The Hardware Implementation of "Smart Gate" for UTAR

By

Navindra Raj

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2015

The Hardware Implementation of "Smart Gate" for UTAR

By

Navindra Raj

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled "The Hardware Implementation of "Smart Gate" for UTAR" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________

Name : ______________________

Date : ______________________

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Goh Hock Guan for his guidance, encouragement and patience through the entire duration of my Final Year Project. I am extremely lucky to have a supervisor who is concerned about my work and one who always responds to my questions promptly. Without his guidance, I would have been stuck at the first obstacle that I faced and would never have been able to come this far.

Next, I would like to thank my parents and family members for their continuous motivation and moral support. Their positivity kept me going till the succesful completion of this project.

Lastly, I would like to thank my teammate, Chan Siew Meng who was willing to undertake the software implementations of this project. His teamwork, constructive feedbacks and views helped me keep things in perspective.

Once again, I would like to express my heartfelt gratitude to all the people mentioned above and also to those who were both directly and indirectly involved in this project.

ABSTRACT

The usage of Active RFID technology is best suited for entry system applications as it allows convenience by working from a long distance and allowing a contactless identification. Hard to duplicate tags, low running-costs, convenience, secure access, and high efficiency are the important points to look out for in entry systems. This work aims to address all that issues by developing an Intelligent Gate system that is fully customized to suit University applications.

However, due to relatively high costs of purchasing actual Active RFID hardware, this project emulates the 2.4GHz Active RFID technology using MICAz motes. The MICAz platform is used to develop the base station which acts as the Active reader interfaced with a weight sensor designed to detect vehicles. Once detected, the base station will try to verify the ID of the Active tag inside the vehicle. If the vehicle does not have a valid tag containing the correct ID, a video sequence of the vehicle and its number plate is captured using a webcam and an alarm is raised to alert the security guards for further action to be taken. Most of the existing solutions in the market were reviewed and their shortcomings were noted.This helped to derive the system requirements needed to develop this project.

The technologies used in this project are Active RFID emulated using MICAz motes, Crossbow 520 Programming board, Arduino Uno R3, Servos, Webcam, OpenCV, TinyOS, Java and MySQL.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

p./pp.	page/pages

LIST OF ABBREVIATIONS

GHZ	Gigahertz
GUI	Graphical User Interface
ID	Identification
IDE	Integrated Development Environment
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
PC	Personal Computer
RFID	Radio-Frequency Identification
SQL	Structured Query Language
USB	Universal Serial Bus
UTAR	Universiti Tunku Abdul Rahman

CHAPTER 1 INTRODUCTION

1.1 Motivation and Problem Statement

With the expansion of teaching facilities and growing number of students each year, Universiti Tunku Abdul Rahman (UTAR), Perak Campus faces a major problem - which is regarding motorized vehicle (motorcycles and cars) entry system for vehicles owned by students. Although UTAR is built up on a vast piece of land, however the number of parking lots are limited. At the beginning of every new semester, approximately 16,000 students of UTAR apply a ballot number for their vehicle sticker application. The balloting process is carried out by a computer which picks random ballots. Successful ballots chosen by the computer allows a student to purchase a sticker which grants the student's vehicle to access campus grounds for the rest of the semester. This is a highly inefficient and insecure system that involves high costs and also consumes unnecessary use of resources.

Firstly, the usage of stickers as a method of identification to grant entry to the campus is not foolproof. It has security flaws such as ease of duplication. The stickers issued by the university is easily duplicated and guards manning the entrance experience a hard time differentiating the fake stickers from genuine ones. In fact, most of the fake stickers used by students goes undetected. Not only that, but as the stickers are highly sought after, students also resort to selling their stickers illegally, at a higher price to other students. This happens because the vehicle number is written on the sticker with a marker pen and is easily erased by using certain solvents.

Besides that, the usage of guards to physically inspect the stickers involves heavy use of manpower which directly contributes to high cost and is also highly inefficient because guards spend lots of time scanning through to inspect each vehicle's sticker one by one before granting them access. This often creates a traffic jam especially during peak hours (8 am - 9 am, on weekdays) and as a result, students and lecturers end up being late for classes. In addition to that, the guards are forced to endure extreme weather conditions such as intense heat and also light drizzles just to enforce the current entry system. Under heavy rain situations, no inspection is carried out and all vehicles are allowed to enter. This situation poses a big security risk because any vehicles can enter and exit the campus grounds freely and this timeframe could be used to carry out unlawful activities by unscrupulous individuals.

This project attempts to create a fundamental prototype of the hardware implementations of a “smart gate” vehicle access system in UTAR in order to solve the problems mentioned above.

1.2 Project Objectives

This project aims to develop a "Smart-Gate" system for UTAR which solves the problems present in the currently deployed manual vehicle entry and parking management system.

The system would consist of Active RFID readers and tags (emulated by MICAz motes), weight sensors, servo-assisted boom gates, Arduino Uno R3 microcontroller board and also webcams that are linked up to a PC which serves as a base station. The PC would contain the software to control and coordinate the sensors and readers. A database would record entries and exits.

The following are the elaboration of the objectives in this project:

The first objective is to link the Active RFID reader to the PC and start sampling information from the active tags. This is done to verify that the Active RFID reader can actually read the information contained in the tags before proceeding any further. The position of the reader should be adjusted to an optimum position so that it is able to detect tags without any problems.

The next objective is to link the weight sensor and program it to trigger the reader when it detects a vehicle. This is done to ensure that the weight sensor is able to detect vehicles that are present. Fine-tuning of the sensor should be applied to ensure accurate detection of vehicles.

The third objective is to link the webcams to the system and ensure that one is able to display a constant live feed of the entry lane while the other is able to capture a video sequence when an unauthorized access is detected. This ensures that any vehicles that attempt to enter illegally is able to be identified. The angle of the webcam must be adjusted to proper position so that the number plate of the vehicle is clearly captured.

The fourth objective is to link the servo to the Arduino Uno microcontroller board to operate the boom-gate. The boom-gate must open or close based on inputs received from the MICAz mote. An additional manual override switch is also required to facilitate manual intervention of the system.

The fifth objective is to set up a database server using the WampServer platform. Relevant tables must be created and the appropriate data-types must be chosen for the columns. The database is required to store a list of authorised tags and also to record the entries and exits of vehicles.

The sixth objective is to ensure that the hardware is properly interfaced with the software and the system functions flawlessly. Synchronisation issues between hardware and software must be sorted out. This is to prevent delays in detecting and recording video feeds of an unauthorised entry.

The last objective is to produce a low-cost and user-friendly prototype that is easily operated by the guards to enforce the entry system. The guards could focus on other areas and respond to unauthorized access only when alarms are raised. This system frees up their time and also workload, and also gives a positive impact towards their health.

In a nutshell, this project focuses on the hardware implementations only, software implementations such as GUI and Java Program are taken care by my team-member, Chan Siew Meng. Both our projects combined, aims to deliver a convenient, non-proprietary, cost-effective, highly efficient, secure and tailor-made entry system that is suitable to be implemented in UTAR. In other words, the combined outcome of this project would be a major improvement compared to the manual system that currently exists.

1.3 Project Scope

The ultimate goal of this project is to produce a fully working prototype of the hardware implementations of "Smart Gate" for UTAR. The system is designed to solve the problems that exist in the manual vehicle entry and parking management system that currently exists in the University. In other words, the system aims to be a convenient, low-cost, highly efficient and secure entry system for the University.

This project would be implemented using MICAz motes to emulate Active RFID readers and tags. MICAz is a 2.4 GHz Mote module used for enabling low-power, wireless sensor networks, and it is used to emulate Active RFID due to the the relatively high costs of actual Active RFID hardware.

The rough overview of this project involves a MICAz mote emulating an Active RFID reader positioned at the guardhouse (base station) which reads Active RFID tags emulated by MICAz motes that are positioned near the front windscreen of vehicles. Another mote attached to a weight sensor is used to detect incoming vehicles which then triggers the reader to attempt to read the tag in the vehicle. If the reader does not detect a valid tag, a webcam is triggered to capture a video sequence of the vehicle and its number plate. The system allows the car to pass through and at the same time, it raises an alarm to alert the guards so that further actions could be taken. Although a vehicle with invalid tag is granted access at the entry point, it would be blocked at the exit point using the boom gate. This ensures a congestion-free entry system.

The usage of active RFID technology allows convenience because it works from a long distance and allows for a contactless identification. The active RFID reader used in this project is able to detect active tags from a range of approximately 6 metres. In other words, it means students do not have to wind down their windows and touch their ID cards on a reader. In fact, this system attempts to work similar to the "Smart Tag" Toll system implemented in Malaysia. As a result, students are protected from harsh weather conditions.

The "smart gate" system is also very cost-effective because once implemented, the maintenance costs are kept to a minimum. Not only that, but the number of guards employed to man the guardhouse could be reduced because vehicles are identified by the system automatically and does not require any physical intervention from the guards unless an alarm is triggered. Each vehicle would just take a few milliseconds to be identified and granted access and this would prevent unnecessary jams or congestion from taking place. This prevents students and lecturers from being late to classes.

Since the entry system is automated, there is virtually no security flaws and the tags cannot be duplicated to gain illegal access. Each tag is checked-in at entry point and checked-out at exit point. If a tag is detected to exit without checking-in or vice versa, the system triggers an alarm. The only point where human intervention is required in this "smart gate" system is when alarms are raised. Alarms raised are to be handled by guards where further actions are to be taken. Lastly, the implementation of this system reduces the health hazards faced by guards because they do not have to stand outside and endure the extremities of the weather.

The heart of the system would be a personal computer (PC) which is linked up to the hardware components mentioned above. The PC would also contain the software and database which allows the whole "smart gate" system to function.

The actual title of this project "The implementation of "Smart Gate" for UTAR however, my part is focused to the hardware implementations where I have to take care of hardware and interfacing before passing it on to my team-member, Chan Siew Meng to continue the software implementations for this project.

1.4 Impact, Significance and Contribution

This project would greatly impact the wireless technology field of RFID because none of the existing solutions in the market are able to deliver a smart entry system that is convenient, non-proprietary, cost-effective, highly efficient, secure and tailor-made for University applications. Most importantly, this project solves all the problems present in the current manual entry system used in UTAR and most of the other educational institutions nationwide.

The main highlights of this project are as follows:

A) It solves the issue of using fake stickers to gain access

-Usage of active RFID technology with secure identification and authentication mechanism means that the active tags cannot be easily duplicated. Besides, there would be no illegal entries due to human error.

B) It solves the issue of wastage and high costs due to hiring manpower

-The system costs less than hiring many guards to carry out inspection. Not only that, but the system also does not need expensive maintenance.

C) It is convenient and efficient

-Students do not have to wind down their windows to tap their tags at the reader. It is a contactless system that is both fast and efficient. Each detection takes a few milliseconds only. In this way, it ensures there is no congestion during peak hours.

D) It is weatherproof

-Security at entry point is enforced at all times, including during rainy season or bad weather. Thus, the campus is always secure. At the same time, the system provides a healthier working environment for guards as they do not have to face the extremities of the harsh weather condition outdoors.

One of the innovative highlights of this project is the usage of webcam to record a video sequence to identify the vehicle that attempts an illegal access. None of the existing systems nor solutions in the market offer the integration of a camera to capture a video footage of the unauthorized vehicle. By integrating the IP camera into this project, we have a solid video evidence of the wrongdoer in action.

Next, this system is the most efficient entry system for Universities because it is customized and tailor-made to be specifically used in Universities. However, this system could also be altered to suit various other applications that require a smart entry system such as residential areas, factories, offices and etc (if required).

The low-cost of implementing this system further makes it more attractive to institutions and organizations that want a state-of-the-art smart entry system that does not cost a bomb to be implemented.

1.5 Background Information

Radio-frequency identification (RFID) technology uses radio-frequency electromagnetic fields to transfer data, and is primarily used to identify and track the movements and whereabouts of objects (Zakaria M. 2008).

The concept of RFID was first developed as early as in the 1960s and was used for military applications in the second World War (Mario W. 2003). However, the developments of this technology up to the present time has dedicated its use to areas such as E-payment, asset managements, inventory systems, product tracking, access control, human/animal identification and etc.

A modern RFID system typically consists of a tag which contains a small chip and an antenna that allows it to be identified and also a reader linked to a computer which allows information to be retrieved from or relayed to the tags (Zakaria M. 2008). In other words, a tag is basically a radio transponder that is attached to the object that is being tracked whereas the reader is used in places where access control or asset tracking is required. An RFID tag performs modulation and demodulation of the radio frequency signal between itself and the reader and also contains memory to store information of the object that it is designed to identify. An RFID reader on the other hand, sends a signal to the tag and reads the response.

In modern applications, there are two types of RFID tags that are being used, active and passive. Active tags have their own power source (usually battery) and are suited for long-range applications such as vehicle and assets tracking whereas, Passive tags are suited for short-range applications (usually below 100cm).

In this project, MICAz, a 2.4 GHz Mote module used for enabling low-power, wireless sensor networks, is used to emulate Active RFID due to the the relatively high costs of Active RFID hardware.

The usage of active RFID technology allows convenience because it works from a long distance and allows for a contactless identification. The active RFID reader used in this project is programmed to detect active tags from a range of approximately 6 metres. This range is deemed as the best range for vehicle-entry systems.

1.6 Organisation of the Report

This report consists of a total of 7 chapters. The first chapter is Introduction. It describes the motivation behind the development of this project, main objectives, scope and also the contributions of this project.

The second chapter is Literature Review where relevant technologies to the project are reviewed and the suitable ones be chosen. Existing systems in the market are also reviewed and its advantages and weaknesses are contrasted against the benefits of this project.

The third chapter is System Methodology where multiple system development models are reviewed and a suitable one to be used for this project is chosen. The hardware and software and software requirements of this project are also elaborated. This chapter also describes the functional requirements of this project, work schedule against timeline and also estimated costs of development.

The fourth chapter is System Design where the architecture of the system is defined. Functions required are divide into separate modules and the overall flow of the system is shown. Design specifications of GUI and database are also elaborated in this chapter.

The fifth chapter is System Implementation. This chapter explains how the hardware and software of this system is set up. The final operation of the system is also shown.

The sixth chapter is System Evaluation and Discussion. The performance metrics and system verification plans are specified and test results obtained are explained and elaborated. A Strength, Weakness, Opportunity, Threat (SWOT) Analysis and objectives analysis is done.

The last chapter is Conclusion where concluding remarks for the entire project is made. Recommendations for future developments are also explained in this chapter.

CHAPTER 2 LITERATURE REVIEW

2.1 Review of the Technologies

RFID is the most common technology used to develop entry systems and it can be divided into two categories, namely Active RFID and Passive RFID. Both technology use radio frequency for communication between tags and reader, however the similarity ends there. Active RFID uses an internal power source such as from batteries to continuously power the tag and the radio frequency circuitry whereas Passive RFID uses energy transferred from reader to tag to power the tag.

Active RFID allows for a “contactless” operation (long-range) whereas passive RFID is fairly restrictive and requires physical touch (short-range) of the tag on the reader to operate. Besides, Active RFID tags can also be interfaced with external sensors to perform temperature monitoring, etc. Figure 2.1.1 shows the comparison between Active RFID and Passive RFID technologies.

	Active RFID	**Passive RFID**
Tag Power Source	Internal powered by batteries, etc	Externally powered by energy transfer from the reader
Tag Battery	Yes	No
Required Signal Strength (Reader to Tag)	Very Low	Very High
Available Signal Strength (Tag to Reader)	High	Very Low
Communication Range	Long range (up to 100m)	Short range (up to 10m)
Sensor Capability	Yes	No

Table 2.1.1 Comparison between Active RFID and Passive RFID technologies

Based on the requirements and objectives of this project, Active RFID technology is deemed more suitable to be used mainly due to the long-range communication and fast operation speed that it offers. Fast operation speed in terms of detection and identification is crucial in fulfilling one of our main goals which is to reduce congestion at entry-gate during peak hours.

Note: The Active RFID technology would be emulated using MICAz motes due to the relatively high costs of purchasing actual Active RFID hardware.

2.2 Review of Existing Systems

Throughout the years, there have been many studies carried out and also systems developed based on the Active RFID technology in order to be implemented as vehicle entry systems. Out of the many systems, three relevant ones were selected to be reviewed in detail as follows.

2.2.1 Gate Access System by Evizal (2013, p. 727)

In Evizal's implementation, active RFID technology is used as a checkpoint or gate access system. The user uses an active tag as identification to gain access to a designated private area. By using active RFID technology, the tag can be read up to few metres and is suitable as a gate access system because user does not need to manually take a ticket as an access token or tap a tag at the reader.

RFID readers are installed at the entrance and exit lanes and the system communicates using radio frequency to the RFID tags mounted in each vehicle. The reader captures information contained in the RFID tag when the vehicle comes into range of the reader. The reader then sends the information to the computer system for verification to be carried out using the database system. If the information is not valid or registered in the database, an alert or an alarm is raised. However, if the information is valid, the gate barrier is released and the vehicle is allowed to pass through. All valid information is recorded for documentation purposes to generate a transaction log report.

Unfortunately however, there are several downsides to this system. Firstly, when an invalid entry is detected, the gate barrier blocks the vehicle from entering and this would cause a jam or congestion. In UTAR, hundreds of cars pass through during peak hours and an invalid entry that is blocked causes congestion for the cars behind it.

This situation is taken care of in our implementation of the "smart gate" entry system, by allowing the car to pass through at the entry point. An alarm is raised to alert the guards and a sequence of video is captured with the webcam to identify the vehicle. The vehicle is eventually blocked at the exit point. As a result, congestion and delays caused by invalid entries are prevented.

Besides that, Evizal's system also does not have a specific device to detect vehicles. It just detects the tags and opens the gate barrier. This is highly insecure because there could be a scenario where there are two cars, the first car with a valid tag and the second car without any tags which is following the first car very closely behind. The system would most likely allow both cars to pass through. In our project however, we have used a weight sensor to detect the passing vehicles and therefore a risk of having a second vehicle tailgating the front vehicle is prevented.

2.2.2 MAGNET Premium Long Range Parking Access System (MAGNET Electronic & Automation Sdn. Bhd.)

Another vehicle access system that is famous and readily available in the market is called the MAGNET Premium Long Range Parking Access System (MAGNET Electronic & Automation Sdn. Bhd.). It provides a hands-free parking access that allows users to enter the premise effortlessly. There is no need to flash cards and wait at long queues in front of the entrance. The system automatically detects and verifies the tag in the vehicle from a range of up to 7 metres. However, this system has a few downsides. First, the system only works with its proprietary software called "Soyal Etegra". The software is designed to support a wide variety of applications and therefore it has a lot of unnecessary functions and the user interface could be very complex for beginners. Guards would require special training just to operate the software and troubleshooting it would cause a major headache.

Next, the whole system is very expensive and is not cost-effective to be implemented in a University. Thirdly, the system also requires hacking of the road to install a loop magnet which is used to identify the passing of vehicles.

In our implementation of this project however, cost is kept to a minimum by utilizing cost-effective yet highly efficient hardware and software that is not proprietary and tied down to any standards. We use hardware and software that allows customization and programming to suit our application needs. As a result, the software and graphical user interface (GUI) could be tailor-made and simplified to provide a user friendly environment for the guards to operate. Besides that, our system does not require hacking of the roads because we use "above-ground" weight sensors to detect the presence of vehicles.

2.2.3 RFID Vehicle Access Control System (M-Tech Innovations Ltd.)

This system uses the passive RFID technology which is highly inconvenient as the user would have to physically touch the tag on the reader for entry verification (M-Tech Innovations Ltd.). This results in slow detection process which might lead to congestion when heavy traffic is involved. Besides, users are also forced to wind down their windows in bad weather conditions just to tap the tag on the reader. Lastly, passive tags are easily duplicated with duplicating devices obtainable easily in the market.

In our project, the usage of active RFID allows for a fast detection rate which makes it highly efficient and reduces the possibility of a congestion in the entry lane. On top of that, because of the contactless nature of active RFID, users do not face the inconvenience of winding down their windows to physically tap their tags on the readers. As such, users are protected from the extremeties of weather. Active RFID tags are relatively harder to duplicate due to the rarity of duplicating devices.

2.2.4 Summary of the Existing Systems

Existing System	**Advantages**	**Disadvantages**	**Critical Comments**
Gate Access System by Evizal (2013, p. 727)	- Provides secure, contactless and convenient access. - Fast detection.	- Boom-gate causes congestion in case of invalid entry (lane is blocked). - No specific device to detect vehicles, tailgating issue poses security risk.	- Boom-gate implemented at exit lanes prevents congestion, video feed of invalid entry is recorded and blocked at exit. - Weight sensor detects vehicles accurately and prevents tailgating issues.

MAGNET Premium Long Range Parking Access System (MAGNET Electronic & Automation Sdn. Bhd.)	- Provides secure, contactless and convenient access. - Fast detection.	- Uses proprietary software that is expensive, complicated and inefficient. - Requires hacking of roads to install loop-magnets used to detect vehicles. - Expensive costs involved in implementation of the entire system.	- Custom-made, user friendly Java GUI program to control the system. - Usage of above-ground weight sensors that do not require expensive hacking of roads - Relatively cheaper costs to implement the entire system.
RFID Vehicle Access Control System (M-Tech Innovations Ltd.)	- Provides secure access.	- Inconvenient because it requires physical tapping of tags on to readers. - Slow and has a high possibility of causing congestion during peak hours. - Passive tags can be easily duplicated with cheap duplicating devices.	- Usage of Active RFID allows for convenient, contactless and secure access. - Fast detection rate prevents congestion. - Active tags are relatively harder to duplicate due to rarity of duplicating devices.

Table 2.2.4.1 Summary of Existing Systems

2.3 Concluding Remarks

As a summary, the best approach for our "smart gate" entry system would be by using MICAz motes to emulate 2.4 GHz Active RFID tags and readers, a weight sensor to detect the presence of vehicles and a webcam to record unauthorized access. These hardware components would be controlled by a software which is connected to a database. By utilizing the hardware and software components and linking them up into an entry access system, our system would serve as the best non-proprietary, cost-effective, convenient, efficient, secure and tailor-made entry system that is best suited to be implemented in UTAR.

CHAPTER 3 SYSTEM METHODOLOGY

3.1 System Development Models

In this section, some of the available system development models would be discussed and the one deemed most suitable for this project is chosen.

3.1.1 Waterfall Model

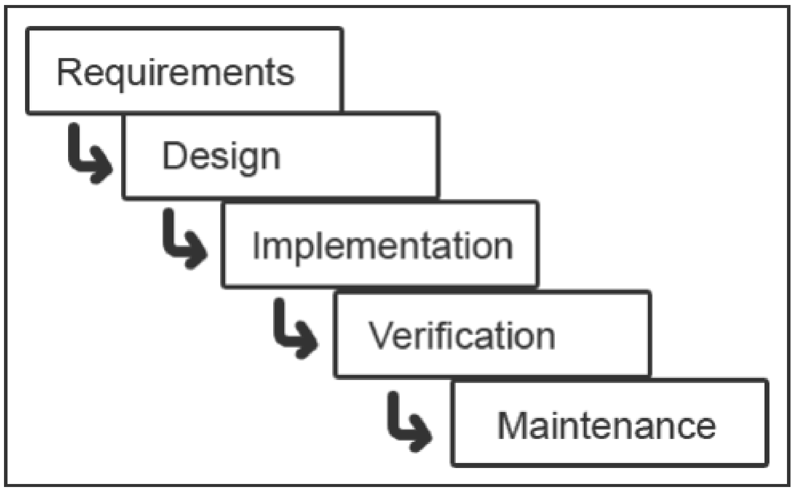


Figure 3.1.1.1 Waterfall Model (CompSci.ca, 2007)

The waterfall model is a traditional system development model which uses a sequential approach. As it can be seen from figure 3.1.1.1, progress through each phase flows downwards like a waterfall. In this model, each phase has its deliverables that is required for the next phase to proceed. This model is the easiest to understand and implement however it is highly unsuitable for our project because it is strictly non-iterative, meaning if there is an error or mistake in one of the phases, we cannot go back and rectify the error before resuming work to the next phase. In Information Technology projects, there is a high possibility of uncertainty and therefore a phase might have to go through extensive changes to iron out possible bugs. Therefore, the waterfall model cannot be chosen for our project.

3.1.2 Iterative and Incremental Development

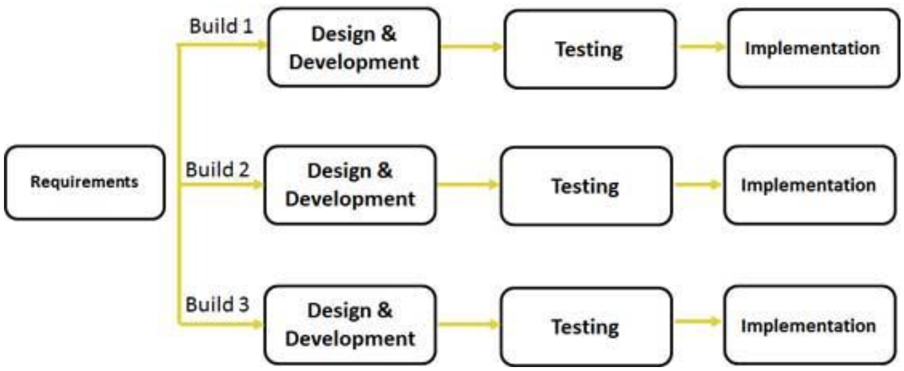


Figure 3.1.2.1 Iterative Model (istqbexamcertification.com, 2001)

The iterative and incremental model starts with a vague description of requirements. Project is developed by specifying and implementing a part of the software only. The outcome is then reviewed and further requirements are identified and gathered. The process is repeated iteratively and a new version of software is produced for each cycle of the model.

In other words, a rough product or outcome is created in each iteration of the model. Each iteration is then reviewed and improved until a satisfactory outcome is obtained. Despite the fact the we can get reliable user feedback of the system and fast development times, this model is risky because all system requirements are not gathered upfront. As a result, scope creep could occur where the project grows bigger and bigger uncontrollably, causing possible overruns in terms of cost and schedule.

In our project, we have limited time and cost to build and deliver the deliverables. As such, we cannot afford any overruns especially in terms of schedule. The project scope must be clearly defined in the beginning of the project so that scope creep can be prevented and the project is able to be completed within the given deadlines.

3.1.3 Agile Model

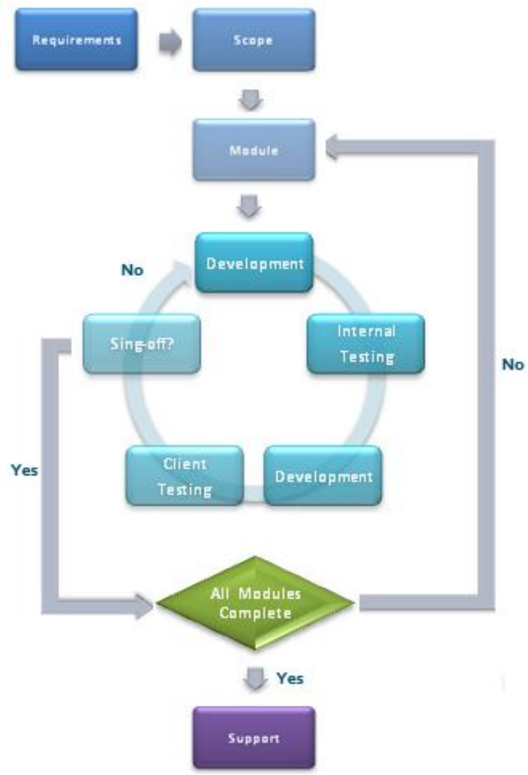


Figure 3.1.3.1 Agile Model (BayAmp, 2011)

The agile model is a type of iterative and incremental model where the project is developed in incremental and rapid cycles. Project tasks are organised according to priority in a "to-do" list and each tasks can be executed according to the iterative model. Each iteration is defined as a self-contained mini-project that comprises of analysis, development and testing. The completion of each iteration produces a sub-system that is ready to integrate with other sub-systems of the project defined earlier. Iterations are performed on each sub-system until no further changes or requirements are requested from the customer or stakeholders. The final step is to integrate all finalised sub-systems into a fully working system that is ready to be deployed.

This model ensures rapid development times, with emphasis on customer satisfaction, high productivity, good project design, high success rates and reduced risks. However, due to the iterative approach where changes are welcome even in the last stages of the project, it can result in the project exceeding the deadlines.

3.1.4 Selected Model

Among the three system development models discussed above, the Agile Model is deemed most suitable for this project and therefore it would be used to develop this project. The agile model allows us to divide the tasks in our objectives to smaller and feasible tasks or sub-systems. These sub-systems would then be iterated until all requirements and changes are met.

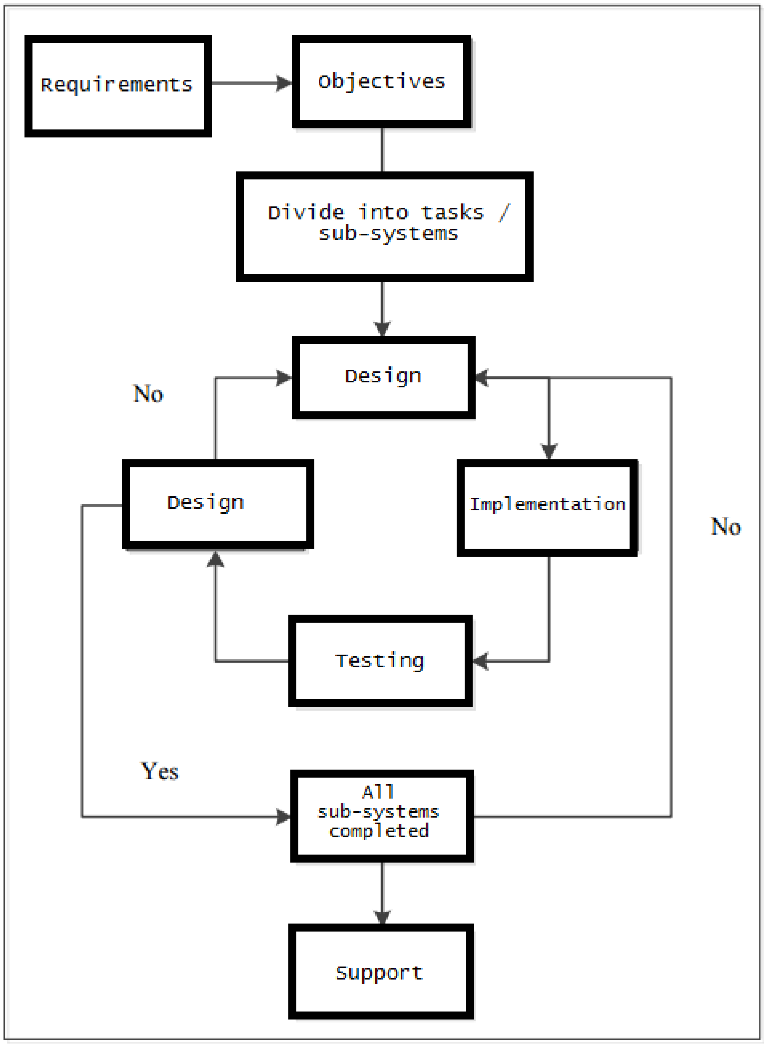


Figure 3.1.4.1 System Development model chosen for this project

With reference to the figure above, the project starts with the gathering of requirements. The requirements gathered are then used to define the scope and objectives of the project. The objectives are divided into smaller tasks / sub-systems in accordance to the agile model. The stakeholder of this project would be the supervisor of this project, and each completed sub-system would be presented to the supervisor to gather feedback and perform changes if necessary. Once the sub-systems are complete, they are integrated into a fully functional system. This indicates the completion of the project and the support phase is where maintenance works are carried out.

3.2 System Requirements

The following are the minimum system requirements for this project sorted into hardware and software categories.

3.2.1 Hardware

The following are the hardware used in this project:

A) MIB 520 Mote Interface Board (Programming board) - by Crossbow Technology Inc.

-Has a USB port and is used for programming as well as communication purposes. The MICAz mote attaches to the top of the board.

B) MICAz Motes - by Crossbow Technology Inc.

-A 2.4 GHz Mote module used for enabling low-power, wireless sensor networks. Used to emulate Active RFID readers and tags. It operates on the 2.4Ghz ISM band and is thus free from any licensing issues. Attached to the MIB 520 board for programming and communication purposes. Also attached to the .

C) 51-pin Breakout Board (also known as butterfly board)

-The breakout board has a 51-pin expansion connector that connects to the MICAz motes which allows us to interface various sensors and external embedded devices.

D) Personal Computer (PC)

-Pre-installed with TinyOS, Java SDK and MySQL database. Should also be installed with the “Smart Gate” program coded by my team-member. Acts as a base station to read the Active RFID tags once a vehicle is detected through the weight sensor. All entry and exit attempts are recorded in a database and a video sequence is marked and captured for any unauthorized entry. An alarm is also raised for unauthorized entry.

E) Arduino Uno R3 microcontroller board

- It is a microcontroller board which runs on an open-source prototyping platform to create interactive electronic projects. In this project it is used to control the servo which acts as a boom-gate. Receives instructions from MICAz mote to open or close the boom-gate and has a manual override push-button switch.

F) TowerPro SG90 Micro Servo

- A tiny and lightweight servo with high output power which rotates up to 180 degrees. Receives input from the Arduino board to open or close the boom-gate.

G) Webcam

- Any generic brand of webcam with a USB port would work for this project. This is because OpenCV has a wide range of libraries that readily interfaces with the webcams that are installed in the computer.

H) Weight Sensor

- A simple circuit that allows current to flow when both terminals are connected together. Uses the concept similar to a push-button switch that detects voltage when it is pressed. In this case, it is used to detect the presence of vehicles when the vehicle passes through.

Note: Software implementations are to be carried out by my team-member which includes a user-friendly front end software with GUI that is able to display video feeds from the webcam and also a database system designed to record entries and exits.

3.2.2 Software

A) TinyOS

- TinyOS is an open source operating system that provides software abstraction for low-power wireless devices used in wireless sensor networks and embedded network devices. In this project, we will use the XubunTOS operating system, configured to run as a virtual machine. This is because XubunTOS has TinyOS and other necessary software required for development preconfigured. As such, it simplifies the job of programming and running codes on the MICAz platform as compared to installing and configuring TinyOS on the Windows environment.

B) WampServer

- WampServer is a windows based web development environment that allows us to run web applications with PHP and a MySQL database. It also contains PhpMyAdmin which allows easy database management. Authorised tags are stored in the database along with details of the tag owners. Not only that but entry and exit details are also recorded in the database.

C) Oracle VM VirtualBox

- It is a cross-platform virtualization application that allows us to run another operating system simultaneously on top of the host operating system. In our case, The TinyOS is run through the VM VirtualBox software which runs on a Windows operating system.

D) OpenCV

- It is a library of programming functions that focuses on image processing. In our project, it is used to get live feed from the webcam and also record a video sequence of an invalid entry. Besides that, it is also used for number plate recognition.

E) Windows 7/8.1 Operating System

- A GUI operating system that is widely used throughout the globe. It is the host operating system in our project. The frontend GUI software and database server runs on Windows, whereas the backend MICAz software runs as a virtual machine via the VirtualBox application.

F) Java SE Development Kit 7

- It is a software development environment used to develop Java applications and applets. Both the frontend GUI software and backend software are compiled and run on the Java platform.

G) Arduino IDE

- It is an Integrated Development Environment used for writing and uploading codes into the Arduino microcontroller boards. The boom-gate is operated based on code written with the Arduino IDE.

3.3 Functional Requirements

The following are the functional requirements of this project:

A) Ability to add or delete authorised tags

- Administrators of the system must be able to add or delete a list of authorised tags on the database.

B) Ability to perform queries on the entry and exit logs

- Administrators of the system must be able to perform various queries such as invalid entries, etc and view them through the GUI.

C) Ability to display realtime live video-feed and record video sequence of invalid entry simultaneously

- The system must be able to simultaneously run two webcams, one to display live feed and the other to record a video sequence of an invalid entry.

D) Ability to detect vehicles with the weight sensors

- The system must be able to detect the presence of vehicles using the weight sensor which would trigger the base station to reading information from the tags.

E) Ability to read tag information and determine if it is authorised to enter

- The base station must be able to read tag information from the tags and perform a query on the database to determine if the particular tag is authorised or not.

F) Ability to open and close boom-gate when needed

- The Arduino operated servo must be able to open and close based on signal received from the MICAz mote and also based on the inputs received from a manual override push-button.

G) Ability to provide simple GUI and ease of operation with a mild learning curve

- The frontend GUI software running on the Windows operating system must be user-friendly and easy to use and have a mild learning curve.

H) Ability to perform queries via the local area network (LAN)

- The management of the university must be able to perform queries to retrieve entry and exit logs from the PC that controls the base station remotely as long as it is connected to the University LAN.

3.4 Project Milestones

Project Tasks	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Determining FYP title	■	■												
Data Collection and Requirements Gathering			■	■										
Determining Project Scope and Objectives				■	■									
Doing Literature Review				■	■	■	■	■						
Determining the correct Technology to be used								■						
Finalizing a complete Preliminary Report								■	■					
Determining the correct Methodology to be used										■				
Enhancements on Final report based on mistakes in preliminary report											■			
Determining System Requirements and setting up Timeline												■	■	
FYP 1 Documentation			■	■	■	■	■	■	■	■	■	■	■	
FYP 1 Presentation														■

Table 3.4.1 Gantt Chart depicting milestones for FYP 1

Project Tasks	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Program base station and active tags, and sample data to ensure base station is able to receive data from tags														
Develop weight sensor circuit and connect it to the MICAz mote to detect the presence of vehicles														
Develop Java code to run 2 webcams simultaneously, one to Display live feed and another to record invalid entry. Ensure recording format is clear and appropriate														
Develop Arduino source code to Operate the servo which controls the boom-gate and configure circuits to include a manual override push-button														
Setup database server using WampServer and create required Tables and perform queries to ensure it is in working order														
Interface hardware components With software developed by team-member, sort out synchronisation issues														

Final Testing and rectification Of errors and bugs														
FYP 2 Documentation														
FYP 2 Presentation														

Table 3.4.2 Gantt Chart depicting milestones for FYP 2

3.5 Estimated Costs

The estimated cost for FYP development is shown in Table 3.5.1 whereas the estimated cost to commercialise this project is show in Table 3.5.2. Due to the relatively high costs of purchasing actual Active RFID hardware, the project is emulated using MICAz motes during the FYP development phase. The concepts and algorithms used to program the MICAz motes to emulate the Active RFID technology are easily transferrable to the actual Active RFID platform during the commercialisation / real-life deployment phase.

During the commercialisation phase, weight sensor and boom-gate equipments designed for heavy-duty usage are purchased to ensure a reliable and low-maintenance system. Although the initial costs for commercialisation might appear to be steep, however one should take note that the costs of the tags (RM50.00 per unit) can be easily recouped within one semester of implementation in UTAR because the traditional car sticker is sold for RM100.00 which is double the cost of a unit of the Active RFID tag. The tags could be leased to students with a security deposit fee of RM50.00 in case of damage or loss. At the end of each semester, the tags could be recollected and reprogrammed to be ready for use during the start of a new semester.

Item	**Quantity**	**Price per Unit (RM)**	**Total Price (RM)**
MICAz Mote	5	Supplied by UTAR	-
MICAz Programming Board (MIB520)	1	Supplied by UTAR	-
A pair of AA batteries	5	Supplied by UTAR	-
Webcam	3	Self-Owned	-
51-pin butterfly board	1	Supplied by UTAR	-

PCB and weight sensor components	4	Supplied by UTAR	-
Arduino Uno R3 development board	1	Self-Owned	-
TowerPro SG90 Micro servo	1	Self-Owned	-
Personal Computer (PC)	1	Supplied by UTAR	-
WampServer, MySQL, TinyOS, Java JDK 1.7, Oracle VM VirtualBox, OpenCV	1	Freeware / Open Source / BSD-License	-
Total:			0.00

Table 3.5.1 Estimated Cost for FYP Development

Item	Quantity	Price per Unit (RM)	Total Price (RM)
Active RFID Tag*	3000	50.00	150,000.00
Active RFID Reader	2	5000.00	10,000.00
Batteries**	3000	5.00	15,000.00
Webcam	3	200.00	600.00
Above-ground weight sensor	4	2500.00	10,000.00
Boom-gate	1	2800.00	2800.00
Personal Computer (PC)	1	1500.00	1500.00
WampServer, MySQL, TinyOS, Java JDK 1.7, Oracle VM VirtualBox, OpenCV	1	Freeware / Open Source / BSD-License	-
Total:			189,900.00

Table 3.5.2 Estimated Cost for Commercialisation / Real-life deployment

(*, **) Note: The figure of 3000 units was derived from the total allocation of 3000 stickers each trimester.

3.6 Concluding Remarks

As a conclusion, the Agile Model is chosen as the most suitable system development model for this project. This project requires most of the hardware and software that are easily obtainable from the market with minimal cost. A projection of milestones in terms of Gantt charts helps keep project on track and on schedule. Estimated cost of commercialisation might seem high initially, however the major cost is contributed by the purchase of tags which can be easily recouped within the first semester of implementation. (cost of 1 Active Tag - RM50.00 vs cost of traditional sticker - RM100.00)

CHAPTER 4 SYSTEM DESIGN

4.1 System Architecture

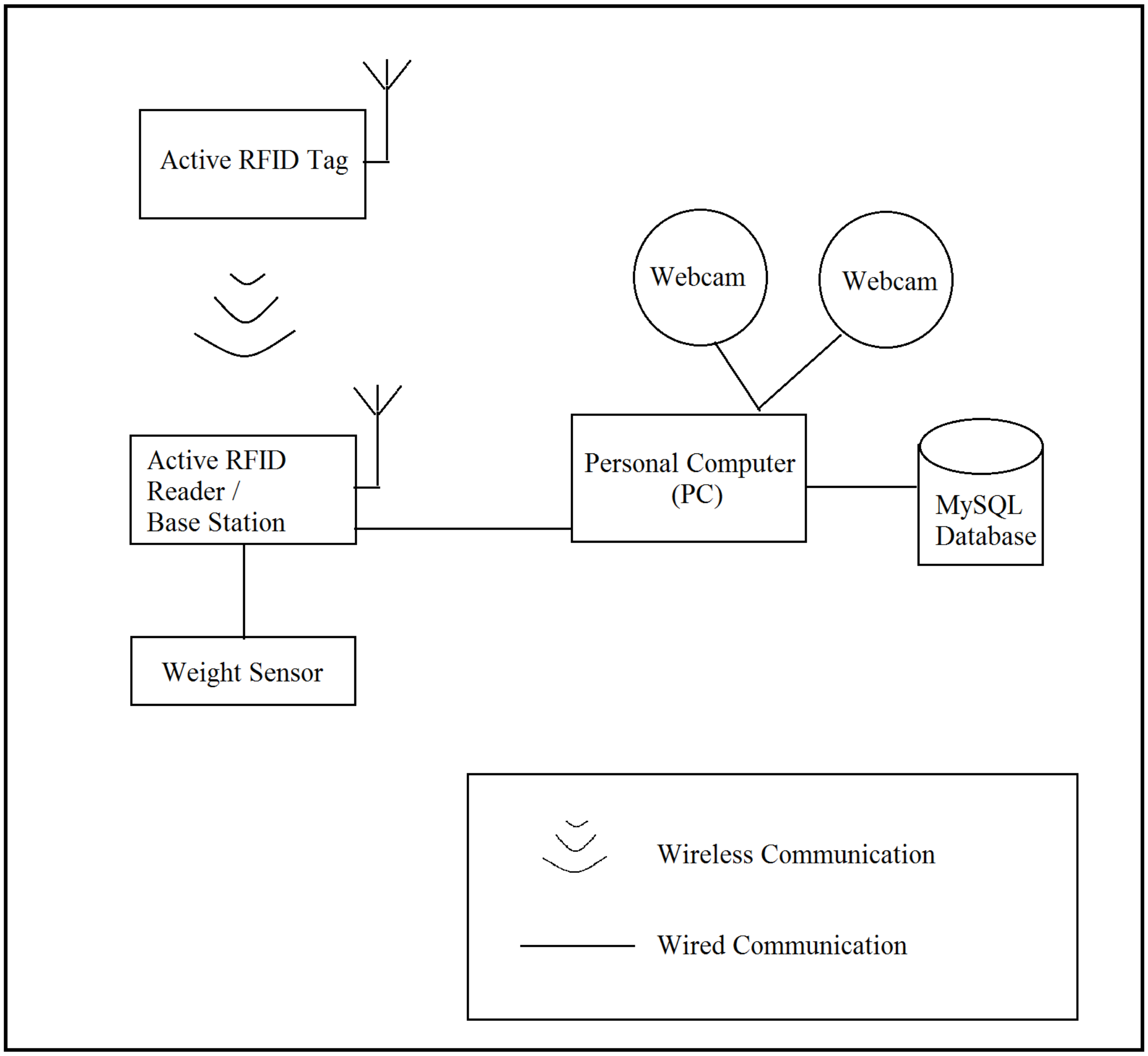


Figure 4.1.1 Overview of the System Architecture of this Project

Figure 4.1.1. shows the overview of the system architecture of this project. Active RFID tags transmit tag information wirelessly to the base station. The base station is triggered to read tags when vehicle passes through the weight sensor. The base station is connected to the personal computer where the frontend Java GUI software and backend Java program runs. Two webcams are connected to the personal computer, one to relay live feed and

another to record a video sequence of an invalid entry. An invalid entry is when the weight sensor at the base station senses a vehicle's presence, however it is not able to determine the ID of the vehicle through the active tag. The Java GUI displays the live feed and entry and exit logs of vehicles. Entry and exit logs are stored in the MySQL database which is hosted on the personal computer. Custom queries can be performed to obtain required information from the database through the Java GUI.

4.2 Functional Modules in the System

4.2.1 Active RFID Tag Module

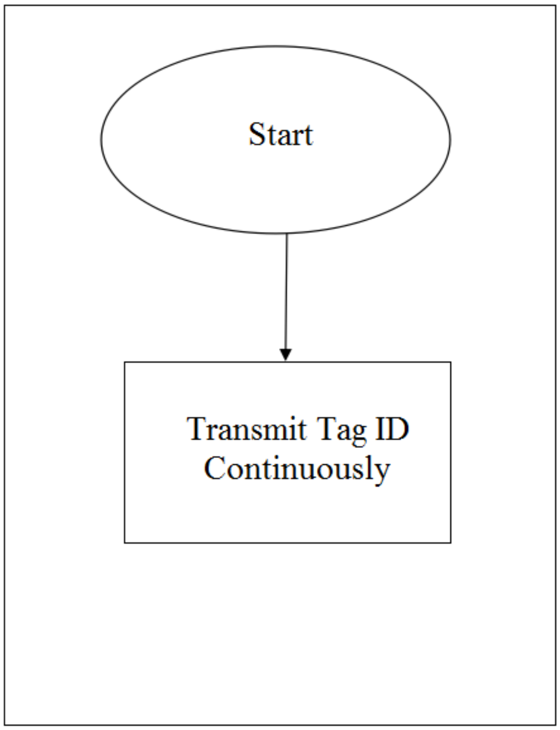


Figure 4.2.1.1 Flow Diagram of Active RFID Tag

Figure 4.2.1.1 shows a simple flow diagram for the Active RFID tag. Its job is to continuously transmit ID information back to the base station where the Active RFID reader is present.

4.2.2 Active RFID Reader / Base Station Module

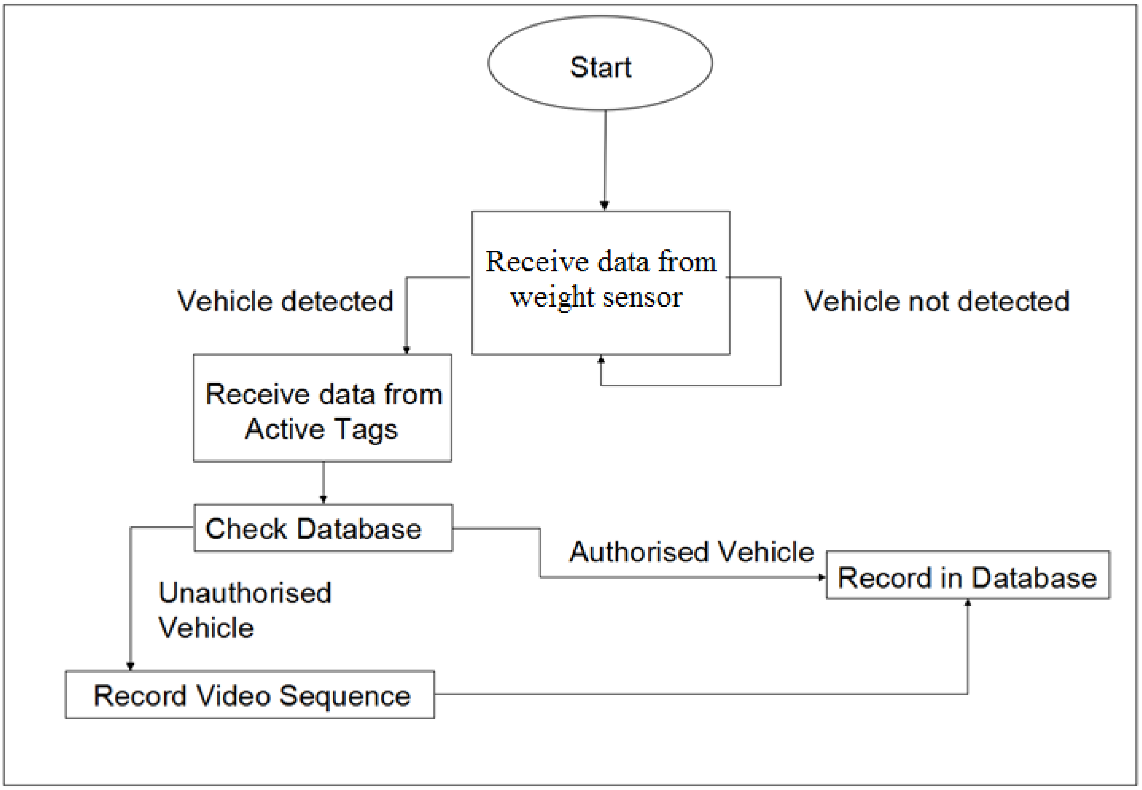


Figure 4.2.2.1 Flow Diagram of Active RFID Reader / Base Station

Figure 4.2.2.1 shows the flow diagram of the Active RFID reader which is also known as the Base Station. The Base Station continuously samples data from the weight sensor to detect the presence of vehicles. Once detected, it tries to read the ID information contained in the Active RFID tag located inside the vehicle. If it is unable to receive a positive identification, the vehicle is assumed to be unauthorized and a video sequence indicating the vehicle's number plate is recorded. Both entry and exit logs are recorded into the database.

4.2.3 Boom-Gate Module

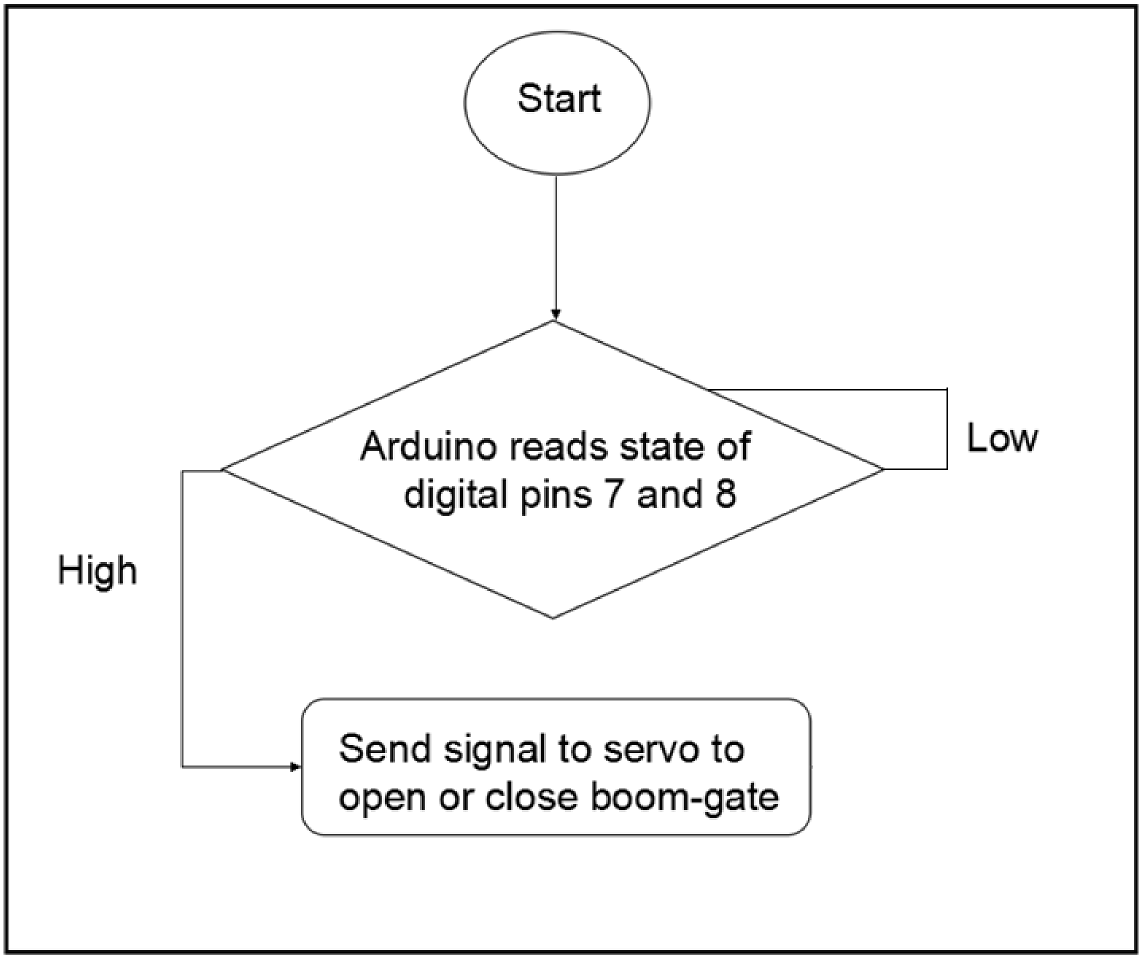


Figure 4.2.3.1 Flow diagram of the Boom-Gate module

Figure 4.2.3.1 illustrates the working concept of the boom-gate module. Once the Arduino microcontroller board is switched on, it continuously reads the state of digital pins 7 and 8. Digital pin 7 is connected to the PW pin of MICAz mote whereas digital pin 8 is connected to a push button which is a manual override switch. If a High-state is detected on either one of the pins, the boom-gate is closed or opened, depending on its last position.

4.2.4 Weight Sensor Module

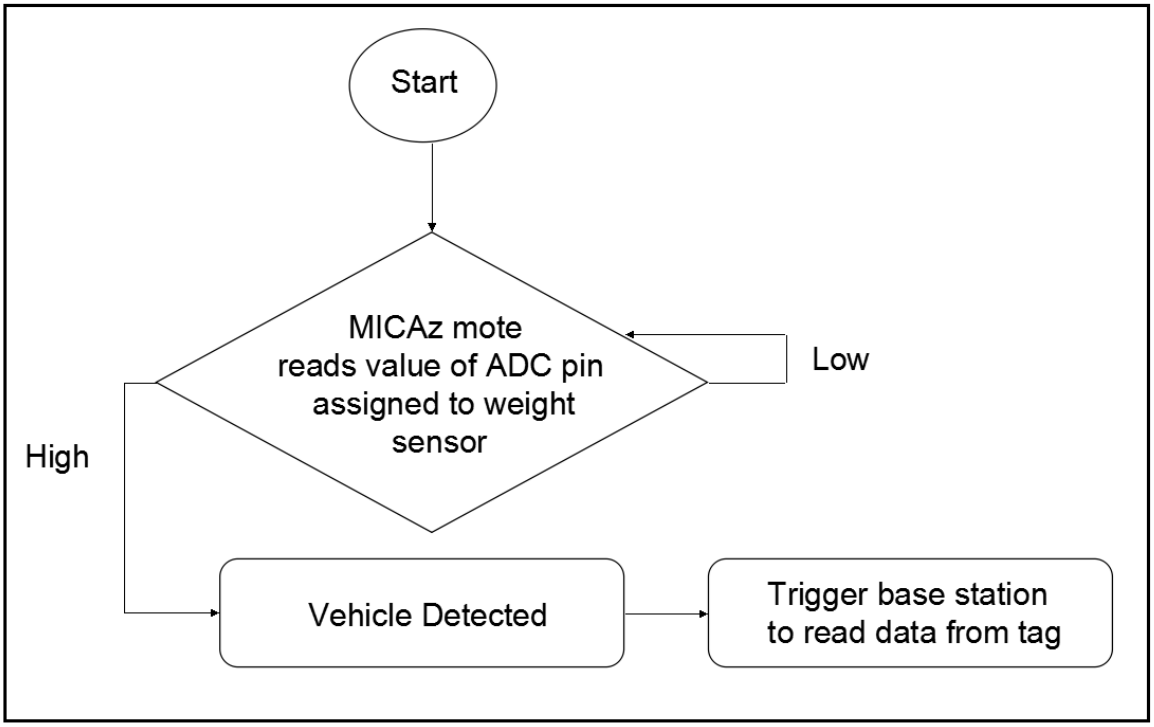


Figure 4.2.4.1 Flow diagram of the weight sensor module

Figure 4.2.4.1 illustrates how the weight sensor works. Once started, it continuously reads the value of the ADC pin that is assigned to the weight sensor. If a High state is detected (ADC value of 1023), the base station is informed that a vehicle is passing through the system and the base station is triggered to read data from tag positioned in the vehicle.

4.2.5 Webcam Module

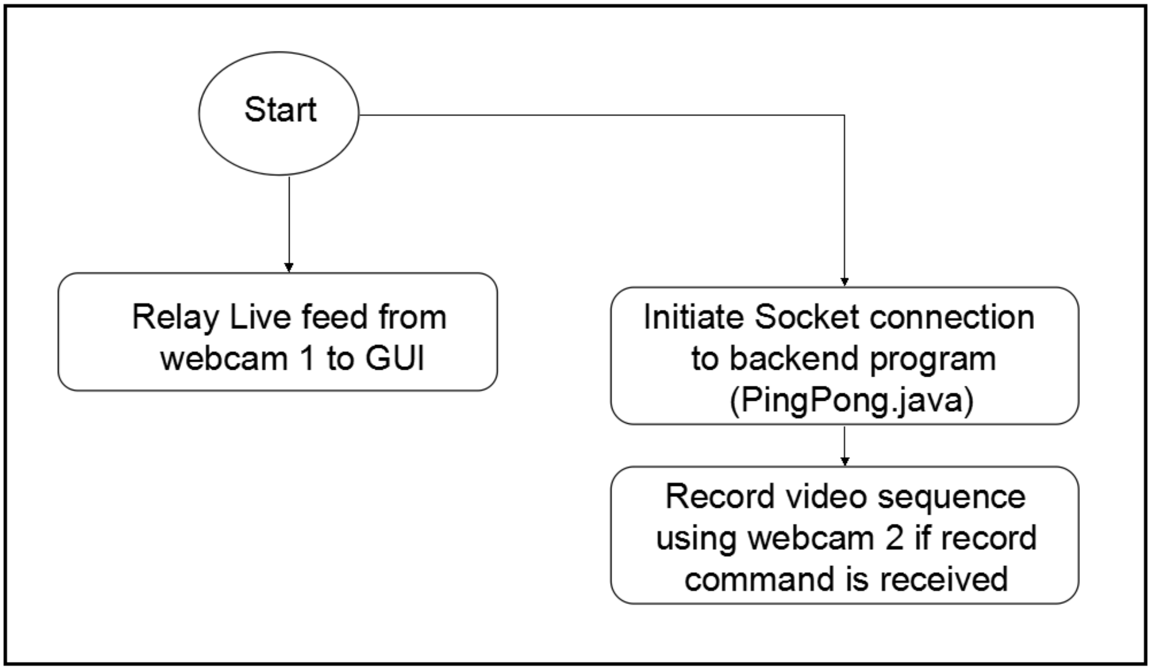


Figure 4.2.5.1 Flow diagram of the webcam module

Figure 4.2.5.1 shows how the webcam module works. The webcam module consists of two webcams connected to the personal computer. The first webcam installed in the personal computer (default camera) is designed to relay a live feed of the vehicles entering the system to the GUI frontend software whereas the second webcam installed in the system is designed to record a video sequence of an invalid entry. The backend Java program triggers the record function in the frontend GUI software via a socket, utilising a client-server architecture for communication.

4.3 System Flow

The system flow for this project is divided into two system flow diagrams. Figure 4.3.1 depicts the system flow on the entry lane whereas Figure 4.3.2 depicts the system flow on the exit lane.

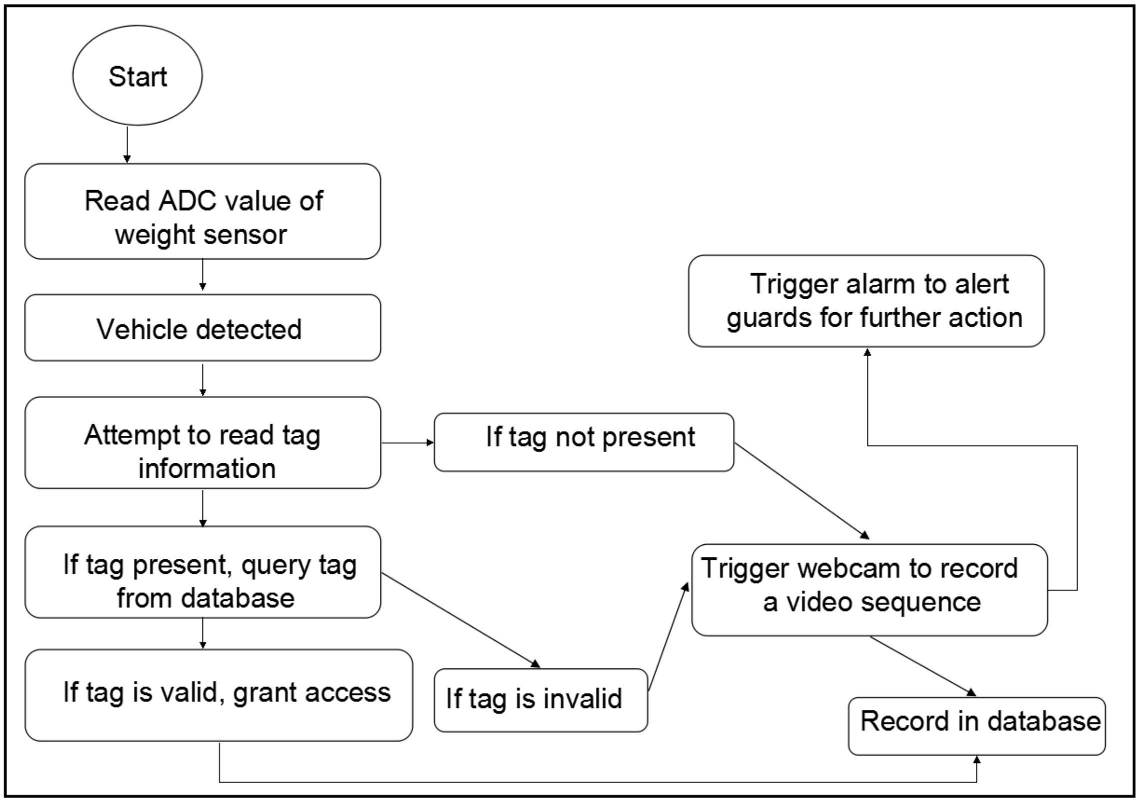


Figure 4.3.1 System Flow for the Entry Lane

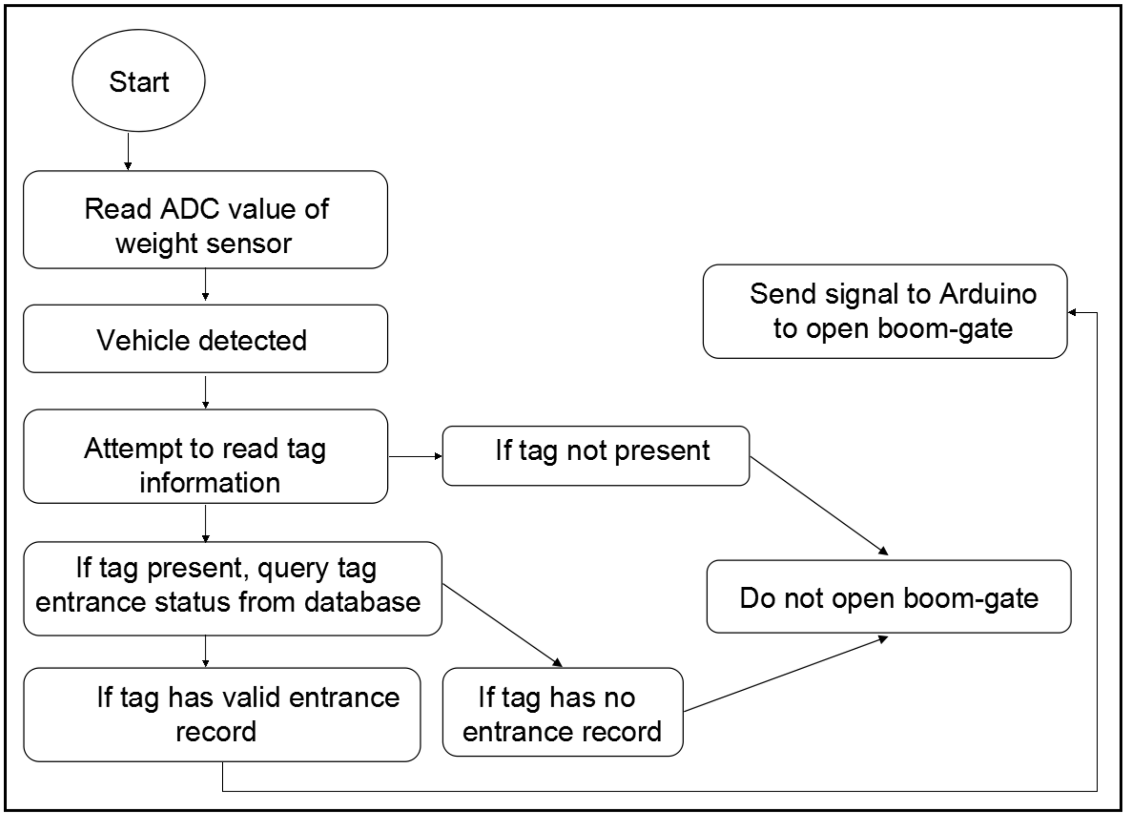


Figure 4.3.2 System Flow for the Exit Lane

4.4 Database Design

Field	Data Type
TagID (PK)	int
VNumber	varchar
VModel	text
VColour	text
StuName	text
StuID	varchar
StuContact	varchar
Status	int

Table 4.4.1 Records table design

Field	Data Type
Timestamp	datetime
TagID	int

Table 4.4.2 Entry table design

Table 4.4.1 shows the design of the records table and Table 4.4.2 shows the design of the entry table. Records table contains student info and vehicle info that is attached to a unique Tag. The entry table is used to store records of entry and exit time of vehicles. A join table query is performed and the outcome is displayed on the frontend Java GUI program when the system is started.

4.5 GUI Design

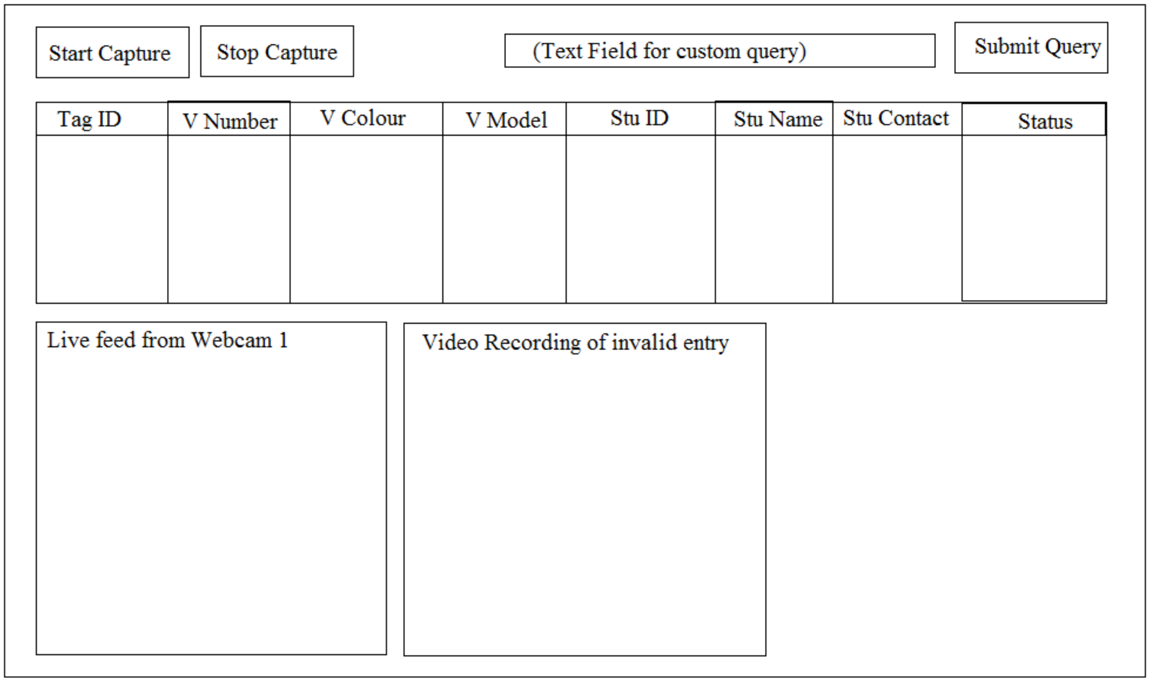


Figure 4.5.1 Simple GUI design for Java Frontend software

Figure 4.5.1 shows the final design of what the Java GUI frontend software is supposed to look like. It is designed in a very user-friendly manner that allows for simple navigation and operation.

Note: The software would be developed by my team-member, Chan Siew Meng.

4.6 Concluding Remarks

System design and architecture is elaborated and explained from a hardware viewpoint, inline with the title of this project. The entire system is divided into functional modules that can be later integrated together. System is divided into modules to provide an easy understanding of concepts and also to facilitate easier development and debugging process.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

5.1.1 51-pin Butterfly Board

We need the 51-pin butterfly board to interface the MICAz mote with external circuits or sensors. In the case of this project, we need it to connect to a weight sensor circuit and also to the Arduino micrcontroller board.

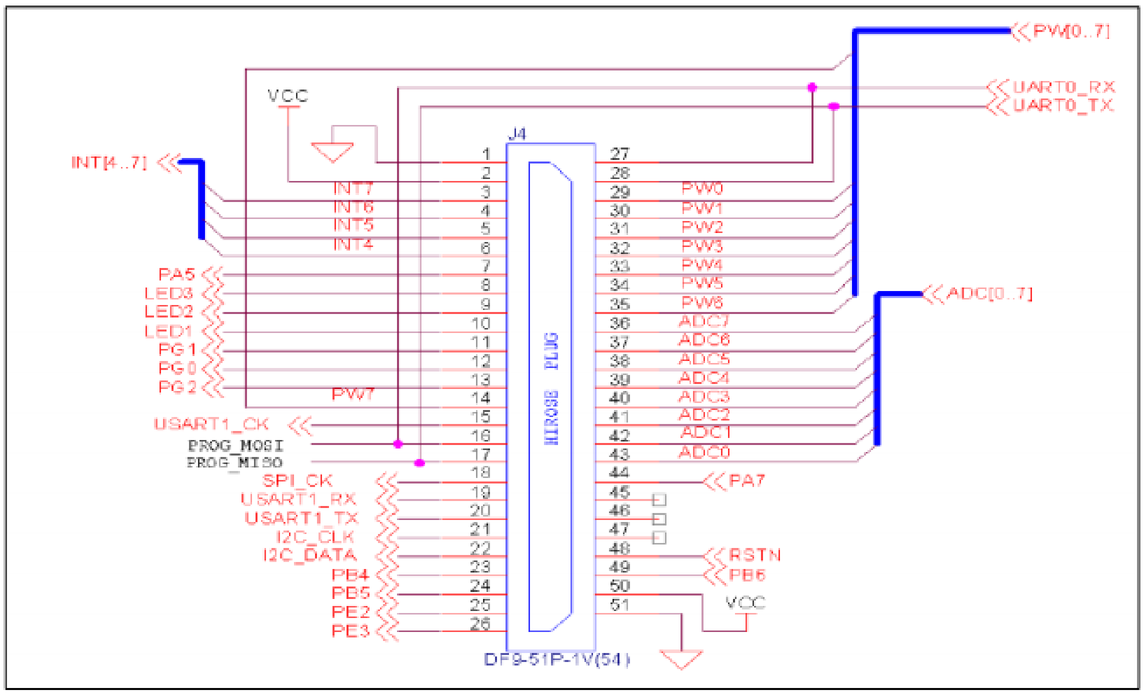


Figure 5.1.1.1 MICAz 51-pin expansion connector (Crossbow Technology, n.d.)

Figure 5.1.1.1 shows a pin-out of the 51-pin expansion connector on the MICAz motes. In this project, we only need to use few pins as shown in Table 5.1.1.1.

Pin	Usage
VCC	To power the external circuits / sensors from the MICAz mote
GND	Connection to ground
ADC5	To read values from weight sensor circuit 1
ADC6	To read values from weight sensor circuit 2
ADC7	To read values from weight sensor circuit 3
ADC8	To read values from weight sensor circuit 4
PW0	To send signals to Arduino microcontroller board which operates the boom-gate

Table 5.1.1.1 Table of pins needed and their usage

Figure 5.1.1.2 below shows the layout of the actual 51-pin butterfly board. In our project, wires are soldered onto the butterfly board according to table 5.1.1.1 to enable the MICAz mote to interface with the weight sensor circuits and the Arduino board.

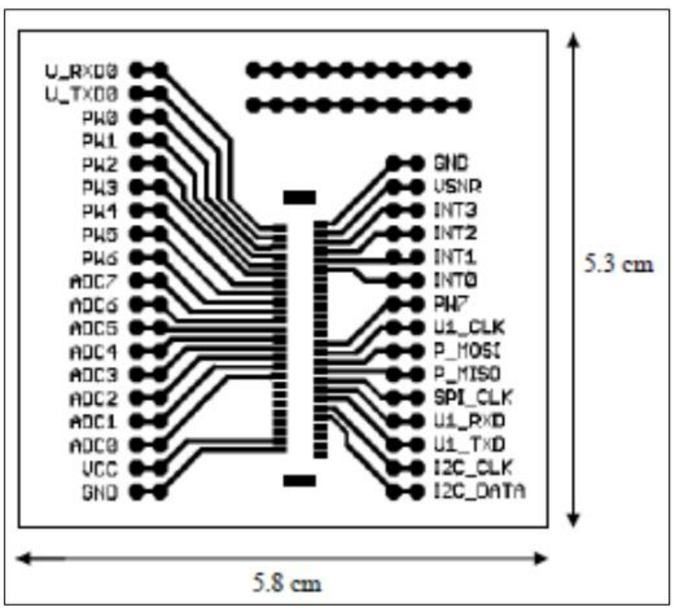


Figure 5.1.1.2 Layout of the 51-pin butterfly board

5.1.2 Weight Sensor Circuit

In this project, we need a total of 4 weight sensor circuits, 2 for the entry lane and another 2 for the exit lane. 2 sensors are deemed necessary to determine the position of the car in the lane and perform necessary actions. The first weight sensor circuit in each lane marks a vehicle entering the lane whereas the second sensor indicates that a vehicle has exited the lane.

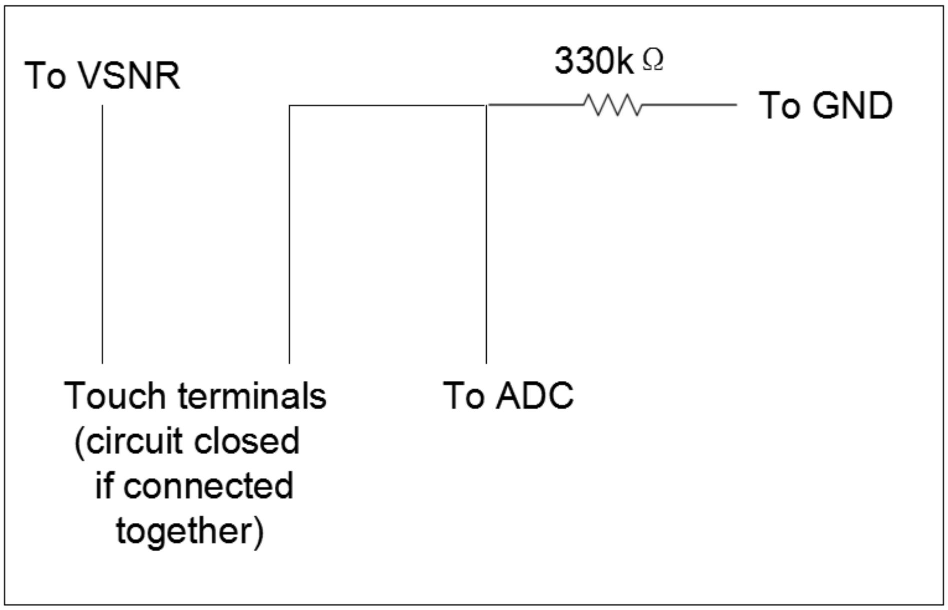


Figure 5.1.2.1 Simple circuit diagram of the Weight sensor

Figure 5.1.2.1 shows how the weight sensor circuit is designed. When the touch terminals are connected together, the circuit is closed and ADC reads a High value else circuit is open and ADC reads a Low value.

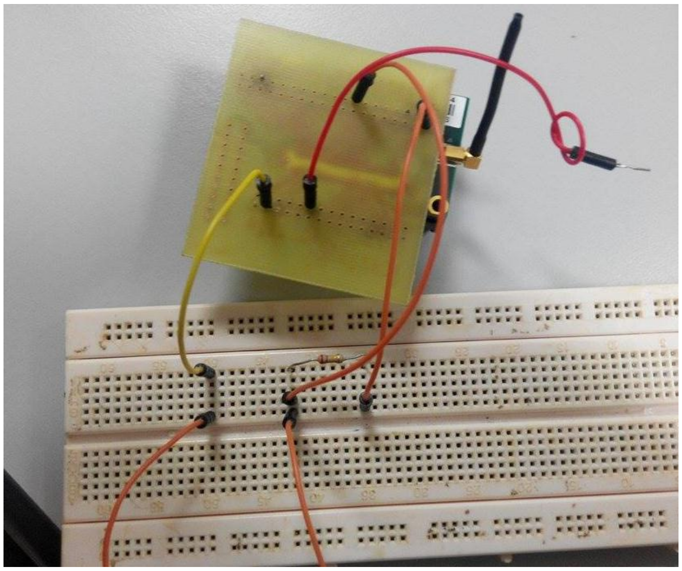


Figure 5.1.2.2 Actual setup of weight sensor circuit

The actual implementation of a single weight sensor circuit is shown in Figure 5.1.2.2 above. To make it more practical and reliable, the touch terminals are soldered onto two separate metal strips that are separated by a static-free foam that is 0.5cm thick. During simulation, when a vehicle's front axle and rear axle travels through the metal strips, the sensor circuit reads two High readings which indicates that it has succesfully sensed a vehicle.

5.1.3 Boom-Gate Setup

A TowerPro SG90 micro servo is used to operate the boom-gate. It has 3 pins as shown in Figure 5.1.3.1 below.

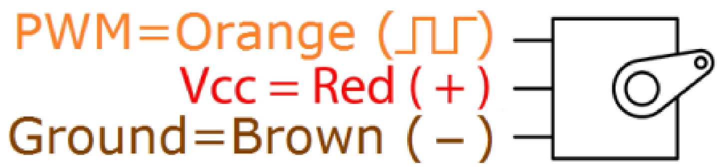


Figure 5.1.3.1 TowerPro SG90 micro servo terminals

The VCC and Ground are connected to the Arduino board's VCC (5V) and GND respectively. This means that the servo is powered up by the Arduino board itself. The PWM pin which is also known as the control pin is connected to Digital pin 7 of the Arduino board. Servo controls are established using the servo control library that is built-in the Arduino IDE.

Figure 5.1.3.2 shows the actual implementation of the boom-gate circuit interfaced with the Arduino board.

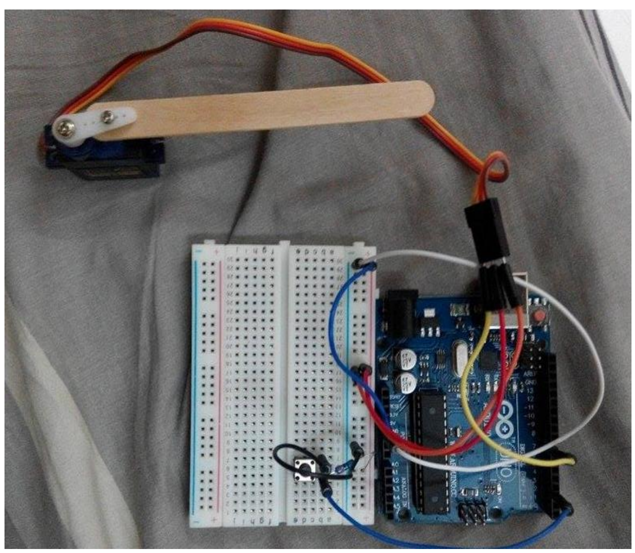


Figure 5.1.3.2 Actual Implementation of the boom-gate circuit

A push-button is connected to Digital pin 2 to function as a manual override to operate the boom-gate.

5.1.4 Webcam Setup

In this project, any brand of webcams would work, provided that the appropriate drivers have been installed. This flexibility is because of the high-level abstraction provided by the OpenCV library. Note: First webcam (default webcam) that is installed in the system relays live feed, whereas the second webcam provides video recording function.

Set up of webcams is relatively easy, once appropriate drivers are installed, the webcam can be connected to the PC via USB. Once the PC detects the webcam, the frontend Java GUI is ready to display a live feed.

5.2 Software Setup

5.2.1 Installation of Oracle VM VirtualBox and XubunTOS

1) Download the executable (.exe) file from https://www.virtualbox.org/wiki/Downloads

2) Follow on-screen instructions to install the VirtualBox application.

3) Obtain the XubunTOS Virtual Machine Image Disk (.vmdk) file from UTAR FICT's Networking Lab.

4) If step 3 is not feasible, proceed to download from the following link:

http://tinyos.stanford.edu/tinyos-wiki/index.php/Installing_XubunTOS_(with_TinyOS_from_tp-freeforall/prod_repository)_in_VirtualBox

5) Click on "New" and fill up the following details in the pop-up window as shown below:

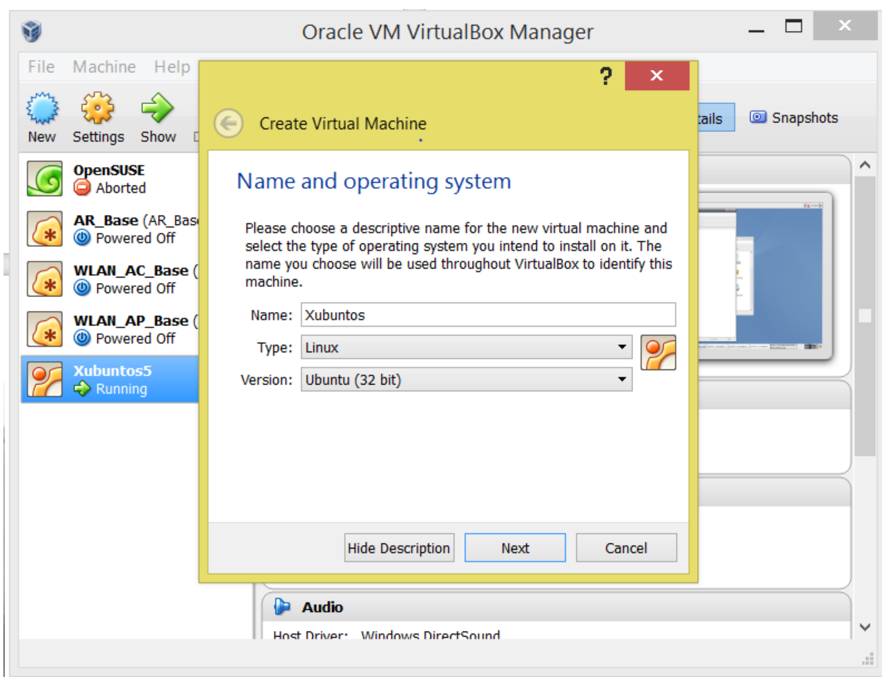


Figure 5.2.1.1 Creating Virtual Machine

6) Set an appropriate memory size to be used by the virtual OS and click "Next"

7) Select "Use and existing virtual hard drive" and navigate to the folder that contains the Xubuntos image file obtained earlier as shown below

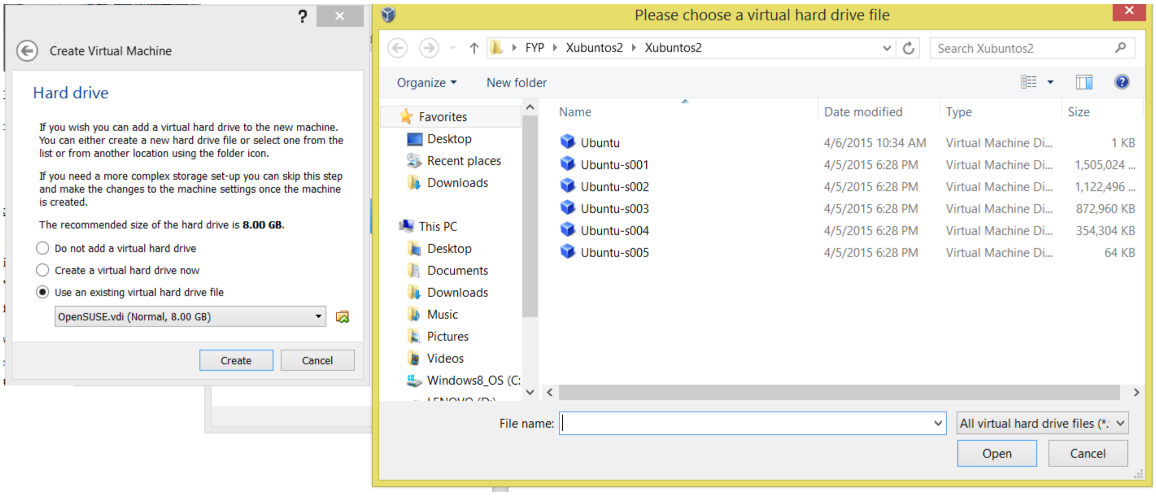


Figure 5.2.1.2 Selecting Virtual Image

7) Select "Ubuntu.vmdk" and click "Open"

8) Click "Create" and the XubunTOS has been configured successfully on the VirtualBox application

Note: XubunTOS has TinyOS preinstalled and relieves us of the extensive configurations needed otherwise.

5.2.2 Installation and Configuration of OpenCV

1) Download the OpenCV Library for Windows from:
http://sourceforge.net/projects/opencvlibrary/files/

2) Run the executable (.exe) file.

3) Extract the contents of the folder to a location, eg: C:\OpenCV2.4.x

4) Click the "Start-menu", right-click "Computer", then select "Properties". Click on "Advanced System Settings" and then click "Environment Variables"

5) Double-click the Path system variable and add one of the following as the variable value as shown below:

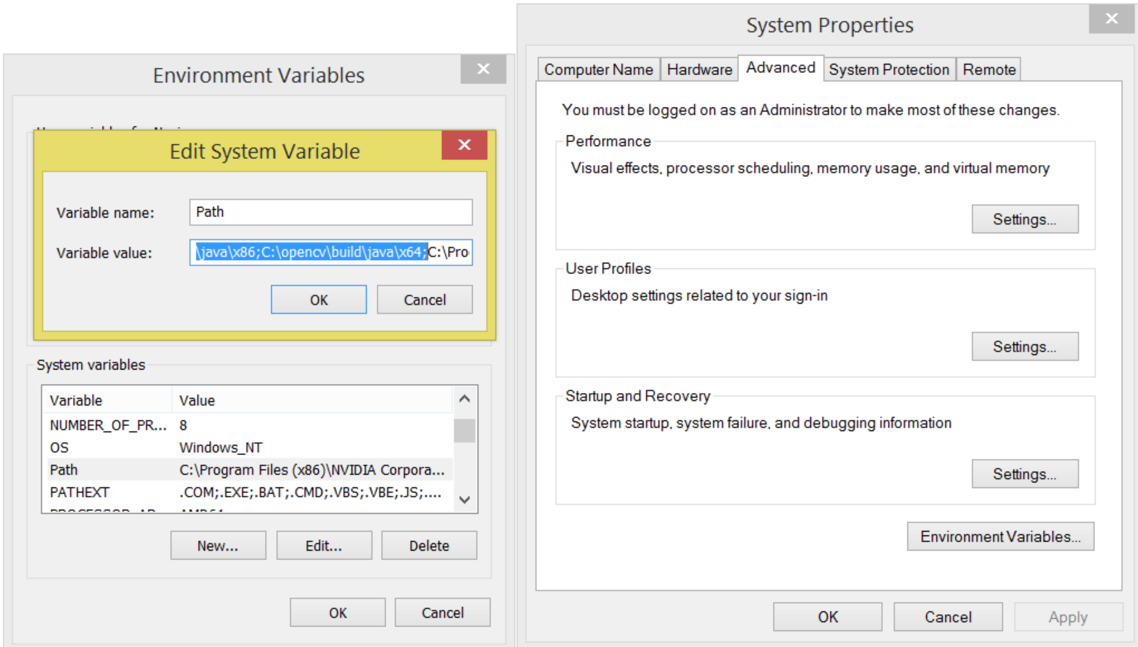


Figure 5.2.2.1 Setting Path for System Variables

- 64-bit path: C:\opencv\build\java\x64;

- 32-bit path: C:\opencv\build\java\x86;

Note: Paths above depends on the location where Step 3 is performed. Navigate to the Java sub-folder because our frontend software that deals with OpenCV is developed in Java.

8) OpenCV has been installed successfully.

5.2.3 MICAz Programming and Running Backend Software

There are 2 folders included in the CD which contains the MICAz source codes needed for both the Reader / Base station and the Active Tags.

1) Connect the MICAz mote to the MIB 510 Programming Board via USB

2) Copy both folders to XubunTOS. Navigate to the folder that contains the source code for base station and open the terminal. Key in the command "make micaz install,1 mib510,/dev/ttyUSB0"

3) Navigate to the folder that contains source code for Active Tags and open a terminal window in that location. Key in the command "make micaz install,102 mib510,/dev/ttyUSB0"

Note: a) The "102" in the command indicates the Tag ID and it has to be unique for every tag in the system. Tag ID can be programmed from a range of 102 to 4294967295.

b) Tag ID 101 is reserved for the mote which is interfaced with the weight sensor circuit and Arduino board. This particular mote is programmed with the same source code for tags, using the same command shown above.

4) Navigate to the folder called "java_app" and a open a terminal in that location.

5) Key in the following commands:

"export MOTECOM=serial@/dev/ttyUSB1:57600"

"javac *.java"

"java PingPong"

6) System resynchronises and the backend software is now running successfully.

5.2.4 Installation of WampServer and Database Setup

1) Download WampServer installer from http://www.wampserver.com/en/

2) Double-click the executable (.exe) file and follow onscreen instructions to install the software.

3) Once installation is completed successfully, select "Run WampServer" from the Start-Menu.

4) Open the web-browser and navigate to “127.0.0.1” which is also known as the localhost. Click on “phpmyadmin” and it will open the database management portal.

5) Navigate to the “mysql” database which is the default database created during the installation of WampServer.

6) Create two tables called “Records” and “Entry” according to the parameters below:

Field	Data Type
TagID (PK)	int
VNumber	varchar
VModel	text
VColour	text
StuName	text
StuID	varchar
StuContact	varchar
Status	int

Table 5.2.4.1 “Records” table specifications

Field	Data Type
Timestamp	datetime
TagID	int

Table 5.2.4.2 “Entry” table specifications

7) Database has been configured successfully and both Java frontend and backend applications can be run.

5.2.5 Installation of Arduino IDE and Programming the Arduino Board

1) Download the Arduino IDE from http://arduino.cc/en/main/software

2) Run the executable (.exe) file and follow onscreen instructions to install.

3) Once installation is completed successfully, run the Arduino IDE.

4) Connect the Arduino board via USB and open the Arduino sketch (source code) included in the CD. Then click on the “upload” button as shown below:

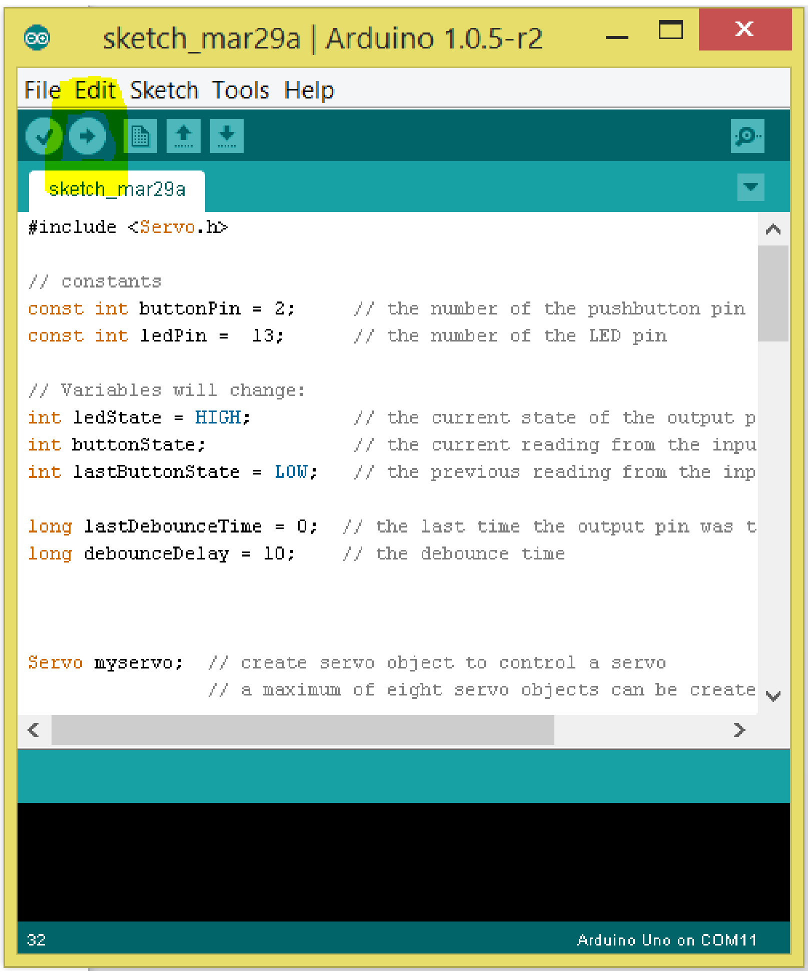


Figure 5.2.5.1 Uploading Code into Arduino board

5) Arduino has been programmed successfully

5.3 System Operation

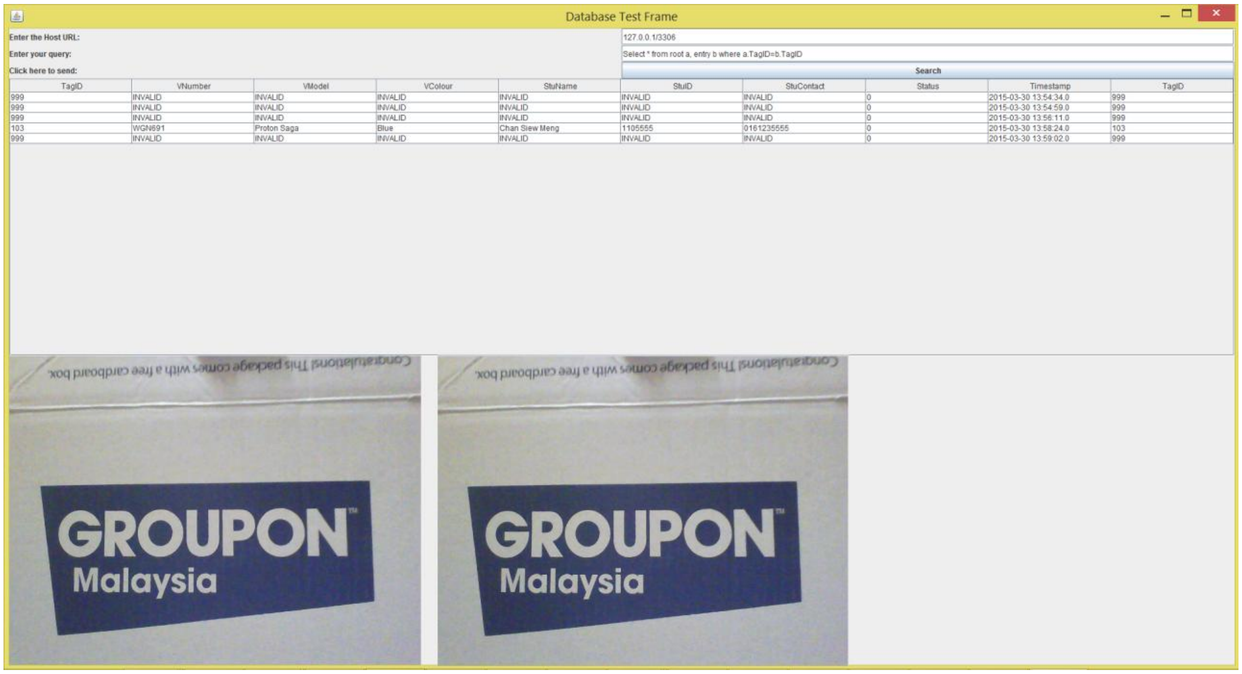


Figure 5.3.1 Java Frontend GUI Application in Action

Figure 5.3.1 shows the frontend GUI application running. If a vehicle attempts to enter, the database would be queried to determine the validity of tags. Both valid and invalid entries are logged into the database and displayed on the GUI. The first image shows a live video feed whereas the second one shows the recording that takes place in case of an invalid entry.

5.4 Concluding Remarks

The process involved in the hardware part of the system is fairly straighforward and simple although some knowledge in basic electronics is required for the set up of certain circuits. The software part is a little long-winded, however detailed steps are elaborated to ensure users do not face any difficulty in the setup of software. All required software are included in the CD.

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

There are three parameters that define the system performance of this project. The first parameter is reliability of the weight sensor to detect vehicles accurately and also to prevent false detection. The next one is the synchronisation between the webcam and frontend software to ensure recording is executed in a timely manner. The last parameter is to ensure that the webcam is able to capture a reasonably clear video depicting the number plate of the unauthorized entry.

To ensure the weight sensor is able to detect vehicles accordingly, the system must be deployed in a lab environment and readings must be obtained. From the readings obtained, the reliability of the weight sensor can be deetermined and the feasibility of using a weight sensor to detect vehicles can be determined. Next, the broadcasting power of the reader and tags must be fine tuned to allow an optimal detection range and also to reduce overlapping between the entry and exit lanes.

The time taken for webcam to start recording from the moment it is triggered by the frontend software must be determined. Necessary adjustments must be made according to the time taken to ensure recording of invalid entry is started at the exact moment during which the invalid entry occurs. Lastly, the webcam must be set up and deployed at multiple areas to determine the correct angle so that the video sequence captured identifies vehicles’ number plate clearly.

6.2 Testing Setup and Results

6.2.1 Reliability of Weight Sensor

Speed of Vehicle across Weight Sensor	Successful Detection
Slow	Yes
Slow	Yes
Slow	Yes
Moderate	Yes
Moderate	Yes
Moderate	Yes
Fast	No
Fast	Yes
Fast	No

Table 6.2.1 Reliability of weight sensor at different vehicle speeds

The Figure 6.2.1 shows the reliability of the weight sensor in detecting presence of vehicles at different speeds. The weight sensor experiences no difficulty in sensing vehicles at slow to moderate speeds, however at fast vehicle speeds, it is a hit-or-miss affair. This is due to the fact that the simple weight sensor used was not sensitive enough. Despite the fact that the weight sensor circuit does not perform well under fast vehicle speeds, it is not a major problem because the actual implementation of this project would be using heavy-duty, full-fledged weight sensor that are specifically made to detect vehicles.

6.2.2 Synchronisation Between Webcam and Frontend Software

After 20 attempts of triggering the webcam to record a video sequence, it is found that if the first recording is triggered within 30 seconds of starting the Frontend program, it takes up to 8 seconds to start recording (counting from the time it is triggered). The subsequent recordings however start almost instantaneously upon triggering. It can be concluded that this is due to initialisation time needed for OpenCV and the Webcam hardware. Our project is unaffected on one condition - that no invalid entry must occur within the 30 seconds of starting the Frontend program.

6.2.3 Quality and Usability of Recorded Video

Quality of recorded video is set at 640x480 in .mp4 format at 30 frames per second. This setting strikes the best balance between quality and size and the recording produced is clear enough to determine a vehicle's number plate. The webcam must be placed at an angle of 30 to 45 degrees. This angle is determined to be the best angle to capture a vehicle along with its number plate in case of an invalid entry.

6.3 Project Challenges

Among the many issues and challenges in this project are:

A) To ensure accurate detection of vehicles and Active RFID tags.

B) To emulate Active RFID technology with MICAz motes.

C) To design and implement the boom-gate with 2 controls (1 from MICAz mote and 1 from push-button)

D) To design and implement the weight sensor circuit that is used to detect entry and exit of vehicles in each lane.

E) To install OpenCV and develop Java code to run two webcams simultaneously.

F) To determine the angle of webcam so that it produces a clear video feed of the unauthorized vehicle's number plate.

G) To sort out synchronisation issues between the software and hardware and ensure that the system runs together as intended.

6.4 SWOT Analysis

Figure 6.4.1 SWOT Analysis

6.5 Objectives Evaluation

All 7 objectives of this project are achieved successfully as indicated below:

A) The MICAz motes are able to emulate the Active RFID technolgy very well.

B) The weight sensor circuit developed is able to detect the entry and exit of vehicles into a lane.

C) 2 webcams are able to run simultaneously with no issues or delay.

D) Boom-gate is able to open and close from signals received from the MICAz mote and also the manual push-button

E) Database server is successfully setup and records can be inserted and queried without any issues

F) There is minimal delay between triggering of recording and actual recording which is negligible because invalid entry is recorded clearly.

G) The entire system was developed with minimal cost and the frontend software is user-friendly.

6.6 Concluding Remarks

System testing based on performance metrics yielded satisfactory results. Despite the multiple challenges faced throughout the execution of this project, all objectives were met. In a nutshell, the final outcome of this project is highly successful.

CHAPTER 7 CONCLUSION

7.1 Conclusion

The Final Year Project aims to train students to think out of the box and come up with solutions to problems that are existent in our daily lives. As mentioned previously in the Motivation and Problem Statement, there were many flaws and faults with the current manual vehicle entry system in the University. However, none of the ready-made offerings in the market were able to fulfill the requirements of this project. Thus, this project was designed to solve all the problems present. The ultimate goal of this project was to create a fully working prototype of the hardware implementations of "Smart Gate" for UTAR and the final outcome was successful. In other words, the system is a convenient, low-cost, highly efficient and secure entry system for the University that increases security and reduces congestion at the same time.

The actual implementation (real-world deployment) would require minimal effort because the concepts and architecture are already derived during the development of this project. All that there is to be done is to produce codes that are recognised by the actual Active RFID hardware and the system is ready for deployment.

7.2 Recommendation

A parking management module could be added as a future enhancement to this project. This would use sensors at the parking lots located at various blocks of the University to facilitate users to find an empty lot easily. Besides that, by performing data mining on the entry and exit logs obtained, usage pattern of the users could be obtained which would then facilitate in the total allocation of tags. In other words, more tags could be allocated to users compared to the current rigid system because it takes into account the entry pattern of users.

REFERENCES

BayAmp, 2011, Agile Software Development Process. Available from: <http://www.bayamp.com/methodology.html>. [14 August 2012].

ComputerSci.ca, 2007, Educational flaws: Programming with the waterfall model. Available from:<http://compsci.ca/blog/educational-flaws-programming-with-thewaterfall-model/>. [14 August 2012].

Crossbow n.d., MICAZ: Wireless Measurement System. Available from: <http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf>. [19 July 2012].

EcoSensa Technologies Sdn Bhd. 2011, Passenger Tracking - Active RFID Vehicle & Passenger Tracking System. Available from: <http://rfid.ecosensa.com/images/RFIDPT/eco20activerfid20brochure.pdf> [23rd March 2014].

Evizal, Tharek, A. R. and Sharul Kamal, A. R. (2010) Active RFID Technology for Asset Tracking and Management System. Available from: <http://iaesjournal.com/online/index.php/IJECE/article/view/3855> Accessed on 29th March 2014, 2.11 pm.

MAGNET Electronic & Automation Sdn. Bhd. 2013, Our Solutions - Premium Long Range Parking Access System. Available from: <http://www.magnet.com.my/our-solutions/> [5th March 2014].

M-Tech Innovations Ltd. 2013, RFID Vehicle Access Control System. Available from: <http://iaesjournal.com/online/index.php/IJECE/article/view/3855> [21st March 2014].

Zahari Zakaria, M. (1998) The Malaysian RFID scenario. *Mohd Zahari Zakaria has the story on its current applications market and trends.* Available from: <http://myconvergence.com.my/main/images/stories/PDF_Folder/july2008/MYC03_p34_37_RFID.pdf> Accessed on 25th March 2014, 1.10pm.

APPENDICES

Appendix A

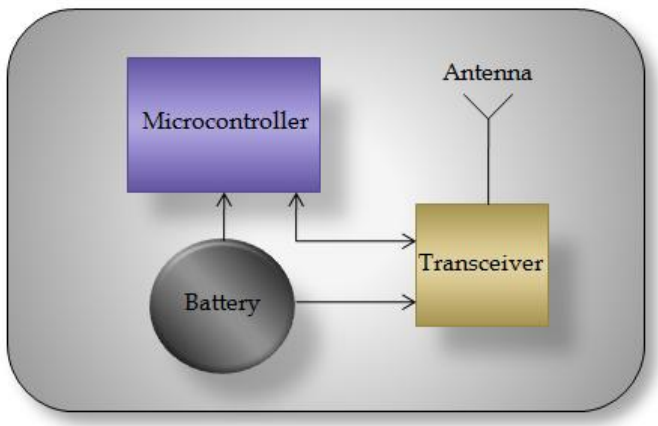


Figure A-1: An illustration of an active RFID Tag (Evizal, 2013)

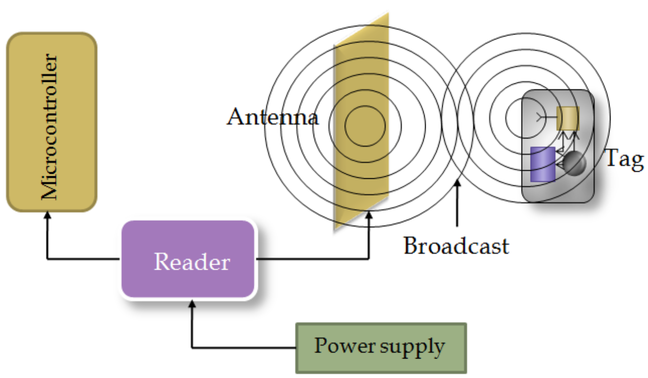
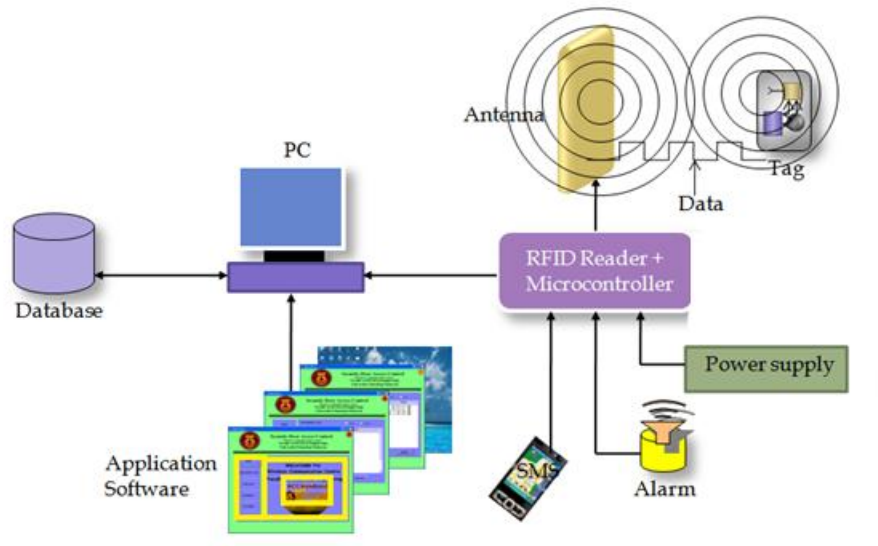


Figure A-2: An illustration of the basic concepts of an Active RFID System

(Evizal 2013)



A1

Figure A-3: An overview of an existing project utilising Active RFID Technology

(Evizal, 2013)