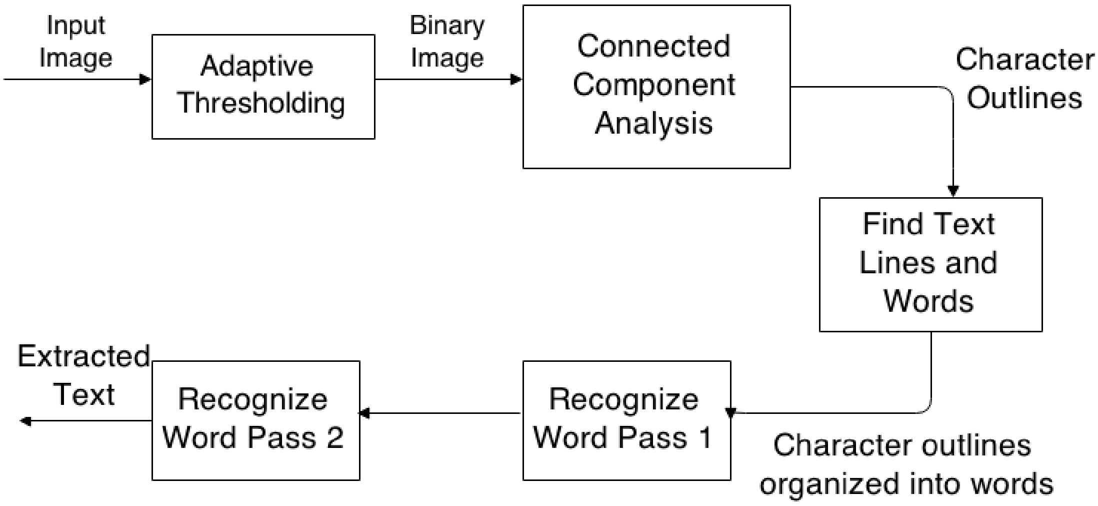
The smoothing implies both filling and thinning. Filling eliminates small breaks, gaps and holes in the digitized characters, while thinning reduces the width of the line. The most common techniques for smoothing, moves a window across the binary image of the character, applying certain rules to the contents of the window.

In addition to smoothing, preprocessing usually includes normalization. The normalization is applied to obtain characters of uniform size, slant and rotation. To be able to correct for rotation, the angle of rotation must be found. For rotated pages and lines of text, variants of Hough transform are commonly used for detecting skew. However, to find the rotation angle of a single symbol is not possible until after the symbol has been recognized. Image Preprocessing steps are outlined in Figure 42 below.

Figure Image Preprocessing

Figure 42

Image Preprocessing



Source: Yuan, et al, (2023)

ii) Feature Extraction

The objective of feature extraction is to capture the essential characteristics of the symbols, and it is generally accepted that this is one of the most difficult problems of pattern recognition. The most straight forward way of describing a character is by the actual raster image. Another approach is to extract certain features that still characterize the symbols, but leaves out the unimportant attributes. The techniques for extraction of such features are often divided into three main groups, where the features are found from:

i) The distribution of points.

ii) Transformations and series expansions.

iii) Structural analysis.

The different groups of features may be evaluated according to their sensitivity to noise and deformation and the ease of implementation and use. The results of such a comparison are as listed below

Robustness.

i. Noise. Sensitivity to disconnected line segments, bumps, gaps, filled loops etc.

ii. Distortions. Sensitivity to local variations like rounded corners, improper protrusions, dilatation and shrinkage.

iii. Style variation. Sensitivity to variation in style like the use of different shapes to represent the same character or the use of serifs, slants etc.

iv. Translation. Sensitivity to movement of the whole character or its components.

v. Rotation. Sensitivity to change in orientation of the characters.

The practicability and usage of OCR has great benefits including;

i) Speed of recognition.

ii) Complexity of implementation.

iii) Independence. The need of supplementary techniques

iv) Template-matching and correlation techniques.

These techniques are different from the others in that no features are actually extracted. Instead the matrix containing the image of the input character is directly matched with a set of prototype characters representing each possible class. The distance between the pattern and each prototype is computed, and the class of the prototype giving the best match is assigned to the pattern.

The technique is simple and easy to implement in hardware and has been used in many commercial OCR machines. However, this technique is sensitive to noise and style variations and has no way of handling rotated characters.

v) **Feature Based Techniques**

In these methods, significant measurements are calculated and extracted from a character and compared to descriptions of the character classes obtained during a training phase.

The description that matches most closely provides recognition. The features are given as numbers in a feature vector, and this feature vector is used to represent the symbol.

vi) **Distribution of points.**

This category covers techniques that extract features based on the statistical distribution of points. These features are usually tolerant to distortions and style variations. Some of the typical techniques within this area are listed below.

Zoning -The rectangle circumscribing the character is divided into several overlapping, or non-overlapping, regions and the densities of black points within these regions are computed and used as features.

Moments - The moments of black points about a chosen centre, for example the centre of gravity, or a chosen coordinate system, are used as features.

Crossings and distances - In the crossing technique features are found from the number of times the character shape is crossed by vectors along certain directions. This technique is often used by commercial systems because it can be performed at high speed and requires low complexity. When using the distance technique certain lengths along the vectors crossing the character shape are measured. For instance the length of the vectors within the boundary of the character.

N-tuples - The relative joint occurrence of black and white points (foreground and background) in certain specified orderings, are used as features.

Characteristic loci - For each point in the background of the character, vertical and horizontal vectors are generated. The number of times the line segments describing the character are intersected by these vectors are used as feature.

vii) **Classification**

The classification is the process of identifying each character and assigning to it the correct character class. In the following sections two different approaches for classification in character recognition are discussed. First decision-theoretic recognition is treated.

These methods are used when the description of the character can be numerically represented in a feature vector.

We may also have pattern characteristics derived from the physical structure of the character which are not as easily quantified. In these cases, the relationship between the characteristics may be of importance when deciding on class membership. For instance, if we know that a character consists of one vertical and one horizontal stroke, it may be either an ―L or a ―T‖, and the relationship between the two strokes is needed to distinguish the characters. A structural approach is then needed.

viii) **Decision-Theoretic Methods**

The principal approaches to decision-theoretic recognition with minimum distance classifiers, statistical classifiers and neural networks. Each of these classification techniques are briefly described below.

Matching - Matching covers the groups of techniques based on similarity measures where the class is calculated. Different measures may be used, but the common is the Euclidean distance. This minimum distance classifier works well when the classes are well separated, that is when the character is used as input to the classification, and no features are extracted (template-matching), a correlation approach is used. Here the distance between the character image and prototype images representing each character class is computed.

i. Optimum statistical classifiers - In statistical classification a probabilistic approach to recognition is applied. The idea is to use a classification scheme that is optimal in the sense that, on average, its use gives the lowest probability of making classification errors.

A classifier that minimizes the total average loss is called the Bayes' classifier. Given an unknown symbol described by its feature vector, the probability that the symbol belongs to class c is computed for all classes c =1...N. The symbol is then assigned the class which gives the maximum probability.

For this scheme to be optimal, the probability density functions of the symbols of each class must be known, along with the probability of occurrence of each class. The Latter is usually solved by assuming that all classes are equally probable. The density function is usually assumed to be normally distributed, and the closer this assumption is to reality, the closer the Bayes' classifier comes to optimal behaviour.

The minimum distance classifier described above is specified completely by the mean vector of each class, and the Bayes classifier for Gaussian classes is specified completely by the mean vector and co-variance matrix of each class. These parameters specifying the classifiers are obtained through a training process. During this process, training patterns of each class is used to compute these parameters and descriptions of each class are obtained.

1. Neural networks - Recently, the use of neural networks to recognize characters (and other types of patterns) has resurfaced. Considering a back-propagation network, this network is composed of several layers of interconnected elements. A feature vector enters the network at the input layer. Each element of the layer computes a weighted sum of its input and transforms it into an output by a nonlinear function.

During training the weights at each connection are adjusted until a desired output is obtained. A problem of neural networks in OCR may be their limited predictability and generality, while an advantage is their adaptive nature.

ix) **Structural Methods**

Within the area of structural recognition, syntactic methods are among the most prevalent approaches. Other techniques exist, but they are less general and will not be treated here.

Measures of similarity based on relationships between structural components may be formulated by using grammatical concepts. The idea is that each class has its own grammar defining the composition of the character. A grammar may be represented as strings or trees, and the structural components extracted from an unknown character is matched against the grammars of each class. Suppose that we have two different character classes which can be generated by the two grammars G1 and G 2 respectively. Given an unknown character, we say that it is more similar to the first class if it may be generated by the grammar G1 but not by G2.

x) **Post processing.**

i. Grouping - The result of plain symbol recognition on a document is a set of individual symbols. However, these symbols in themselves do usually not contain enough information. Instead we would like to associate the individual symbols that belong to the same string with each other, making up words and numbers. The process of performing this association of symbols into strings, is commonly referred to as grouping. The grouping of the symbols into strings is based on the symbols' location in the document. Symbols that are found to be sufficiently close are grouped together. For fonts with fixed pitch the process of grouping is fairly easy as the position of each character is known. For typeset characters the distance between characters are variable. However, the distances between words are usually significantly larger than the distance between characters, and grouping is therefore still possible. The real problems occur for handwritten characters or when the text is skewed.

ii. Error-detection and correction - Up until the grouping each character has been treated separately, and the context in which each character appears has usually not been exploited. However, in advanced optical text-recognition problems, a system consisting only of single-character recognition will not be sufficient. Even the best recognition systems will not give 100% percent correct identification of all characters, but some of these errors may be detected or even corrected by the use of context. There are two main approaches, where the first utilizes the possibility of sequences of characters appearing together.

This may be done by the use of rules defining the syntax of the word, by saying for instance that after a period there should usually be a capital letter. Also, for different languages the probabilities of two or more characters appearing together in a sequence can be computed and may be utilized to detect errors. For instance, in the English language the probability of a k‖ appearing after an —h‖ in a word is zero, and if such a combination is detected an error is assumed.

iii. Another approach is the use of dictionaries, which has proven to be the most efficient method for error detection and correction. Given a word, in which an error may be present, the word is looked up in the dictionary. If the word is not in the dictionary, an error has been detected, and may be corrected by changing the word into the most similar word. Probabilities obtained from the classification, may help to identify the character which has been erroneously classified. If the word is present in the dictionary, this does unfortunately not prove that no error occurred. An error may have transformed the word from one legal word to another, and such errors are undetectable by this procedure. The disadvantage of the dictionary methods is that the searches and comparisons implied are time-consuming.

4.4 Infra-Red Sensors in image binarization framework in the implementation of Media Auto-stream.

Infrared technology addresses a wide variety of wireless applications. The main areas are sensing and remote controls. In the electromagnetic spectrum, the infrared portion is divided into three regions: near infrared region, mid infrared region and far infrared region. The wavelengths of these regions and their applications are shown below.

i. Near infrared region — 700 nm to 1400 nm — IR sensors, fiber optic

ii. Mid infrared region — 1400 nm to 3000 nm — Heat sensing

iii. Far infrared region — 3000 nm to 1 mm — Thermal imaging

The frequency range of infrared is higher than microwave and lesser than visible light. For optical sensing and optical communication, photo optics technologies are used in the near infrared region as the light is less complex than RF when implemented as a source of signal. Optical wireless communication is done with IR data transmission for short range applications.

An infrared sensor emits and/or detects infrared radiation to sense its surroundings. The working of any Infrared sensor is governed by three laws:

Planck‘s Radiation law, Stephen – Boltzmann law and Wien‘s Displacement law. Planck‘s law states that ―every object emits radiation at a temperature not equal to 0^0K‖. Stephen – Boltzmann law states that ―at all wavelengths, the total energy emitted by a black body is proportional to the fourth power of the absolute Temperature. According to Wien’s Displacement law, ―the radiation curve of a black body for different temperatures will reach its peak at a wavelength inversely proportional to the temperature. The basic concept of an Infrared Sensor which is used as Obstacle detector is to transmit an infrared signal, this infrared signal bounces from the surface of an object and the signal is received at the infrared receiver.

There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LED’s of specific wavelength can be used as infrared sources. The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibers. Optical components are used to focus the infrared radiation or to limit the spectral response.

Optical lenses made of Quartz, Germanium and Silicon are used to focus the infrared radiation. Infrared receivers can be photodiodes, phototransistors etc. some important specifications of infrared receivers are photosensitivity, detectivity and noise equivalent power. Signal processing is done by amplifiers as the output of infrared detector is very small. Planck's law states that - every object emits radiation at a temperature not equal to 0^0K.

4.5 Pre-processing of the prototype of an RFID microcontroller based real time media auto-streaming model for vehicle parking.

This area explores the pre-processing phase of developing a prototype for an RFID microcontroller-based real-time media auto-streaming model for vehicle parking. The effective pre-processing of data and system components is a critical foundation for the success of the entire system. This topic will provide a comprehensive analysis of the pre-processing steps, including RFID tag data collection, data cleaning, data transformation, and the preparation of the media auto-streaming component. This research contributes to the understanding of the crucial initial phases in the development of modern parking management systems. In the era of urbanization and rapidly increasing vehicle populations, efficient parking management has become an imperative need. Innovations in technology, particularly the integration of Radio-Frequency Identification (RFID) technology, have brought forth a revolutionary approach to parking management. This research focused on the pre-processing phase of developing a prototype for an RFID microcontroller-based real-time media auto-streaming model for vehicle parking. The pre-processing phase is a critical juncture in the development of any technological system, setting the stage for efficient data handling and real-time functionality.

i. RFID Technology and Parking Management

Before delving into the specifics of the pre-processing phase, it is essential to understand the fundamentals of RFID technology and its applications in parking

ii. RFID Tags

RFID tags are passive devices containing a unique identification number. They are typically affixed to vehicles and can take the form of passive, active, or semi-passive tags. Passive RFID tags do not require a power source; instead, they relied on the energy emitted by RFID readers to transmit data. Active tags, on the other hand, have their power source, allowing for longer-range data transmission. Semi-passive tags utilize a battery for onboard processing.

iii. RFID Readers

RFID readers, or interrogators, play a pivotal role in the RFID system. They emit radio waves, activating passive RFID tags and collecting data from them. This collected data is then used for various applications, including vehicle tracking and parking management.

iv. Pre-processing Steps

Efficient pre-processing of data and system components is essential for the successful development of a real-time media auto-streaming model for vehicle parking using RFID technology. This section presents a detailed analysis of the key pre-processing steps involved.

v. Tag Data Collection

The initial phase of pre-processing revolves around the collection of RFID tag data from the vehicles entering the parking facility. This data is critical for tracking and managing vehicles effectively. The data collection process typically involves RFID readers positioned at entry and exit points, which communicate with the RFID tags on vehicles.

In this research, data collection was conducted systematically, capturing unique vehicle identifiers and time stamps upon entry and exit. The data was stored for further processing and analysis.

vi. Data Cleaning

Data collected from RFID tags often requires cleaning to eliminate inconsistencies and inaccuracies. Cleaning of data was a pivotal pre-processing step in this research. It involved the identification and rectification of errors, such as missing or duplicate entries, and the handling of anomalies caused by environmental factors or technical glitches.

Data cleaning was performed using data cleaning algorithms, ensuring that the collected RFID data was reliable and free from anomalies. The goal was to create a clean dataset for subsequent processing and analysis.

vii. Data Transformation

Data transformation is a crucial pre-processing step, involving the conversion of data into a structured format that can be readily utilized in the development of the media auto-streaming model. In the context of the prototype, data transformation encompassed several aspects:

- Normalization: RFID data was normalized to ensure consistency and comparability across different tags and readers.
- Encoding: Vehicle identifiers were encoded for efficient storage and retrieval. The encoding scheme allowed for rapid access to specific data records.
- Aggregation: Data was aggregated to create a comprehensive parking log, including entry and exit timestamps, vehicle identifiers, and other relevant information.

viii. Preparation of the Media Auto-Streaming Component

One of the distinctive features of the prototype was the media auto-streaming component, allowing vehicle owners to remotely monitor their vehicles using real-time audio and video streams. The preparation of this component involved several critical pre-processing steps:

- Authentication and Authorization: Pre-processing included the development of authentication and authorization mechanisms, ensuring that only authorized vehicle owners could access the media auto-streams. User data and permissions were integrated into the system.
- Stream Data Collection: Data collection for the media auto-streams involved configuring cameras and audio devices at key points within the parking facility. Data from these devices were synchronized with the RFID data to ensure real-time accuracy.
- Media Format Standardization: To enable efficient streaming and playback, media formats were standardized. This involved the conversion of audio and video data into compatible formats for web and mobile access.

ix. Significance of Pre-processing

The pre-processing phase is foundational for the success of the RFID microcontroller-based real-time media auto-streaming model for vehicle parking. It lays the groundwork for data reliability, accuracy, and accessibility, all of which are crucial for an effective parking management system. The significance of pre-processing can be summarized as follows:

- Data Quality: Pre-processing ensures that the data collected from RFID tags is of high quality and free from errors, ensuring the accuracy of parking records.
- Efficiency: Cleaned and transformed data is more efficient for system processing and analysis, reducing the computational load on the central server.
- User Experience: Properly prepared media auto-streaming components enhance the user experience, allowing vehicle owners to securely monitor their vehicles in real-time.
- Security: Authentication and authorization processes implemented during pre-processing enhance security, ensuring that only authorized users can access sensitive media data.

The pre-processing phase is a critical foundation in the development of a prototype for an RFID microcontroller-based real-time media auto-streaming model for vehicle parking. Efficient data collection, cleaning, transformation, and the preparation of media components set the stage for a robust and functional system. The significance of pre-processing cannot be understated, as it ensures data quality, system efficiency, an enhanced user experience, and heightened security. This research-based approach provides valuable insights into the importance of effective pre-processing in the development of modern parking management systems.

4.6 Subprocess.py Code

```
Import sub-process import numpy as np import cv2
From copy import deepcopy from PIL import Image import pytesseract as tess
defpreprocess_image(img):
imgBlurred = cv2.GaussianBlur(img, (5,5), 0)
gray = cv2.cvtColor(imgBlurred, cv2.COLOR_BGR2GRAY) sobelx
= cv2.Sobel(gray,cv2.CV_8U,1,0,ksize=3)
ret2, threshold_img     =
cv2.threshold(sobelx,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)return
threshold_img
```

i. Number Plate Localization

The number plate localization is the phase in which mainly focuses on ROI (Region of Interest) where we find the contour region.

ii. Contour Tracing

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

Here we use contours in rectangle shape. For better accuracy, we use binarized Images and before finding the contours, we apply threshold.

4.7 Code for Contour Tracing

```
defget_contours(threshold_img):
element = cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(17, 3))
morph_img_threshold = threshold_img.copy()
```

```
cv2.morphologyEx(src=threshold_img,op=cv2.MORPH_CLOSE,kernel=element,
dst=morph_img_threshold)im2,contours,hierarchy=cv2.findContours(morph_img_thr
eshold,mode=cv2.RETR_EXTERNAL,method=cv2.CHAIN_APPROX_NONE)
return contours.
```

For the puposes of image segmentation and data extraction edge detection was undertaken. This helped to detect the edges present in an images therefore revealing the discontinuities in brightness on selected images relayed from cameras.

```
defscrap_plate(plate): gray = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)
_ , thresh = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY) im1,contou rs,hier
archy=cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPRO
X_NONE)

if contours:
areas = [cv2.contourArea(c) for c in contours] max_index = np. argmax(areas)
max_cnt = contours[max_index] max_cntArea = areas[max_index] x, y,w,h =
cv2.boundingRect(max_cnt) if not check_ratio(max_cntArea,w,h):return plate, None
cleaned_final = thresh[y:y+h, x:x+w] return cleaned_final,[x,y,w,h] else: return
plate,None
```

iii. Morphological Operations

In these study it applied to the structuring of the input image, an output image of the same size was created. The morphological operations are shown below;

```
<snip> defget_text_from_plate (img, contours):
for i, cnt in enumerate(contours): min_rect = cv2.minAreaRect(cnt)
if validate_image_rotation_and_ratio(min_rect): x, y, w, h = cv2.boundingRect(cnt)
plate_img = img [y: y+h, x:x+w] if(get_max_white(plate_img)):
```

```
clean_plate, rect = scrap_plate(plate_img) if rect:
x1, y1, w1, h1 = rect
x, y, w, h = x+x1, y+y1,w1,h1
</snip>
```

entifying the License Plate

To identify the region containing the license plate, two features are defined

a) Aspect ratio

b) Edge Density

Iv. Aspect Ratio

The aspect ratio is defined as the ratio of the width to the height of the region. Aspect Ratio = width/height in this phase we have to assume the width and height of an image. In this we assume width and height is 16/9.

```
defcheck_ratio (area, width, height):
ratio = float(width) / float(height) if ratio < 1:
ratio = 1 / ratio aspect = 4.7272
min = 15*aspect*15 # minimum area max = 125*aspect*125 # maximum area rmin = 3 rmax = 6
if (area < min or area > max) or (ratio <rmin or ratio >rmax): return False
return True
```

The edge density is the process in which we mainly determine the pixel of a square image of vehicle. Here also assume the area covered in whole image. We assume the 6.25 square of area of vehicles.

```
defvalidate_image_rotation_and_ratio(rect): (x, y), (width, height), rect_angle = rect
if(width > height):
angle = - rect_angle
```

```
else:
angle = 90 + rect_angle
if angle>15:
return False
if height == 0 or width == 0: return False
area = height*width
if not check_ratio(area,width,height): return False
else:
return True
```

i. Character Segmentation

Character segmentation is the technique in which individual character present in the image is separated out. Here all character is checked out individually.

ii. Character Recognition

An automatic license plate recognition system must recognize alphanumeric characters. The character image is compared with the training set and the best similarity is measured accordingly, the recognized character is the displayed to the LCD screen. displayed.

iii. Character Segmentation and Character Recognition

For purposes of simplifying representation of an image into something that is more meaningful and easier to analyze segmentation is done. In this regard it has been achieved using py the pytesseract module. Putting all the snippets together for a test run.

```
import subprocess import numpy as np import cv2
import picamera
```

```
from copy import deepcopy from PIL import Image import pytesseract as tess
defpreprocess_image(img):
imgBlurred = cv2.GaussianBlur(img, (5,5), 0)
gray = cv2.cvtColor(imgBlurred, cv2.COLOR_BGR2GRAY) sobelx =
cv2.Sobel(gray, cv2.CV_8U,1,0, ksize=3)
ret2, threshold_img
cv2.threshold(sobelx,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU) return
threshold_img
defscrap_plate(plate):
gray = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)
_ , thresh = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)
im1,contours,hierarchy =
          cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
if contours:
areas = [cv2.contourArea(c) for c in contours] max_index = np.argmax(areas)
max_cnt = contours[max_index]
max_cntArea = areas[max_index] x,y,w,h = cv2.boundingRect(max_cnt) if not
check_ratio(max_cntArea,w,h):
return plate,None cleaned_final = thresh[y:y+h, x:x+w] return cleaned_final,[x,y,w,h]
else:
return plate,None
defget_contours(threshold_img):
element = cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(17, 3))
morph_img_threshold = threshold_img.copy()
```

```
cv2.morphologyEx(src=threshold_img,        op=cv2.MORPH_CLOSE,
                kernel=element, dst=morph_img_threshold)
im2,contours,hierarchy=
cv2.findContours(morph_img_threshold,mode=cv2.RETR_EXTERNAL,method=cv2
.CHAI N_APPROX_NONE)
return contours
defcheck_ratio(area, width, height):
ratio = float(width) / float(height) if ratio < 1:
ratio = 1 / ratio aspect = 4.7272
min = 15*aspect*15 # minimum area max = 125*aspect*125 # maximum area
rmin = 3
rmax = 6
if (area < min or area > max) or (ratio <rmin or ratio >rmax): return False
return True
defget_max_white(plate): avg = np.mean(plate) if(avg>=115):
return True
else:
return False
defvalidate_image_rotation_and_ratio(rect): (x, y), (width, height), rect_angle = rect
if(width > height):
angle = - rect_angle
else:
angle = 90 + rect_angle
if angle>15:
return False
```

```
if height == 0 or width == 0: return False
area = height*width
if not check_ratio(area,width,height): return False
else:
return True
defcmd_tesseract(image_name): cmd=['tesseract'] cmd.append(image_name)
cmd.append('stdout') output=subprocess.check_output(cmd) return output.split('\n')[0]
defget_text_from_plate(img,contours): for i,cnt in enumerate(contours):
min_rect = cv2.minAreaRect(cnt)
if validate_image_rotation_and_ratio(min_rect): x,y,w,h = cv2.boundingRect(cnt)
plate_img = img[y:y+h,x:x+w] if(get_max_white(plate_img)):
clean_plate, rect = scrap_plate(plate_img) if rect:
x1,y1,w1,h1 = rect
x,y,w,h = x+x1,y+y1,w1,h1 cv2.imshow("Cleaned Plate",clean_plate)
cv2.imwrite("plate.jpeg", clean_plate) cv2.waitKey(0)
plate_im = Image.fromarray (clean_plate)
text = tess.image_to_string(plate_im, lang='eng') print "Detected Text : ",text
deftake_pic():
filename=str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"))+'.jpg'
camera = PiCamera(resolution=(1280, 720), framerate=30)
camera.iso = 100 time.sleep(5)
camera.shutter_speed = camera.exposure_speed camera.exposure_mode = 'off'
g = camera.awb_gains camera.awb_mode = 'off' camera.awb_gains = g
camera.capture(filename) process_car_plates(filename)
return filename
```

```
if__name__== '_main__':
print "Procesing plate . . ." img = cv2.imread(take_pic())
threshold_img = preprocess_image(img) contours= get_contours(threshold_img)
get_text_from_plate(img,contours)
```

Image Processing Challenges

iii. Circumvention

Vehicle owners have used a variety of techniques in an attempt to evade ANPR systems and road-rule enforcement cameras in general. One method increases the reflective properties of the lettering and makes it more likely that the system will be unable to locate the plate or produce a high enough level of contrast to be able to read it. This is typically done by using a plate cover or a spray, though claims regarding the effectiveness of the latter are disputed. Other users have attempted to smear their license plate with dirt or utilize covers to mask the plate.

iii) Plate inconsistency and jurisdictional differences

Many ANPR systems claim accuracy when trained to match plates from a single jurisdiction or region, but can fail when trying to recognize plates from other jurisdictions due to variations in format, font, color, layout, and other plate features

iv) External Influences

There are obvious legal and civil liberty concerns associated with tracking, recording and storing of license plate information.

v) Low image resolution

The prime difficulty is the low resolution of the number plates for vehicles in video frames under typical surveillance systems. Low resolution lead to empty number plates or mis-interpreted information.

The expected solutions for this problem is to develop the sequential coordination of image processing tasks. This processing sequence may include data filtering and image enhancing algorithms. object tracking, segmentation, locating the license plate area, detecting number and its color are also considered.

4.8 Car Parking System Slot Management

The car parking system slots will be checked and managed by the infrared sensors with a time interval of 2.5 seconds and a mathematical function derived.

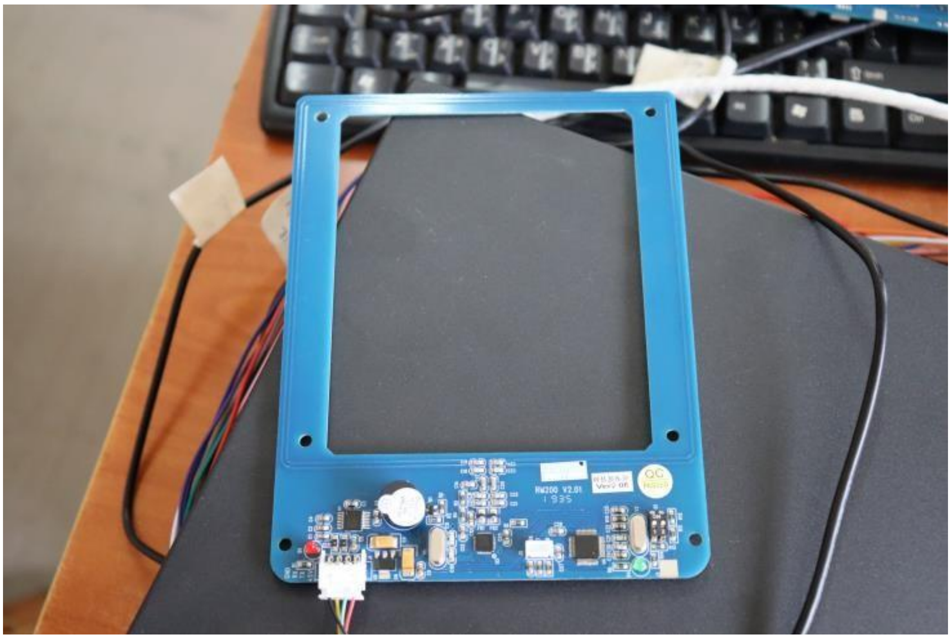
```
if av> 0.4:
return 1 else:
return 0
defspace_checker():
unpacked_spaces=[]
cs0=car_sensors(carsensor0)
cs1=car_sensors(carsensor1)
cs2=car_sensors(carsensor2)
cs3=car_sensors(carsensor3)
cs4=car_sensors(carsensor4)
cs5=car_sensors(carsensor5) if cs0==1:
pass else:
unpacked_spaces.append('cs0')
if cs1==1:
pass else:
unpacked_spaces.append('cs1')
if cs2==1:
pass else:
```

```
unpacked_spaces.append('cs2')

if cs3==1:

pass else:

unpacked_spaces.append('cs3')

if cs4==1:

pass else:

unpacked_spaces.append('cs4')

if cs5==1:

pass else:

unpacked_spaces.append('cs5')

if len(unpacked_spaces) > 0: return 1,unpacked_spaces else:

return 0,unpacked_spaces

def display(row1='->',row2='->'): try:

while True:

print("Writing to display") display.lcd_display_string(row1, 1)
```

Output: of IR Motion Sensor Detection

Figure 43

IR Motion Ssensor Detection



As you can see the space_checker() function calls the car_sensors() function with the parameter sensor pin which checks the state of the infrared sensor pin. If an object is detected a boolean true is returned and false if not.

All empty car sensor plates are then marked and printed at the lcd with the function display() as unparked parking slots and also keeps track of how many parking slots are available. This is achieved by calling len(unpacked_spaces).

The opening and closing of the gate Controlled by the function move_servo():

move_servo syntax has two parameters: servoPin and desired_angle.

The servoPin in the pin attached to the servo and what angle to move the servo in by parameter desired_angle.

This function too is used to to move the camera too to take a picture of an incoming car or outgoing car.

```
defmove_camera(servoPin,desired_angle):
pwm=GPIO.PWM(servoPin,50)
dc=1./18.*(desired_angle)+2
pwm.ChangeDutyCycle(dc)
```

```
defmove_camera(servoPin,desired_angle):
pwm=GPIO.PWM(servoPin,50)
dc=1./18.*(desired_angle)+2
pwm.ChangeDutyCycle(dc)
```

4.9 Car Parking Authetication Management Using Rfid

The system uses arduino to check and read RFID cards for security management to allow only authenticated users into the car parking garage. The info from the arduino is handled by the pyserial module.

```
#include <SPI.h> #include <MFRC522.h> #define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN)
void setup()
{
Serial.begin(9600); // Initiate a serial communication SPI.begin();// Initiate SPI bus
mfrc522.PCD_Init(); // Initiate MFRC522
Serial.println("Approximate your card to the reader...");
//Serial.println();
}
void loop()
{
 // Look for new cards
```

```
if ( ! mfrc522.PICC_IsNewCardPresent())
{
return;
}
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
{
return;
}
//Show UID on serial monitor
//Serial.print("UID tag :");
String content= "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
//Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
//Serial.print(mfrc522.uid.uidByte[i], HEX); content.concat (String
(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ")); content.concat
(String(mfrc522.uid.uidByte[i], HEX));
}
//Serial.println();
//Serial.print("Message : "); content.toUpperCase();
if (content.substring(1) == "F1 1F C5 19" || content.substring(1) == "D6 54 FE 1A" ||
content.substring(1) == "7D 54 AA C0" || content.substring(1) == "26 AE 59 D3")
//change here the UID of the card/cards that you want to give access
```

```
{
Serial.println("Authorized access");
//Serial.println(); delay(3000);
}
else {
Serial.println(" Access denied"); delay(3000);
}
}
```

AES 256 encryption/decryption code used in this researcher. import base64

```
from Crypto.Cipher import AES from Crypto import Random
from Crypto.Protocol.KDF import PBKDF2
BLOCK_SIZE = 16
pad = lambda s: s + (BLOCK_SIZE - len(s) % BLOCK_SIZE) * chr(BLOCK_SIZE
- len(s) % BLOCK_SIZE)
unpad = lambda s: s[:-ord(s[len(s) - 1:])]
password = —mys3cr3tp4ssw0rd‖ defget_private_key(password):
    salt = b"this is a salt"=
kdf   =   PBKDF2(password,   salt,   64,
    1000)key = kdf[:32]
    return key
def encrypt(raw, password):
private_key                              =
    get_private_key(password)   raw    =
    pad(raw)
    iv = Random.new().read(AES.block_size)
```

```
  cipher = AES.new(private_key, AES.MODE_CBC,
  iv) return base64.b64encode(iv +
  cipher.encrypt(raw))
def decrypt(enc, password):
private_key =
get_private_key(password) enc =
base64.b64decode(enc)
  iv = enc[:16]
  cipher = AES.new(private_key, AES.MODE_CBC,
  iv)return unpad(cipher.decrypt(enc[16:]))
encrypted = encrypt("NO.3261",
password)print(encrypted)
decrypted = decrypt(encrypted, password)
print(bytes.decode(decrypted))
```

4.10 To validate and verify the RFID microcontroller based real time media auto- streaming model for vehicle parking.

i) Formula for calculating Empty slots

To find if a parking slot is available we add each sensor value being high or low to an array for ten times then find the mode. The functions used in this thesis to find the mode is defined below;

```
 int mode(int a[],int n) { intmaxValue = 0, maxCount = 0, i, j;
   for (i = 0; i < n;
 ++i) {int count = 0;
```

```
for (j = 0; j < n; ++j) {if
  (a[j] == a[i])
  ++count;
}
if (count >maxCount) { maxCount = count; maxValue = a[i];
}
}
return maxValue;
}
intget_sensor_four(){ int state[10];
//adding all values to the array for(int i=0;i<10;i++){
val4 = digitalRead(inputPin4);
//Serial.println(val6); state[i]=val4; delay(12);
}
// pass the array to the mode function which returns the mode intmode_val =
mode(state,10);
if(mode_val==1){
//empty slot return 0;
}else{
//car parked
   return 1;
```

i. The RFID tag Authentication

This function searches for RFID tag number in the Database and send either ok or failed value to the slave from the master:

The Slave Code

```
intautheticate()
{

                                                  // listen for 10 secs for rfid card

  for(  uint32_t  tStart  =  millis();  (millis()-tStart)  <  period;  ){  if  (  !
   mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  String content= "";byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     //Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     //Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
    content.toUpperCase();
    Serial.println("rfid_check "+content.substring(1));
    uint32_t period = (1 * 60000L)/6;   // listen for 10 secs for rfid card
for( uint32_t tStart = millis(); (millis()-tStart) < period; ){ String

str=Serial.readStringUntil('\n');

Serial.println(str);

if(str=="rfid_ok"){ return 1;

}

}

}

return 0;

}
```

<u>Output: for the entry sensor indictor</u>

i. The master code

```
defautheticate_rfid(rfid_card):

            qry="SELECT * FROM car_parking WHERE rfid_id="+str(rfid_card)

            cursor.execute(qry)

            rows = cur.fetchall()

            if len(rows) > 0:

              # send rfid_ok to the slave through serial
```

```
ser.write('rfid_ok')

else:

#sends rfid_ok to the slave through serial

 ser.write('rfid_denied')/
```

i. Analytics on the Data captured

Each card number, number plate, arrival time, departure time is recorded and stored in the database for future analysis.

The system is also able to analyze very accurately the number of the parking slots available, this is achieved by:

```
<................snip 1........> int mode(int a[],int n) {
int maxValue = 0, maxCount = 0, i, j; for (i = 0; i < n; ++i) {
int count = 0;
for (j = 0; j < n; ++j) { if (a[j] == a[i])
++count;
}
if (count > maxCount) { maxCount = count; maxValue = a[i];
}
}
return maxValue;
}
void lcd_print_out(String row1,String row2){ lcd.clear();
lcd.print(row1); lcd.setCursor(0,1); lcd.print(row2);
//delay(1000);
}
int get_sensor_two(){ int state[10];
for(int i=0;i<10;i++){
val2 = digitalRead(inputPin2);
//Serial.println(val6); state[i]=val2; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
```

```
//car parked return 1;

}

}

int get_sensor_three(){

int state[10];

for(int i=0;i<10;i++){

val3 = digitalRead(inputPin3);

//Serial.println(val6); state[i]=val3; delay(12);

}

int mode_val = mode(state,10); if(mode_val==1){

//empty return 0;

}else{

//car parked return 1;

}

}

int get_sensor_four(){ int state[10];

for(int i=0;i<10;i++){

val4 = digitalRead(inputPin4);

//Serial.println(val6); state[i]=val4; delay(12);

}

int mode_val = mode(state,10);

if(mode_val==1){

//empty return 0;

}else{
```

```
//car parked return 1;

}

}

int get_sensor_five(){ int state[10];

for(int i=0;i<10;i++){

val5 = digitalRead(inputPin5);

//Serial.println(val6); state[i]=val5; delay(12);

}

int mode_val = mode(state,10); if(mode_val==1){

//empty return 0;

}else{

//car parked return 1;

}

}

int get_sensor_six(){ int state[10];

for(int i=0;i<10;i++){

val6 = digitalRead(inputPin6);

//Serial.println(val6); state[i]=val6; delay(12);

}

int mode_val = mode(state,10); if(mode_val==1){

//empty return 0;

}else{

//car parked return 1;

}

}
```

```
int get_sensor_seven(){ int state[10];
for(int i=0;i<10;i++){
val7 = digitalRead(inputPin7);
//Serial.println(val6); state[i]=val7; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}
int get_sensor_eight(){ int state[10];
for(int i=0;i<10;i++){
val8 = digitalRead(inputPin8);
//Serial.println(val6); state[i]=val8; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{ return 1;
}
}
//intialialize a string which is basically send to the master through serial
//the strings are then maniputale
String Car_parking_Values="";
String Car_parking_status="";
```

```
//form the string command by mode of sensor values
 int r2 = get_sensor_two();
if (r2){
//Serial.println("Car Parked");
Car_parking_status+="1 ";
Car_parking_Values+="S2 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
Car_parking_Values+="S2 ";
}
int r3 = get_sensor_three(); if (r3){
//Serial.println("Car Parked");
 Car_parking_status+="1 ";
Car_parking_Values+="S3 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
Car_parking_Values+="S3 ";
}
int r4 = get_sensor_four(); if (r4){
//Serial.println("Car Parked");
Car_parking_status+="1 ";
 Car_parking_Values+="S4 ";
}else{
```

```
//Serial.println("Empty Slot");

Car_parking_status+="0 ";

Car_parking_Values+="S4 ";

}

int r5 = get_sensor_five(); if (r5){

//Serial.println("Car Parked");

Car_parking_status+="1 ";

Car_parking_Values+="S5 ";

}else{

//Serial.println("Empty Slot");

Car_parking_status+="0 ";

Car_parking_Values+="S5 ";

}

int r6 = get_sensor_six();

if (r6){

//Serial.println("Car Parked");

Car_parking_status+="1 ";

Car_parking_Values+="S6 ";

}else{

//Serial.println("Empty Slot");

Car_parking_status+="0 ";

Car_parking_Values+="S6 ";

}

int r7 = get_sensor_seven(); if (r7){

//Serial.println("Car Parked");
```

```
Car_parking_status+="1 ";
Car_parking_Values+="S7 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
 Car_parking_Values+="S7 ";
}
int r8 = get_sensor_eight(); if (r8){
//Serial.println("Car Parked");
 Car_parking_status+="1 ";
Car_parking_Values+="S8 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
 Car_parking_Values+="S8 ";
}
//The final resulting string is send to the master for further analysis
Serial.println(Car_parking_status+','+Car_parking_Values);
//print empty parking slots
// waits for empty_slots from the master with a ten seconds timeout
uint32_t period = (1 * 60000L)/6;
for( uint32_t tStart = millis(); (millis()-tStart) < period; ){ String
str=Serial.readStringUntil('\n'); Serial.println(str);
if(str.startsWith("empty_slots")){ int commaIndex = str.indexOf(' ');
int secondCommaIndex = str.indexOf(' ', commaIndex + 1);
```

```
 String firstValue = str.substring(0, commaIndex);
String secondValue = str.substring(commaIndex + 1,
secondCommaIndex); String thirdValue = str.substring(secondCommaIndex + 1);
Serial.println(secondValue);
Serial.println(thirdValue);
//Prints the empty slots to the screen
 lcd_print_out (secondValue, thirdValue);
continue;
}
}
```

The sample of RFID Authentication Tags to be used for assigning the empty slots by giving them to the authorized drivers to park at the facility are has shown below in Figure 44.

Figure 44

RFID Authentication Tags

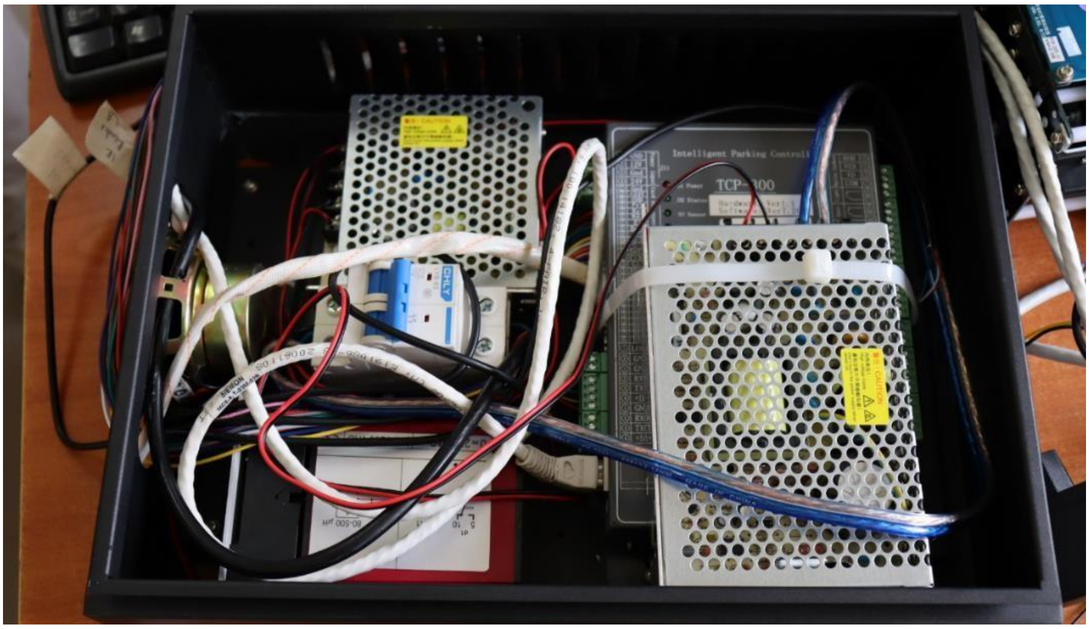
Snip 1 and Snip 2 is the analytical part for finding empty slots and forwards the values to the raspberry pi through serial the results of sensor values and then listens for responds from the master i.e display if any of the parking slots are available:

```
Serial.println (Car_parking_status+','+Car_parking_Values);
The master python snip codes def get_empty_slots(slot_status):
try:
available_slots=[]
available_slots_names=[]
for slot in slot_status.split(',')[0].split(' '): if len(slot) > 0:
available_slots.append(slot)
for slot_name in slot_status.split(',')[-1].split(' '): if len(slot_name) > 0 :
available_slots_names.append(slot_name)
return available_slots,available_slots_names[:-1]
except Exception as e:
return None , None
while True:
action = ser.readline().strip('\n')
a,b= get_empty_slots(action)
if a is not None or b is not None: try:
empty_slots_ints=[]
if a[1] == '1' and a[2]== '1' and a[3] =='1' and a[4] == '1'
and a[0]=='1':
ser.write('empty_slots No parking') print 'empty_slots No parking available'
else:
for slotx,spacex in zip(a,b): if slotx == '0':
```

```
empty_slots_ints.append(spacex)

ser.write('empty_slots '+

str(len(empty_slots_ints)) +' available')

#debug output

print 'empty_slots '+ str(len(empty_slots_ints))

+' available'
```

Figure 45

Output of the Microcontroller assembled Experiment



The function below captures the entrance and the exit time, by what car number plate and by which person through the RFID card reader followed by time analysis function used in the master.py

```
#this functions saves number plate, status(arrival or depature time) date and time of the rfid number

defsave_to_db(number_plate,rfid_number,status): time_now=datetime.datetime.now()

if status=="depart":

qry="INSERTINTOcar_parking
```

```
 (rfid_id,status,datetimestamp,number_plates)
VALUES
("+str(rfid_number)+",‘departure‘,"+str(time_now)+","+str(number_plates)+")";
cursor.execute(qry)
elif status=="arival":
qry="INSERTINTO  car_parking (rfid_id,status,datetimestamp,number_plates
VALUES ("+str(rfid_number)+",‘arrival‘,"+str(time_now)+","+str(number_plates)+")";
cursor.execute(qry)
#convert seconds to hours minutes and seconds for payment analysis def
secs2hours(secs):
mm, ss = divmod(secs, 60) hh, mm = divmod(mm, 60)
return "%d h %02d m %02d s" % (hh, mm, ss)
#get the time the user has spend in the car parking def get_user_hours(rfid_card):
qry="SELECT * FROM car_parking WHERE rfid_id="+str(rfid_card)
cursor.execute(qry)
rows = cur.fetchall() arrival_time=rows[-1][2][:-7]
ar_tm = datetime.datetime.strptime(a[:-7], '%Y-%m-%d %H:%M:%S') right_now=
datetime.datetime.now()
timedelta=right_now-ar_tm timeused=secs2hours(timedelta)
ser.write('lcd_print Card:'+str(rfid_card)+' '+str(timeused))
User against frequency graph code snippets #
def get_freq(user):
qry2="SELECT * FROM car_parking WHERE rfid_id="+str(user)
cursor.execute(qry2)
rows_users = cursor.fetchall() freq=int(len(rows_users)/2) form=str(user)+' '+str(freq)
```

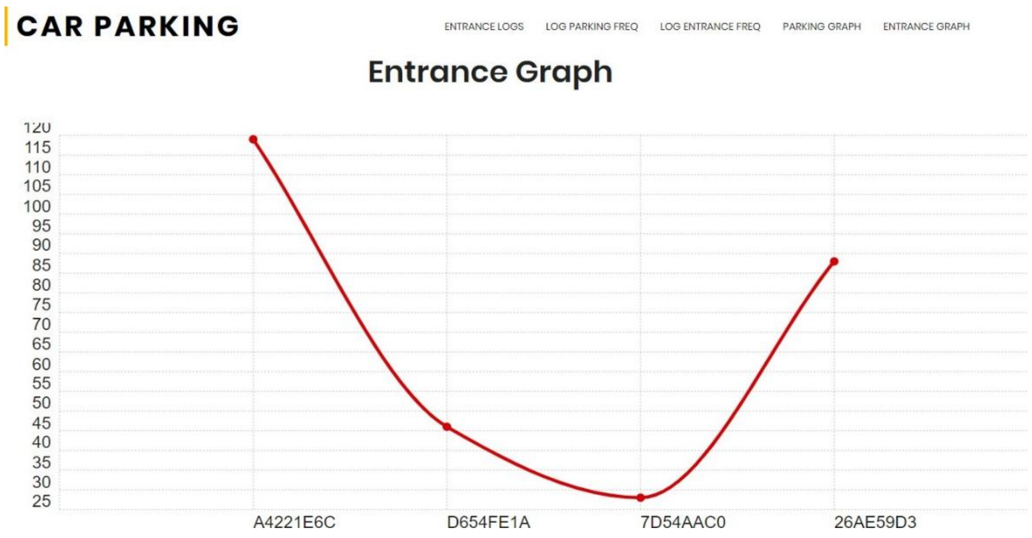
```
userfreq=tuple(form.split(' ')) return user,freq

def get_users_frequency(): u=[];f=[]

qry="SELECT * FROM car_parking" cursor.execute(qry)

rows = cursor.fetchall() users=set()

for row in rows:

users.add(row[0]) for user in users:

a,b=get_freq(user) u.append(a) f.append(b)

return u , f
```

Output

A graph frequency of users against number of visits in the car parking system.

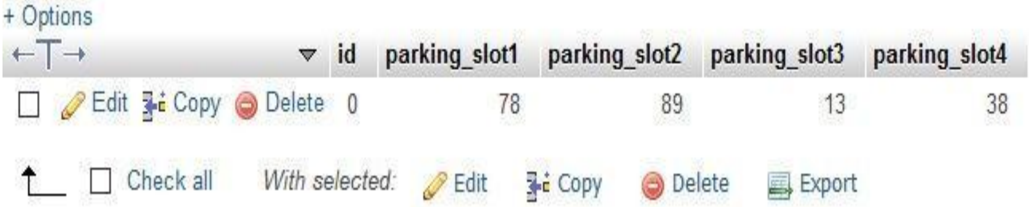
Figure 46

Slot frequency report line chart interface



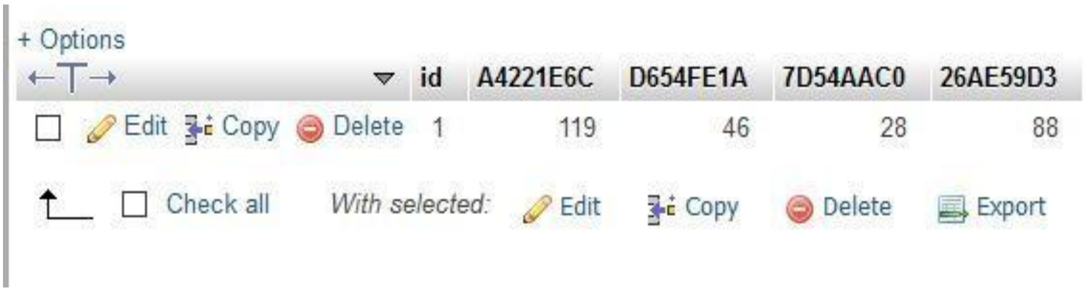
4.11 Datasets for storage

The databases were used to store data from cars. As shown below is an expert slot sectioning from the database.



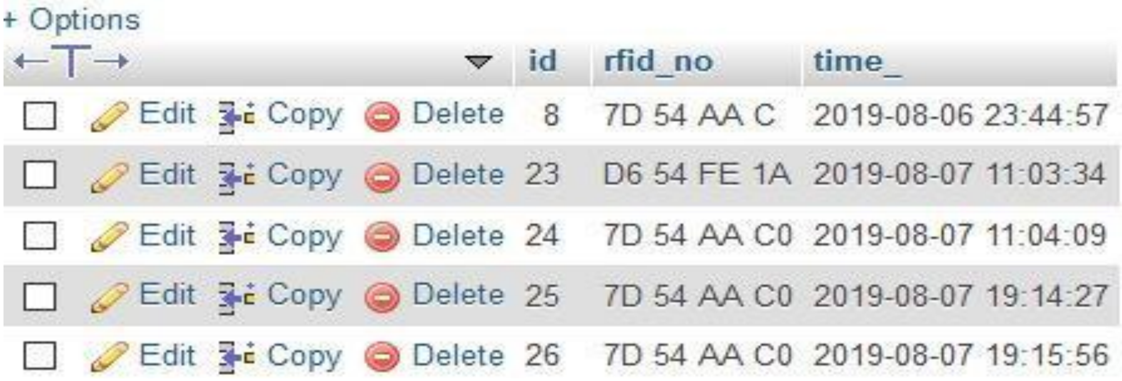
	id	parking_slot1	parking_slot2	parking_slot3	parking_slot4
□ Edit Copy Delete	0	78	89	13	38

RFID datasets that input data captured from tags is as shown below in this expert sample showing list of tags.



	id	A4221E6C	D654FE1A	7D54AAC0	26AE59D3
□ Edit Copy Delete	1	119	46	28	88

The following dagram below is illustration of datacapture of RFID sources associated with time and the tag taking the reading.



	id	rfid_no	time_
□ Edit Copy Delete	8	7D 54 AA C	2019-08-06 23:44:57
□ Edit Copy Delete	23	D6 54 FE 1A	2019-08-07 11:03:34
□ Edit Copy Delete	24	7D 54 AA C0	2019-08-07 11:04:09
□ Edit Copy Delete	25	7D 54 AA C0	2019-08-07 19:14:27
□ Edit Copy Delete	26	7D 54 AA C0	2019-08-07 19:15:56

To validate and verify the RFID microcontroller based real time media auto- streaming model for vehicle parking**.**

Validation does not imply verification, nor does verification imply validation. However, in practice, validation is often blended with verification, especially when measurement data is available for the system being modelled.

4.12 Embedded parking system testing summery Objectives

The Embedded testing summery Test supports the following specific objectives:

i. Ensure conformance to functional and non-functional requirement

ii. Prove necessary functions, communications, and operational interfaces

iii. Indicate how each testable need will be proven, including the test's procedure.

iv. Determine the outcomes for each test that will indicate success

v. Determine the resources needed to finish each test.

vi. List all the products that will be produced after the testing is over.

vii. Issues discovered during testing should be documented, along with a resolution plan.

viii. Ensure that problems are identified and fixed before go-live

The embedded system was constructed to meet the user needs and features identified in the functional system Operations and the requirements in the System Requirements. This testing plan is also supported by other system design documents, including System Architecture and Standards Plan.

4.13 Testing Approach

This section covers the overall approach to testing. Each significant group of features or feature combinations will be sufficiently tested due to the testing strategy. This section defines the essential procedures, methods, and equipment used to test the specified categories of features.

4.13.1 Testing Approach

Test Plan describes the approach for acceptance of the integrated Embedded system. Two methods were combined: a systems engineering V-model was used to develop system requirements, and agile development was used to deliver those requirements. A key aspect of documentation is software tools that enable development, tracking, and traceability throughout the process.

Agile development was selected to deliver the Electronic Parking Management (EPM) system to allow for iterative and incremental delivery and respond quickly to changes during the development process.

4.13.2 Test Deliverables

During testing, additional artifacts extended this plan to support and enhance the testing process. The following deliverables were the results of the testing process:

i. Test cases (efficiency and functional Requirements)

ii. Test scenarios

iii. Requirements Traceability Matrix

4.13.3 Test Types

The project had the following features: user accounts, parking reservations, parking availability, user preferences, parking amenities, and real-time data transmission. Test types include functional and non-functional testing to validate the requirements. Various test types were used in this to test each project's features. Each methodology has a defined test objective, strategy, and pass/fail criterion (expected result). The following are examples of test types to be used:

i. System testing – System testing involves testing the embedded system to verify that it meets specified requirements.

ii. Integration testing – reveals flaws in the interactions and interfaces of Integrated systems or components.

iii. Installation testing – Installation testing follows a set of procedures before the software and the embedded system can be used and tested.

iv. Unit testing – Unit tests are performed by the developer. Individual units or software and electronic components are tested to validate that each division or part performs as designed.

4.13.4 Verification Approaches

Each embedded system requirement requires a verification method, which details the plan for verifying the condition based on its stated definition. For each test case defined in the Requirements Traceability Matrix (RTM), one of the verification methods listed in Table 7 is assigned.

Table 7

Methods of Verification

Type	Description
Inspection	Verification through a visual, auditory, or tactile observation and comparison of observations with required attributes and characteristics of the system.
Demonstration	Verification that exercises the system software or hardware designed to be used, without external influence, to verify that the system behaves as specified by the requirement.
Test	Verification using controlled and predefined inputs and other external elements (e.g., data, triggers) that influence or induce the system to produce the output specified by the requirement.
Analysis	Verifying indirect and logical conclusions using mathematical analysis, models, calculations, and derived outputs based on validated data sets.

4.14 Test Criteria

4.14.1 Test Status

Each test case has several unique properties that should be taken into account while conducting the testing examination. Properties comprise, but are not limited to, test identifier (ID), test objective, procedure, expected outcome, number of test runs that must be completed, and status. The RTM maintains the following level for each test case:

i. Planned – The test case is known already, its role has been identified, testers have been assigned, and the test case is logged in the system as ready for testing.

ii. In progress –The test case is underway but has not been completed.

iii. Passed – A —passed‖ value indicates the defined number of tests that various testers have completed without error, and the expected result has been achieved. It is likely that each time this test is performed, independent of who is testing, the same successful results will be achieved.

iv. Failed – A test case is characterized as —failed‖ when it does not meet part or all of its expected outcomes. In this case, a defect would be logged, and a brief note would be entered in the comments column listing the defect ID for traceability. One or more defects must be reported for each failed test case to record the specifics of the failure and monitor its progress.

v. Deferred – A test case is characterized as —deferred‖ when it cannot be performed at the time of testing or when requirements change. If a test is deferred, the tester must explain briefly in the RTM's comments section. The test manager is in charge of keeping track of deferred cases and determining the best course of action to address them.

vi. Canceled- Test cases are labeled —canceled‖ when the associated requirement is no longer applicable to the project; that test case should be logged in the Change Request Log.

4.14.2 Entry/Exit microcontroller test

i. Entry criteria:

a. Successful completion

b. Test cases and scenarios added to the system logs Exit criteria:

ii. All planned test cases and scenarios have been executed

iii. Test scenarios achieve a 100% pass ratio

iv. All defects found have been recorded in the defect management tool

v. All high-severity faults have been resolved and retested

4.14.3 Testing Suspension and Resumption

There could be cases in which a critical, severe defect is detected that is significant enough that, if not addressed, it would require one or more iterations of the same tests to be performed again or could even require suspension of testing.

The following situations would cause testing to be suspended:

i. **App/ web crash** – Inability to execute a test case without crashing the app or web.

ii. **Inaccurate information** – Incorrect information associated with the parking availability, reservation, and/or payment information.

iii. **Network failure** – Failure of network connection required for regular operation**.**

iv. **Injury** – Any situation that could lead to bodily injury or significant damage to property

4.14.4 Test Tools

Requirements Traceability Matrix

An RTM was used to ensure that 100% of the requirements were tested. The RTM is a spreadsheet that links requirements to test cases. In coordination with other testers, the experiment engineer will be responsible for updating the RTM to indicate the status of test cases and record any functional defects, notes, or observations. Test status includes planned, in progress, passed, failed, deferred, and canceled. A test case may be deferred because (1) it is not possible to test specific functionality yet because of dependencies on requirements, or (2) the functionality is not available yet for testing, so testing must be deferred to a subsequent release. For the embedded system RTM, many of Release 3 is still in development, especially those that require integration into external systems such as the Common Payment System or Operating System. The schedule for testing Release 3 requirements is shown in Table 7 the Release 3 test cases are currently presented in a separate matrix in Table 8 since the test cases do not have test procedures.

A test case may be canceled if its written requirement is obsolete. Obsolete provisions must follow a change request process that explains why the condition is no longer needed. Canceled test cases do not count toward ensuring that 100% of the requirements are tested.

Defect Log

The test engineer will monitor the defect log for corrective action. A defect log will be used during testing to capture, track, monitor, and address anomalies observed during UAT. For each entry, the development team will work to understand and reproduce (where possible) the defects, identify the root cause, summarize a response, and log the activities undertaken to resolve the issue. Deferred tests must be assigned to a subsequent release to ensure 100% of the requirements are tested. The severity of the defect needs to be assessed, and the fault must be assigned to the category of critical, high, medium, or low, where ―critical‖ is the most severe classification, causing the feature or product to be unusable. Defects of this severity should be brought to the immediate attention of the test manager for further inspection, coordination, and decision-making. Low severity indicates a cosmetic observation.

The defect tracker will also be leveraged (with the RTM) to measure the feasibility and readiness of the software to be promoted to production. Additional information can be found in.

4.14.5 Features to be Tested

The following features and capabilities of EPM will be tested:

i. Traveler web portal

ii. User profile/account Individual Fleet

  a. Parking availability and restrictions
  b. Event parking
  c. Reservation

iii. Microcontroller embedded system

  a. Parking sensor functionality
  b. RFID functionality
  c. Camera integration functionality

iv. Embedded system integration

  d. Parking software
  e. Parking availability
  f. Performance efficiency measurement

Acceptance Test Cases

The EPM elements will be tested across four channels [Web, App (Apple and Android), Parking Operator Application, and system administrator]. The test cases in the tables below are crucial for the EPM system's acceptance since they assess the system's capacity to satisfy the criteria holistically, completely, and from the perspectives of all active participants (traveler, parking operators, City of Columbus, and system administrator).

The tables below list the Release 1 and 2 requirements that make up for the operational readiness for the EPM elements.

i. Test cases related to the Parking web portal are listed in **TABLE** 1

ii. Test cases related to microcontroller access are listed in TABLE 2

iii. Test cases related to RFID ACESS are listed in TABLE 3

iv. Test cases related to Parking AI CAMERA in TABLE 4

The test cases outlined below are reflective of the capabilities the system functionalities must be able to perform to receive the optimum efficiency.

Web Portal Test Cases

The web system was an essential channel for accessing real-time data to access and transfer. The following essential test cases were planned for the system.

Table 8

Web Portal Test Case

Test Case Id	Test Type	Verification Method	Feature	Test Objective	Metric
WEB002-V01	System	Demonstration	Real-time data capture	Verify that WEB requests could capture data in real-time. Logs. Camera video and pictures	Web loading datacapture PASS
Test Case Id	Test Type	Verification Method	Feature	Test Objective	Metric
WEB001-V01	System	Demonstration	Multi-language	Verify the login and components areloading	Web loading PASS

Test cases related to microcontroller access are listed in Table 9

Table 9

Test Cases Related to Microcontroller Access

Test Case Id	Test Type	Verification Method	Feature	Test Objective	Metric
Microcontroller -v01	Hardware	Demonstration	Signal transfer	Verify that the micro controller transfers signal in real-time	Integrated devices responding PASS

Test cases related to RFID ACESS are listed in Table 10

Table 10

Test Cases Related to RFID ACESS

Test CaseId	Test Type	Verification Method	Feature	Test Objective	Metric
Rfid Test - V01	Hardware	Demonstration	RFID Authenti Cation	Verify that The RFID securityallows real-time Authentication	RFID devices responding PASS

Test cases related to Parking AI CAMERA in Table 11

Table 11

Test Cases Related to Parking AI CAMERA

Test Case ID	Test Type	Verification Method	Feature	Test Objective	Metric
AI CAMERA-V01	HARDWARE	Demonstration	AI-Camera Video Picture Real-Time Data Capture	Verify That Ail Camera Real-Time Data Capture	AIL Camera devices responding PASS

4.15 Test Result Summary

This section identifies the test results for each test scenario, a defect managementtool, test metrics, exit criteria, and test results conclusions and sign-offs.

4.15.1 Test Results

a. EPM Test Case Report

This section identifies the test results for each of the test cases. The test cases were tofocus on the functionality of each essential function. All functions were throughout this process, any bugs, inconsistencies, errors, or the like that were detected were captured in the defect tool and reported to the development team. The development part modified, updated, and enhanced the system and embedded system to address issues during testing. Provides a detailed log of the final testing results for each test case.

Table 12

Test Case Results Matrix

Test ID	Function	Test Objective	MATRIX Status
1	Web system	System Functioning	PASS
2	Microcontroller	Hardware Functioning	PASS
3	RFID	Hardware Functioning	PASS
4	AI CAMERA	Hardware Functioning	PASS

b. Model Verification Process

Verification was intended to ascertain that the model does what it was intended to do. The verification process entailed checking that the model met the specifications. Evaluated items in the verification phase included plans, requirement specifications, design specifications, code, and test cases. A test plan outlined the test schedule phases and tasks. Deliverables from initial System Life Cycle Development (SLCD) stages like.

Requirement Specification Document (RSD), Functional Requirement Specification (FRS) and the Use Case helped develop a test plan. The FRS provided functional requirements specifications in a detailed manner. The Software Requirements

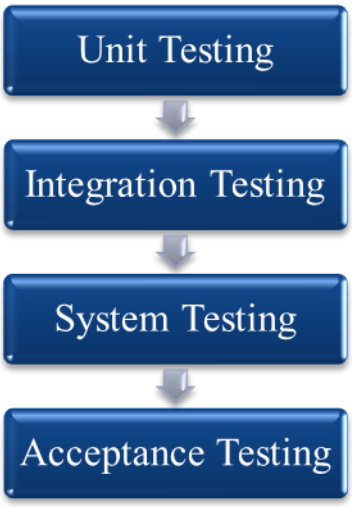
Specifications (SRS) provided information about the expected hardware, software, middle-ware, non-functional requirements properties. A test report for the model was produced at the end of this stage. Deliverables in this phase included test cases, test scripts, and test results.

i. Testing Strategy

A testing strategy is a strategy plan for defining the testing approach, what you want to accomplish and how you are going to achieve it. The testing strategy adopted during the verification of the developed model entailed undertaking unit testing, integration testing, system testing and acceptance testing successively as shown in Figure 47 below.

Figure 47

Testing Strategy



ii. Unit Testing

In the first phase of this testing strategy, individual units/ components of the model were tested. This the most important level of testing. Unit testing involved the initial testing of new and/or changed code in the model. The purpose of unit testing was to allow the developer to confirm the functionality provided by a single unit or component of code. Unit testing also helped validate that each unit performed as designed and fixed bugs early in development cycle. A unit was the smallest testable part of the model. Additionally, wherein one component of the model could not function without interacting with another component, the test included limited interactions. The developer of the unit was responsible for the creation of the unit test scripts in accordance to the unit test plan. They were also be responsible for execution of the test scripts and certifying that the unit testing was complete. The open box testing method was used for testing paths within units. In open box testing, testing commenced at an earlier stage of development and needed not wait for the availability of the (GUI) Graphical User Interface. Various valid and invalid inputs were chosen to exercise paths through the code and determine the appropriate output. Individual hardware components of the parking model were also tested. Unit testing couldn‘t be expected to catch every error in the model. Unit testing by its very nature focuses on a unit of code/component e.g. the RFID reader. Hence it can't catch integration errors or broad system level errors.

iii. Integration Testing

In the second testing phase, integration testing was performed to expose defects in the interfaces and in the interactions between the integrated components of the parking management model. The individual units of the model were combined and tested as a group. Integration testing is performed to expose defects in the interfaces and interactions between integrated components. This testing level used the gray box testing method, which is a combination of black box testing method and white box testing method. The structures for any two units/modules were studied (White Box Testing method) for designing test cases and actual tests are conducted using the exposed interfaces (Black Box Testing method). The Bottom Up approach was used when testing this phase. Bottom level units were tested first then upper level units were tested. This approach was adopted since the bottom-up development approach was also followed when developing the parking model. A guide outlining the Interactions between the various hardware and software units were defined through detail design specifications.

iv. System Testing

System testing is the third level of software testing. It was performed after integration testing and just before acceptance testing. System testing was performed to verify the model‘s compliance with the specified requirements. The complete and already integrated model was tested. The black box/ behavioral testing method were adopted. In this type of testing, the internal design/ structure/ implementation of the model was deemed not to be known by the tester thus testing was done without reference to the internal structures of the complete model. The parking management model had several modules e.g. the radio frequency identification reader, sensors and the software application. System testing was done by combining all the modules and testing it as a complete system is termed as System testing.

v. Beta/Acceptance Testing

The model was finally tested for acceptability. Any defects found while acceptance testing is considered as a failure of the Product. This evaluated the model‘s compliance to the current policies and procedures practiced in the parking areas. Acceptance testing was done with the reference to the expected deployment environment. This determined whether or not the parking model developed met the acceptance criteria and whether users and other industry stakeholders would accept it. Black box testing was used. It did not follow a strict procedure but rather an extemporaneous approach.

vi. Testing Environment

A test environment consists of elements that support test execution with software, hardware and their integration. The test environment configuration mimicked the production environment in order to uncover any environment/configuration related issues of the parking management model. Setting up a right test environment ensures software-testing success. Potential flaws such as insufficient fidelity to the actual system being tested may lead to the hardware and software under test to behave differently during testing than during production. Testing the prototype in a poor environment would have exposed the study to validity threats. Key area to set up included: -

i. HTTP Server- Xampp web server was used. Xampp is a free and open source cross-platform web server solution package developed by Apache Friends, consisting of the Apache HTTP Server, MariaDB database.

ii. Database server- A MariaDB database was part of the Xampp software installed.

iii. Operating System- The operating system is a low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals. Windows 10 operating system was used. A Microsoft window 10 is one operating system of a group of several graphical operating system families, all of which are developed, marketed and sold by Microsoft.

iv. Browsers: Netscape, Mozilla Firefox, Google Chrome, Internet Explorer and Opera web browser.

vii. Test Report and Defect Report

A defect report documents an anomaly discovered during testing. It includes all the information needed to reproduce the problem, including the author, release/build number, open/close dates, problem area, problem description, test environment, defect type, how it was detected, who detected it, priority, severity, status, etc. After uncovering a bug, a formal defect report was created. This report helped developers easily replicate the defect and fix it. Graphical representation using tables made the presentation better and easy to understand. The format of the defect report template used is shown in Table 12 while the results of the Validation are shown in the test report in Table 13.

Table 13

Model Validation Defect Report Template

ID	Unique identifier
Detected Build Version	*Build Version of model where the defect was detected*
Date and Time	*Date/time when defect was realized*
Module/Component	*Specific module where defect was detected*
Summary of defect	*Clear and concise summary of the defect*
Description	*Detailed description of the defect*
Steps to Replicate	*Step by step description to reproduce the defect*
Actual result	*The actual result received*
Expected Result	*The expected results*
Attachments	*Attach additional information e.g. logs, screenshots*
Severity	*Severity of defect (1-10)*
Priority	*Priority to be given to the defect (1-10)*
Reported by	*Name of person who reported the defect*
Assigned to	*Name of person assigned to fix the defect*
Status	*Status of the defect*
Fixed Build Version	*Build Version of model where the defect wasfixed*

Table 14

Validation Test Report

Attribute	Remarks	Result
Unit Testing		
Statements/ Expressions	Individual hardware components were examined for fault. Code statements and expressions were executed independently and the observed results reflected the expected results.	Successful
Integration Testing		
Functions and Integration	The model's independent hardware components interaction with other hardware components and software functionalities were executed and the outcome reflected our success criteria.	Successful
System Testing		
Conformity to Standards and Requirements	The complete and already integrated model was validated as a whole without reference to the internal structures of the complete model. The model was robust all vital system operations were achieved seamlessly.	Successful
Acceptance Testing		
Adoption and ease of use	User manuals were available and the model could easily be mastered. The steps to be followed when using the system were also optimized for simplicity and mimicked actual parking operations.	Successful

b. Model Validation Process

It is entirely possible that a product passes the verification but fails validation. This can happen when, say, a product is built as per the specifications but the specifications themselves fail to address the user‘s needs. Validation is the task of demonstrating that the model is a reasonable representation of the actual system and that it reproduces system behavior with enough fidelity to satisfy analysis objectives. Validation ensured that the product actually met the user‘s needs and that the specifications were correct in the first place it helped demonstrate that the model fulfilled its intended use when placed in its intended environment. Validation was not performed with the purpose to find fault or lay blame for incompetence or carelessness. Validation process in this research attempted to assess the effectiveness of the model in the management of parking lots. This validation phase helped: -

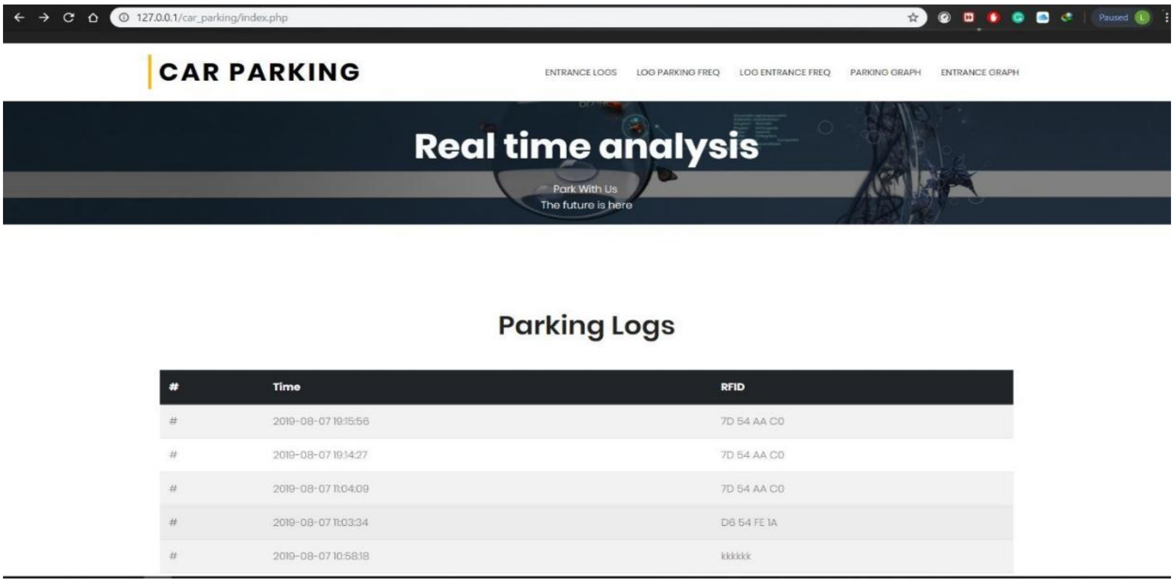
i. Review and validate the POC (Proof of Concept) results with the model designers, developers etc.
ii. Contrast the results to the success criteria in order to develop findings summary.
iii. Attain calibration on a stride forward ‘resolution and actualize an execution plan.
iv. Deliverables in the evaluate ‘phase included: evaluation model, findings summary and an execution plan.

i. Test Regime

The model testers validated the developed model for a period of four weeks. The users were required to first interact with the RFID components. RFID methods utilize radio waves to identify objects, collect data about them, and enter those data directly into computer systems with little or no human intervention. This model consisted of three basic RFID components: an RFID tag or smart label, an RFID reader and an antenna. RFID tags contained an integrated circuit, coupled with an antenna, they were used to transmit data to the RFID reader (also called an interrogator). The reader then converted the radio waves to a more usable form of data. Information collected from the tags was then transferred through a communications interface to a host computer system, where the data can be stored in a database and analyzed at a later time. Database analysis interface was clear, concise, familiar, responsive, consistent, attractive and efficient. The lead page mainly consisted of a navigation bar and a table interface to show all the parking log entries as shown in Figure 48 below.

Figure 48

Real time Analysis Report Interface



Parking Logs

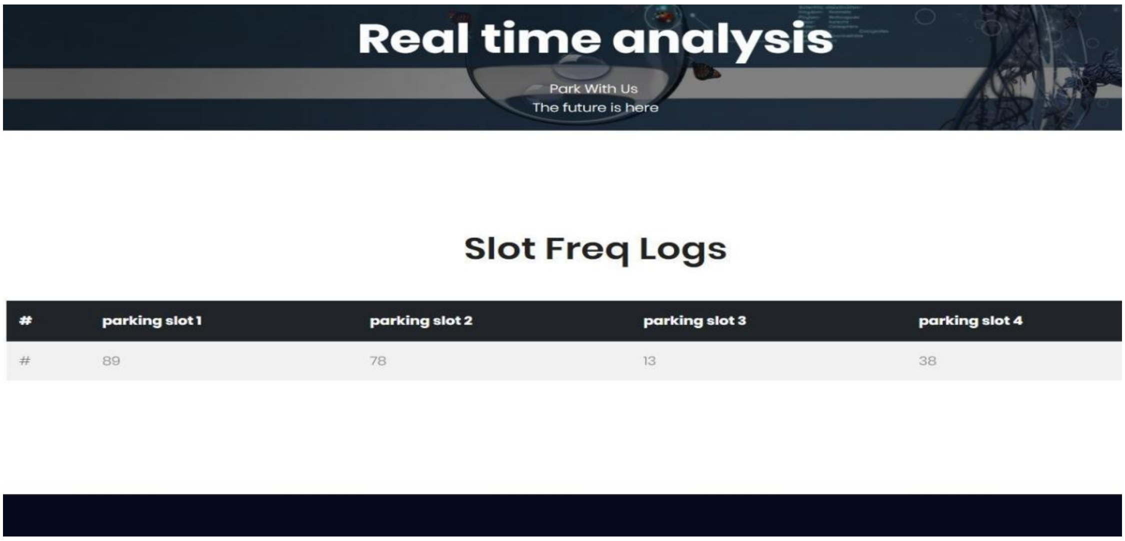
#	Time	RFID
#	2019-08-07 19:15:56	7D 54 AA C0
#	2019-08-07 19:14:27	7D 54 AA C0
#	2019-08-07 11:04:09	7D 54 AA C0
#	2019-08-07 11:03:34	D6 54 FE 1A
#	2019-08-07 10:58:18	kkkkkk

The parking entry logs were arranged in a descending order from the most recent to the earliest log entries. The navigation bar enabled people to interact with the model by providing meaningful links to all the models reports. If people could not figure out how the application works or where to go on the analysis platform they would get confused and frustrated.

Frequency tables are useful in describing the number of occurrences of a particular type of datum within a dataset. The model had an interface to show the slot frequency couners of all the parking slots as shown in Figure 48 below.

Figure 49

Slot Frequency Logs Report Interface



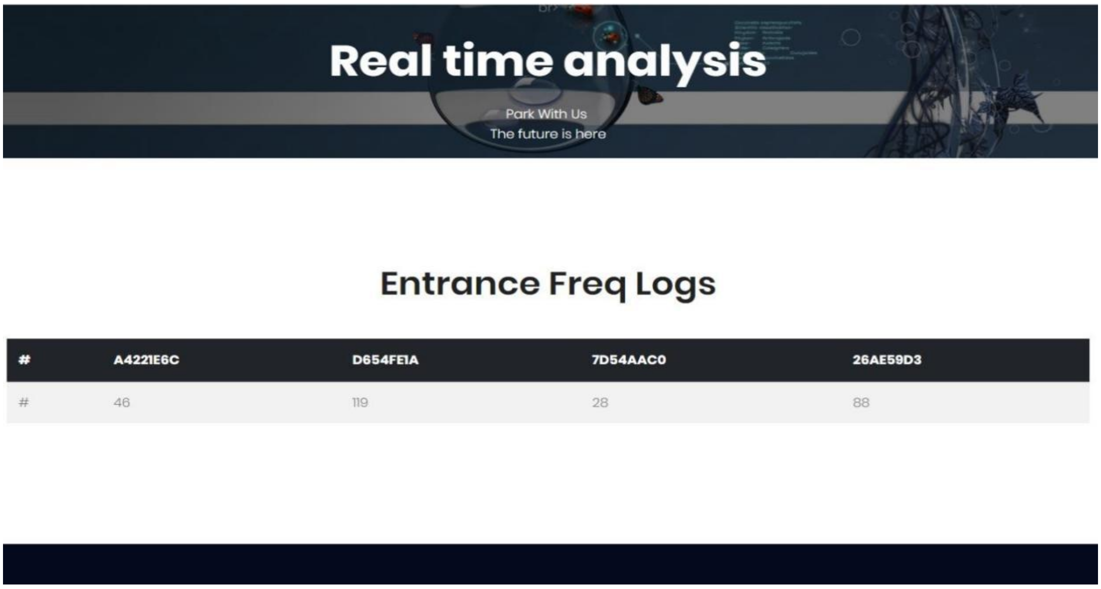
Slot Freq Logs

#	parking slot 1	parking slot 2	parking slot 3	parking slot 4
#	89	78	13	38

Slot frequency tables quickly revealed outliers and even significant trends within the parking slots with not much more than a cursory inspection. The parking slots used during this validation test were parking slot 1, parking slot 2, parking slot 3 and parking slot 4.. Another useful real-time frequency report that the model showed was the slot frequency counters of all the parking slots as shown in Figure 50 below.

Figure 50

Real Time Analysis Logs Report Interface



Entrance Freq Logs

#	A4221E6C	D654FE1A	7D54AAC0	26AE59D3
#	46	119	28	88

Slot frequency log reports helped examine the relative abundance of each particular target data within the sample RFID cards. Relative abundance represented how much of the data set was comprised of the target data. The RFID cards used during this validation test had the following numbers A4221E6C, D654FE1A, 7D54AAC0 and 26AE59D3. The model also had the capability to present the two realtime frequency reports in line charts. The slot frequency logs report chart and entrance frequency logs report charts interfaces are shown in Figure 50 and Figure 51 respectively below.

Figure 51

Slot Frequency Report Line Chart Interface

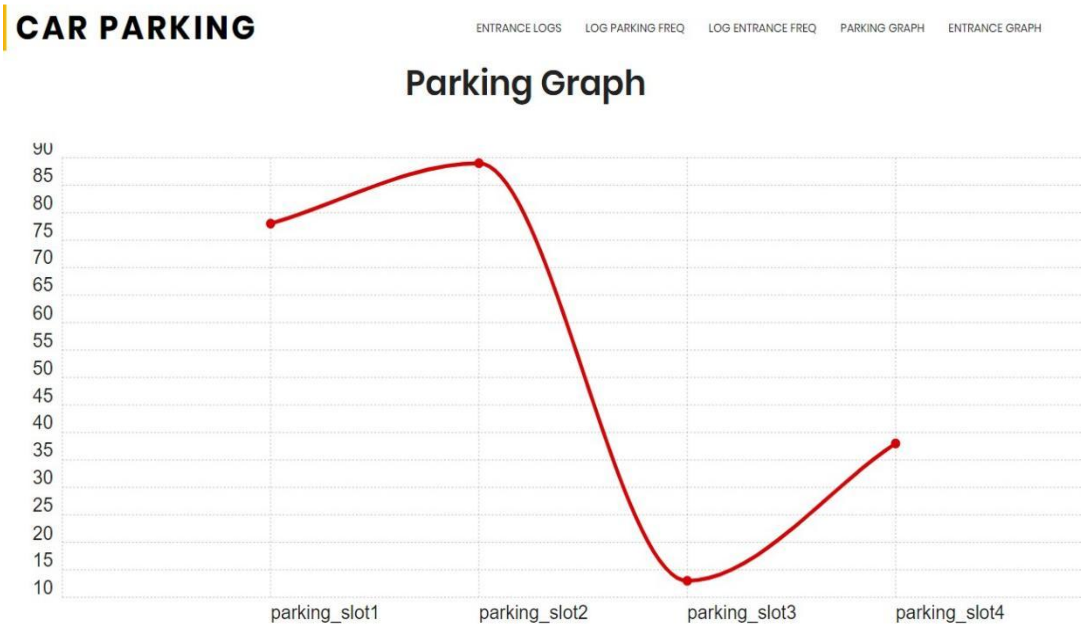
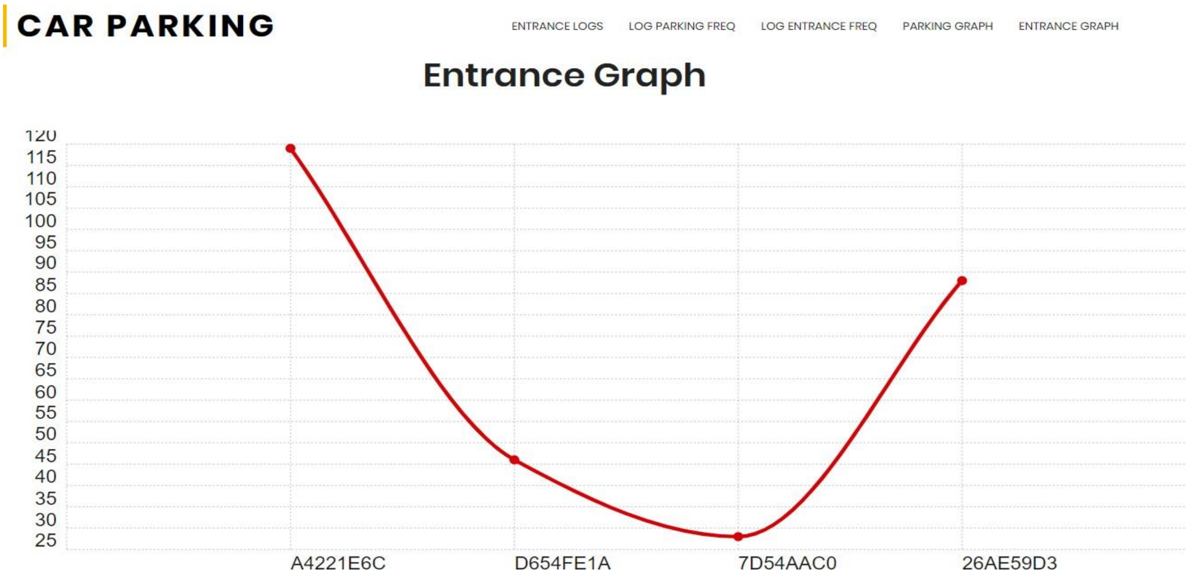


Figure 52

Entrance Frequency Line Chart Interface



The longitudinal aptitude of the line graph was instrumental in helping display the real-time data in an easily interpretable format. The frequencies for the tracked events were easily plotted along the y axis. Line graphs showed a graphic representation of the rise or fall of the data points. Missing data could also be plotted along the line with some degree of certainty or error probability.

The IPC uses ARM chips. An ARM processor is based on Reduced Instruction Set (RISC) architecture developed by Advanced RISC Machines (ARM). Essentially, the IPC is based on an ARM microcontroller that forms the projects functional requirement. At the center of the IPS, the ARM-compatible instruction set architecture produces outstanding performance by processing millions of instructions per second (Walls, 2012). ARM microcontrollers support few transistors and integrated circuits offering, leading to low power consumption. Walls (2012) affirmed that a key motivating factor for adopting a multicore design strategy in processor- based systems significantly reduces power consumption. According to the author, single-core processors have higher power expenditure despite the complex architecture to reduce power consumption. The projects ARM microcontroller is an excellent component relative to multicore embedded systems. It utilizes Ethernet with full TCP/IP stack and real-time embedded operating system (OS) design.

While the projects IPS may require minimal to no Internet connection, the system can run entirely offline and yield expected results. The controller can support non- contact type IC and ID cards; however, only IC cards are relevant to this project. The IC card dispensing system supports two types of users: a visitor and a VIP parker. While the visitor has to request a card at the entry point by pressing a button on the card system parking box and inserting it into the exit parking box, the VIP parker only swipes their registered card at entry and exit points.

Intelligent Parking Controller Modules

The ARM microcontroller forms the IPC's central design. Each controller has the following functioning modules:

i. IPC Power Supply

ii. Dial Code Switch

iii. TLD-110 Loop Detector

iv. Card Dispenser/Collector

v. Wiegand Reader Interface

vi. Notification Module

vii. Barrier Gate

viii. Ethernet with full TCP/IP stack

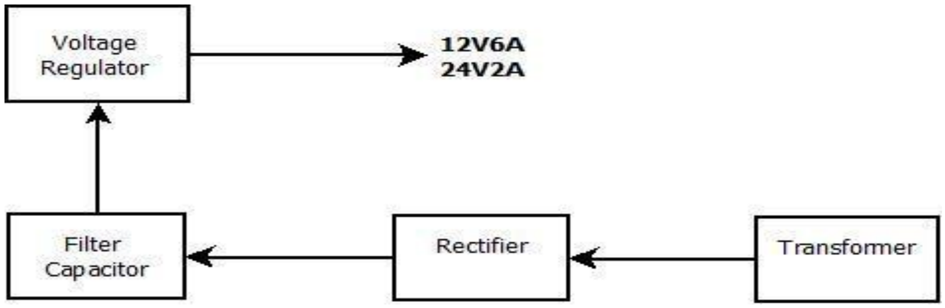
ix. Reset Button

i. IPC Power Supply

The IPC power supply module ensures a constant supply of electric power. J11 module on the controller registers power input from the power supply controlled by the power switch with an output of +12V6A, +24V2A. The regulated AC-DC power supply is necessary for stepping voltage up or down as needed to prevent damage to the controller. Essentially, the switching power supply allows the passage of correct voltage through a rectifier, minimizing output voltage ripples. Another overarching advantage of the switching AC-DC power supply is to avoid the dissipation of excess power, depicting energy efficiency. The figure below befits the power switch connecting to J11 (Power input) illustrated on the IPC.

Figure 53

Power Switch



i. Dial Code Switch

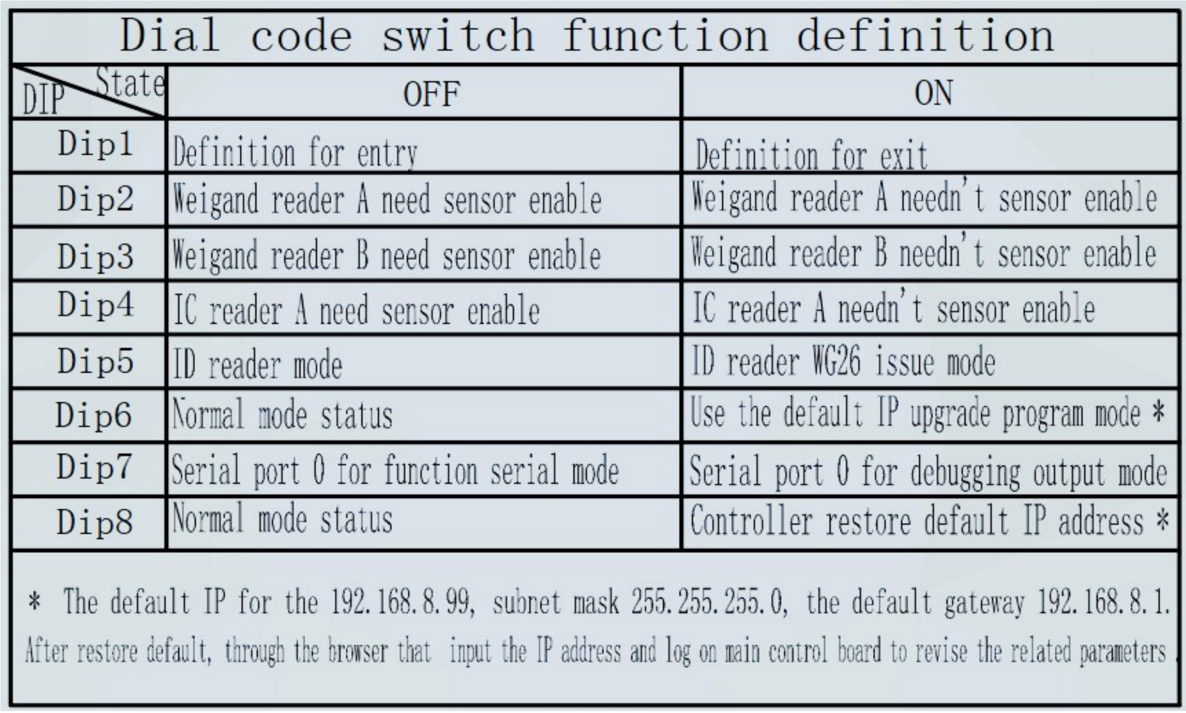
The dial code switch labeled S1 on the IPC has 8 DIP switches, each configured to perform specific functions. The DIP (dual in-line package) switch is a set of manual electric switches essential in controlling the flow of electricity around the IPC and other peripherals, changing the operating mode of devices. For example, Dip2, Dip3, and Dip4 control card dispensing and collecting. The DIP switches perform similar functions as jumper blocks; they both control the behavior of electric devices in specific situations, such as lighting LEDs and beeping buzzers.

Choosing DIP switches over jumpers was because they are quicker to change with no parts to lose (Westcott & Westcott, 2020), making them more suitable for the experimental purpose. Besides, the DIP switches are cheaper than programable chips and offer a permanent connection to the circuit board. However, the underlying compromise is that the DIP switches require manual configurations. J2 on the IPC illustrates manual configuration.

The DIP switch settings configure the IPC. Some DIP switches must perform multiple functions since the available number of I/O (or ON/OFF) lines is insufficient to assign each switch a unique purpose. Accordingly, the microcoulombs (μC) must read several DIPs switches, leveraging a multiplexer IC. The multiplexer IC interfaces multiple pins using a single pin, allowing resource sharing among several input signals. The multiplexer IC is essential in sharing one I/O port with multiple switches. For example, Dip3 is applicable in signal selection by the Wiegand reader interface and determining the time range for taking back the card at entry. Notably, these projects multiplexer IC was integral in sharing resources, not devices. The Figure 51 below defines the unique functions of each DIP switch.

Figure 54

Dial Code Switch Function Definition



Dial code switch function definition

DIP \ State	OFF	ON
Dip1	Definition for entry	Definition for exit
Dip2	Weigand reader A need sensor enable	Weigand reader A needn't sensor enable
Dip3	Weigand reader B need sensor enable	Weigand reader B needn't sensor enable
Dip4	IC reader A need sensor enable	IC reader A needn't sensor enable
Dip5	ID reader mode	ID reader WG26 issue mode
Dip6	Normal mode status	Use the default IP upgrade program mode *
Dip7	Serial port 0 for function serial mode	Serial port 0 for debugging output mode
Dip8	Normal mode status	Controller restore default IP address *

* The default IP for the 192.168.8.99, subnet mask 255.255.255.0, the default gateway 192.168.8.1.
After restore default, through the browser that input the IP address and log on main control board to revise the related parameters

ii. TLD-110 Loop Detector

Loop detectors connected to J15 (Sensor) are instrumental in tracking parking availability. The system uses a TLD-110 series inductive loop detector, a highly sensitive single-channel loop detector with 11 pins. Its responsiveness and adjustable sensitivity prevent false triggers caused by the difference in vehicle type and environmental conditions like temperature changes. The ARM microcontroller picks and evaluates output signals to control operating barriers at entry and exit points. The detector has a ground reference (Gnd), two relays, pins connecting to the power supply, and a loop.

When a vehicle enters the loop, it causes an increase in frequency, triggering the relay. However, the barrier cannot be lifted until the driver uses the IC card to access or exit the facility. The relay circuit output stays open or closed as long as a vehicle rests on the loop due to loop inductance changes. Frequency alteration involves the Dip6 switch to eliminate interference between wire loops, distribution cables, and loop detectors nearby.

The loop detectors sensitivity is adjustable to high (suitable for bicycles), medium (ideal for automobiles), and low (preferable for trucks). Its technical data is as follows:

i. Supply voltage AC: 220V, 110V

ii. Supply voltage DC: 24V, 12V

iii. Frequency range: 20 kHz to 170 kHz

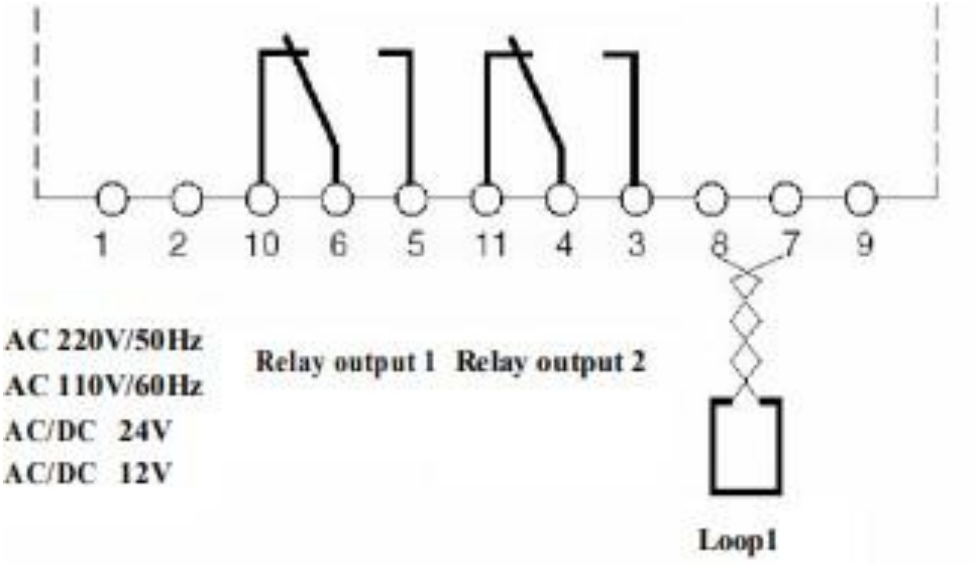
iv. Reaction time: 100ms

v. Loop inductance: Ideal:80µH to 300µH

vi. Operating temperature: -20°C to +65°C

Figure 55

Loop Detector Connection Diagram



iii. Card Dispenser/Collector

A card dispenser (at entry) and collector (at exit) each have 12 control pins for mechanically supporting and connecting electronic components. Though different in structural appearance, the card dispenser and collector share multiple technical similarities, including a working voltage of DC24V and a maximum current of 2A consistent with the controllers' power supply. Besides the design for automatic card dispensing/collecting, both devices provide RS232 (two RS232 reader interfaces) and I/O voltage interfaces. The RS232 standard is essential for serial communication. The dispenser/collector can accept all kinds of paper cards and PVC/Mylar cards at an operating temperature of -40°C to 85°C. When an error occurs or there is no card to dispense, the controller initiates an automatic buzz alarm.

Direct connectivity to the inductor (consistent with loop inductance changes due to the loop detector) ensures the dispensation/collection of one IC card per vehicle. Should the user fail to take the card within the pre-programmed duration, the dispenser will take back the card to prevent its loss; then, the user will have to press the ―Take card‖ button again. Dip1 and Dip3 on the dial code switch (S1) determine the time range for taking back the card at entry and rejecting the card at the exit. Dip4 controls quitting if the card is jammed. Essentially, Dip2, Dip3, and Dip4 are instrumental in card dispensing and collecting as they need sensor-enabling, which links to loop detection.

iv. Wiegand Reader Interface (IC Card Reader)

Wiegand technology provides the wired communication interface for interaction between the card reader (Wiegand output device) and the controller (RS232-based serial device) using an external power supply. It creates an access control mechanism, allowing the card reader to capture ID numbers from IC cards used in the IPS. The project uses two Wiegand reader interfaces. The Wiegand converter is suitable for Original Equipment Manufacturers (OEMs) that require Wiegand output from the IC cards (proximity cards). Conversion involves RS232 data (card reading) and RS485 data (LED display). The interfaces support Wiegand output reader format 26, 34, 44. The project‘s Wiegand output reader format 26, relevant to Bluetooth and proximity cards, uses a regulated power supply consistent with the IPC‘s voltage (+12V). The Wiegand reader interfaces utilize the Dip2 switch and Dip3 switch for signal selection, facilitating communication with the ARM microcontroller. Each interface leverages five pins with the following assignments:

i. Gnd (Pin1), black LED screen, no current/0V

ii. En (Pin2)

iii. Wg0 (Pin3), green LED screen

iv. Wg1 (Pin4), brown LED screen

v. 12V (Pin5), red LED screen

Figure 56

Wiegand Output Reader Format 26

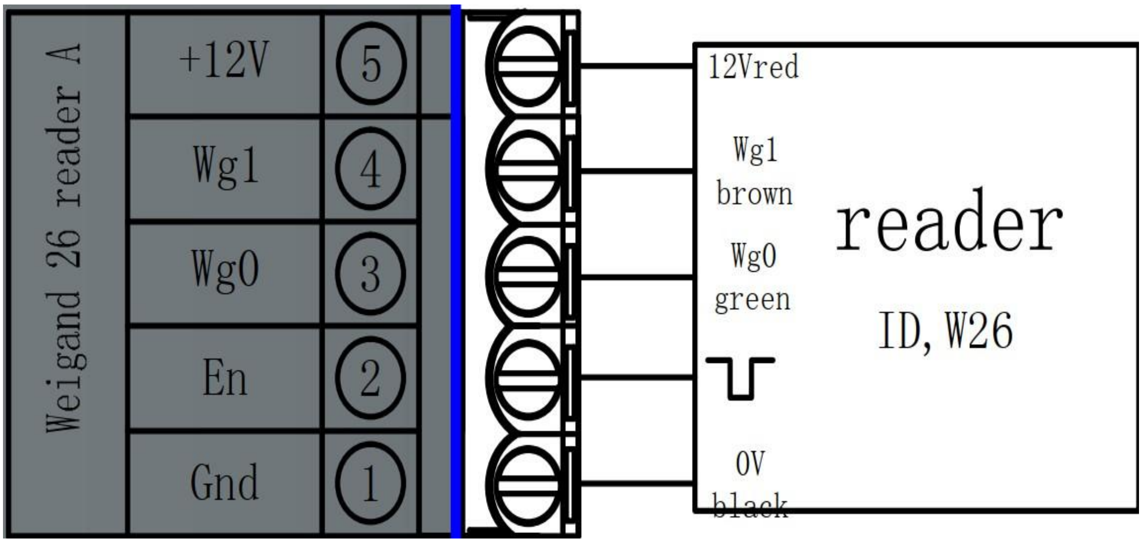
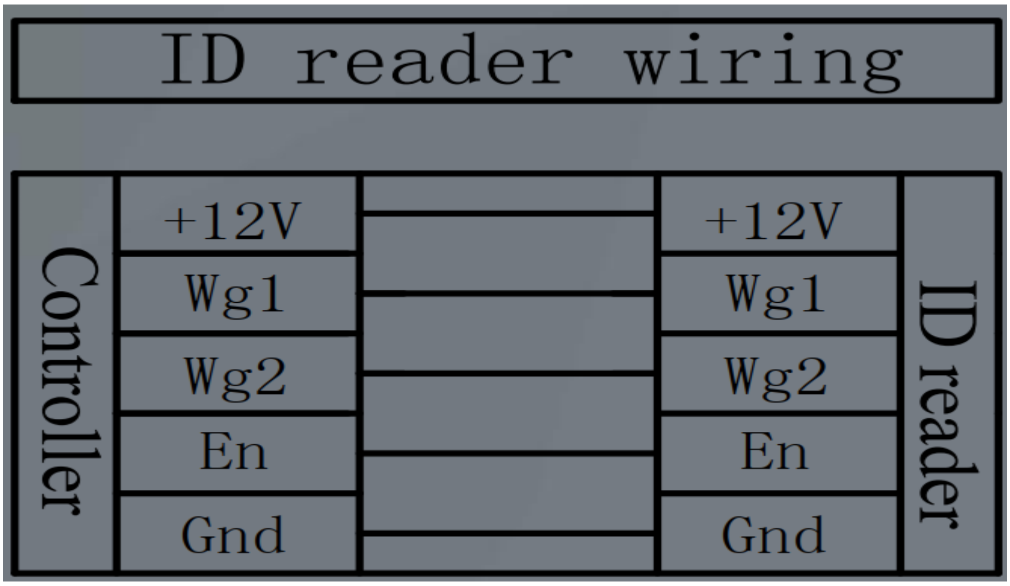


Figure 57

ID reader wiring for reading card ID numbers

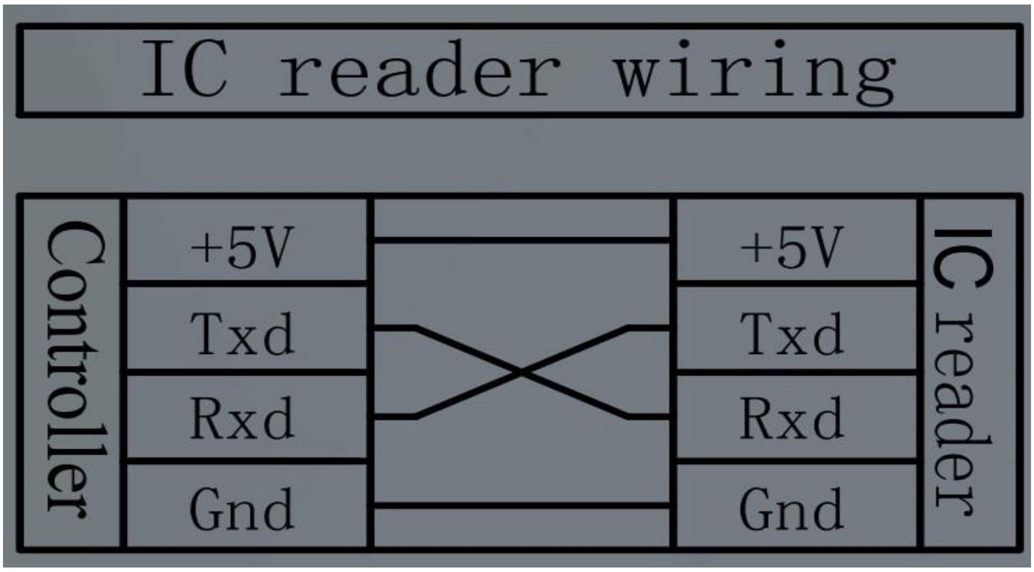


The controller module represented by J13 has three buttons, each connecting to two pins: Take card, Intercom, Take card 2. One button facilitates card issuing at the entry point (Take card), while another (Intercom) informs the system user during entry through an intercommunication device. Upon arrival at the entrance, the user presses a physical ―Take card‖ button on the card system parking box linked to the IPC to get an IC card.

J16 (IC reader A) and J17 (IC reader B) have two signal lines: 1) Txd (transmitted data) and Rxd (received data) and 2) Gnd (ground reference). The controller, through Rxd, receives and interprets transmitted data from an IC card to an IC reader on the parking box panel. Essentially, Rxd and Txd lines transmit received data and receive transmitted data across IC cards and IC readers, respectively. RS232 data transmission is integral in card reading. When a card is rejected, the buzzer unit comes in handy. The IC reader on the parking box panel links to J16 (IC reader A), while the card dispenser/receiver IC reader links to J17 (IC reader B). The IC readers use a 5VDC electromechanical relay instrumental in granting entry or exit. Control bits involve the Dip4 switch to determine pins assigned to Rxd/Txd. The Dip switch selects signals for the ARM microcontroller.

Figure 58

IC Reader Wiring



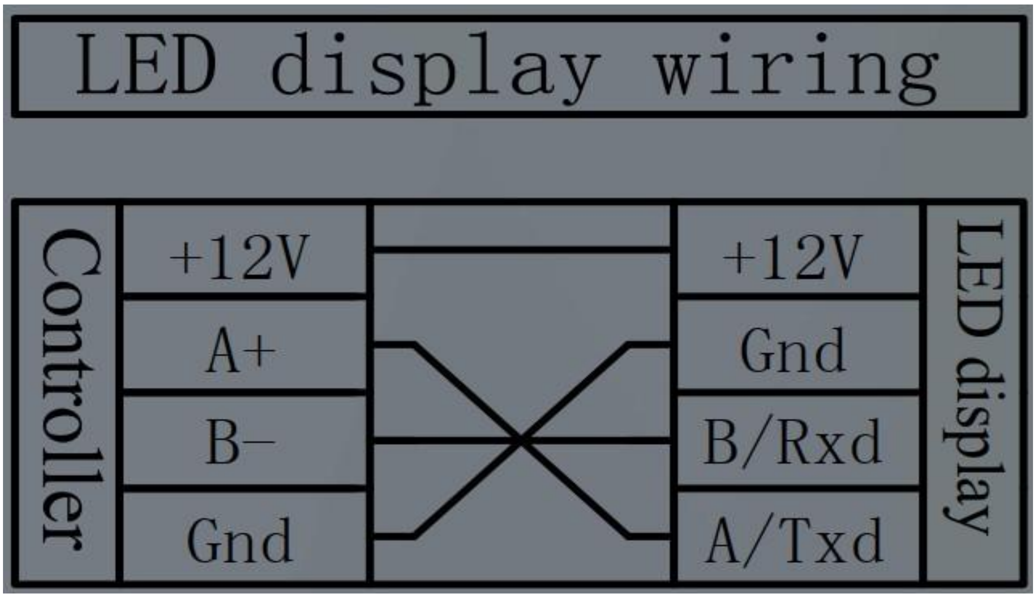
vi. Notification Module

The display interface includes Lights output (J8) and LED display (J4). J8 uses a 6-pin package to regulate the IPS lighting mechanism; this involves light display upon IC card rejection and acceptance during exit and entrance and light indicators linked to the loop detectors. For instance, pin 1 is responsible for the +12V power supply, pin 4 for lighting response when the gate barrier opens and closes, and pin 3 for loop detectors. Essentially, Lights output under J8 is an indicator is system responsiveness to motion and card reading. Motion involves loop detectors, while card reading involves Intercom (J5) with a 3-pin package. J5 used the RS232 standard to read card data and links to the LED display to notify the user accordingly using the RS485 standard. The entire notification module uses +12V, consistent with the IPC power supply.

The IPS exhibit different forms of light notifications. Small green and red LED light outputs blink as the user interacts with the system using an IC card (found at card system parking boxes and ticket system parking boxes). Some diodes detect the presence of a power supply. The diodes are D1, D2, D3, D4, and D5. Another form of light notification is the LED display (J4) that shows readings like the available number of parking slots through the car system parking box at the entrance. However, the entire green and red parking guidance LED display uses the RS485 communication with settings alterable by changing DIP based on S1 configurations.

Figure 59

LED Display Wiring (Display Notification)



The buzzer unit represented by J3 (Voice) has a 10W speaker. It has a 2-pin package powered by 5V DC for sounding the alarm when a vehicle breaks through the barrier without standard procedure. Pin 2 is connected to a transistor base to power the buzzer.

This Testing Process aims to conduct testing activities and mode of functionalities. The plan was to facilitate processes.

The primary goal was to evaluate how well the system conforms to the allocated requirements and satisfies the system of interest envisioned use and user needs. The evaluation will consist of inspection, analysis, demonstration, and testing of various functional systems and data to provide final acceptance of the system and allow movement forward to the next phase in the project. After addressing Reliability, real-time data communication, efficiency and circuit complexity, as explained earlier under validation.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

In this thesis, interfacing micro-controller and RFID towards a secure embedded car parking model is discussed. It guides drivers when parking their cars at the exact place in a specified period of time by simplifying the monitoring and also intelligence gathering of parking occupancy. The integrated RFID microcontroller functions as a vehicle parking meter, eliminating the need for parking guides directing cars on where to park.

5.2 Intelligent Car Parking Technology Interfacing Micro-Controller and RFID

According to already existing technological solutions, all of the sophisticated smart parking systems are proposed in academia (Fraifer & Fernström, 2016). The solutions agreed mostly depend on the knowledge of real-time parking information, based on which the system makes and apportions allocations for drivers. Current sensing technologies provide several options to monitor parking spots. In general, the design from this study provisions a guideline for designing any smart parking system that depends on sensors working either by sensors being deployed in each parking spot or by the construction of a network of wireless sensors with sink, which connects all sensors together and transmits sensing data to the gateway and then to the driver through GUI via the server. Typically, several Mesh networks supports multi- hop routing through which data packets can be relayed from one to another (Rashid & Rehmani, 2016). Thus, in these solution sensors provides a mechanism to relay signals from a distance.

5.2.1 Research Question 1: What are security components in existing technological solution for managing public parking lots?

The technologies that are used in administration and management of car parks include Automatic identification, or auto ID for short, is the broad term given to a host of technologies that are used to help machines identify objects. Auto identification is often coupled with automatic data capture (koh *et al*, 2003).

RFID (Radio Frequency Identification) is a means of identifying an item based on radio frequency transmission (Zumsteg and Qu., 2018). This technology can be utilized to identify, track and detect a wide variety of objects. Communication takes place between a reader and a transponder (derived from Transmitter/responder- Silicon Chip connected to an antenna), customarily called ―tag‖.

The use of far field backscatter modulation introduces problems that are not present in HF and lower frequency systems (Lopez and Conley, 2019).

Barcode is most commonly used in retail (the first UPC barcode was scanned on June 26, 1974 (Harris, 2009) and logistics as it is inexpensive to use (i.e. a barcode costs half a cent each‖ (Shih, 2009) and can be easily integrated into packaging; to this day barcodes are still in use in many different areas using RFID the embedded system was more efficient and with more security features compared to Barcode technology,

5.2.2 Research Question 2: How can an architectural model based on RFID microcontroller integrating real time media auto-streaming for vehicle packing be designed?

The design was created using a Qemu and simuli system running Raspbian on Windows 10. The source code provided for in this simulation did not deviate with the final model results even as much as there were various input sources from real data.

Using Qemu system and simuli was able to desighn the circuit block and do the simulation testing on simuli system.

5.2.3 Research Question 3: How can a prototype of an RFID microcontroller based real time media auto-streaming model for vehicle parking be implemented?

Foremost a block diagram for implementation was obtained from Qemu simulator as illustrated in block diagram of Figure 30 section 4.1.4 indicating all the embedded components. First camera configuration for taking images was done as discussed in section 4.4.1.4 Table 4. Secondly, a function to simulate if a car is parked sensor output was configured. The function returns a random value with 0 being no car and 1 being a car present. The function averages 5 input values for more certain results. If the input value of a car present averages more than 0.4 of 1 is considered a steady input of a car present else if it falls below the car is assumed not to be there. Also, realized the main script used in emulating the automatic license plate recognition as well as automatic parking system. The random module was used to emulate sensor values. This does not present the actual input from the hardware sensors such as infrared sensors. Import all modules needed for simulation included ours modified scripts. The scripts must be in the same folder as the main project.py

5.2.4 Research Question 4: How can a model of RFID microcontroller based on real time media auto-streaming model for vehicle parking validated and verified?

While the program is running, it checks if their exists empty parking slots and displays to the lcd necessary information, then checks the inputs of the check_for_outgoing_cars followed by check_for_incoming_cars input to check if there are cars at the entrance or exit of the parking system. Then depending on the output of the exit and entrance sensors, the camera is adjusted accordingly to either position with the vehicle, takes a picture for the number plate to be processed. Then the picture number plate text is processed and then stored at the database as well as the date time.

After automatic car number place licence recognition, the user is the prompted to place their RFID card, on detection off the card the gate is opened for 20 seconds to allow the car to pass and then jumps right back to the beginning and starts all the process again.

5.3 Concept Validity

The validity of the prototype implementation of the proposed real time media auto stream model interfacing micro-controller and RFID towards a secure embedded car parking model was described and validated. A simple proof of concept has been developed in order to demonstrate the validity of the solution. The considered scenario, discussed in chapter 4 above, illustrates a car mimic including an experimental board fitted with sensors at different locations: a permission scenario to emulate a real live scenario of car movements is achieved by the experiments.

In implementation, the car is equipped with a passive UHF RFID tag containing the Electronic Production Code (EPC) and information about the grant. In the prototypal implementation the Alien ALN-9654 G RFID tags have been used. This choice has been mainly done due to their extreme low-cost and compliance with the EPC standard. Let us observe that the aim of this research is to demonstrate the feasibility of just one of the several possible use-case scenarios of offering an efficient solution for car parking. Therefore, finding the best tag to use to detect a car is outside the scope of this work. Furthermore, a microcontroller interfacing UHF RFID reader, connected to cameras and sensors has been achieved by this study. Cards enabled with RFID Technology were used as portable reader.

5.4 Challenges in Car Parking Technologies

Accordingly with previous studies there is indeed a myriad of challenges in car parking. From findings in this study, it appears that there is no consensus on the best technological solution package, as each possible solution has deficiencies in one area or another. My findings in my academic research show that each solution has different technological requirements in relation to all the components that are integrated to provision solutions. Every approach has its advantages and disadvantages in terms of the following criteria: cost, reliability, scalability, accuracy, communication type, circuit complexity, reliability, method of operation, and ease of installation/usability of the system. Many types of research have influenced the ways in which the existing systems have been designed, measuring their performance and quality from the perspective of the designer, engineer, or developer, and not the stakeholders. The real time media auto stream model interfacing micro-controller and RFID towards a secure embedded car parking model bridges the gap by minimizing challenges experienced in car parks.

5.5 Areas of Further Research

Future research work would be the extension of this system by calculating the duration of stay of a vehicle in a parking lot as well as deduction of the parking charges on the basis of time spent. The tag will be recharged with a certain amount and this amount will be deducted at each visit. For realization of this a time recording technique is to be used. By using this kind of system the manual work will be minimized at a great extent. From a HCI perspective, there is an inadequate research base to inform fully how to design and evaluate smart parking systems for stakeholders; therefore, this issue requires further investigation.

An extension of this solution in order to allow stakeholders to undertake booking before arriving to parking bays will characterize future work.

5.6 Recommendation

In the light of the number of vehicles rising consistently and parking space is fast becoming a major issue in urban and semi urban cities so there is a need to design parallel parking. It is essentially a stacked car park. Instead of cars parking onthe streets, a more modern and a fast operating parking-lot system should be developed. In this study, a solution has been provided for the problems encountered in parking-lot management systems via RFID technology interfacing 8087 micro- controllers. The output of this research makes parking effective, convenient and safe. The RFID system is used to park the cars automatically in parking area.

The use of parking management system, micro-controllers, RFID tags, readers and antennas makes it easier to automate the 'in and out' privileges of parking subscribers. Personnel costs will be reduced considerably using this technology. Towards the future there is a need for scholars to upscale parking technology through unmanned, secure and atomized parking-lots functioning with RFID technology. Check-ins and checkouts should be handled in a faster manner without stopping the cars. By this the traffic congestion related issues should be avoided during parking. Drivers should not stop at the circulation points. The ticket jamming problems for the ticket processing machines should be avoided as well.

REFERENCES

A. P. (2018). Machine learning for Internet of Things data analysis: A survey. Digital Communications and Networks, 4(3), 161-175.

Adler, J. L., Satapathy, G., Manikonda, V., Bowles, B., and Blue, V. J. 2005. "A multi-agent approach to cooperative traffic management and route guidance," Transportation Research Part B: Methodological (39:4), pp 297-318.

Advanced Fuzzy logic technologies in industrial applications," Electronics and Information Engineering (ICEIE), 2010 International Conference On, IEEE2010, pp. V2-268-V262-272.

Aftab, H., Gilani, K., Lee, J., Nkenyereye, L., Jeong, S., & Song, J. (2019). Analysis of identifiers in IoT platforms. Digital Communications and Networks.

Agarwal, V., Murali, N. V., and Chandramouli, C. 2009. "A cost-effective ultrasonic sensor-based driver-assistance system for congested traffic conditions," IEEE transactions on intelligent transportation systems (10:3), pp 486-498.

Ahmed, S. (2018). *Integration of an RFID reader antenna with a work glove* (Master's thesis).

Ako, E. (2019). The Impact of Good Coding and Referencing on Warehouse Performance of Manufacturing Industries in Cameroon. *Available at SSRN 3403325*.

Alhammad, A., Siewe, F., and Al-Bayatti, A. H. Year. "An InfoStation-based context-aware on-street parking system," Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on, IEEE2012, pp. 1-6.

Al-Turjman, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. *Sustainable Cities and Society*, 101608.

Al-Turjman, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. Sustainable Cities and Society, 101608.

Awad, G., Jawale, S., Student, P. G., & EPS, A. P. (2017). Hardware Model of Single Phase Dynamic Voltage Restorer. *International Journal of Engineering Science*, *10520*.

Baichtal, J. (2018). *10 LED Projects for Geeks: Build Light-up Costumes, Sci-fi Gadgets, and Other Clever Inventions*. No Starch Press.

Baker, R. J. (2019). *CMOS: circuit design, layout, and simulation*. Wiley-IEEE press.

Balbo, F., and Pinson, S. 2005. "Dynamic modeling of a disturbance in a multi-agent system for traffic regulation," Decision Support Systems (41:1), pp 131-146.

Banerjee, S., Choudekar, P., and Muju, M. Year. "Real time car parking system using image processing," Electronics Computer Technology (ICECT), 2011 3rd International Conference on, IEEE2011, pp. 99-103.

Barata E. et al. (2011). ‗Parking at the UC campus: Problems and solutions'. Cities 28 (2011) 406–413

Başyazicioğlu, H. N., & Karamustafa, K. (2018). Marketing 4.0: Impacts of Technological Developments on Marketing Activities. Kırıkkale Üniversitesi Sosyal Bilimler Dergisi, 8(2), 621-640.

Bensky, A. (2019). *Short-range wireless communication*. Newnes.

Benson, J. P., O'Donovan, T., O'Sullivan, P., Roedig, U., Sreenan, C., Barton, J., Murphy, A., and O'Flynn, B. Year. "Car- park management using wireless sensor networks," Proceedings. 2006 31st IEEE Conference on Local Computer Networks, IEEE2006, pp. 588-595.

Bernhart, S., & Leitgeb, E. (2018, July). Evaluations of Low-Cost Decoding Methods for 1090 MHz SSR Signals. In *2018 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)* (pp. 1-5). IEEE.

Bilstrup, K., Uhlemann, E., Strom, E. G., and Bilstrup, U. Year. "Evaluation of the IEEE 802.11 p MAC method for vehicle-to-vehicle communication," Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th, IEEE2008, pp. 1-5.

Boda, V. K., Nasipuri, A., and Howitt, I. Year. "Design considerations for a wireless sensor network for locating parking spaces," Proceedings 2007 IEEE SoutheastCon, IEEE2007, pp. 698-703.

Bong, D., Ting, K., and Lai, K. 2008. "Integrated approach in the design of car park occupancy information system (COINS)," IAENG International Journal of Computer Science (35:1), pp 7-14.

Bong, D., Ting, K., and Lai, K. 2008. "Integrated approach in the design of car park occupancy information system (COINS)," IAENG International Journal of Computer Science (35:1), pp 7-14.

Busaba, F. Y., Gschwind, M. K., & Schwarz, E. M. (2017). *U.S. Patent No. 9,575,890*. Washington, DC: U.S. Patent and Trademark Office.

Cai, J. (2019). *U.S. Patent No. 10,325,128*. Washington, DC: U.S. Patent and Trademark Office.

Caicedo et al (2012). ‗Prediction of parking space availability in real time'. Expert Systems with Applications 39 (2012) 7281–7290

Caicedo F. (2009) ‗The use of space availability information in ‗‗PARC‖ systems to reduce search times in parking facilities'. Transportation Research Part C 17 (2009) 56–68

Caicedo F. (2010). ‗Real-time parking information management to reduce search time, vehicle displacement and emissions'. Transportation Research Part D 15 (2010) 228–234

Cairó, J. I., Bonache, J., Paredes, F., & Martín, F. (2018). Interference Sources in Congested Environments and its Effects in UHF-RFID Systems: A Review. IEEE Journal of Radio Frequency Identification, 2(1), 1-8.

Caliskan et al (2007). ‗Predicting parking lot occupancy in vehicular ad-hoc networks'. 65th IEEE Vehicular Technology Conference, VTC 2007, IEEE Press, Dublin, Ireland, 2007, pp. 277– 281

Cervantes, L. F., Lee, Y.-S., Yang, H., and Lee, J. Year. "A hybrid middleware for RFID-based parking management system using group communication in overlay networks," Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on, IEEE2007, pp. 521-526.

Chen Z. et al (2013). _Development of Fuzzy Logic Forecast Models for Location-Based Parking Finding Services'. Hindawi Publishing Corporation. Mathematical Problems in Engineering. Volume 2013, Article ID 473471, pages 6

Chen, F. C., & Jahanshahi, M. R. (2017). NB-CNN: Deep Learning-based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion. *IEEE Transactions on Industrial Electronics*.

Cheung, S. Y., Ergen, S. C., and Varaiya, P. Year. "Traffic surveillance with wireless magnetic sensors," Proceedings of the 12th ITS world congress2005, pp. 173-181.

Chinrungrueng, J., Sunantachaikul, U., and Triamlumlerd, S. Year. "Smart parking: An application of optical wireless sensor network," Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on, IEEE2007, pp. 66-66.

Chon, H. D., Agrawal, D., and El Abbadi, A. Year. "NAPA: Nearest available parking lot application," Data Engineering, 2002. Proceedings. 18th International Conference on, IEEE2002, pp. 496-497.

Chou, S.-Y., Lin, S.-W., and Li, C.-C. 2008. "Dynamic parking negotiation and guidance using an agent-based platform," Expert Systems with Applications (35:3), pp 805-817.

Doan, B. (2018). Radio frequency identification (RFID) technology and its impacts on logistics activities.

Dorsey, J., McCauley, N. P., Cummins, J. P., Monica, D., Quigley, O. S. C., & McKelvey, J. M. (2018). U.S. Patent No. 9,916,581. Washington, DC: U.S. Patent and Trademark Office.

Drozd, J., Drozd, A., & Antoshchuk, S. (2017). Green IT Engineering in the view of resource-based approach. In *Green IT Engineering: Concepts, Models, Complex Systems Architectures* (pp. 43-65). Springer International Publishing.

Druml, N., Krieg, A., Dielacher, M., LOBNIK, R., & Mark, M. (2019). *U.S. Patent Application No. 10/177,846*.

Düll, M., Haase, B., Hinterwälder, G., Hutter, M., Paar, C., Sánchez, A. H., & Schwabe, P. (2015). High-speed Curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. Designs, Codes and Cryptography, 77(2-3), 493-514.

Duroc, Y., & Tedjini, S. (2018). RFID: A key technology for Humanity. Comptes Rendus Physique, 19(1-2), 64-71.

Elfergani, I., Hussaini, A. S., Rodriguez, J., & Abd-Alhameed, R. (Eds.). (2018). Antenna Fundamentals for Legacy Mobile Applications and Beyond. Springer International Publishing.

Emergency Alerts through SMS.

Eppler, W. R., Erden, M. F., & Hernandez, S. (2019). *U.S. Patent Application No.10/311,897*.

Fabian, T. Year. "An algorithm for parking lot occupation detection," Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th, IEEE2008, pp. 165-170.

Fahmy, A., Altaf, H., Al Nabulsi, A., Al-Ali, A., & Aburukba, R. (2019). Role of RFID Technology in Smart City Applications. In 2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA) (pp. 1-6). IEEE.

Fereyre, P., Mayer, F., Douine, P., Ligozat, T., Prevost, V., & Diasparra, B. (2018). *U.S. Patent Application No. 15/538,294*.

Fernández Caballero, M. A. (2017). Interfaz de control can-bluetooth para robots Lego-NXT.

Finkenzeller, K. (2010). *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons.

Foschini, L., Taleb, T., Corradi, A., and Bottazzi, D. 2011. "M2M-based metropolitan platform for IMS-enabled road traffic management in IoT," IEEE Communications Magazine (49:11), pp 50-57.

Fraifer, M., & Fernström, M. (2016). Investigation of smart parking systems and their technologies. In *Thirty Seventh International Conference on Information Systems. IoT Smart City Challenges Applications (ISCA 2016), Dublin, Ireland* (pp. 1-14).

Fraifer, M., and Fernström, M. 2014. "Smart Car parking system prototype utilizing CCTVs," in Intel Ireland Research Conference 2014: At Dublin-Ireland.

Funck, S., Mohler, N., and Oertel, W. Year. "Determining car-park occupancy from single images," Intelligent Vehicles Symposium, 2004 IEEE, IEEE2004, pp. 325-328.

Ganesan et al. (2007). ‗Automated Parking Slot Allocation using RFID Technology'.1-4244-0779-6/07/$20.00 ©2007 IEEE

Geist, W. D., & Borras, J. A. (2018). U.S. Patent No. 9,864,882. Washington, DC:U.S. Patent and Trademark Office.

Geng et al (2012). ‗A new ―Smart Parking‖ System infrastructure and Implementation'. 15th meeting of EURO Working Group on Transportation, Elsevier Publications, Procedia – Social and Behavioral Sciences 54. Pages – 1278 – 1287.

Geng, Y., and Cassandras, C. G. 2012. "A new ―smart parking‖ system infrastructure and implementation," Procedia-Social and Behavioral Sciences (54), pp 1278-1287.

Ghassemlooy, Z., Popoola, W., & Rajbhandari, S. (2019). *Optical wireless communications: system and channel modelling with Matlab®*. CRC press.

Giuffre T. et al. (2012). ‗A novel architecture of parking management for Smart Cities'. Procedia - Social and Behavioral Sciences 53 (2012) 16 – 28

Grazioli et al. (2013). ‗Collaborative Mobile Application and Advanced Services for Smart Parking'. 2013 IEEE 14th International Conference on Mobile Data Management. 978-0-7695-4973-6/13

Gueaieb, W., and Miah, M. S. 2008. "An intelligent mobile robot navigation technique using RFID technology," IEEE Transactions on Instrumentation and Measurement (57:9), pp 1908-1917.

Guerra, R. (2013). Design and development of an embedded flash memory integrated simulator for the automotive microcontroller firmware validation.

Gupte, S., & Younis, M. (2018). Intelligent Parking Management in Smart Cities. *Transportation and Power Grid in Smart Cities: Communication Networks and Services*, 443-483.

Guri, M., & Bykhovsky, D. (2019). air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (ir). *Computers & Security*, *82*, 15-29.

Guru Prasad et al (2012). ‗Fiber Braggs Grating Sensor instrumentation for parking Space occupancy management'. 2013 international Conference on Optical Engineering.

Hakim, C. U., Surjati, I., & Nurwijayanti, K. N. (2017). PENGANTRIAN MEJA PADA RESTORAN MENGGUNAKAN FASILITAS SHORT MESSAGE SERVICE. *TESLA*, *15*(2), 140-152.

Hanif, N. H. H. M., Badiozaman, M. H., and Daud, H. Year. "Smart parking reservation system using short message services (SMS)," Intelligent and Advanced Systems (ICIAS), 2010 International Conference on, IEEE2010, pp. 1-5.

Haoui A. et al (2008). ‗Wireless magnetic sensors for traffic surveillance'. Elsevier journal. Transportation Research part C 16. Pages 294 – 306

Harbi, Y., Aliouat, Z., Harous, S., Bentaleb, A., & Refoufi, A. (2019). A Review of Security in Internet of Things. Wireless Personal Communications, 1-20.

Hasan, M., Islam, M., & Ahmed, S. R. (2019). RFID System Adaptivity in Supply Chain Management for Bangladesh. *World Scientific News*, *138*(2), 113-140.

Hockney, R. W., & Jesshope, C. R. (2019). *Parallel Computers 2: architecture, programming and algorithms*. CRC Press.

Holfelder,W. Year. "Vehicle-to-vehicle and vehicle-to-infrastructure communication: recent developments, opportunities and challenges," Workshop: Future Generation Software Architectures in the Automotive Domain, La Jolla2004.

Horng G. et al. (2012). ‗Using Cellular Automata on Recommendation Mechanism for Smart Parking in Vehicular Environments'. IEEE 978-1-4577-1415-3/12.

Hsieh, W.-H., Ho, C.-J., and Jong, G.-J. Year. "Vehicle information communication safety combined with mobile rfid system," Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on, IEEE2008, pp. 1021-1024.

Huayan S. et al. (2007). ‗Empirical Study of Parking Problem on University'. J Transpn Sys Eng & IT, 2007, 7(2), 135−140.

Hussein, A. E. (2009). Design and building of homemade 8051 micro controller emulator. Inf. Sci. 7, No. 1L, 55-59 (2013)

Ihlenburg, J., Steigerwald, J., & Schöppner, M. D. (2019). U.S. Patent Application No. 10/284,764.

Intel 2015. "Intel® Galileo Gen 2 Development Board."

Jacobus, C. J., Beach, G. J., & Rowe, S. (2018). *U.S. Patent Application No.15/439,301*.

Jian, M.-S., Yang, K. S., and Lee, C.-L. 2008. "Modular RFID parking management system based on existed gate system integration," WSEAS Transactions on systems (7:6), pp 706-716.

Jian, M.-S., Yang, K. S., and Lee, C.-L. 2008. "Modular RFID parking management system based on existed gate system integration," WSEAS Transactions on systems (7:6), pp 706-716.

John, D. O., Ahmad, A. A., Lawan, S., & Ladodo, A. R. (2019). Design and Implementation of a Venue Overcrowding Prevention System. *Advances in Electrical and Telecommunication Engineering (AETE) ISSN: 2636- 7416*, *2*(1), 17-28.

Jones, M. A., Jones, N. A., Vasgaard, A., & Wilkinson, B. W. (2018). *U.S. Patent Application No. 15/878,807*.

Jones, M. A., Jones, N. A., Vasgaard, A., & Wilkinson, B. W. (2018). U.S. Patent Application No. 15/878,807.

Joshi P. et al. (). ‗Global Review Of Parking Management Systems & Strategies‘. 2012 Northeast Decision Sciences Institute Conference Proceedings. Pages 477 – 496

Juffa, N., & Beebe, N. H. (2017). A Bibliography of Publications on Floating-Point Arithmetic.

Jung, H. G., Choi, C. G., Kim, D. S., and Yoon, P. J. 2006. "System configuration of intelligent parking assistant system," Proc. 13th World Congr. Intell. Transp. Syst. Services, London, UK, Oct), pp 8-12.

Jung, H. G., Kim, D. S., Yoon, P. J., and Kim, J. Year. "Light stripe projection based parking space detection for intelligent parking assist system," 2007 IEEE Intelligent Vehicles Symposium, IEEE2007, pp. 962-968.

Jung, H. G., Kim, D. S., Yoon, P. J., and Kim, J. Year. "Parking slot markings recognition for automatic parking assist system," 2006 IEEE Intelligent Vehicles Symposium, IEEE2006, pp. 106-113.

Kalyan, B. S., Singh, B., & Jalandhar, P. (2018). Quantum dot cellular automata (QCA) based 4-bit shift register using efficient JK flip flop. *International Journal of Pure and Applied Mathematics*, *118*(19), 143-157.

Kamath, A. M., Ambiel, C. A., Deshmukh, U., Castle, A. R., & Haverkate, C. M. (2019). *U.S. Patent Application No. 16/212,125*.

Kant, K. (2014). Microprocessors and microcontrollers. Place of publication not identified:

Karaksina, E. V., Shiryaev, V. S., Kotereva, T. V., Velmuzhov, A. P., Ketkova, L. A., & Snopatin, G. E. (2016). Preparation of high-purity Pr3+ doped Ge–As–Se– In–I glasses for active mid-infrared optics. *Journal of Luminescence, 177*, 275-279.

Khoukhi, A. Year. "An intelligent multi- agent system for mobile robots navigation and parking," Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on, IEEE2010, pp. 1-6.

Kianpisheh, A., Mustaffa, N., Limtrairut, P., and Keikhosrokiani, P. 2012. "Smart parking system (SPS) architecture using ultrasonic detector," International Journal of Software Engineering and Its Applications (6:3), pp 55-58.

Kilzer, K., Julicher, J., & Van Eeden, J. A. (2018). *U.S. Patent No. 9,946,482*. Washington, DC: U.S. Patent and Trademark Office.

Kim, Y. (2017). *WCET-Aware Scratchpad Memory Management for Hard Real-Time Systems* (Doctoral dissertation, Arizona State University).

Klappenecker et al. (2014). ‗Finding available parking spaces made easy'. Ad Hoc Networks 12 (2014) 243–249.

Klappenecker, A., Lee, H., and Welch, J. L. 2014. "Finding available parking spaces made easy," Ad Hoc Networks (12), pp 243-249.

Krishnan, A., Nadu, T., Nair, R. R., & Danithraj, N. (2018). Auto switching car door-lock system based on facial recognition using MATLAB.

Kuang, G., & Xu, H. (2018, October). Construction of Sports Competition Management System Based on Key Technologies of Internet of Things. In 8th International Conference on Management and Computer Science (ICMCS 2018). Atlantis Press.

Kumar, N. S., Saravanan, M., & Jeevananthan, S. (2011). Microprocessors and Microcontrollers.

Kumar, R., Chilamkurti, N. K., and Soh, B. Year. "A comparative study of different sensors for smart car park management," Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on, IEEE2007, pp. 499-502.

Lan et al. (2014). ‗An intelligent driver location system for smart parking'. Elsevier Publications, Expert Systems with Applications, pages – 2443 - 2456.

Lee, C., Han, Y., Jeon, S., Seo, D., and Jung, I. Year. "Smart parking system for Internet of Things," 2016 IEEE International Conference on Consumer Electronics (ICCE), IEEE2016, pp. 263-264.

Lee, S., Yoon, D., and Ghosh, A. Year. "Intelligent parking lot application using wireless sensor networks," Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on, IEEE2008, pp. 48-57.

Leephakpreeda T. (2007). ‗Car-parking guidance with fuzzy knowledge-based decision making'. Building and Environment 42 (2007) 803–809

Lele, A. (2019). Big Data. In Disruptive Technologies for the Militaries and Security (pp. 155-165). Springer, Singapore.

Li, C. H., Lao, K. W., & Tam, K. W. (2018). A Flooding Warning System based on RFID Tag Array for Energy Facility. In 2018 IEEE International Conference on RFID Technology & Application (RFID-TA) (pp. 1-4). IEEE.

Li, C.-C., Chou, S.-Y., and Lin, S.-W. Year. "An agent-based platform for drivers and car parks negotiation," Networking, Sensing and Control, 2004 IEEE International Conference on, IEEE2004, pp. 1038-1043.

Liang, W., Xie, S., Long, J., Li, K. C., Zhang, D., & Li, K. (2019). A Double PUF-based RFID Identity Authentication Protocol in Service-Centric Internet of Things Environments. Information Sciences.

Liu, H., Bolic, M., Nayak, A., and Stojmenovi, I. 2010. "Integration of RFID and wireless sensor networks," Encyclopedia On Ad Hoc And Ubiquitous Computing: Theory and Design of Wireless Ad Hoc Sensor, and Mesh Networks), pp 319-348.

LLC, R. J. 2016. RFID Journal LLC) 2016, p FAQ.

Longfei, W., Hong, C., and Yang, L. Year. "Integrating mobile agent with multi-agent system for intelligent parking negotiation and guidance," 2009 4th IEEE Conference on Industrial Electronics and Applications, IEEE2009, pp. 1704-1707.

Lopez, G., & Conley, D. C. (2019). Comparison of HF Radar Fields of Directional Wave Spectra Against In Situ Measurements at Multiple Locations. Journal of Marine Science and Engineering, 7(8), 271.

Lu, R., Lin, X., Zhu, H., and Shen, X. 2010. "An intelligent secure and privacy-preserving parking scheme through vehicular communications," IEEE Transactions on Vehicular Technology (59:6), pp 2772-2785.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth,

Mahmud, S., Khan, G., Rahman, M., and Zafar, H. 2013. "A survey of intelligent car parking system," Journal of applied research and technology (11:5), pp 714- 726.

Malagi, M. (2019). RFID based prepaid card for petrol station.

Mano, M. M. (2017). *Digital logic and computer design*. Pearson Education India.

Martínez-Sala, A. S., Egea-López, E., García-Sánchez, F., & García-Haro, J. (2009). Tracking of returnable packaging and transport units with active RFID in the grocery supply chain. *Computers in Industry*, *60*(3), 161-171.

Mateo, R. M. A., Lee, Y.-s., and Lee, J. Year. "Collision detection for ubiquitous parking management based on multi-agent system," KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Springer2009, pp. 570-578.

Mathur, S., & Panda, J. (2016). Microprocessors and microcontrollers. Delhi: PHI Learning.

Mathur, S., Kaul, S., Gruteser, M., and Trappe, W. Year. "ParkNet: a mobile sensor network for harvesting real time vehicular parking information," Proceedings of the 2009 MobiHoc S 3 workshop on MobiHoc S 3, ACM2009, pp. 25-28.

Mbacke, A. A., Mitton, N., & Rivano, H. (2018). A survey of RFID readers anticollision protocols. IEEE Journal of Radio Frequency Identification, 2(1), 38-48.

McCathie, L. (2004). The advantages and disadvantages of barcodes and radio frequency identification in supply chain management.

McKeon, R. (2018). *Neural networks for electronics hobbyists: a non-technical project-based introduction*. Apress.

Mei Z. (2012). ‗Analysis of Parking Reliability Guidance of Urban Parking Variable Message Sign System'. Hindawi Publishing Corporation. Mathematical Problems in Engineering. Volume 2012, Article ID 128379, 10 pages

Meurer, J., Stein, M., Randall, D., & Wulf, V. (2018). Designing for way-finding as practices–A study of elderly people's mobility. International Journal of Human-Computer Studies, 115, 40-51.

Mohammadi, S., Tavassoli, M., and Rajabi, A. Year. "Authoritative Intelligent Perfect Parallel Parking Based on Fuzzy Logic Controller for Car-Type Mobile Robot," Information Technology: New Generations (ITNG), 2011 Eighth International Conference on, IEEE2011, pp. 135-138.

Musey, J. A., & Keener, B. (2018). The Spectrum Handbook 2018. *The Spectrum Handbook.*

Nguyen, S., Salcic, Z., & Zhang, X. (2018). Big Data Processing in Fog-Smart Parking Case Study. In 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDC loud/SocialCom/SustainCom) (pp. 127-134). IEEE.

O'Flynn, B., Bellis, S., Delaney, K., Barton, J., O'Mathuna, S. C., Barroso, A. M., Benson, J., Roedig, U., and Sreenan, C. Year. "The development of a novel minaturized modular platform for wireless sensor networks," IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005., IEEE2005, pp. 370-375.

Olarewaju, I. K., Ayodele, O. E., Michael, F. O., Alaba, E. S., & Abiodun, R. O. (2017). Design and Construction of an Automatic Home Security System Based on GSM Technology and Embedded Microcontroller Unit. Computer Engineering, 1(1), 25-32.Oxford University Press, Inc..

Pala, Z., and Inanc, N. Year. "Smart parking applications using RFID technology," RFID Eurasia, 2007 1st Annual, IEEE2007, pp. 1-3.

Pala, Z., and Inanc, N. Year. "Smart parking applications using RFID technology," RFID Eurasia, 2007 1st Annual, IEEE2007, pp. 1-3.

Panayappan, R., Trivedi, J. M., Studer, A., and Perrig, A. Year. "VANET-based approach for parking space availability," Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks, ACM2007, pp. 75-76.

Park, W.-J., Kim, B.- S., Seo, D.-E., Kim, D.-S., and Lee, K.-H. Year. "Parking space detection using ultrasonic sensor in parking assistance system," Intelligent Vehicles Symposium, 2008 IEEE, IEEE2008, pp. 1039-1044.

Pedersen, M. H. R. (2017). Interfacing an SD Card with Patmos.

Petrie, G., & Toth, C. K. (2018). Introduction to laser ranging, profiling, and scanning. In *Topographic laser ranging and scanning* (pp. 1-28). CRC Press.

Pham, T. N., Tsai, M.-F., Nguyen, D. B., Dow, C.-R., and Deng, D.-J. 2015. "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies," IEEE Access (3), pp 1581-1591.

Pichorim, S., Gomes, N., & Batchelor, J. (2018). Two solutions of soil moisture sensing with RFID for landslide monitoring. Sensors, 18(2), 452.

Polycarpou E. et al. (2013). ‗Smart parking Solutions for Urban Areas'. 978-1-4673-5828-6/13/$31 ©2013 IEEE.

Pooley, D. (2019). U.S. Patent Application No. 10/177,608.

Praveen, B. L. N. (2017). *Bachelor of Technology In Electronics and Communication Engineering* (Doctoral dissertation, Jawaharlal Nehru Technological University, Kakinada). Prentice-Hall Of India.

Priyanka, S. S., Krishna, S. R., Durga, N. L., Kumar, G. S., Govardhan, S., & LIET, J. (2017). A High Speed Low Power Content Addressable Memory Design with A Parity Bit and Power Gated Ml Sensing using Match Line. *International Journal of Engineering Science*, *5921*.

Pullola, S., Atrey, P. K., and El Saddik, A. Year. "Towards an intelligent GPS-based vehicle navigation system for finding street parking lots," Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on, IEEE2007, pp. 1251-1254.

Qin et al. (2013). _An intelligent parking system based on GSM module'. Appl. Math.

Qureshi, S. T. (2018). *Passive UHF RFID Tag for Wireless Detection of Compression of Materials* (Master's thesis).

Rai, R. S., Madan, G., Prakash, K. R., & Krishna, G. R. (2019). Automation of Object Segregation.

Raj, P., Kumar, R., Sharma, N., & Puthoran, E. (2018). *Wireless Controlled Robot* (Doctoral dissertation, Jaypee University Of Information Technology, Waknaghat).

Raj, P., Kumar, R., Sharma, N., & Puthoran, E. (2018). *Wireless Controlled Robot* (Doctoral dissertation, Jaypee University of Information Technology, Waknaghat).

Ramya, N., Sandhya, U., & Gayathri, L. (2018, February). Biometric Authentication to ensure security in ePassports. In 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT) (pp. 342-346). IEEE.

Rashid, B., & Rehmani, M. H. (2016). Applications of wireless sensor networks for urban areas: A survey. *Journal of network and computer applications*, *60*, 192-219.

Recouly, M., CATALANOTTO, C., TALON, M., & Loussert, C. (2019). *U.S. Patent Application No. 10/289,878*.

Recouly, M., CATALANOTTO, C., TALON, M., & Loussert, C. (2019). U.S. Patent Application No. 10/289,878.

Rehman, M. H., Yaqoob, I., Salah, K., Imran, M., Jayaraman, P. P., & Perera, C. (2019). The role of big data analytics in industrial Internet of Things. Future Generation Computer Systems, 99, 247-259.

Reve, S. V., and Choudhri, S. 2012. "Management of car parking system using wireless sensor network," Int.J. Emerg. Technol. Adv. Eng (2), pp 262-268.

Rezaei, M., Park, M. S., Rabinowitz, C., Tan, C. L., Wheaton, S., Ulmer, M., & Mohseni, H. (2019). InGaAs based heterojunction phototransistors: Viable solution for high-speed and low-noise short wave infrared imaging. *Applied Physics Letters*, *114*(16), 161101.

Rhoads, G. B., Brunk, H. L., & Lord, J. D. (2018). *U.S. Patent Application No.10/027,868*.

Robinette, J., & Gernandt, T. (2016). *U.S. Patent No. 9,311,586*. Washington, DC: U.S. Patent and Trademark Office.

Rui, L. Y. U. (2019). *U.S. Patent No. 10,243,643*. Washington, DC: U.S. Patent and Trademark Office.

Selvini, T. (2018). Design of a logistics network in an omni-channel context: development of a conceptual model and implementation in a real case.

Shah, S. (2017). *Design and Implementation of a Scalable Real-Time Motor Controller Architecture for Humanoid Robots and Exoskeletons* (Doctoral dissertation, Virginia Tech).

Sharafi, M., and Nikpoor, S. Year. "Intelligent parking method for truck in presence of fixed and moving obstacles and trailer in presence of fixed obstacles:

Sharma, A., Chaki, R., and Bhattacharya, U. Year. "Applications of wireless sensor network in Intelligent Traffic System: A review," Electronics Computer Technology (ICECT), 2011 3rd International Conference on, IEEE2011, pp. 53-57.

Silva, B. N., Khan, M., & Han, K. (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society*, *38*, 697-713.

Singh, Harkishen. (2015). Voice Enabled Device Switching for Physically Challenged & Šolić, P., Marasović, I., Stefanizzi, M. L., Patrono, L., and Mainetti, L. Year. "RFID- based efficient method for parking slot car detection," Software, Telecommunications and Computer Networks (SoftCOM), 2015 23rd International Conference on, IEEE2015, pp. 108-112.

Smith, A. D., Rupp, W. R., & Manna, D. R. (2019). Case Studies of Green Supply Chains and Enabling RFID Technology.

Somervell, A. R. D., Alami, N., Cheung, D. C. L., Quill, C. S. J., & Payton, M. J. (2019). U.S. Patent Application No. 16/157,002.

Song, I., Gowan, K., Nery, J., Han, H., Sheng, T., Li, H., and Karray, F. Year. "Intelligent parking system design using FPGA," 2006 International Conference on Field Programmable Logic and Applications, IEEE2006, pp. 1- 6.

Stibor, L., Zang, Y., and Reumerman, H.-J. Year. "Evaluation of communication distance of broadcast messages in a vehicular ad-hoc network using IEEE 802.11 p," 2007 IEEE Wireless Communications and Networking Conference, IEEE2007, pp. 254-257.

Sun, J. S., Wu, C. H., Sie, Y. T., & Hsieh, M. H. (2018). Quasi-Yagi antenna with PBG structure for dual band RFID applications. In 2018 7th International Symposium on Next Generation Electronics (ISNE) (pp. 1-4). IEEE.

Sundstrom, K. E., & Iyer, J. (2018). U.S. Patent No. 9,881,186. Washington, DC: U.S. Patent and Trademark Office.

Takizawa, H., Yamada, K., and Ito, T. Year. "Vehicles detection using sensor fusion," Intelligent Vehicles Symposium, 2004 IEEE, IEEE2004, pp. 238-243.

Tan, H.-C., Zhang, J., Ye, X.-C., Li, H.-Z., Zhu, P., and Zhao, Q.-H. Year. "Intelligent car-searching system for large park," 2009 International Conference on Machine Learning and Cybernetics, IEEE2009, pp. 3134-3138.

Tanaka, Y., Saiki, M., Katoh, M., and Endo, T. 2006. "Development of image recognition for a parking assist system," Proc. 13th World Congr. Intell. Transp. Syst. Services), pp 1-7.

Tao, S., & Dubrova, E. (2017, March). Temperature aware phase/frequency detector-basec RO-PUFs exploiting bulk-controlled oscillators. In *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 686-691). IEEE.

Taraate, V. (2017). *PLD Based Design with VHDL: RTL Design, Synthesis and Implementation*. Springer.

Teodorovic D. et al (2006). ‗Intelligent parking systems'. European Journal of Operational Research 175 (2006) 1666–1681

Trangkanukulkij, R., Kim, T., & Kim, W. S. (2018). A 3D Printed Flexible Passive RFID for Temperature Sensing. In 2018 International Flexible Electronics Technology Conference (IFETC) (pp. 1-3). IEEE.

Tubaishat, M., Zhuang, P., Qi, Q., and Shang, Y. 2009. "Wireless sensor networks in intelligent transportation systems," Wireless communications and mobile computing (9:3), pp 287-302.

Ung, K. Y., Fredrickson, J., Kelsey, W. D., Yu, J. J., Li, E., & Smith, B. J. (2018). *U.S. Patent No. 10,163,046*. Washington, DC: U.S. Patent and Trademark Office.

Vargas, J., Schneider, K. K., Palmer, C., & Chamandy, P. A. (2017). *U.S. Patent Application No. 15/343,584*.

Vianna M. et al. (2004). ‗Intelligent transportation systems and parking management: implementation potential in a Brazilian city'. *Cities,* Vol. 21, No. 2, p. 137– 148, 2004

Vollmer, M., & Möllmann, K. P. (2017). *Infrared thermal imaging: fundamentals, research and applications*. John Wiley & Sons.

Von Mueller, C., Bellare, M., Yale, S., Hazel, P. K., & Catinella, P. E. (2018). *U.S. Patent No. 10,007,910*. Washington, DC: U.S. Patent and Trademark Office.

Waradkar, G., Ramina, H., Maitry, V., Ansurkar, T., Rawat, A., Das, P., ... & Sion, P.P. P. C. O. E. (2016). Automated room light controller with visitor counter. *Imperial Journal of Interdisciplinary Research (IJIR)*, *2*(4).

Wenzhi, C., and Bai, L. Year. "A smart roadside parking navigation system based on sensor networks for ITS," Wireless, Mobile and Multimedia Networks, 2006 IET International Conference on, IET2006, pp. 1-4.

Weston, D. C. (2018). U.S. Patent Application No. 15/907,734. Sadr, R. (2018). U.S. Patent Application No. 10/127,412.

White, B., Kamga, H., Kordyzon, M., Packheiser, F., Troy, J., Ufacik, E., ... & Lascu, O. (2017). *IBM Z Connectivity Handbook*. IBM Redbooks.

Xekalakis, P., Marcuello, P., Martinez, A. V., Kotselidis, C. E., Magklis, G., Latorre, F., ... & Gonzelez, A. (2018). *U.S. Patent No. 10,157,063*. Washington, DC:U.S. Patent and Trademark Office.

Xu et al (2013). ‗Real-time Street Parking Availability Estimation'. 2013 IEEE 14th International Conference on Mobile Data Management. 978-0-7695-4973-6/13$26.00 © 2013 IEEE

Xu, J., Chen, G., and Xie, M. Year. "Vision-guided automatic parking for smart car," Proceedings of the IEEE Intelligent Vehicles Symposium2000, pp. 725-730.

Yan, G., Olariu, S., Weigle, M. C., and Abuelela, M. Year. "SmartParking: A secure and intelligent parking system using NOTICE," 2008 11th International IEEE Conference on Intelligent Transportation Systems, IEEE2008, pp. 569-574.

Yan, G., Weigle, M. C., and Olariu, S. Year. "A novel parking service using wireless networks," Service Operations, Logistics and Informatics, 2009. SOLI'09. IEEE/INFORMS International Conference on, IEEE2009, pp. 406-411.

Yang, L., Rongguo, M., and Longfei, W. Year. "Intelligent Parking Negotiation Based on Agent Technology," Information Engineering, 2009. ICIE'09. WASE International Conference on, IEEE2009, pp. 265-268.

Ye, D., Wang, Y., Xiang, Y., Lyu, L., Min, H., & Shi, C. J. R. (2019). A Wireless Power and Data Transfer Receiver Achieving 75.4% Effective Power Conversion Efficiency and Supporting 0.1% Modulation Depth for ASK Demodulation. *IEEE Journal of Solid-State Circuits*.

Yeh, H.-T., Chen, B.-C., and Wang, B.-X. 2016. "A City Parking Integration System Combined with Cloud Computing Technologies and Smart Mobile Devices," Eurasia Journal of Mathematics, Science & Technology Education (12:5), pp 1231-1242.

Yousefi, S., Mousavi, M. S., and Fathy, M. Year. "Vehicular ad hoc networks (VANETs): challenges and perspectives," 2006 6th International Conference on ITS Telecommunications, IEEE2006, pp. 761-766.

Zadeh, L. A. 1996. Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, (World Scientific.

Zhai, J. (2014). *Efficient and reliable simulation, memory protection, and driver generation in embedded network systems* (Doctoral dissertation, Clemson University).

Zhang, L., and Wang, Z. Year. "Integration of RFID into wireless sensor networks: architectures, opportunities and challenging problems," 2006 Fifth international conference on grid and cooperative computing workshops, IEEE2006, pp. 463-469.

Zhang, Y., Gao, F., Fan, L., Lei, X., & Karagiannidis, G. K. (2018). Backscatter Communications Over Correlated Nakagami-$ m $ Fading Channels. IEEE Transactions on Communications, 67(2), 1693-1704.

Zhao, Y., and Collins, E. G. 2005. "Robust automatic parallel parking in tight spaces via fuzzy logic," Robotics and Autonomous Systems (51:2), pp 111-127.

Zheng, L., Song, C., Cao, N., Li, Z., Zhou, W., Chen, J., & Meng, L. (2018). A new mutual authentication protocol in mobile RFID for smart campus. IEEE Access, 6, 60996-61005.

Zheng, Y., and Cao, J. Year. "An intelligent car park management system based on wireless sensor networks," 2006 First International Symposium on Pervasive Computing and Applications, IEEE2006, pp. 65-70.

Zhu, B., Lei, Y., Peng, Y., & He, T. (2017). Low Latency and Low Error Floating-Point Sine/Cosine Function Based TCORDIC Algorithm. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *64*(4), 892-905.

Zhu, Y., Goel, M., & Bittlestone, C. (2017). *U.S. Patent No. 9,760,301*. Washington, DC: U.S. Patent and Trademark Office.

Zhu, Z., Zhao, Y., and Lu, H. Year. "Sequential architecture for efficient car detection," 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE2007, pp. 1-8.

Zumsteg, P., & Qu, H. (2018). U.S. Patent No. 9,892,289. Washington, DC: U.S. Patent and Trademark Office.

APPENDICES

Appendix I: Model Implementation

Master.py

```
import datetime
import sqlite3
import serial
import time
import numpy as npimport cv2
from copy import deepcopy
from PIL import Image
import pytesseract as tess

#initialize db connection

con = sqlite3.connect('parking_register.db3')
cursor = con.cursor()
#qry="CREATE TABLE users_analytics (rfid TEXT PRIMARY KEY, tot_time TEXT NOT NULL,);"
#cursor.execute(qry)

ser = serial.Serial('/dev/ttyACM0',9600)
def preprocess(img):
        #cv2.imshow("Input",img)
        imgBlurred = cv2.GaussianBlur(img, (5,5), 0)
        gray = cv2.cvtColor(imgBlurred, cv2.COLOR_BGR2GRAY)


        sobelx      =      cv2.Sobel(gray,cv2.CV_8U,1,0,ksize=3)

        #cv2.imshow("Sobel",sobelx)

        #cv2.waitKey(0)

        ret2,threshold_img                                         =
cv2.threshold(sobelx,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

        #cv2.imshow("Threshold",threshold_img)
        #cv2.waitKey(0)
        return threshold_img


    def cleanPlate(plate):
        print "CLEANING PLATE. . ."

        gray = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)

        kernel = cv2.getStructuringElement(cv2.MORPH_CROSS, (3, 3)) thresh=
        cv2.dilate(gray, kernel, iterations=1)



        _, thresh = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)

        im1,contours,hierarchy
cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
                                                                   =

        if contours:

            areas = [cv2.contourArea(c) for c in contours]

            max_index = np.argmax(areas)
```

```
max_cnt = contours[max_index] max_cntArea = areas[max_index] x,y,w,h =
cv2.boundingRect(max_cnt) if not ratioCheck(max_cntArea,w,h):

return plate,None

cleaned_final = thresh[y:y+h, x:x+w] #cv2.imshow("Function Test",cleaned_final)
return cleaned_final,[x,y,w,h]

else:

return plate,None

def extract_contours(threshold_img):

element = cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(17, 3))

morph_img_threshold = threshold_img.copy()

cv2.morphologyEx(src=threshold_img, op=cv2.MORPH_CLOSE, kernel=element,
dst=morph_img_threshold)

cv2.imshow("Morphed",morph_img_threshold) cv2.waitKey(0)

im2,contours, hierarchy=

cv2.findContours(morph_img_threshold,mode=cv2.RETR_EXTERNAL,method

=cv2.CHAIN_APPROX_NONE)

return contours

def ratioCheck(area, width, height):

ratio = foat(width) / float(height)

if ratio < 1:

ratio = 1 / ratio aspect = 4.7272
```

```
min = 15*aspect*15 # minimum area max = 125*aspect*125 # maximum area rmin = 3
rmax = 6
if (area < min or area > max) or (ratio < rmin or ratio > rmax): return False
return True
def isMaxWhite(plate):
avg = np.mean(plate) if(avg>=115):
return True
else:
return False
def validateRotationAndRatio(rect):
(x, y), (width, height), rect_angle = rect
if(width>height):
angle = -rect_angle
else:
angle = 90 + rect_angle
if angle>15:
return False
if height == 0 or width == 0: return False
area = height*width
if not ratioCheck(area,width,height): return False
else:
```

```
return True
def cleanAndRead(img,contours): #count=0
for i,cnt in enumerate(contours): min_rect = cv2.minAreaRect(cnt)
if validateRotationAndRatio(min_rect): x,y,w,h = cv2.boundingRect(cnt) plate_img = img[y:y+h,x:x+w]
if(isMaxWhite(plate_img)): #count+=1
clean_plate, rect = cleanPlate(plate_img)
if rect:
x1,y1,w1,h1 = rect
x,y,w,h = x+x1,y+y1,w1,h1 cv2.imshow("Cleaned Plate",clean_plate)cv2.waitKey(0)
plate_im = Image.fromarray(clean_plate)
text=tess.image_to_string(plate_im,
lang='eng')
print "Detected Text : ",text
img=cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
cv2.imshow("Detected Plate",img)cv2.waitKey(0)
return text
return None
#print "No. of final cont : " , count
def process_pic():
print "DETECTING PLATE . . ."
```

```
img = cv2.imread("4.jpg") threshold_img = preprocess(img)

contours= extract_contours(threshold_img)return cleanAndRead(img,contours)

 def get_empty_slots(slot_status)

 :try:

 available_slots=[] available_slots_names=[]

 for slot in slot_status.split(',')[0].split(' '): if len(slot) > 0:

 available_slots.append(slot)

 for slot_name in slot_status.split(',')[-1].split(' '): if len(slot_name) > 0 :

 available_slots_names.append(slot_name)

 return available_slots,available_slots_names[:-1]

 except Exception as e:

 return None , None

 def autheticate_rfid(rfid_card):

 qry="SELECT * FROM car_parking WHERE rfid_id="+str(rfid_card)
         cursor.execute(qry)

 rows = cursor.fetchall() if len(rows) > 0:

 ser.write('rfid_ok')

 else:

 ser.write('rfid_denied')

 def secs2hours(secs):

 mm, ss = divmod(secs, 60) hh, mm = divmod(mm, 60)

 return "%d h %02d m %02d s" % (hh, mm, ss)

 def get_freq(user):

 qry2="SELECT * FROM car_parking WHERE rfid_id="+str(user)
         cursor.execute(qry2)
```

```
rows_users = cursor.fetchall() freq=int(len(rows_users)/2) form=str(user)+' '+str(freq)
        userfreq=tuple(form.split(' ')) return user,freq

        def get_users_frequency(): u=[];f=[]

qry="SELECT * FROM car_parking" cursor.execute(qry)

rows = cursor.fetchall() users=set()

for row in rows:

users.add(row[0]) for user in users:

a,b=get_freq(user) u.append(a) f.append(b)

return u , f

def get_user_hours(rfid_card):

#dbcur.execute("INSERT INTO av_md5 (virus_md5) VALUES (?);", (line.strip('\n'),))

qry="SELECT * FROM car_parking WHERE rfid_id="+str(rfid_card)
        cursor.execute(qry)

rows = cursor.fetchall() arrival_time=rows[-1][2][:-7]

ar_tm = datetime.datetime.strptime(a[:-7], '%Y-%m-%d %H:%M:%S') right_now=
        datetime.datetime.now()

timedelta=right_now-ar_tm timeused=secs2hours(timedelta)

ser.write('lcd_print Card:'+str(rfid_card)+' '+str(timeused)) #time.sleep(3)

        def save_to_db(number_plate,rfid_number,status):
        time_now=datetime.datetime.now()

if status=="depart":

qry="INSERT INTO car_parking (rfid_id,status,datetimestamp,number_plates)
                        VALUES
        ("+str(rfid_number)+",'departure',"+str(time_now)+","+str(number_plates)+")";

cursor.execute(qry) elif status=="arival":

qry="INSERT INTO car_parking (rfid_id,status,datetimestamp,number_plates)
                        VALUES
        ("+str(rfid_number)+",'arrival',"+str(time_now)+","+str(number_plates)+")";
```

```
 cursor.execute(qry)
 def take_pic():
 pass
 while True:
 action = ser.readline().strip('\n') print action
         if action.startswith("rfid_check"): rs=action.slip(" ")[-1] autheticate_rfid(rs)
 else:
 a,b= get_empty_slots(action)
 if a is not None or b is not None: try:
 empty_slots_ints=[]
 if a[1] == '1' and a[2]== '1' and a[3] =='1' and a[4]
 == '1' and a[0]=='1':
 ser.write('empty_slots No parking') print 'empty_slots No parking available'
 else:
 for slotx,spacex in zip(a,b): if slotx == '0'
 empty_slots_ints.append(spacex)
 ser.write('empty_slots       '+ str(len(empty_slots_ints)) +' available')
 print   'empty_slots  '+ str(len(empty_slots_ints)) +' available'
 except Exception as e:
 pass
 if a[5]=='1':
 print 'out going car' ser.write('read_rfid') time.sleep(3)
 while True:
 rfid_number=ser.readline()
 if rfid_number.startswith('rfid_no: '): if check_db_auth():
 break ser.write('read_ok') time.sleep(0.5) ser.write('move_camera_left') take_pic()
         number_plate=process_pic()


if number_plate is not None:

save_to_db(number_plate,rfid_number,'depart') else:

pass
```

```
ser.write('open_gate') time.sleep(7) ser.write('close_gate')

if a[6]=='1':

print 'incoming car' ser.write('read_rfid') time.sleep(3)

while True:

rfid_number=ser.readline()

if rfid_number.startswith('rfid_no: '): if check_db_auth():

break ser.write('read_ok') time.sleep(0.5) ser.write('open_gate')
ser.write('move_camera_right') take_pic() number_plate=process_pic()

if number_plate is not None:

save_to_db(number_plate,rfid_number,'arival')

else:

pass time.sleep(7) ser.write('close_gate')

else:

print action
```

Slave.ino

```
//slave code

#include <Servo.h>

#include <LiquidCrystal_I2C.h> #include <Wire.h>

#include <SPI.h> #include <MFRC522.h>

#define SS_PIN 10

#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);

LiquidCrystal_I2C lcd(0x27,20,4);

byte armsDown[8] = { 0b00100,

0b01010,

0b00100,
0b00100,
0b01110,
0b10101,
0b00100,
0b01010
};
byte armsUp[8] = { 0b00100,
0b01010,
0b00100,
0b10101,
0b01110,
0b00100,
0b00100,
0b01010
};
Servo gate;
Servo camera_hor; Servo camera_vert;
int pos1=0; int pos3=0; int pos2=0;
int val2=0; int val3=0; int val4=0; int val5=0; int val6=0;
int val7=0; int val8=0;
int inputPin2=2; int inputPin3=3; int inputPin4=4; int inputPin5=5; int inputPin6=6;
int inputPin7=7; int inputPin8=8;
//int servo_pin1=A0;
//int servo_pin2=10;
//int servo_pin3=11;
void setup() {
```

```
// put your setup code here, to run once: Serial.begin(9600);

lcd.init();

lcd.backlight();

pinMode(inputPin2, INPUT); pinMode(inputPin3, INPUT); pinMode(inputPin4,
INPUT); pinMode(inputPin5, INPUT); pinMode(inputPin6, INPUT);

pinMode(inputPin7, INPUT); pinMode(inputPin8, INPUT); gate.attach(A0);
camera_hor.attach(A1); camera_vert.attach(A2); lcd.createChar(3, armsDown);
lcd.createChar(4, armsUp);

}

int autheticate()

{

uint32_t period = (1 * 60000L)/6;      // listen for 10 secs for rfid card for( uint32_t tStart
= millis(); (millis()-tStart) < period; ){

if ( ! mfrc522.PICC_IsNewCardPresent())

{

return;

}

if ( ! mfrc522.PICC_ReadCardSerial())

{

return;

}

String content= ""; byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

//Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
//Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
content.toUpperCase();
Serial.println("rfid_check "+content.substring(1));
uint32_t period = (1 * 60000L)/6;      // listen for 10 secs for rfid card for( uint32_t tStart
= millis(); (millis()-tStart) < period; ){
String str=Serial.readStringUntil('\n'); Serial.println(str);
if(str=="rfid_ok"){ return 1;
}
}
}
return 0;
}
int mode(int a[],int n) {
int maxValue = 0, maxCount = 0, i, j; for (i = 0; i < n; ++i) {
int count = 0;
for (j = 0; j < n; ++j) { if (a[j] == a[i])
++count;
}
if (count > maxCount) { maxCount = count; maxValue = a[i];
}
}
return maxValue;
}
void lcd_print_out(String row1,String row2){ lcd.clear();
```

```
lcd.print(row1); lcd.setCursor(0,1); lcd.print(row2);
//delay(1000);
}
int get_sensor_two(){ int state[10];
for(int i=0;i<10;i++){
val2 = digitalRead(inputPin2);
//Serial.println(val6); state[i]=val2;
delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}
int get_sensor_three(){ int state[10];
for(int i=0;i<10;i++){
val3 = digitalRead(inputPin3);
//Serial.println(val6); state[i]=val3; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked
return 1;
}
```

```
}
int get_sensor_four(){ int state[10];
for(int i=0;i<10;i++){
val4 = digitalRead(inputPin4);
//Serial.println(val6); state[i]=val4; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}
int get_sensor_five(){ int state[10];
for(int i=0;i<10;i++){
val5 = digitalRead(inputPin5);
//Serial.println(val6); state[i]=val5; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}
int get_sensor_six(){ int state[10];
for(int i=0;i<10;i++){
```

```
val6 = digitalRead(inputPin6);

//Serial.println(val6); state[i]=val6; delay(12);

}

int mode_val = mode(state,10); if(mode_val==1){

//empty return 0;

}else{

//car parked return 1;

}

}

int get_sensor_seven(){ int state[10];

for(int i=0;i<10;i++){

val7 = digitalRead(inputPin7);

//Serial.println(val6); state[i]=val7; delay(12);

}

int mode_val = mode(state,10); if(mode_val==1){

//empty return 0;

}else{

//car parked return 1;

}

}

int get_sensor_eight(){ int state[10];

for(int i=0;i<10;i++){
val8 = digitalRead(inputPin8);
//Serial.println(val6); state[i]=val8; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
```

```
}else{ return 1;
}
}
int open_gate(){
for (pos1 = 0; pos1 <= 180; pos1 += 1) {
// in steps of 1 degree gate.write(pos1);
delay(15);
}
Serial.print("Gate Opened"); return 1;
}
int close_gate(){
for (pos1 = 180; pos1 >= 0; pos1 -= 1) { gate.write(pos1);
delay(15);
}
Serial.print("Gate Closed"); return 1;
}
int move_camera_right_pos(){
for (pos2 = 0; pos2 <= 180; pos2 += 1) { camera_vert.write(pos2);
delay(15);
}
for (pos3 = 0; pos3 <= 180; pos3 += 1) { camera_hor.write(pos3);
delay(15);
}
for (pos2 = 180; pos2 >= 0; pos2 -= 1) {
camera_vert.write(pos2);
delay(15);
}
return 1;
}
int move_camera_left_pos(){
for (pos2 = 0; pos2 <= 180; pos2 += 1) {camera_vert.write(pos2); delay(15);
}
for (pos3 = 180; pos3 >= 0; pos3 -= 1) { camera_hor.write(pos3);
```

```
delay(15);
}
for (pos2 = 180; pos2 >= 0; pos2 -= 1) { camera_vert.write(pos2);
delay(15);
}
return 1;
}
void loop() { String str; char character;
while (Serial.available()) { str=Serial.readStringUntil('\n'); Serial.println(str);
if(str=="open_gate"){
Serial.println("ha burn"); lcd.clear(); lcd.setCursor(4, 1); lcd.write(4); delay(2000);
if(open_gate()){ Serial.println("gate opened");
lcd_print_out("          Welcome        ","Enjoy your stay "); delay(10000);
}
}else if(str=="close_gate"){ Serial.println("ha burn2"); if(close_gate()){
Serial.println("gate closed");
}
}else if(str=="move_camera_right"){
Serial.println("ha burn3");
if (move_camera_right_pos()){
Serial.println("cam to right pos");
}
}
else if(str=="move_camera_left"){
Serial.println("ha burn4");
if (move_camera_left_pos()){
 Serial.println("cam to left pos");
}
}
else if(str=="read_rfid"){
 Serial.println("ha burn5"); if (autheticate()){
Serial.println("Autheticated");
lcd_print_out("          WELCOME    ","       AUTHETICATED      ");
}else{
Serial.println("Card Error");
lcd_print_out("          TRY AGAIN  "," CARD ERROR ");
}
}
else if(str.startsWith("lcd_print")){
 int commaIndex = str.indexOf(' ');
int secondCommaIndex =
 str.indexOf(' ', commaIndex + 1);
String firstValue =
 str.substring(0, commaIndex);
String secondValue =
 str.substring(commaIndex + 1,
secondCommaIndex);
String thirdValue =
str.substring(secondCommaIndex + 1);
 Serial.println(secondValue);
Serial.println(thirdValue);
 lcd_print_out(secondValue,thirdValue);
}
```

```
}
String Car_parking_Values="";
String Car_parking_status="";
int r2 = get_sensor_two(); if (r2){
//Serial.println("Car Parked");
Car_parking_status+="
1 "; Car_parking_Values+="S2 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
 Car_parking_Values+="S2 ";
}
int r3 = get_sensor_three(); if (r3){
//Serial.println("Car Parked");
 Car_parking_status+="1 ";
 Car_parking_Values+="S3 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
 Car_parking_Values+="S3 ";
}
int r4 = get_sensor_four(); if (r4){
//Serial.println("Car Parked");
 Car_parking_status+="1 ";
Car_parking_Values+="S4 ";
}else{
```

```
//Serial.println("Empty Slot");
 Car_parking_status+="0 ";
 Car_parking_Values+="S4 ";
}
int r5 = get_sensor_five(); if (r5){
//Serial.println("Car Parked");
 Car_parking_status+="1 ";
Car_parking_Values+="S5 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
Car_parking_Values+="S5 ";
}
int r6 = get_sensor_six(); if (r6){
//Serial.println("Car Parked");
Car_parking_status+="1 ";
Car_parking_Values+="S6 ";
}else{
//Serial.println("Empty Slot");
 Car_parking_status+="0 ";
Car_parking_Values+="S6 ";
}
int r7 = get_sensor_seven(); if (r7){
//Serial.println("Car Parked");
Car_parking_status+="1 ";
Car_parking_Values+="S7 ";
```

```
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
Car_parking_Values+="S7 ";
}
int r8 = get_sensor_eight(); if (r8){
//Serial.println("Car Parked");
Car_parking_status+="1 ";
 Car_parking_Values+="S8 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
Car_parking_Values+="S8 ";
}
Serial.println(Car_parking_status+',
'+Car_parking_Values);
//print empty parking slots
uint32_t period = (1 * 60000L)/6;
// listen for 10 secs for rfid card for( uint32_t tStart = millis();
(millis()-tStart) < period;  ){
String str=Serial.readStringUntil('\n');
Serial.println(str);
if(str.startsWith("empty_slots")){
int commaIndex = str.indexOf(' ');
int secondCommaIndex = str.indexOf(' ', commaIndex + 1);
 String firstValue = str.substring(0, commaIndex);
String secondValue = str.substring(commaIndex + 1, secondCommaIndex);
String thirdValue = str.substring(secondCommaIndex + 1);
 Serial.println(secondValue);
Serial.println(thirdValue);
lcd_print_out(secondValue,thirdValue); continue;}}}
```

Appendix

Master-slave serial communications

Master and serial communications between the arduino and the raspberry pi uses Master syntax to send communications to arduino

ser.write(val) with val being a string

Master syntax to send communications to arduino

```
action=ser.readline()
```

Slave syntax to send communications to raspberry pi

```
Serial.println(String str);
```

Slave syntax to receive communications to raspberry pi String str=Serial.readStringUntil('\n');

Mater communications

```
# while actively running

while True:

     #listen for any output from the serial rx (command)

         action = ser.readline().strip('\n')

         # if action is reading rfid card passif

         action.startswith("rfid_check"):

                   rs=action.strip("    ")[-1]

                   autheticate_rfid(rs)

         else:
                   a,b= get_empty_slots(action)

                   if a is not None or b is not None:

                             try:

                                       empty_slots_ints=[]

if a[1] == '1' and a[2]== '1' and a[3] =='1' and a[4] == '1' and a[0]=='1':

ser.write('empty_slots No parking') print 'empty_slots No parking available'

                                       else:

for slotx,spacex in zip(a,b):if slotx == '0':
empty_slots_ints.append(spacex)

ser.write('empty_slotsstr(len(empty_slots_ints)) +' available')



                                                                                     '+
print    'empty_slots    '+ str(len(empty_slots_ints)) +' available'

except Exception as e:

pass

if a[5]=='1':

print 'out going car' ser.write('read_rfid') time.sleep(3)

while True:

rfid_number=ser.readline()

if rfid_number.startswith('rfid_no: '):

 if check_db_auth():

break ser.write('read_ok') time.sleep(0.5)

 ser.write('move_camera_left') take_pic()

 number_plate=process_pic()

 if number_plate is not None:

save_to_db(number_plate,rfid_number,'depart') else:

pass

er.write('open_gate')

time.sleep(7) ser.write('close_gate')

if a[6]=='1':

print 'incoming car' ser.write('read_rfid') time.sleep(3)

while True:

rfid_number=ser.readline()

if rfid_number.startswith('rfid_no: '): if check_db_auth():

break ser.write('read_ok') time.sleep(0.5) ser.write('open_gate')
ser.write('move_camera_right')

take_pic() number_plate=process_pic()

if number_plate is not None:
```

```
save_to_db(number_plate,rfid_number,'arival')
else:
pass time.sleep(7) ser.write('close_gate')
else:
print action
Slave Communication String str;
char character;
while (Serial.available())
 { str=Serial.readStringUntil('\n');
Serial.println(str); if(str=="open_gate"){
lcd.clear(); lcd.setCursor(4, 1); lcd.write(4); delay(2000);
if(open_gate())
{ Serial.println("gate opened");
lcd_print_out("           Welcome
","Enjoy your stay "); delay(10000);
}
}else if(str=="close_gate")
{ if(close_gate()){
 Serial.println("gate closed");
}
}else if(str=="move_camera_right"){ Serial.println("ha burn3");
if (move_camera_right_pos()){ Serial.println("cam to right pos");
}
}
```

```
else if(str=="move_camera_left"){
 if (move_camera_left_pos())
{ Serial.println("cam to left pos");
}
}
else if(str=="read_rfid"){ if (autheticate()){
Serial.println("Autheticated");
lcd_print_out("          WELCOME    ","
AUTHETICATED     ");
}else{
Serial.println("Card Error");
lcd_print_out("
TRY AGAIN  ","  CARD ERROR ");
}
}
else if(str.startsWith("lcd_print")){
 int commaIndex = str.indexOf(' ');
int secondCommaIndex = str.indexOf(' ', commaIndex + 1);
 String firstValue = str.substring(0, commaIndex);
String secondValue = str.substring(commaIndex + 1, secondCommaIndex);
String thirdValue = str.substring(secondCommaIndex + 1); Serial.println(secondValue);
Serial.println(thirdValue); lcd_print_out(secondValue,thirdValue);

}

}
```

Full codes

Master.py

```
import datetime import sqlite3 import serial import time

import numpy as np import cv2

from copy import deepcopy from PIL import Image import pytesseract as tess

#initialize db connection

con = sqlite3.connect('parking_register.db3') cursor = con.cursor()

#qry="CREATE TABLE users_analytics (rfid TEXT PRIMARY KEY, tot_time

TEXT NOT NULL,);"

#cursor.execute(qry)

ser = serial.Serial('/dev/ttyACM0',9600)

def preprocess(img):

#cv2.imshow("Input",img)

imgBlurred = cv2.GaussianBlur(img, (5,5), 0)

gray = cv2.cvtColor(imgBlurred, cv2.COLOR_BGR2GRAY)

sobelx = cv2.Sobel(gray,cv2.CV_8U,1,0,ksize=3) #cv2.imshow("Sobel",sobelx)

#cv2.waitKey(0)

ret2,threshold_img      =

cv2.threshold(sobelx,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

#cv2.imshow("Threshold",threshold_img)
```

```
#cv2.waitKey(0) return threshold_img
def cleanPlate(plate):
print "CLEANING PLATE. . ."
gray = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)
kernel = cv2.getStructuringElement(cv2.MORPH_CROSS, (3, 3)) thresh=
cv2.dilate(gray, kernel, iterations=1)
_, thresh = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)
im1,contours,hierarchy          =
cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
if contours:
areas = [cv2.contourArea(c) for c in contours] max_index = np.argmax(areas)
max_cnt = contours[max_index]
max_cntArea = areas[max_index] x,y,w,h = cv2.boundingRect(max_cnt)
if not ratioCheck(max_cntArea,w,h): return plate,None
cleaned_final = thresh[y:y+h, x:x+w] #cv2.imshow("Function Test",cleaned_final)
return cleaned_final,[x,y,w,h]
else:
return plate,None
def extract_contours(threshold_img):
element=        cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(17, 3))
morph_img_threshold = threshold_img.copy()
```

```
cv2.morphologyEx(src=threshold_img, op=cv2.MORPH_CLOSE, kernel=element,
dst=morph_img_threshold)
cv2.imshow("Morphed",morph_img_threshold) cv2.waitKey(0)
im2,contours, hierarchy=
cv2.findContours(morph_img_threshold,mode=cv2.RETR_EXTERNAL,method
=cv2.CHAIN_APPROX_NONE)
return contours
def ratioCheck(area, width, height):
ratio = float(width) / float(height) if ratio < 1:
ratio = 1 / ratio aspect = 4.7272
min = 15*aspect*15 # minimum area max = 125*aspect*125 # maximum area rmin = 3
rmax = 6
if (area < min or area > max) or (ratio < rmin or ratio > rmax): return False
return True
def isMaxWhite(plate):
avg = np.mean(plate) if(avg>=115):
return True
else:
return False
def validateRotationAndRatio(rect):
(x, y), (width, height), rect_angle = rect
if(width>height):
```

```
angle = -rect_angle
else:
angle = 90 + rect_angle
if angle>15:
return False
if height == 0 or width == 0: return False
area = height*width
if not ratioCheck(area,width,height): return False
else:
return True
def cleanAndRead(img,contours): #count=0
for i,cnt in enumerate(contours): min_rect = cv2.minAreaRect(cnt)
if validateRotationAndRatio(min_rect): x,y,w,h = cv2.boundingRect(cnt) plate_img =
img[y:y+h,x:x+w]
if(isMaxWhite(plate_img)): #count+=1
clean_plate, rect = cleanPlate(plate_img)
if rect:
x1,y1,w1,h1 = rect
x,y,w,h = x+x1,y+y1,w1,h1 cv2.imshow("Cleaned Plate",clean_plate) cv2.waitKey(0)
plate_im = Image.fromarray(clean_plate)
text      =         tess.image_to_string(plate_im,
lang='eng')

print "Detected Text : ",text

img=

cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

cv2.imshow("Detected Plate",img) cv2.waitKey(0)

return text

return None

#print "No. of final cont : " , count

def process_pic():

print "DETECTING PLATE . . ."

img = cv2.imread("4.jpg") threshold_img = preprocess(img)

contours= extract_contours(threshold_img) return cleanAndRead(img,contours)

def get_empty_slots(slot_status): try:

available_slots=[] available_slots_names=[]

for slot in slot_status.split(',')[0].split(' '): if len(slot) > 0:

available_slots.append(slot)

for slot_name in slot_status.split(',')[-1].split(' '): if len(slot_name) > 0 :

available_slots_names.append(slot_name) return

available_slots,available_slots_names[:-1]

except Exception as e:

return None , None
```

```
def autheticate_rfid(rfid_card):

qry="SELECT * FROM car_parking WHERE rfid_id="+str(rfid_card)
cursor.execute(qry)

rows = cursor.fetchall() if len(rows) > 0:

ser.write('rfid_ok')

else:

ser.write('rfid_denied')

def secs2hours(secs):

mm, ss = divmod(secs, 60) hh, mm = divmod(mm, 60)

return "%d h %02d m %02d s" % (hh, mm, ss)

def get_freq(user):

qry2="SELECT * FROM car_parking WHERE rfid_id="+str(user) cursor.execute(qry2)

rows_users = cursor.fetchall() freq=int(len(rows_users)/2) form=str(user)+' '+str(freq)
userfreq=tuple(form.split(' ')) return user,freq

def get_users_frequency(): u=[];f=[]

qry="SELECT * FROM car_parking" cursor.execute(qry)

rows = cursor.fetchall() users=set()

for row in rows:

users.add(row[0])

for user in users:
a,b=get_freq(user) u.append(a) f.append(b)
```

```
return u , f

def get_user_hours(rfid_card):

#dbcur.execute("INSERT INTO av_md5 (virus_md5) VALUES (?);", (line.strip('\n'),))

qry="SELECT * FROM car_parking WHERE rfid_id="+str(rfid_card)
cursor.execute(qry)

rows = cursor.fetchall() arrival_time=rows[-1][2][:-7]

ar_tm = datetime.datetime.strptime(a[:-7], '%Y-%m-%d %H:%M:%S') right_now=
datetime.datetime.now()

timedelta=right_now-ar_tm timeused=secs2hours(timedelta)

ser.write('lcd_print Card:'+str(rfid_card)+' '+str(timeused)) #time.sleep(3)

def save_to_db(number_plate,rfid_number,status): time_now=datetime.datetime.now()

if status=="depart":

qry="INSERT INTO   car_parking

(rfid_id,status,datetimestamp,number_plates)                    VALUES
("+str(rfid_number)+",'departure',"+str(time_now)+","+str(number_plates)+")";

cursor.execute(qry) elif status=="arival":

qry="INSERT INTO   car_parking

(rfid_id,status,datetimestamp,number_plates)                                VALUES
("+str(rfid_number)+",'arrival',"+str(time_now)+","+str(number_plates)+")";

cursor.execute(qry)
def take_pic():

pass
```

```
while True:

action = ser.readline().strip('\n') print action

if action.startswith("rfid_check"): rs=action.slip(" ")[-1] autheticate_rfid(rs)

else:

a,b= get_empty_slots(action)

if a is not None or b is not None: try:

empty_slots_ints=[]

if a[1] == '1' and a[2]== '1' and a[3] =='1' and a[4]

== '1' and a[0]=='1':

ser.write('empty_slots No parking') print 'empty_slots No parking available'
```

```
                else:
                    for slotx,spacex in zip(a,b):

    empty_slots_ints.append(spacex)
                    ser.write('empty_slots '+
str(len(empty_slots_ints)) +' available')
                    print 'empty_slots '+
str(len(empty_slots_ints)) +' available'
            except Exception as e:

            if a[5]=='1':

                print 'out going car'
                ser.write('read_rfid')time.sleep(3)
                while True:

                    rfid_number=ser.readline()

                    if rfid_number.startswith('rfid_no: '): if
                        check_db_auth():
                            break
                ser.write('read_ok') time.sleep(0.5)
                ser.write('move_camera_left')
                take_pic()
                number_plate=process_pic()
```

```
if number_plate is not None:

save_to_db(number_plate,rfid_number,'depart') else:

pass

ser.write('open_gate') time.sleep(7) ser.write('close_gate')

if a[6]=='1':

print 'incoming car' ser.write('read_rfid') time.sleep(3)

while True:

rfid_number=ser.readline()

if rfid_number.startswith('rfid_no: '): if check_db_auth():

break ser.write('read_ok') time.sleep(0.5) ser.write('open_gate')
ser.write('move_camera_right') take_pic() number_plate=process_pic()

if number_plate is not None:

save_to_db(number_plate,rfid_number,'arival')

else:

pass time.sleep(7) ser.write('close_gate')

else:

print action
```

Slave.ino

```
//slave code

#include <Servo.h>

#include <LiquidCrystal_I2C.h> #include <Wire.h>

#include <SPI.h> #include <MFRC522.h>

#define SS_PIN 10

#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);

LiquidCrystal_I2C lcd(0x27,20,4);
```

```
byte armsDown[8] = { 0b00100,
0b01010,
0b00100,
0b00100,
0b01110,
0b10101,
0b00100,
0b01010
};
byte armsUp[8] = { 0b00100,
0b01010,
0b00100,
0b10101,
0b01110,
0b00100,
0b00100,
0b01010
};
Servo gate;
Servo camera_hor; Servo camera_vert;
int pos1=0; int pos3=0; int pos2=0;
int val2=0; int val3=0; int val4=0; int val5=0; int val6=0; int val7=0; int val8=0;
int inputPin2=2; int inputPin3=3; int inputPin4=4; int inputPin5=5; int inputPin6=6;
int inputPin7=7; int inputPin8=8;
//int servo_pin1=A0;
//int servo_pin2=10;
```

```
//int servo_pin3=11;
void setup() {
// put your setup code here, to run once: Serial.begin(9600);
lcd.init(); lcd.backlight();
pinMode(inputPin2, INPUT); pinMode(inputPin3, INPUT); pinMode(inputPin4,
INPUT);
pinMode(inputPin5, INPUT); pinMode(inputPin6, INPUT); pinMode(inputPin7,
INPUT); pinMode(inputPin8, INPUT); gate.attach(A0); camera_hor.attach(A1);
camera_vert.attach(A2); lcd.createChar(3, armsDown); lcd.createChar(4, armsUp);
}
int autheticate()
{
uint32_t period = (1 * 60000L)/6;      // listen for 10 secs for rfid card for( uint32_t tStart
= millis(); (millis()-tStart) < period; ){
if ( ! mfrc522.PICC_IsNewCardPresent())
{
return;
}
if ( ! mfrc522.PICC_ReadCardSerial())
{
return;
}
String content= ""; byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
//Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
//Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
content.toUpperCase();
Serial.println("rfid_check "+content.substring(1));
uint32_t period = (1 * 60000L)/6;      // listen for 10 secs for rfid card for( uint32_t tStart
= millis(); (millis()-tStart) < period; ){
String str=Serial.readStringUntil('\n'); Serial.println(str);
if(str=="rfid_ok"){ return 1;
}
}}
return 0;
}
int mode(int a[],int n) {
int maxValue = 0, maxCount = 0, i, j; for (i = 0; i < n; ++i) {
int count = 0;
for (j = 0; j < n; ++j) {
if (a[j] == a[i])
++count;
}if (count > maxCount) { maxCount = count; maxValue = a[i];
}
}return maxValue;
}
void lcd_print_out(String row1,String row2){ lcd.clear();
lcd.print(row1); lcd.setCursor(0,1);
lcd.print(row2);
//delay(1000);
}
int get_sensor_two(){ int state[10];
for(int i=0;i<10;i++){
val2 = digitalRead(inputPin2);
//Serial.println(val6); state[i]=val2; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
```

```
}
int get_sensor_three(){ int state[10];
for(int i=0;i<10;i++){
val3 = digitalRead(inputPin3);
//Serial.println(val6); state[i]=val3; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}int get_sensor_four(){ int state[10];
for(int i=0;i<10;i++){
val4 = digitalRead(inputPin4);
//Serial.println(val6); state[i]=val4; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
else{
//car parked return 1;
}
}
int get_sensor_five(){ int state[10];
for(int i=0;i<10;i++){
val5 = digitalRead(inputPin5);
//Serial.println(val6); state[i]=val5; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty
return 0;
}else{
//car parked return 1;
}
```

```
int get_sensor_six(){ int state[10];
for(int i=0;i<10;i++){
val6 = digitalRead(inputPin6);
//Serial.println(val6); state[i]=val6; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}
int get_sensor_seven(){ int state[10];
for(int i=0;i<10;i++){
val7 = digitalRead(inputPin7);
//Serial.println(val6); state[i]=val7; delay(12);
}
int mode_val = mode(state,10); if(moe_val==1){
//empty return 0;
}else{
//car parked return 1;
}
}
int get_sensor_eight(){ int state[10];
for(int i=0;i<10;i++){
val8 = digitalRead(inputPin8);
//Serial.println(val6); state[i]=val8; delay(12);
}
int mode_val = mode(state,10); if(mode_val==1){
//empty
return 0;
}else{ return 1;
}
}
```

```
}
int opengate(){
for (pos1 = 0; pos1 <= 180; pos1 += 1) {
// in steps of 1 degree gate.write(pos1); delay(15);
}
Serial.print("Gate Opened"); return 1;
}
int close_gate(){
for (pos1 = 180; pos1 >= 0; pos1 -= 1) { gate.write(pos1);
delay(15);
}
Serial.print("Gate Closed"); return 1;
}
int move_camera_right_pos(){
for (pos2 = 0; pos2 <= 180; pos2 += 1) { camera_vert.write(pos2);
delay(15);
}
for (pos3 = 0; pos3 <= 180; pos3 += 1) { camera_hor.write(pos3);
delay(15);
}
for (pos2 = 180; pos2 >= 0; pos2 -= 1) { camera_vert.write(pos2);
delay(15);
}
return 1;
}
int move_camera_left_pos(){
for (pos2 = 0; pos2 <= 180; pos2 += 1) { camera_vert.write(pos2);
delay(15);
}
```

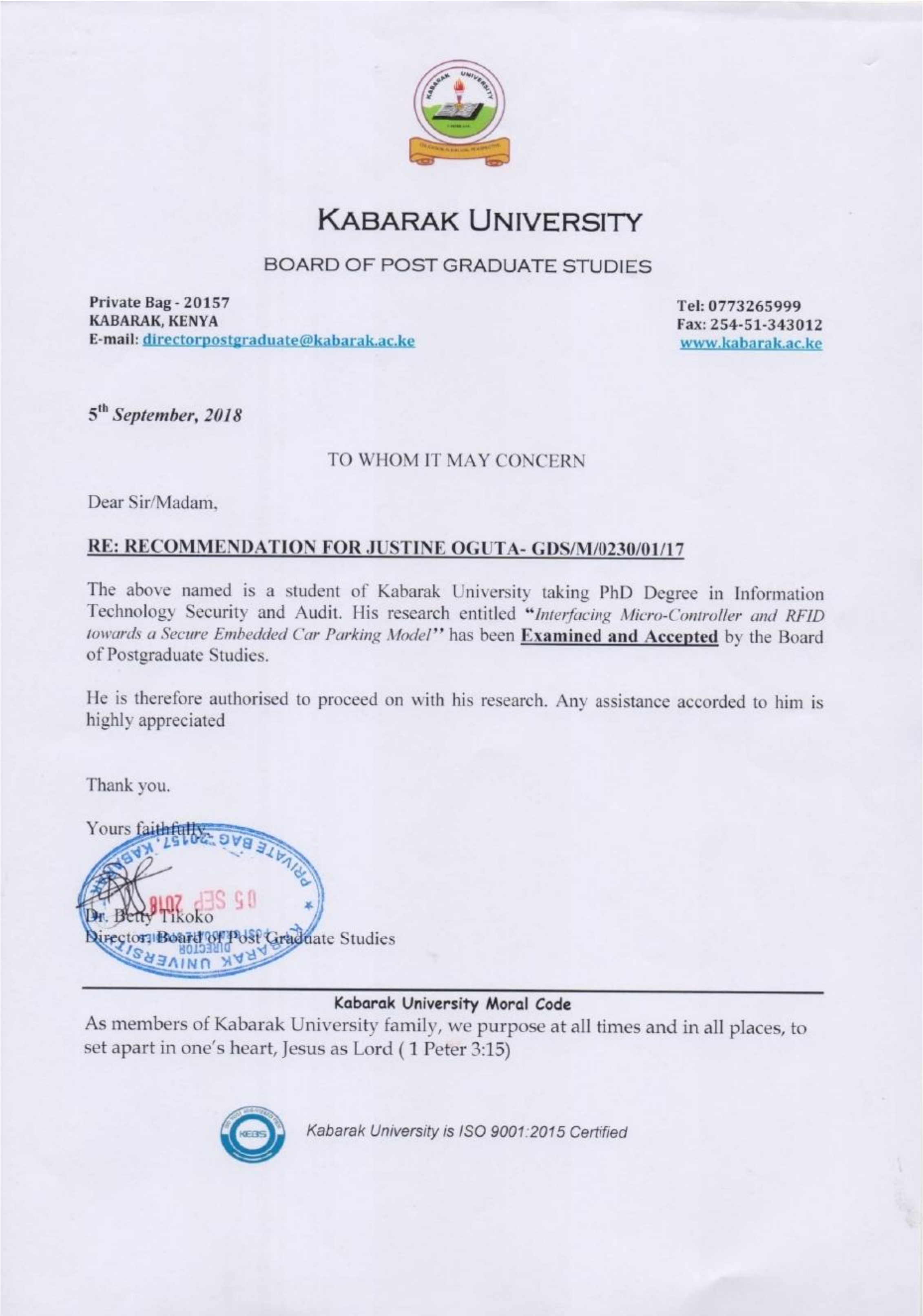
```
for (pos3 = 180; pos3 >= 0; pos3 -= 1) { camera_hor.write(pos3);

delay(15);

}

for (pos2 = 180; pos2 >= 0; pos2 -= 1) { camera_vert.write(pos2);

delay(15);

}

return 1;

}

void loop() { Strig str;

har character;

while (Serial.available()) { str=Serial.readStringUntil('\n'); Serial.println(str);
if(str=="open_gate"){

Serial.println("ha burn"); lcd.clear(); lcd.setCursor(4, 1); lcd.write(4); delay(2000);

if(open_gate()){ Serial.println("gate opened");

lcd_print_out(" Welcome ","Enjoy your stay "); delay(10000);

}

}else if(str=="close_gate"){ Serial.println("ha burn2"); if(close_gate()){
Serial.println("gate closed");

}

}else if(str=="move_camera_right"){ Serial.println("ha burn3");

if (move_camera_right_pos()){ Serial.println("cam to right pos");

}

else if(str=="move_camera_left"){ Serial.println("ha burn4");

if (move_camera_left_pos()){ Serial.println("cam to left pos");

}

}
```

```
else if(str=="read_rfid"){ Serial.println("ha burn5"); if (autheticate()){
Serial.println("Autheticated");
lcd_print_out("          WELCOME    ","       AUTHETICATED     ");
}else{
Serial.println("Card Error");
lcd_print_out("          TRY AGAIN  "," CARD ERROR ");
}
}
else if(str.startsWith("lcd_print")){
int commaIndex = str.indexOf(' ');
int secondCommaIndex = str.indexOf(' ', commaIndex + 1); String firstValue =
str.substring(0, commaIndex);
String secondValue = str.substring(commaIndex + 1, secondCommaIndex); String
thirdValue = str.substring(secondCommaIndex + 1); Serial.println(secondValue);
Serial.println(thirdValue); lcd_print_out(secondValue,thirdValue);
}
String Car_parking_Values=""; String Car_parking_status="";
int r2 = get_sensor_two(); if (r2){
//Serial.println("Car Parked")
; Car_parking_status+="1 ";
Car_parking_Values+="S2 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
Car_parking_Values+="S2 ";
}
```

```
int r3 = get_sensor_three(); if (r3){
//Serial.println("Car Parked");
Car_parking_status+="1 ";
 Car_parking_Values+="S3 ";
}else{
//Serial.println("Empty Slot");
Car_parking_status+="0 ";
 Car_parking_Values+="S3 ";
}
nt r4 = get_sensor_four(); if (r4){
//Serial.println("Car Parked"); Car_parking_status+="1 "; Car_parking_Values+="S4 ";
}else{//Serial.println("Empty Slot"); Car_parking_status+="0 "; Car_parking_
Values+="S4 ";
}
int r5 = get_sensor_five(); if (r5){
//Serial.println("Car Parked"); Car_parking_status+="1 "; Car_parking_Values+="S5 ";
}else{
//Serial.println("Empty Slot"); Car_parking_status+="0 "; Car_parking_Values+="S5 ";
}
int r6 = get_sensor_six(); if (r6){
//Serial.println("Car Parked"); Car_parking_status+="1 ";
Car_parking_Values+="S6 ";
}els{
//Serial.println("Empty Slot"); Car_parking_status+="0 "; Car_parking_Values+="S6 ";
}
int r7 = get_sensor_seven(); if (r7){
//Serial.println("Car Parked"); Car_parking_status+="1 "; Car_parking_Values+="S7 ";
}else{
//Serial.println("Empty Slot"); Car_parking_status+="0 "; Car_parking_Values+="S7 ";
}
int r8 = get_sensor_eight(); if (r8){
//Serial.println("Car Parked"); Car_parking_status+="1 "; Car_parking_Values+="S8 ";
```

```
   }else{
   //Serial.println("Empty Slot"); Car_parking_status+="
   0 ";

    Car_parking_Values+="S8 ";
  }


  Serial.println(Car_parking_status+','+Car_parking_Values);


  uint32_t period = (1 * 60000L)/6;         // listen for 10 secs for rfid card
  for( uint32_t tStart = millis();  (millis()-tStart) < period;  ){String
        str=Serial.readStringUntil('\n'); Serial.println(str);
        if(str.startsWith("empty_slots")){        int
          commaIndex = str.indexOf(' ');
          int secondCommaIndex = str.indexOf(' ', commaIndex + 1);


          String      secondValue      =      str.substring(commaIndex      +      1,
secondCommaIndex);


          String thirdValue = str.substring(secondCommaIndex + 1);
          Serial.println(secondValue);
          Serial.println(thirdValue);
          lcd_print_out(secondValue,thirdValue);continue;
        }
   }
}
```

Appendix II: Authorization Letter from the University



KABARAK UNIVERSITY

BOARD OF POST GRADUATE STUDIES

Private Bag - 20157
KABARAK, KENYA
E-mail: directorpostgraduate@kabarak.ac.ke

Tel: 0773265999
Fax: 254-51-343012
www.kabarak.ac.ke

5th *September, 2018*

TO WHOM IT MAY CONCERN

Dear Sir/Madam,

RE: RECOMMENDATION FOR JUSTINE OGUTA- GDS/M/0230/01/17

The above named is a student of Kabarak University taking PhD Degree in Information Technology Security and Audit. His research entitled "*Interfacing Micro-Controller and RFID towards a Secure Embedded Car Parking Model*" has been **Examined and Accepted** by the Board of Postgraduate Studies.

He is therefore authorised to proceed on with his research. Any assistance accorded to him is highly appreciated

Thank you.

Yours faithfully,

PRIVATE BAG 20157, KABARAK
05 SEP 2018
DIRECTOR
KABARAK UNIVERSITY

Dr. Betty Tikoko
Director, Board of Post Graduate Studies

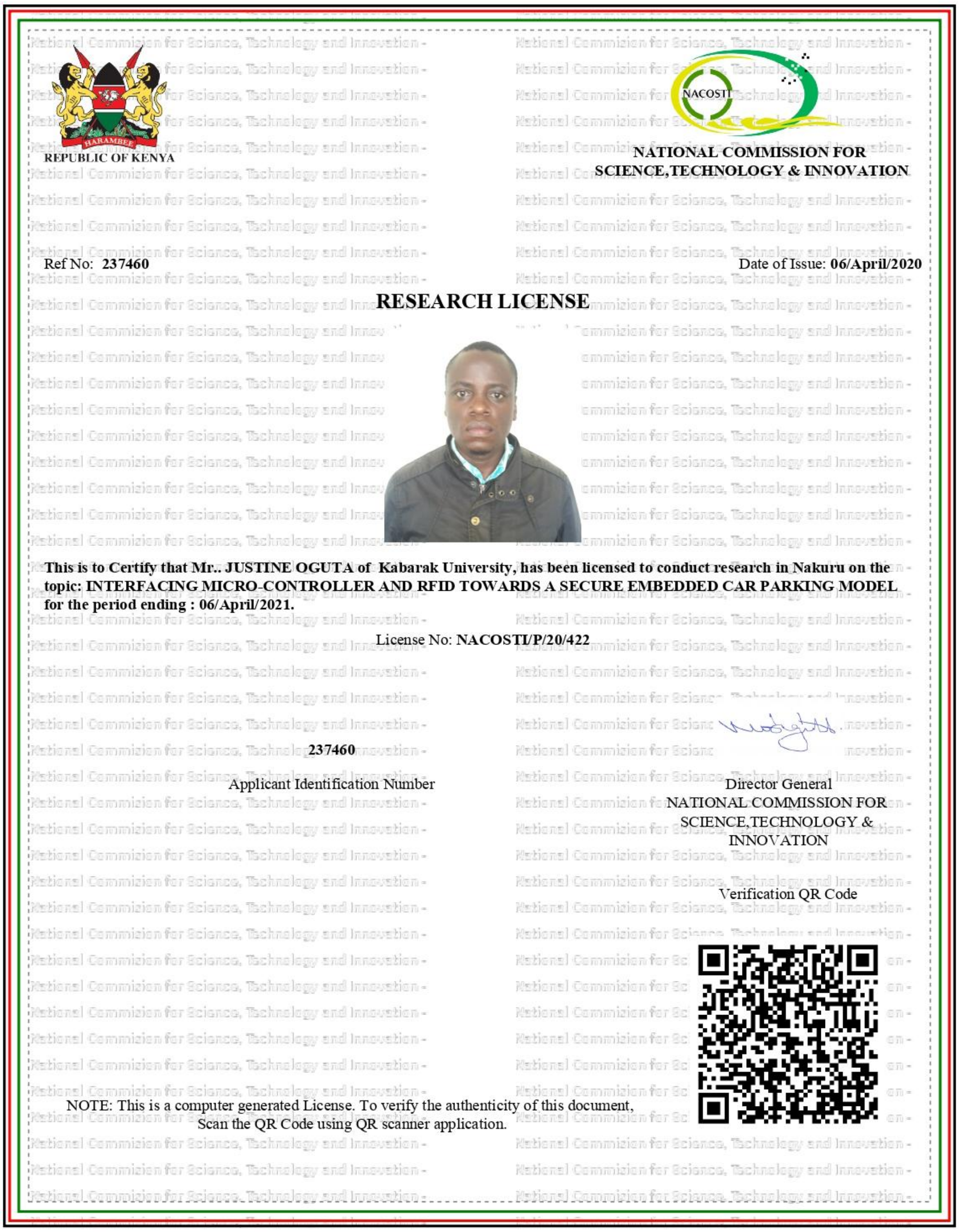
Kabarak University Moral Code

As members of Kabarak University family, we purpose at all times and in all places, to set apart in one's heart, Jesus as Lord (1 Peter 3:15)

KEBS

Kabarak University is ISO 9001:2015 Certified

Appendix III: NACOSTI Research Permit



REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **237460**

Date of Issue: **06/April/2020**

RESEARCH LICENSE

This is to Certify that Mr.. JUSTINE OGUTA of Kabarak University, has been licensed to conduct research in Nakuru on the topic: INTERFACING MICRO-CONTROLLER AND RFID TOWARDS A SECURE EMBEDDED CAR PARKING MODEL for the period ending : 06/April/2021.

License No: **NACOSTI/P/20/422**

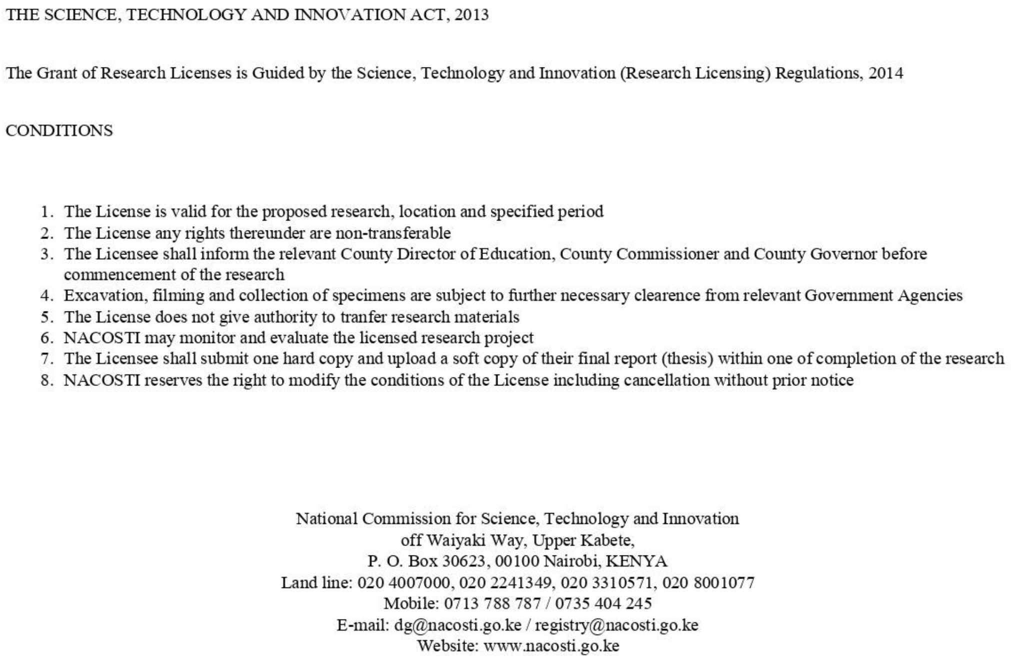
237460

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.



THE SCIENCE, TECHNOLOGY AND INNOVATION ACT, 2013

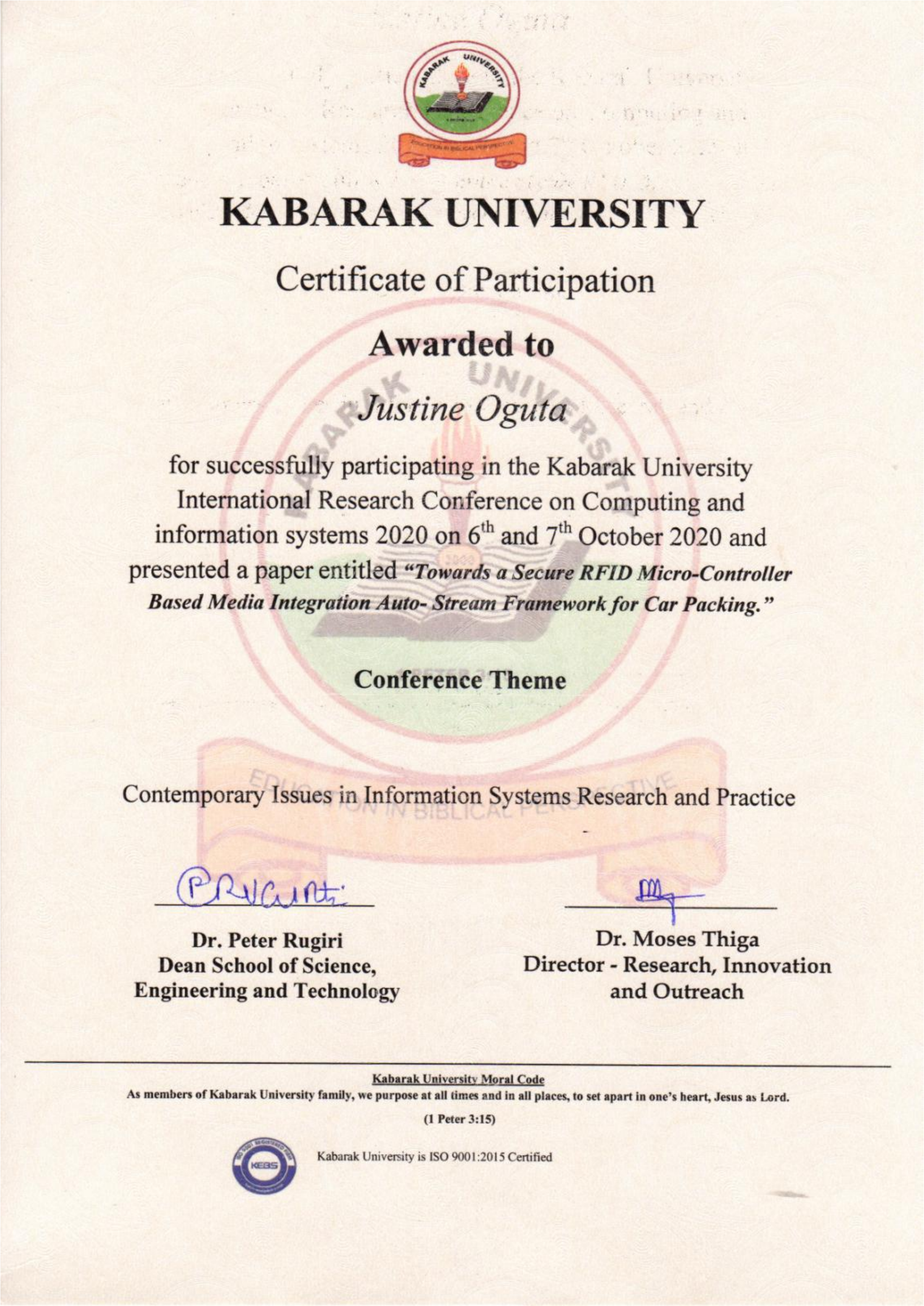
The Grant of Research Licenses is Guided by the Science, Technology and Innovation (Research Licensing) Regulations, 2014

CONDITIONS

1. The License is valid for the proposed research, location and specified period
2. The License any rights thereunder are non-transferable
3. The Licensee shall inform the relevant County Director of Education, County Commissioner and County Governor before commencement of the research
4. Excavation, filming and collection of specimens are subject to further necessary clearence from relevant Government Agencies
5. The License does not give authority to tranfer research materials
6. NACOSTI may monitor and evaluate the licensed research project
7. The Licensee shall submit one hard copy and upload a soft copy of their final report (thesis) within one of completion of the research
8. NACOSTI reserves the right to modify the conditions of the License including cancellation without prior notice

National Commission for Science, Technology and Innovation
off Waiyaki Way, Upper Kabete,
P. O. Box 30623, 00100 Nairobi, KENYA
Land line: 020 4007000, 020 2241349, 020 3310571, 020 8001077
Mobile: 0713 788 787 / 0735 404 245
E-mail: dg@nacosti.go.ke / registry@nacosti.go.ke
Website: www.nacosti.go.ke

Appendix IV: Evidence of Conference Participation



KABARAK UNIVERSITY

Certificate of Participation

Awarded to

Justine Oguta

for successfully participating in the Kabarak University International Research Conference on Computing and information systems 2020 on 6th and 7th October 2020 and presented a paper entitled ***"Towards a Secure RFID Micro-Controller Based Media Integration Auto- Stream Framework for Car Packing."***

Conference Theme

Contemporary Issues in Information Systems Research and Practice

Dr. Peter Rugiri
Dean School of Science,
Engineering and Technology

Dr. Moses Thiga
Director - Research, Innovation
and Outreach

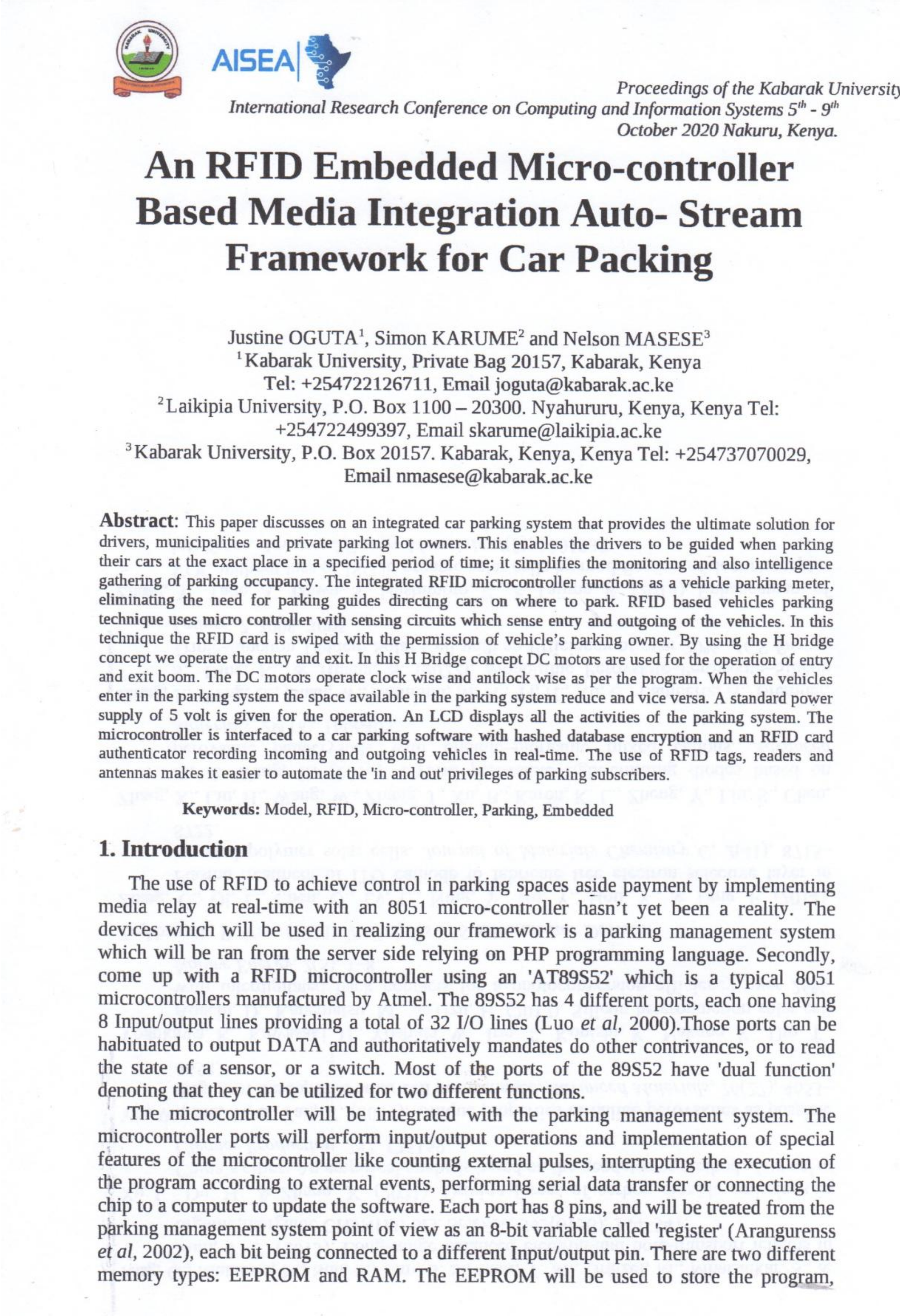
Kabarak University Moral Code

As members of Kabarak University family, we purpose at all times and in all places, to set apart in one's heart, Jesus as Lord.

(1 Peter 3:15)

Kabarak University is ISO 9001:2015 Certified

Appendix V: List of Publications



Proceedings of the Kabarak Universit
International Research Conference on Computing and Information Systems 5th - 9th
October 2020 Nakuru, Kenya.

An RFID Embedded Micro-controller Based Media Integration Auto- Stream Framework for Car Packing

Justine OGUTA[1], Simon KARUME[2] and Nelson MASESE[3]
[1] Kabarak University, Private Bag 20157, Kabarak, Kenya
Tel: +254722126711, Email joguta@kabarak.ac.ke
[2] Laikipia University, P.O. Box 1100 – 20300. Nyahururu, Kenya, Kenya Tel:
+254722499397, Email skarume@laikipia.ac.ke
[3] Kabarak University, P.O. Box 20157. Kabarak, Kenya, Kenya Tel: +254737070029,
Email nmasese@kabarak.ac.ke

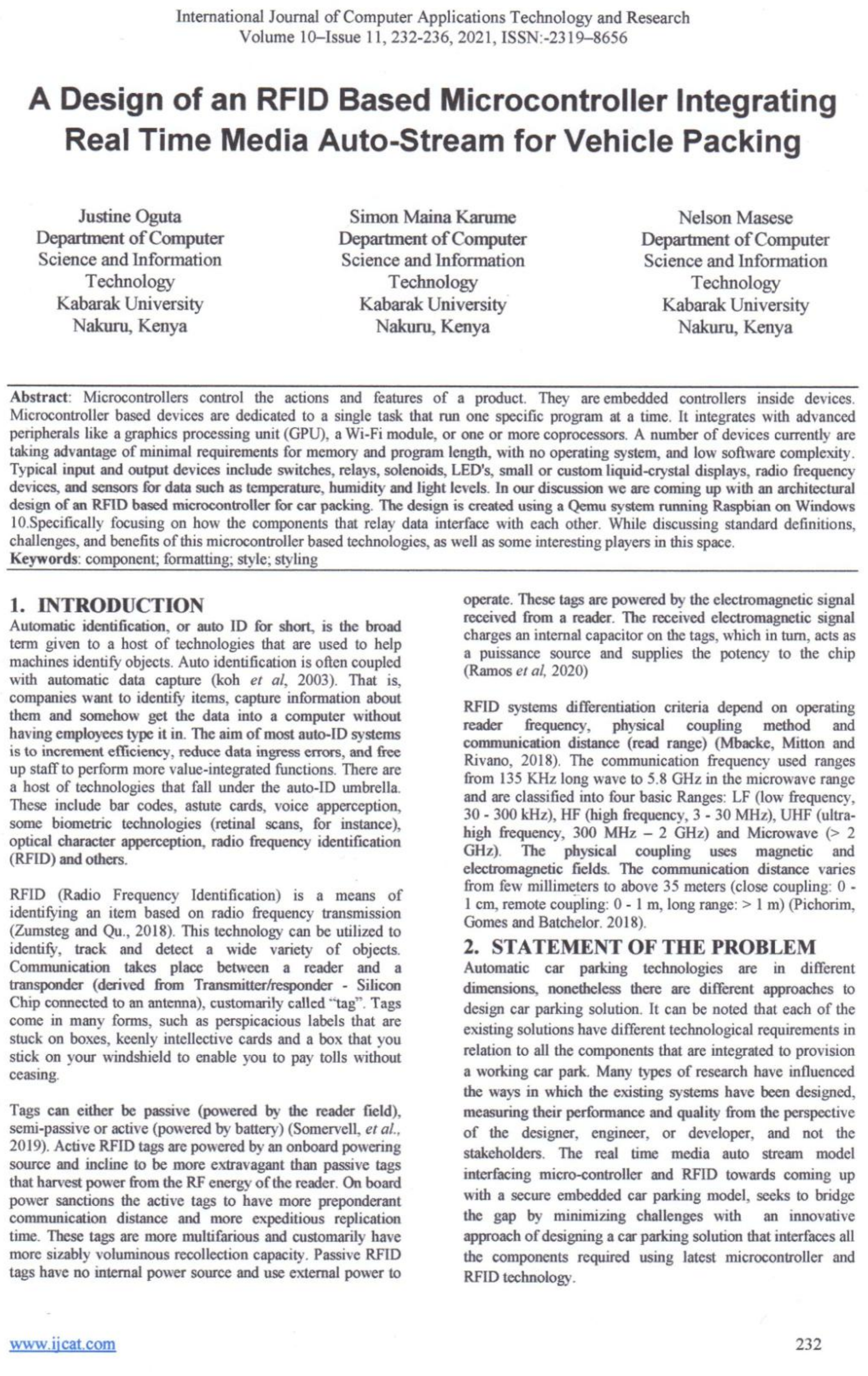
Abstract: This paper discusses on an integrated car parking system that provides the ultimate solution for drivers, municipalities and private parking lot owners. This enables the drivers to be guided when parking their cars at the exact place in a specified period of time; it simplifies the monitoring and also intelligence gathering of parking occupancy. The integrated RFID microcontroller functions as a vehicle parking meter, eliminating the need for parking guides directing cars on where to park. RFID based vehicles parking technique uses micro controller with sensing circuits which sense entry and outgoing of the vehicles. In this technique the RFID card is swiped with the permission of vehicle's parking owner. By using the H bridge concept we operate the entry and exit. In this H Bridge concept DC motors are used for the operation of entry and exit boom. The DC motors operate clock wise and antilock wise as per the program. When the vehicles enter in the parking system the space available in the parking system reduce and vice versa. A standard power supply of 5 volt is given for the operation. An LCD displays all the activities of the parking system. The microcontroller is interfaced to a car parking software with hashed database encryption and an RFID card authenticator recording incoming and outgoing vehicles in real-time. The use of RFID tags, readers and antennas makes it easier to automate the 'in and out' privileges of parking subscribers.

Keywords: Model, RFID, Micro-controller, Parking, Embedded

1. Introduction

The use of RFID to achieve control in parking spaces aside payment by implementing media relay at real-time with an 8051 micro-controller hasn't yet been a reality. The devices which will be used in realizing our framework is a parking management system which will be ran from the server side relying on PHP programming language. Secondly, come up with a RFID microcontroller using an 'AT89S52' which is a typical 8051 microcontrollers manufactured by Atmel. The 89S52 has 4 different ports, each one having 8 Input/output lines providing a total of 32 I/O lines (Luo *et al*, 2000).Those ports can be habituated to output DATA and authoritatively mandates do other contrivances, or to read the state of a sensor, or a switch. Most of the ports of the 89S52 have 'dual function' denoting that they can be utilized for two different functions.

The microcontroller will be integrated with the parking management system. The microcontroller ports will perform input/output operations and implementation of special features of the microcontroller like counting external pulses, interrupting the execution of the program according to external events, performing serial data transfer or connecting the chip to a computer to update the software. Each port has 8 pins, and will be treated from the parking management system point of view as an 8-bit variable called 'register' (Arangurenss *et al*, 2002), each bit being connected to a different Input/output pin. There are two different memory types: EEPROM and RAM. The EEPROM will be used to store the program,



A Design of an RFID Based Microcontroller Integrating Real Time Media Auto-Stream for Vehicle Packing

Justine Oguta
Department of Computer Science and Information Technology
Kabarak University
Nakuru, Kenya

Simon Maina Karume
Department of Computer Science and Information Technology
Kabarak University
Nakuru, Kenya

Nelson Masese
Department of Computer Science and Information Technology
Kabarak University
Nakuru, Kenya

Abstract: Microcontrollers control the actions and features of a product. They are embedded controllers inside devices. Microcontroller based devices are dedicated to a single task that run one specific program at a time. It integrates with advanced peripherals like a graphics processing unit (GPU), a Wi-Fi module, or one or more coprocessors. A number of devices currently are taking advantage of minimal requirements for memory and program length, with no operating system, and low software complexity. Typical input and output devices include switches, relays, solenoids, LED's, small or custom liquid-crystal displays, radio frequency devices, and sensors for data such as temperature, humidity and light levels. In our discussion we are coming up with an architectural design of an RFID based microcontroller for car packing. The design is created using a Qemu system running Raspbian on Windows 10.Specifically focusing on how the components that relay data interface with each other. While discussing standard definitions, challenges, and benefits of this microcontroller based technologies, as well as some interesting players in this space.

Keywords: component; formatting; style; styling

1. INTRODUCTION

Automatic identification, or auto ID for short, is the broad term given to a host of technologies that are used to help machines identify objects. Auto identification is often coupled with automatic data capture (koh *et al*, 2003). That is, companies want to identify items, capture information about them and somehow get the data into a computer without having employees type it in. The aim of most auto-ID systems is to increment efficiency, reduce data ingress errors, and free up staff to perform more value-integrated functions. There are a host of technologies that fall under the auto-ID umbrella. These include bar codes, astute cards, voice apperception, some biometric technologies (retinal scans, for instance), optical character apperception, radio frequency identification (RFID) and others.

RFID (Radio Frequency Identification) is a means of identifying an item based on radio frequency transmission (Zumsteg and Qu., 2018). This technology can be utilized to identify, track and detect a wide variety of objects. Communication takes place between a reader and a transponder (derived from Transmitter/responder - Silicon Chip connected to an antenna), customarily called "tag". Tags come in many forms, such as perspicacious labels that are stuck on boxes, keenly intellective cards and a box that you stick on your windshield to enable you to pay tolls without ceasing.

Tags can either be passive (powered by the reader field), semi-passive or active (powered by battery) (Somervell, *et al.*, 2019). Active RFID tags are powered by an onboard powering source and incline to be more extravagant than passive tags that harvest power from the RF energy of the reader. On board power sanctions the active tags to have more preponderant communication distance and more expeditious replication time. These tags are more multifarious and customarily have more sizably voluminous recollection capacity. Passive RFID tags have no internal power source and use external power to operate. These tags are powered by the electromagnetic signal received from a reader. The received electromagnetic signal charges an internal capacitor on the tags, which in turn, acts as a puissance source and supplies the potency to the chip (Ramos *et al*, 2020)

RFID systems differentiation criteria depend on operating reader frequency, physical coupling method and communication distance (read range) (Mbacke, Mitton and Rivano, 2018). The communication frequency used ranges from 135 KHz long wave to 5.8 GHz in the microwave range and are classified into four basic Ranges: LF (low frequency, 30 - 300 kHz), HF (high frequency, 3 - 30 MHz), UHF (ultra-high frequency, 300 MHz – 2 GHz) and Microwave (> 2 GHz). The physical coupling uses magnetic and electromagnetic fields. The communication distance varies from few millimeters to above 35 meters (close coupling: 0 - 1 cm, remote coupling: 0 - 1 m, long range: > 1 m) (Pichorim, Gomes and Batchelor. 2018).

2. STATEMENT OF THE PROBLEM

Automatic car parking technologies are in different dimensions, nonetheless there are different approaches to design car parking solution. It can be noted that each of the existing solutions have different technological requirements in relation to all the components that are integrated to provision a working car park. Many types of research have influenced the ways in which the existing systems have been designed, measuring their performance and quality from the perspective of the designer, engineer, or developer, and not the stakeholders. The real time media auto stream model interfacing micro-controller and RFID towards coming up with a secure embedded car parking model, seeks to bridge the gap by minimizing challenges with an innovative approach of designing a car parking solution that interfaces all the components required using latest microcontroller and RFID technology.