# SECURITY PARKING SYSTEM

Oleh

**MUHAMMAD ARIF BIN IZAHAR**

**MOHD HAKIM BIN ROZALI**

**Penghantaran Laporan Ini Adalah Untuk Memenuhi Keperluan Untuk Penganugerahan Diploma Kejuruteraan Elektronik (Komputer) di Jabatan Kejuruteraan Elektrik Politeknik Seberang Perai**

**JUNE 2016**

**PENGISTIHARAN**

Di sini saya mengistiharkan bahawa laporan ini adalah berdasarkan hasil kerja saya sendiri dengan bantuan maklumat daripada sumber-sumber yang diberitahu di dalam pengakuan. Saya juga mengistiharkan hasil projek saya ini tidak pernah dihasilkan oleh mana-mana pelajar lain serta dari institusi pengajian yang lain.

.................................................... ....................................................
(Muhammad Arif Bin Izahar) (Mohd Hakim Bin Rozali)

Tarikh : 28.10.16

Di sahkan oleh Penyelia Projek :

....................................................
(Tuan Syed Adnan Bin Syed Othman)
Tarikh : 28/10/2016

SYED ADNAN B SYED OTHMAN
Pensyarah
Jabatan Kejuruteraan Elektrik
Politeknik Seberang Perai
Permatang Pauh
Pulau Pinang.

*(cop pensyarah)

# CONTENTS

## ABSTRACT

Radio-frequency identification (RFID) is an automatic identification method wherein the data stored on RFID tags or transponders is remotely retrieved. The RFID tag is a device that can be attached to or incorporated in to a product or a person for identification and tracking using radio waves. Some tags can be read from several meter away, beyond the line of sight of the reader. RFID technology is used in vehicle parking system. The system normally consists of a vehicle counter, sensors, display board, gate controller, RFID tags and RFID reader. Presented here is an automatic vehicle parking system using Arduino Uno.

This system create to reduce the problem at the parking place. Other than that, this system to reduce the problem at the parking place. Moreover, with this system easy to identify users who often make mistakes in the parking place. Furthermore, to allows users to park the vehicles in an orderly. Lastly, aims to facilitate.

**ABSTRAK**

*pengenalan frekuensi radio (RFID) adalah satu kaedah pengenalan automatik di mana data yang disimpan pada tag RFID atau transponder adalah jauh diambil. Tag RFID adalah alat yang boleh dipasang pada atau digabungkan masuk ke produk atau orang yang pada pengenalan dan pengesanan menggunakan gelombang radio. Beberapa tag boleh dibaca dari beberapa meter jauhnya, di luar garis penglihatan pembaca. teknologi RFID digunakan dalam sistem parkir kenderaan. Sistem ini biasanya terdiri daripada kaunter kenderaan, sensor, memaparkan Lembaga, pengawal pintu, tag RFID dan pembaca RFID. Di sini ialah satu sistem letak kenderaan automatik menggunakan Arduino Uno.*

*Sistem ini mewujudkan untuk mengurangkan masalah itu di tempat letak kereta. Selain daripada itu, sistem ini untuk mengurangkan masalah itu di tempat letak kereta. Selain itu, dengan sistem ini mudah untuk mengenal pasti pengguna yang sering melakukan kesilapan di tempat letak kereta. Tambahan lagi, untuk membolehkan pengguna untuk meletak kenderaan secara teratur. Akhir sekali, bertujuan untuk memudahkan pengguna.*

**ACKNOWLEDGEMENT**

In the name of Allah S.W.T the Most Gracious and Merciful. Thanks and praise to Allah S.W.T for giving me strength and patience to complete our project successfully.

First of all, we would like to take this opportunity to dedicate our highest appreciation to our supervisor, Mr. Syed Adnan Bin Syed Othman for his guidance and idea contributions. He has made a lot of effort and advice in order to complete our project successfully. Many thanks as well for her time of a proofreading process.

Our sincere thanks go to all members of the staff of the Engineering Department, JKE, who helped us in many ways and for their assistance and share the information.

Finally, we express our gratefully to our beloved parents, brothers and sisters that have gave a lot of courage during finished this Final Year Project. we also thanks to all classmates that also give us a lot of support and idea to do this project.

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

## 1.0 INTRODUCTION

Radio-frequency identification (RFID) is an automatic identification method wherein the data stored on RFID tags or transponders is remotely retrieved. The RFID tag is a device that can be attached to or incorporated in to a product or a person for identification and tracking using radio waves. Some tags can be read from several meter away, beyond the line of sight of the reader. RFID technology is used in vehicle parking system. The system normally consists of a vehicle counter, sensors, display board, gate controller, RFID tags and RFID reader. Presented here is an automatic vehicle parking system using Arduino Uno.

This system create to reduce the problem at the parking place. Other than that, this system was specially for polytechnic students at the Parking Lot.
Moreover, to recognize identity of students polytechnic. Lastly, easy to identify users who often make mistakes in the parking.

In this project we also use Microsoft Access and Visual Basic 6.0 as a Database system to make a connection with Arduino and RFID. With this database we can recognise who enter the Parking Lot using their matric card as a Identity for student in Polytechnic of Seberang Perai.

## 1.1 STATEMENT OF ISSUE PROJECT

- Each system requires a program of appropriate and correct.
- Properly fitting components and arranged so that unwanted things do not happen.
- Learn how to use a arduino properly before using.

## 1.2 OBJECTIVES OF THE PROJECT

There are several key objectives as guidelines for this project:

- Identify the way to solve the problem at Parking Lot.
- Recognize identity of user at Parking Lot.
- Identify users who often make mistakes in the parking.

## 1.3 PROJECT SCOPE AND LIMITATIONS

The scope or the limit of the project as a reference should be made to ensure that the implementation of the project is not out of the objectives to be achieved . The scope of the project objectives or goals are set by the project . There are:

- Need time in finding all the component and accessories.
- A little bit high cost.
- Process of installation maybe need more time.
- Must do some research in finding where to find the component and accessories.
- Learn step by step the process in successfully the project.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 INTRODUCTION

This chapter describes the main components used in the production of this project. In addition, this chapter explains the advantages of the major components used.

## 2.2 The Main Controller Used

### 2.2.1 Arduino

#### I. INTRODUCTION TO ARDUINO

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. **Arduino boards** are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

#### II. Why Arduino?

The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than RM50.
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and

people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it.

## III. Arduino Boards

The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. We'll talk about what's on it and what it can do later in the tutorial.

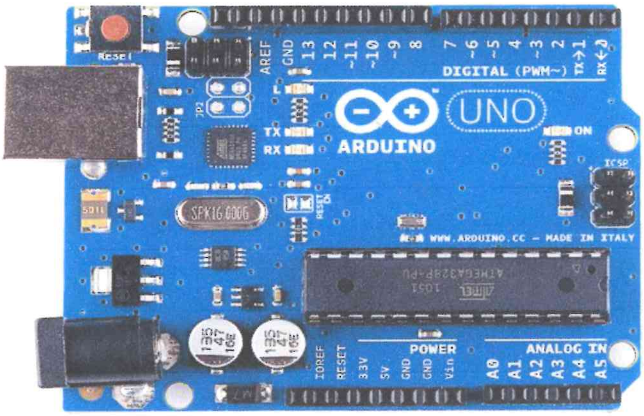


Figure 2.1: Arduino UNO

## IV. Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

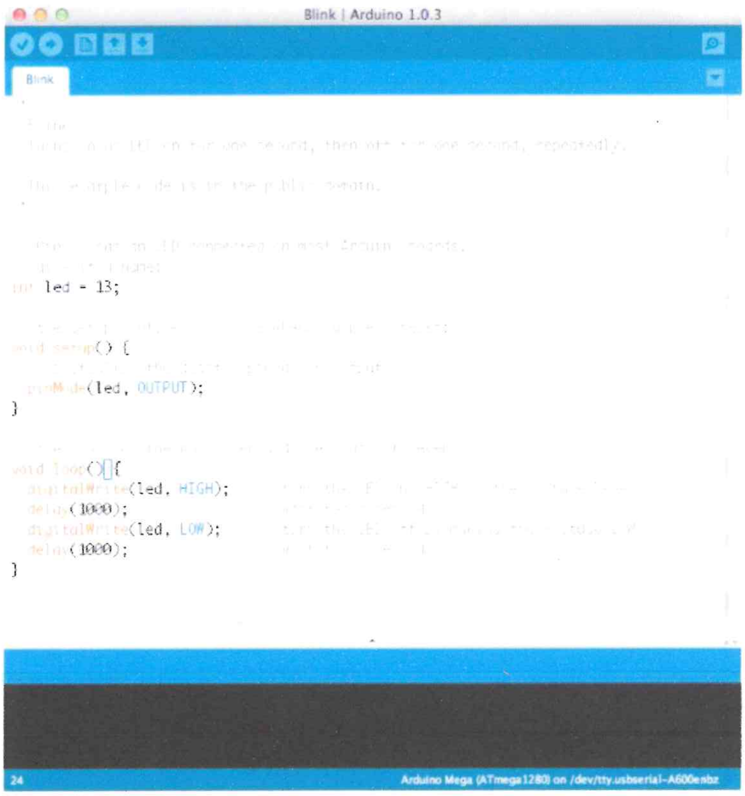


Figure 2.2 : This is a screenshot of the Arduino IDE

## V. What's on the board?

There are many varieties of Arduino boards that can be used for different purposes. Some boards look a bit different from the one below, but most Arduinos have the majority of these components in common:

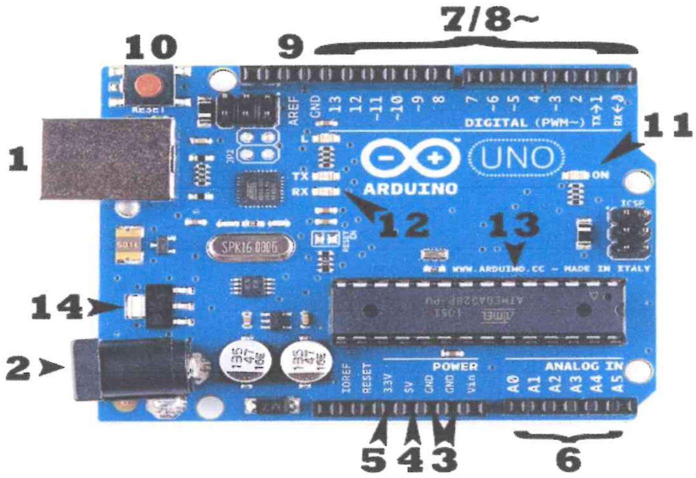


Figure 2.3 : Power (USB / Barrel Jack) (a)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board.

**NOTE:** Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts. Reference (a)

**Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)**

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a **breadboard** and some **wire**). They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3)**: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) is Analog In pins. These pins can read the signal from an analog sensor (like a **temperature sensor**) and convert it into a digital value that we can read.
- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

**Reset Button**

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

**Power LED Indicator**

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

**TX RX LEDs**

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for **serial communication**. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

**Main IC**

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

**Voltage Regulator**

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

In this project 'Arduino UNO' selected based on its low cost, cross-platform, simple and clear programmimg. Although open source and extensible software and hardware but it was chosen because of easily available.

### 2.2.2 Radio Frequency Identification (RFID)

Radio-Frequency Identification (RFID) is the use of radio waves to read and capture information stored on a tag attached to an object. A tag can be read from up to several feet away and does not need to be within direct line-of-sight of the reader to be tracked.

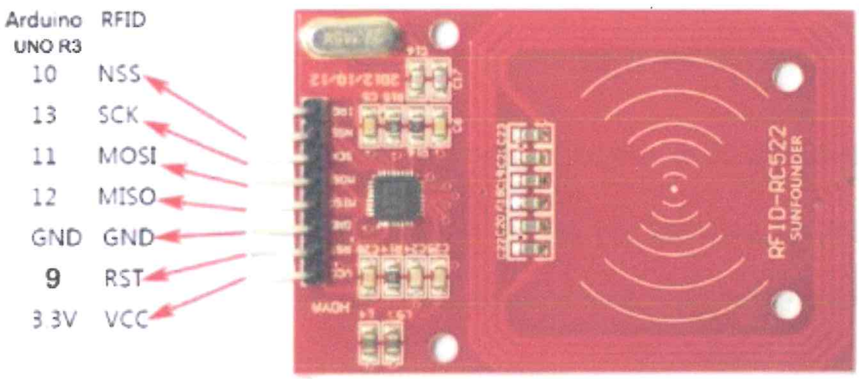


Figure 2.4 : Example of RFID RC522 (e)

| pinout | | |
|---|---|---|
| pin number on the module's header | name | description |
| 1 (labeled "SDA") | SDA | SPI slave-select |
| 2 (labeled "SCK") | SCK | SPI clock |
| 3 (labeled "MOSI") | MOSI | SPI Master Out Slave In "MOSI" |
| 4 (labeled "MISO") | MISO | SPI Master In Slave Out "MISO" |
| 5 (labeled "IRQ") | IRQ | Maskable interrupt pin. Not Connected for most applications |
| 6 (labeled "GND" | $V_{SS}$ | Ground (0 volts) |
| 7 (labeled "RST") | RST | Reset |
| 8 (labeled "3.3V") | $V_{CC}$ | Input Voltage (3.3 volts DC) |

Figure 2.5 : RFID Reader/Writer Module (e)

**How does a RFID system work?**

A RFID system is made up of two parts: a tag or label and a reader. RFID tags or labels are embedded with a transmitter and a receiver. The RFID component on the tags have two parts: a microchip that stores and processes information, and an antenna to receive and transmit a signal. The tag contains the specific serial number for one specific object.

To read the information encoded on a tag, a two-way radio transmitter-receiver called an interrogator or reader emits a signal to the tag using an antenna. The tag responds with the information written in its memory bank. The interrogator will then transmit the read results to an RFID computer program.

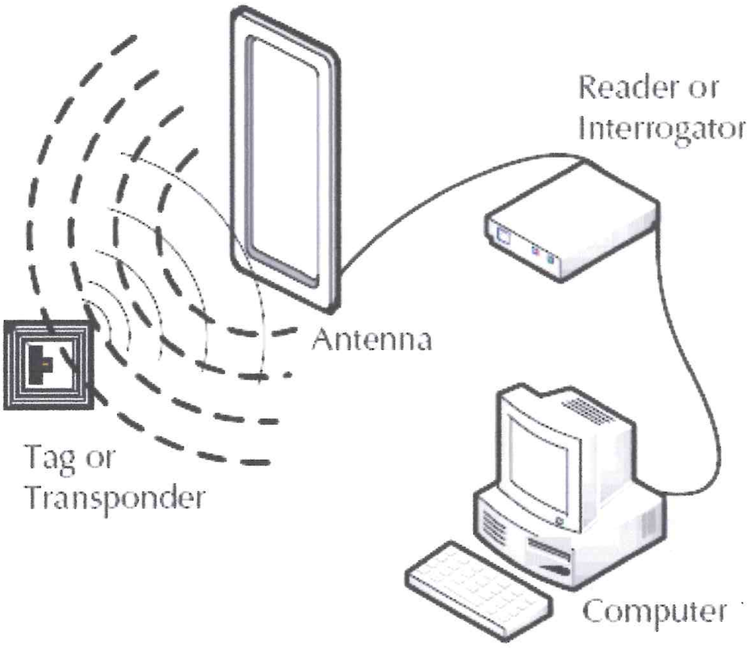


Figure 2.6 : operation of RFID (d)

There are two types of RFID tags: passive and battery powered. A passive RFID tag will use the interrogator's radio wave energy to relay its stored information back to the interrogator. A batter powered RFID tag is embedded with a small battery that powers the relay of information.

In a retail setting, RFID tags may be attached to articles of clothing. When an inventory associate uses a handheld RFID reader to scan a shelf of jeans, the associate is able to differentiate between two pairs of identical jeans based upon the information stored on the RFID tag. Each pair will have its own serial number.
With one pass of the handheld RFID reader, the associate can not only find a specific pair, but they can tell how many of each pair are on the shelf and which pairs need to be replenished. The associate can learn all of this information without having to scan each individual item. Reference (d)

### 2.2.3 Servo Motor

A servomotor, or servo for short, is a self-contained motor, controller, and reduction gear system in a box. Rather than turning continuously, most of them have a limited range of movement, and are normally configured to move a lever from side to side and hold it in position. If this sounds a lot like the kind of stall-motor we've been using to throw turnouts on model railroads for decades, that's because in some ways it is very similar. There are differences, and two of the most important are economy of scale and how they are controlled.

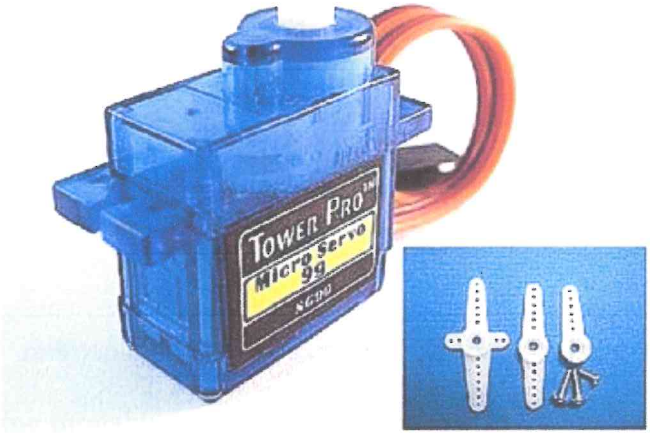


figure 2.7 : example of Servo Motor

The newer micro servers are particularly interesting, because they are very compact compared to current motor-based turnout controls, and are thus suitable for more space-constrained environments. And yet they have sufficient power for throwing typical turnouts, as well as being applied for other purposes such as crossing gates (or semaphore signals for that matter). They are also potentially less expensive (if you are willing to work on your own control system; commercial systems are still somewhat expensive).

They do, however, require specialized controllers. You can't just wire a switch up to one to make it work. A simple analog timer circuit can be used, but in recent years with the rise of inexpensive embedded-processing computers, like the Arduino, new options for control have opened up. The Arduino software comes with a built-in servo control library, and there are external shields and circuit boards for powering servos while controlling them from the Arduino.

A servo is not a stall motor though, and treating one as if it is can destroy it. Like any device they have their strengths and weaknesses, and to best use them, these need to be understood.

Uses in model railroading include throwing turnouts, raising and lowering crossing gates, positioning turntables, creating animated scenes by operating things like doors or moving spouts on steam railroad water tanks, etc. Turnout control tends to be the most common application, due to potential cost savings and because it is relatively easy to do.

Most servos have a limited range of motion. While 180° (+/- 90° from a center point) is commonplace, many have a shorter range (90° is fairly common, and many will do 180° with specialized controllers and 90° with normal ones). There are some than can rotate continuously (such as for operating wheels on a robot), but these are designed somewhat differently from normal ones. It is also possible to modify many limited-range servos to be continuously-rotating ones, although some trade-offs have to be accepted when doing this since the servo isn't likely to have been designed for that purpose.

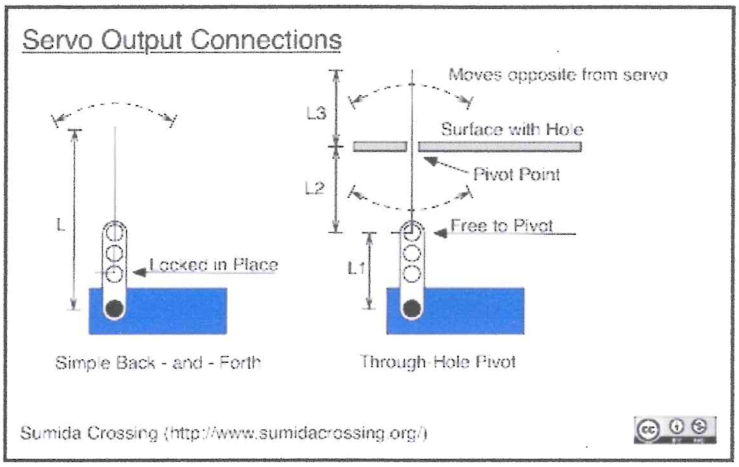


Figure 2.8 : motion of Servo Motor SG90 (b)

Servos almost always operate on 5 volt power, since this is a good fit for modern control systems using digital circuitry. They are rated for a range from about 3 to 6 volts, and many will work on typical 7.2 volt L-Ion batteries, even if technically overloaded. The three wires are + (usually +5V), ground, and a control line. The use of 5V power means they need a special power supply (you can't wire them up to the 12 or more volts used for typical railroad accessory power). Normally the ground wire will be black or brown, and +5V wire will be red, and the control wire will be orange, white or blue (but other colors may be used, consult the documentation for the specific servo if in doubt).

Servos do not draw continuous power, but instead use a series of short pulses 1 - 2 milliseconds (ms) in length spaced about 20 ms apart. This means that their average power draw is considerably less than their peak power draw. And the peak current for even small servos can approach (or exceed) 1 Amp.

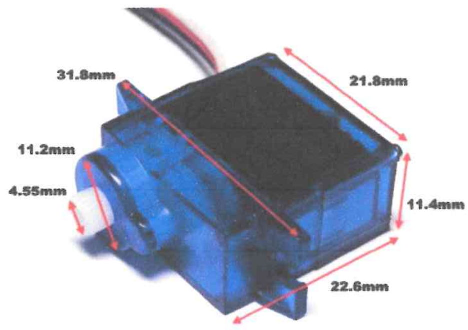


Figure 2.9 : size of Servo Motor SG90 (b)

A servo at these limits will also tend to be more noisy than one in motion, because it is continuously stopping and restarting to push against full load, causing the internal gears to engage and disengage dozens to hundreds of times per second.

Servos move very quickly, going their full side-to-side range as quickly as 0.3 seconds, although the controller could be designed to slow them down to a more realistic speed for railroad use. This is normally given as a rating for motion through 60°, so one that can move 180° in 0.3 seconds would be rated as “0.1s/60°”. Reference (b)

### 2.2.4 BUZZER

A **buzzer** or **beeper** is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

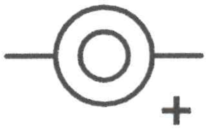


Figure 2.10 : symbol of Buzzer

# CHAPTER 3

## METODOLOGY

### 3.1 INTRODUCTION

Methodology is the ways or the means used to carry out the project in detail . These steps are essential in carrying out this project to ensure the project debuted prepared at a predetermined time . In addition, also , there are ways to test circuits pairs ready .

In a project to produce something , a few steps that need to be passed before a project pertaining ready . These steps need to be done with great precision in order to produce something -quality projects and quality . In producing this project , there are some measures had been taken .

## 3.2 SELECTION OF TITLE

Selection of topics is the very first step was done before the start of the work associated with the project. The project title should be sought as appropriate to the level of Diploma is a final project during the course of study in the Diploma in Electronic Engineering (Computer).

In addition, the selection of appropriate projects to help power their creative and innovative thinking as well as it symbolizes the level of consciousness of a person and how high the level of individual knowledge in aspects involving the use of electrical and electronic equipment. After the project is selected, the title to the project should be selected based on its ability to attract others to know more about the project closely. Title capable of attracting the attention of others symbolizing the beginning of the project status.

After the topics are selected, steps through again is to select circuits associated with the project to be made. In addition, identify the components involved with the circuit should be done properly so that it is readily available and does not pose a huge problem to get it. This is because the components that are difficult to be found will have an impact on the projects to be made because it would probably take a long time to get it.

## 3.3 FLOW CHART