Multifunctional Bike Assist System For Cyclist

BY

Siew Kar Hoe

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY

COMPUTER ENGINEERING
(HONS)

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2017

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____

____

____

Academic Session: ____________

I ____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

________________________	________________________
(Author's signature)	(Supervisor's signature)
Address:	
_________________________	
_________________________	________________________
_________________________	Supervisor's name
Date: ____________________	**Date**: ___________________

Multifunctional Bike Assist System For Cyclist

BY

Siew Kar Hoe

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY

COMPUTER ENGINEERING
(HONS)

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2017

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Multifunctional Bike System For Cyclist"** 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

Appreciation for help and thank for:

Mr. Teoh Shen Khang

Mr. Teoh Shen Khang is my supervisor of this project. He is very nice and able to spend time to teach and supervise me for this project. His kindness and a lot of advices help me to manage and solve the problem that I had faced.

Family
My family, their financial support, mentally support and love being through with me along the way to finish this project. I glad to did this project and thank for their unconditional support and love.

Abstract

This project is to build a multifunctional bike assist system for cyclist by using Arduino based microcontroller. The aim of this project is to assist the cyclist to reduce their risk when they are on the road by using this bike assist system. This bike assist system will turn on the left turn or right turn or brake led light when cyclist has press the left, right or break button. This function helps the cyclist on the road to inform other road user such as road passer and motorcyclist that they want to turn left or turn right or they will be brake, so that the other road user will be aware of the cyclist's next move and take precautions before or after the cyclist. Furthermore, the other road user will prioritize the cyclist on the road such as they will wait for the cyclist to turn first into the road so that they can avoid unnecessary accident. This system has a speedometer to notice the cyclist about their speed at the moment. Furthermore, all of the cyclist's speed data and gps location will send to a database server by using the Arduino Uno platform. All of the data that achieved will send to the server so that cyclist can access into the web page and get the data to refer about their cycling performance.. Nevertheless, the hardware of this system will be in waterproof condition and low power consumption.

Table Contents

List of Figures

List of Tables

List of Abbreviations

IoT	Internet of thing
GPS	Global Positioning System
LCD	liquid crystal display
PCB	Printed Circuit Boards
V_{CC}	plus collector supply line voltage in a common NPN circuit
EEPROM	***e***lectrically ***e***rasable ***p***rogrammable ***r***ead-***o***nly ***m***emory

Chapter 1: Introduction

1.1 Problem Statement

In this modern era, most people will choose to drive motorcar or motorbike rather than cycling although the destination that they want to go is short. In addition, less people choose cycling as sport. One of the reasons is the motorcar and motorbike had good system to assist them, for example, auto-motor systems, brake systems, light systems and so on. In other hands, bike didn't have any systems to assist cyclist, it just have bicycle gears to assist them and sometimes it's dangerous for cyclist to cycle on the road. Here were some problems that cyclist faced, there were bike didn't have any signal light or break light, bike can't give information about speed of cyclist. Besides, bike can't send location that it passed through and bike can't record and send data to database.

1.2 Background and Motivation

Nowadays, broadband internet become more widely available, thus the cost of connecting is decreasing and more devices are being created with Wi-Fi capabilities and sensors built into them. In the others hand, all of these things are creating a ‘perfect storm’ for Internet of Things (IoT). The IoT is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network. It revolves around increase machine-to-machine communication. The machine-to-machine (M2M) data that is generated has a wide range of uses, but commonly seen as a way to determine the health and status of things such as inanimate or living.

Besides, it’s comes together with the connection of sensors and machines. In other words, IoT creates the real value at the intersection of gathering data and leveraging it. Cloud-based applications are the key to using leveraged data. The IoT doesn’t function without cloud-based applications to interpret and transmit the data coming from all these sensors. The cloud is what enables the apps to go to work for you anytime and anyway. The new rule for the future is going to be “Anything that can be connected, will be connected.” Our life style will be radically transformed and become “smarter”. Smart living aims our life-style incorporating with a communications network that connects the key electrical appliances and service, and allows them to be remotely controlled, monitored or accessed (UK Department of Trade and Industry, 2003).

In addition, multifunctional bike assist system for cyclist is one of the smart living, it’s leveraging IoT Technology to make cycling safer and more efficient. Nowadays, cycling carries important characteristic in both healthy life-style and in transportation. Once a cyclist cycling on the road, they need a multifunctional bike assist system to bestead them in many situations. This multifunctional bike assist system stand in give information of signal and break light of the cyclist to other road user that the cyclist will be turn or break after the moment, so that they can prevent to hit the cyclist in the situation with no signal or break light. This will avoid both cyclist and road users in unnecessary accident.

Furthermore, this multifunctional bike assist system can help improve the performance of the cyclist by achieve the cycling performance, for example, speed at a current location, at which location cyclist's speed will decrease due to the road gradient condition that compute, pedalling frequency, uphill and downhill power and so on. After that it will upload onto a cloud platform and the cyclist can retrieve their data on the cloud, so that they can look at their performance and can plan to have a good strategies training to improve their performance. This multifunctional bike assist system assist cyclist no matter they are professional cyclist or non-professional cyclist, this system can make their cycling journey to be more safety and improve their performance more efficient.

In this modern era, while cyclist cycling their normal bike on the road, they will face some problem that may be dangerous for them and thus will lead to accident if not handle correctly. When cyclists want to turn left or right or brake on the road, there will be some car or motorbike behind them. For some of the cyclist that is new learner of cycling they not suitable to give signal by hand like motorcyclist, thus it dangerous for them too if beside them have motorbike or car wants to cross-over them and they didn't keep one's eyes open. Thus, Arduino based multifunctional bike assist system for cyclist may solve the problems and with more functions inside this system with the features below as stated here, the multifunctional bike assist system will turn on or off for the signal LED light or break LED light that build on the bike through by the cyclist press the corresponding button. Besides, the multifunctional bike assist system will able to send location, speed, uphill or downhill speed in different road gradient to a cloud platform. In addition, the multifunctional bike assist system will be low power consumption and installed in a waterproofing body or product.

1.3 Project Objectives

In the market now, it had similar bike assist systems for cyclist, but the system still can be improve and upgrade for cyclist, for instance, market have some red or orange LED light that can be easy install in or dismantle out from the bike. These LED light is useful at night, but it will always been light up, it just can let road user to recognize there is a cyclist there. These LED lights can’t give actual information of cyclist to other road user. In other hands, there have some bike assist system that can turn on or off the LED light by the cyclist, but the system still can be upgrade.

Therefore, here are the objectives of my project to improve and upgrade the system. First, this project will have signal and break LED lights for cyclist that they can control to turn on and off it. Secondly, it will have speedometer for cyclist to check their speed. Thirdly, it will send location while cyclist is cycling onto the database server.

Furthermore, it will have low power consumption and to have this project with waterproof condition.

1.4 Highlight of what have been achieved

1) Push button to display signal light
2) Slide potentiometer to display signal light
3) Speed and distance show in LCD display
4) GPS location trace
5) Data store into database
6) Retrieve data by web page

Chapter 2: Literature Review

2.1 Literature Review and Discussion of strength and weakness of existing product

Based on the research, there have some similar product related to this current project. One of the products known as ‘Smart e-bike monitoring system: real-time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles’ developed by Chris Kiefer1 & Frauke Behrendt in year 2015. By comparing this product, there have some strength and weakness that can be found in this product.

First, this product is a smart e-bike monitoring system (SEMS) which mean it is a show stage for the real time of usage data from e-bike also called electricity bike. It is open source and open hardware with autonomous which mean run out of the bike battery. Besides, it also in different fleet sizes and sensors can be added too, so it can do further research, upgrade and development. The system monitors GPS location and rider control in data and other custom sensor input for real time. The SEMS data did an online interface for data analysis, for riders to monitor their personal ride data and for share in life and social media. The basic system can be duplicated by other researchers and can be further upgrade with modules to fix several serious issues in e-bike system. The strength of this product is they ensure for the gathering data in comparative data and lowers down the development costs for monitoring of e-bike fleets. This system was implemented with replicable therefore it allow for the scalability and modularity of the SEMS. They build the android phone with the hardware interface and power board so that the power consumption issue of the android phone can be solved. There have some weakness like power was a fundamental issue. The e-bike use lithium-ion batteries to run, if too much charge is drawn these will be permanently damage. In addition, as not to significantly affect the range of the e-bike SEM should not draw too much power. (Chris Kiefer1 & Frauke Behrendt 2015)

Second, there has another product that similar with this project which was ‘bicycle light signal’ developed by Von Lighthill in year 1994. It’s relates to lights

used on bicycles. Besides, for the light especially adapted for visibly signaling the presence of
the bicycle to drivers of motor vehicles, other bicycle riders, and pedestrians. It's a new and improved high visibility light signal apparatus for use on a bicycle includes a strobe light assembly which includes a source of electric power and a strobe lamp/driver assembly. A support assembly supports the strobe lamp/driver assembly on the bicycle. The support assembly includes a first end and a second end. There have some strength like the lamp cover includes a first portion and a second portion. The first portion includes a partial reflector, and the second portion includes a transparent portion. The partial reflector portion permits a portion of light landing on the partial reflector to be reflected and a portion of the light landing on the partial reflector be transmitted through the partial reflector. The partial reflector may include a partially silvered surface. In this respect, the lamp cover emits two lighted state signals. The first light signal has a first level of brightness, and a second light signal has a second level of brightness. (Von Lighthill 1994)

Third, there was another product which similar with this project 'turn signal and horn assembly for a bicycle' developed by Izzo in year 1994. This present invention generally relates to a bicycle mounted combination turn-signal and horn apparatus and, more particularly but not by way of limitation, to such an apparatus having telescoping arms or folding arms, on the ends of which are mounted signal lights and lens. The strength of this product are they provide a bicycle signaling device capable of signaling others of their turning intentions which is more easily seen by motorists and pedestrian and provide directional indication, caution indication, stopping indication, and a horn warning to motorists and pedestrians which is relatively compact, easily detachable, and portable. (Izzo 1994)

Besides that, there has a similar product 'signal helmet' which similar with this project developed by Adam Gouda in year 1990. As may be seen on any roadway eye level rear signals are more easily discernible to a motorist following another vehicle. It has been statistically determined that the eye level brake signals have a tendency to avoid rear end collision. On a motorcycle the advantage of such eye level signal is not easily acquired on the bike itself. The helmet of the cyclist provides a convenient way of raising the signals to the level desired. The invention here describes a smooth surface signal helmet with at least one lens for signal lights flush with the helmet and presenting a singular contour between the lens and the surface of the shell of the helmet. The lights can have sources of radiation comprising a choice of incandescent lamps, halogen lamps, and LEDs. One embodiment envisages a radio receiving module adapted to receive telemetered signal information from the motorcycle proper. (Adam Gouda 1990)

In addition, there has a similar product with this project which is ‘Bicycle signal system’ developed by Kovarix Zarko in year 1974. This product invent a stop light system for bicycle brakes embodying a battery pack and an electrical signal lamp appropriately mounted on the bicycle frame and having a contact element positioned for engagement with one of the movable brake shoe levers of the bicycle. The contact, when engaged by the brake shoe lever, completes an electrical circuit through the signal lamp, thus illuminating the same. By comparing with this product, there have some strength and weakness that can be found in this product. The strength of this product is it can generate power for itself to use and in low power consumption. This product has weakness such as it only can have an electrical signal lamp for normal light up, it can’t give more information. (Kovarix Zarko 1974)

At last, a better multifunctional bike assist system for cyclist should be based on low power consumption, high functionality, user friendly and waterproofing.

Chapter 3: System Design

3.1 Hardware Integration

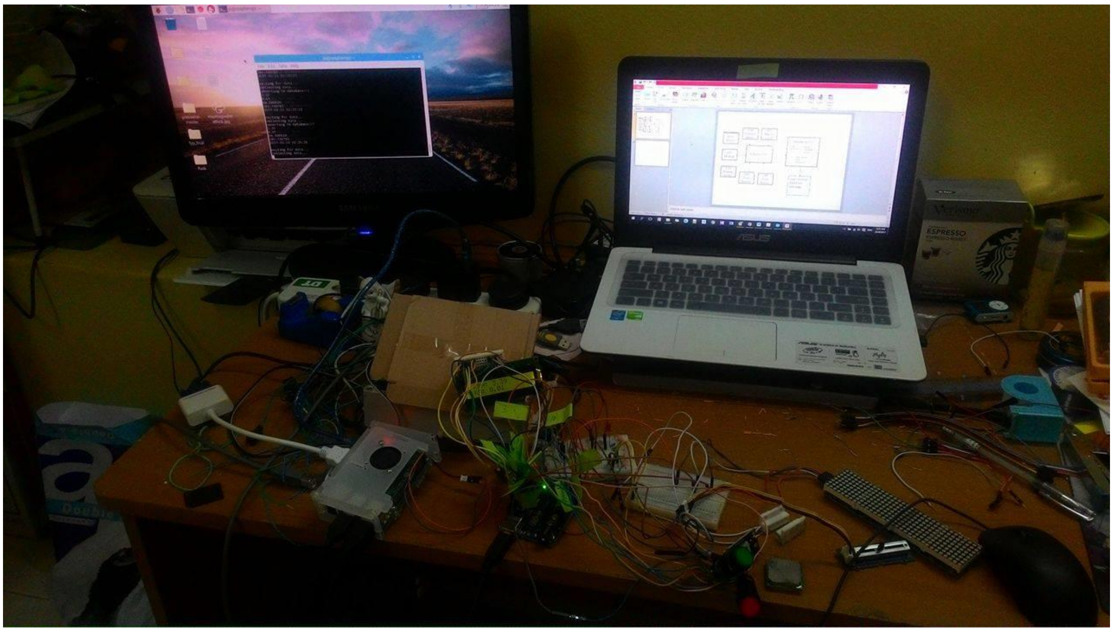


Figure 3-F1 Overview of system

First of all, all of the hardware must be tested well and full function before integrate them together. All components of hardware include Arduino Uno, Raspberry Pi, MAX7219 Dot LED matrix module kit for Arduino and microcontroller, slide potentiometer, door sensor, left and right on/off push button, LCD display screen and GY-NEO6MV2 GPS module for Arduino must be separate to be test their functionality. Furthermore, after all test has been done and no error occur, it can be start to integrate all hardware together in step by step. At beginning, the most important thing in this project is to show left signal and right signal. Therefore, on/off push button integrate with LED matrix module and two LED light is the major part in this project.

I use two digital interrupt pin in Arduino Uno to receive the left and right signal of the button when it push. After receive the signal, Arduino Uno board will turn on one of the LED light and show signal in the LED matrix module. The uses of the LED light is to see which button is being push and is it function well and is its signal transmit to Uno board or whether Uno board received the signal and light up the LED light or not. After all function well, the LED matrix module will then turn on according to its signal that transmit to it, either left signal or right signal.

Besides, due to the LED matrix module signal need to easy to let people get the information, I changed the signal light to be slide right when it change the signal light. Furthermore, not only left and right signal light need to be show, I also let the LED matrix module to show out the brake light by just light up all the LED light in the matrix module. In the other hands, this brake light is the integration between slide potentiometer with the LED matrix module. When the reading of the slide potentiometer is 0 or lower than the value before it store, the Uno board will not display LED matrix module brake light. Otherwise, if the reading of the slide potentiometer is more than the value of it store, it will transmit signal to Uno board and Uno board will turn on LED matrix module to display the brake signal light.

In the other hands, the door sensor uses to calculate the speed of the bicycle. How it calculate? First of all, it read the door sensor signal that transmit when it complete one cycle, Uno board store the current time in milliseconds and update the previous time and increase number of cycle. After that, it calculates out the speed by using formula of diameter of wheel bicycle divide by the time of it use to complete one cycle. Then, uses the answer to multiple 36 will get the kilometer per hour of speed. Furthermore, if want to calculate the distance that a person cycle in kilometer (km), then it need to use the diameter of wheel bicycle multiple with the number of cycle that the person cycle and after that divide by 100000 due to different board different of frequency it generate, therefore different rate will be generate and need to divide by 100000.

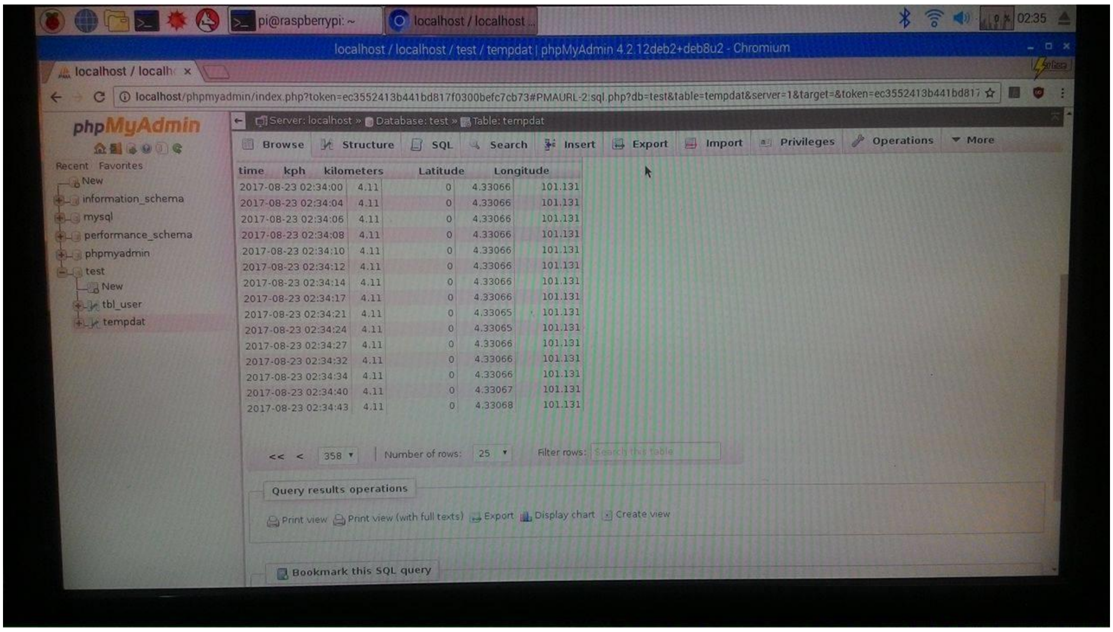
Besides, calculate out the speed and distance is fine to use, but user cannot know and cannot always use serial monitor to show out all results. Therefore, we need the LCD display screen to display out the speed and distance. To communicate between LCD display screen and the door sensor, it just let the LCD display get the value of it and display then on the LCD display screen. If the LCD display screen cannot show up any things, may be its contrast not enough or too high. Slightly go adjust the potentiometer of the LCD display screen till it appears the display.

For the GPS module, it must need a 10k ohm resistor and 4.7k ohm resistor to enhance it to function well to achieve the latitude and longitude of the current position. After manipulate all of these data, Uno board cannot save all of these data into it due to it did not have any sd card or pendrive can communicate with Uno board. Therefore, a serial communication between Arduino Uno with Raspberry Pi 3 is formed. Due to Uno board did not have any WiFi hardware, module or setup therefore, it cannot upload the data via internet or WiFi. A normal AC supply or DC supply like power bank can power up both Arduino Uno and Raspberry pi 3.

3.2 Software Integration

After the grab serial communication in python code between Uno and Pi 3 is successful, the Arduino Uno starts to transfer data to Raspberry Pi 3. Before that, Raspberry Pi 3 need to install MySQL server first before data transmit to Pi 3. After install MySQL server, it needs to create database and table in it so that data can transfer in. After create table, create a correct structure of column also important. Besides, after all complete, the data can be successful transfer from Arduino Uno to Raspberry Pi 3.

Furthermore, install a flask python base web framework in Raspberry Pi is necessary to write a python based web page to retrieve data from MySQL. After all things complete and run well, login into the web page by user. All of the data will be good and safety for store and retrieve. Besides, add port forwarding in the router will be necessary to port forwarding into Raspberry Pi server. In addition, I also install phpmyadmin software to change the structure of the table and managed the data of Raspberry Pi that transfer from the Arduino Uno.



localhost / localhost / test / tempdat | phpMyAdmin 4.2.12deb2+deb8u2 - Chromium
localhost/phpmyadmin/index.php?token=ec3552413b441bd817f0300befc7cb73#PMAURL-2:sql.php?db=test&table=tempdat&server=1&target=&token=ec3552413b441bd817
phpMyAdmin
Server: localhost » Database: test » Table: tempdat
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
More
information_schema
mysql
performance_schema
phpmyadmin
test
tbl_user
tempdat
Query results operations
Print view
Print view (with full texts)
Export
Display chart
Create view
Bookmark this SQL query

time	kph	kilometers	Latitude	Longitude
2017-08-23 02:34:00	4.11	0	4.33066	101.131
2017-08-23 02:34:04	4.11	0	4.33066	101.131
2017-08-23 02:34:06	4.11	0	4.33066	101.131
2017-08-23 02:34:08	4.11	0	4.33066	101.131
2017-08-23 02:34:10	4.11	0	4.33066	101.131
2017-08-23 02:34:12	4.11	0	4.33066	101.131
2017-08-23 02:34:14	4.11	0	4.33066	101.131
2017-08-23 02:34:17	4.11	0	4.33066	101.131
2017-08-23 02:34:21	4.11	0	4.33065	101.131
2017-08-23 02:34:24	4.11	0	4.33065	101.131
2017-08-23 02:34:27	4.11	0	4.33066	101.131
2017-08-23 02:34:32	4.11	0	4.33066	101.131
2017-08-23 02:34:34	4.11	0	4.33066	101.131
2017-08-23 02:34:40	4.11	0	4.33067	101.131
2017-08-23 02:34:43	4.11	0	4.33068	101.131

<< < 358 | Number of rows: 25 Filter rows:

Figure 3-F2 Overview of database

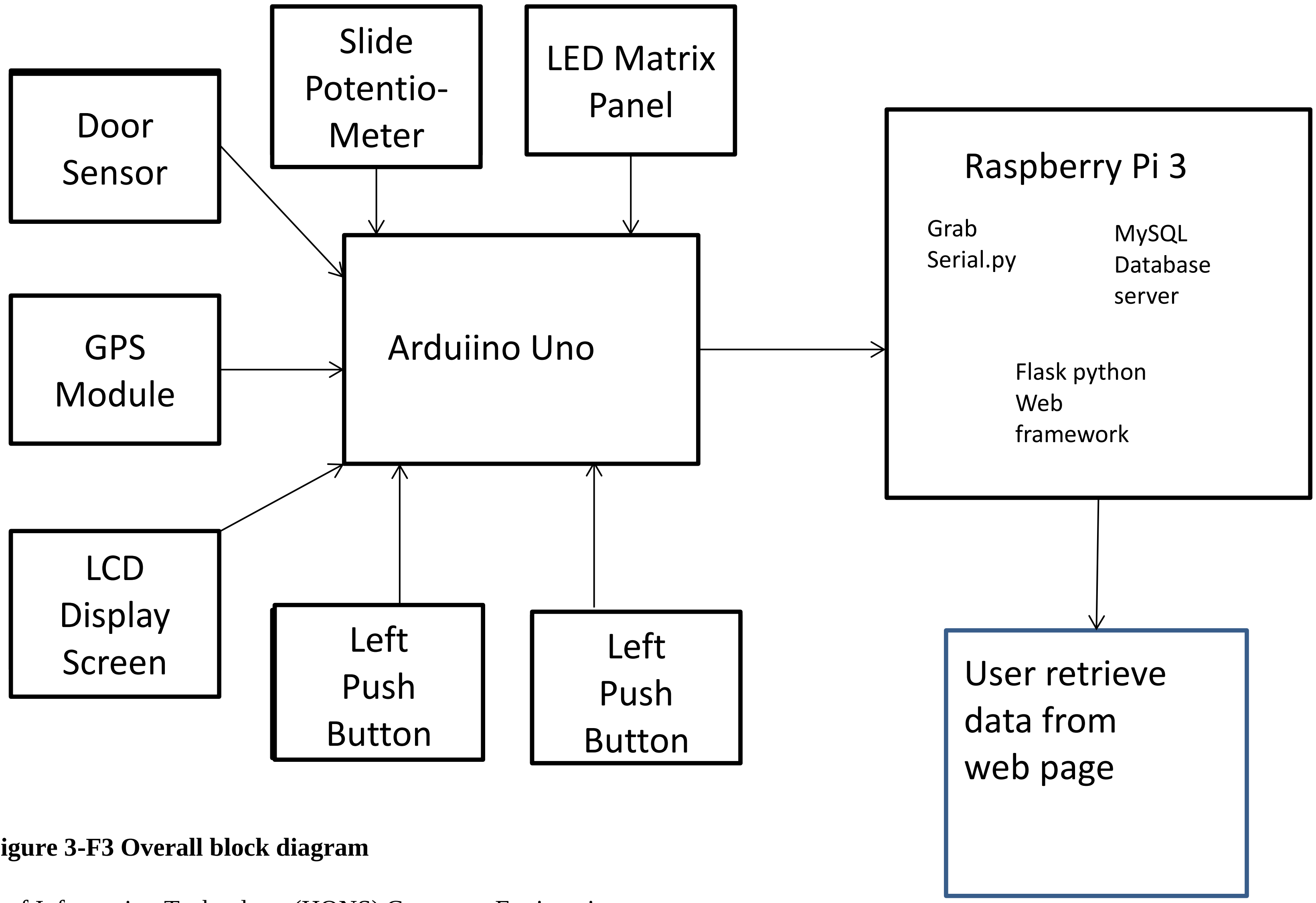


Figure 3-F3 Overall block diagram

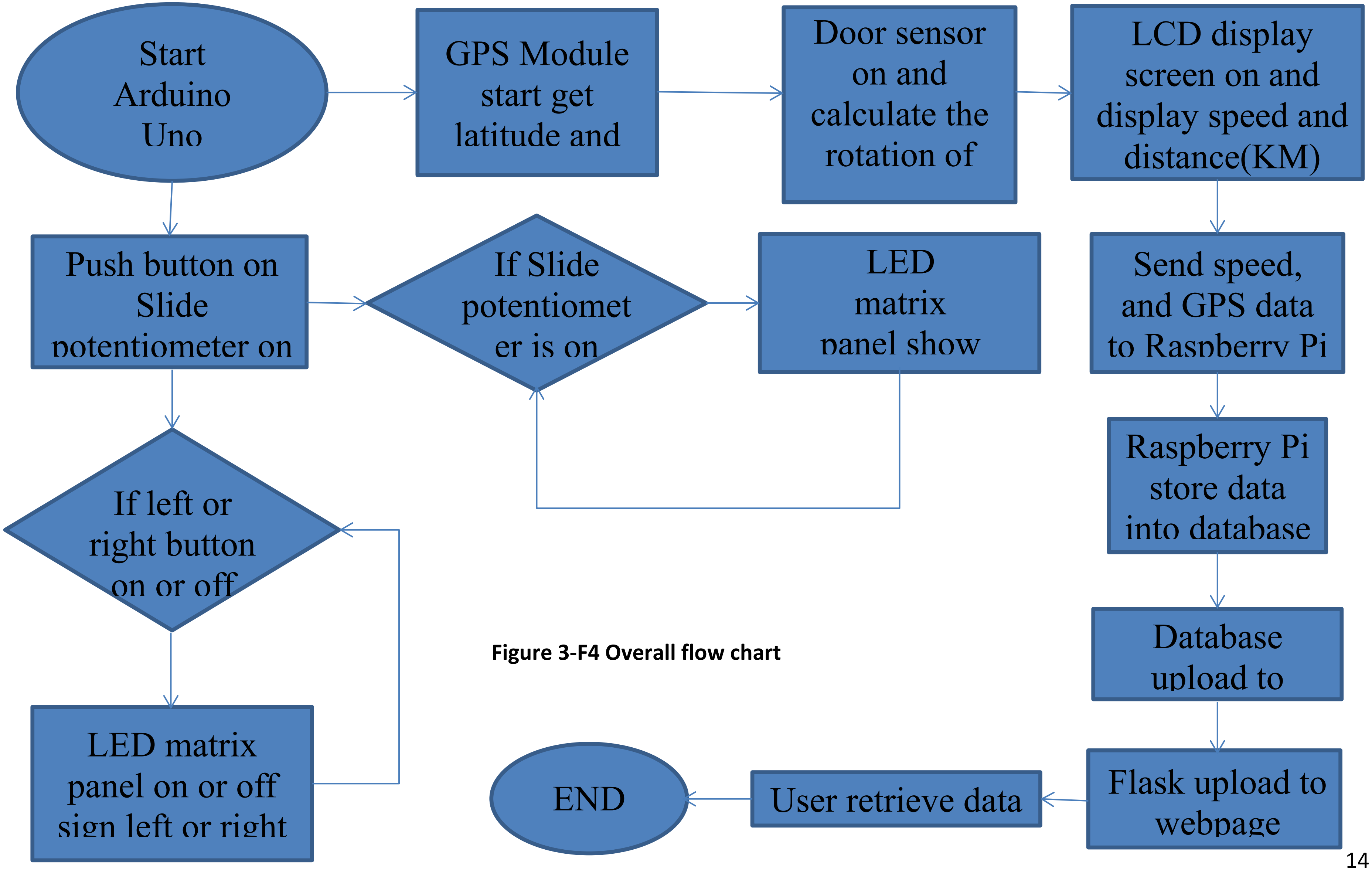

Start Arduino Uno
GPS Module start get latitude and
Door sensor on and calculate the rotation of
LCD display screen on and display speed and distance(KM)
Push button on Slide potentiometer on
If Slide potentiomet er is on
LED matrix panel show
Send speed, and GPS data to Raspberry Pi
If left or right button on or off
Raspberry Pi store data into database
LED matrix panel on or off sign left or right
Database upload to
END
User retrieve data
Flask upload to webpage

Figure 3-F4 Overall flow chart

Chapter 4: Proposed Method/ Approach

4.1 Methodologies and General Work Procedures

Prototyping model

This prototyping model is carried out by the currently known requirements of the project. By developed this prototype, the interaction with prototype can ensure the client get some understanding about the desired system and "actual feel" of the system. Prototyping is an attractive idea for large and complicated system for which there is no existing system or manual process to help determining the requirements.

Prototyping usually are not complete system and may have many of the details are not built in the prototype. The main purpose is to provide a system with overall functionality. There have some advantage of prototyping models, users are actively involved in the development. Besides, since a working model of the system is provided in this methodology, the users can gain a better understanding of the system being developed. In addition, it can be much earlier to detect error and user quick feedback is available leading to a better solutions. Furthermore, functionality that missed can be identified easily.

Besides that, this prototyping model had some disadvantage that will lead to implement and repairing way of building system. If the full system was desgined incomplete or inadequate problem analysis, application may cause application not to be used.

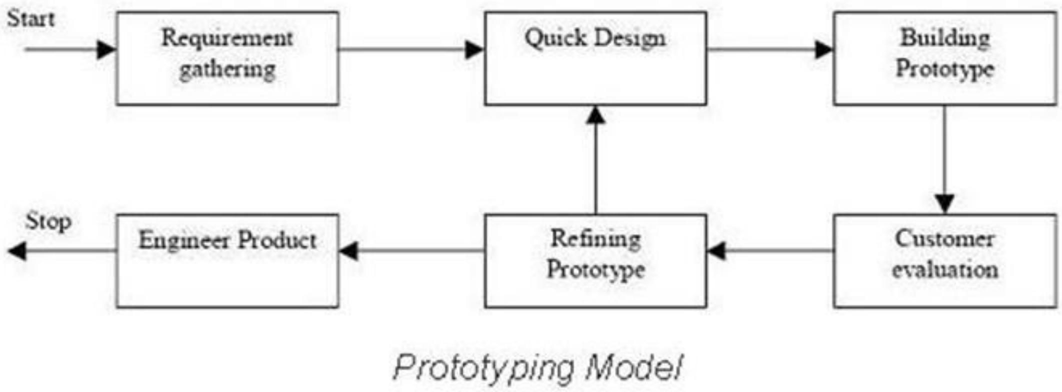


Figure 4-F1 Prototyping model

Chapter 5: Tools

5.1 Hardware Development

5.1.1 MAX7219 Dot Led Matrix Module Kit For Arduino & MicroController

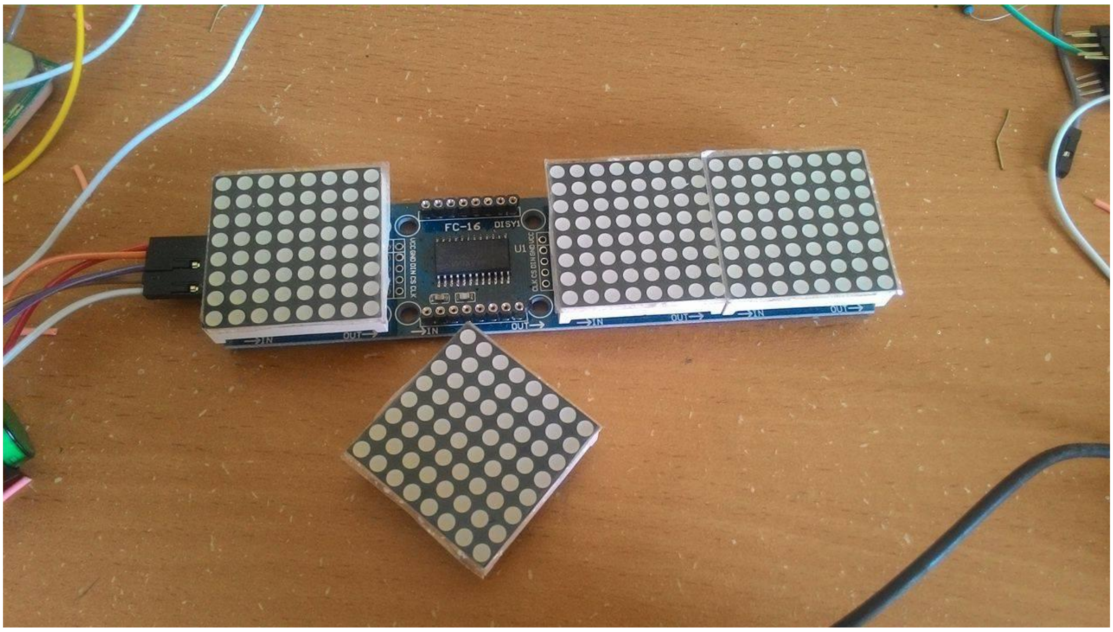


Figure 5-1-F1 LED matrix module front

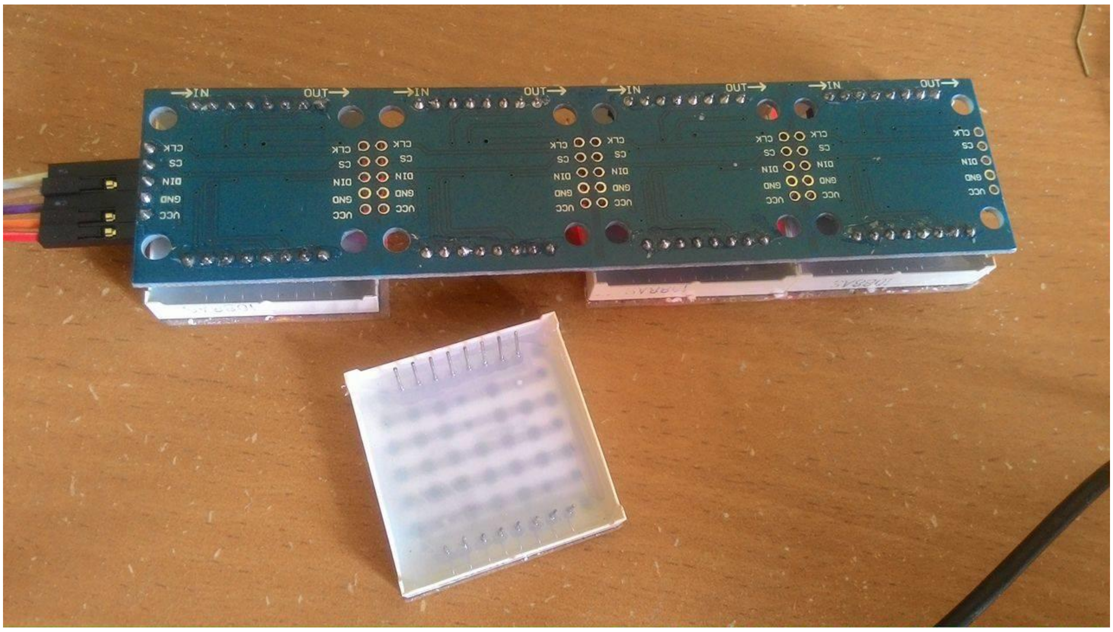


Figure 5-1-F2 LED matrix module back

The MAX7219/MAX7221 are compact, serial input/output common-cathode display drivers that interface microprocessors (µPs) to 7-segment numeric LED displays of up to 8 digits, bar-graph displays, or 64 individual LEDs. Included on-chip are a BCD code-B decoder, multiplex scan circuitry, segment and digit drivers, and an 8x8 static RAM that stores each digit. Only one external resistor is required to set the segment current for all LEDs.

The MAX7221 is compatible with SPI™, QSPI™, and MICROWIRE™, and has slewrate-limited segment drivers to reduce EMI. A convenient 4-wire serial interface connects to all common µPs. Individual digits may be addressed and updated without rewriting the entire display. The MAX7219/MAX7221 also allow the user to select code- B decoding or no-decode for each digit. The devices include a 150µA low-power shutdown mode, analog and digital brightness control, a scanlimit register that allows the user to display from 1 to 8 digits, and a test mode that forces all LEDs on.

It is a 4 in 1 display kit that each single module can drive a 8x8 dot matrix common cathode. This led matrix panel need 5V as its operating voltage. It is a module with input and output interfaces thus support for cascading multiple modules. It has 5 pins, the first one is VCC that connect 5V, second is GND that connect to ground, third is DIN that connect to digital input pin of 13 in Arduino Uno, fourth is CS that also connect to digital input pin of 11 in Arduino Uno and for the last is CLK that also connect to digital input pin of 12 in Arduino Uno. Meanwhile, it also needs software part to test this hardware. I used Arduino Software (IDE) to test this MAX7219 Dot Led Matrix Module Kit For Arduino & MicroController. There have some important code that necessary to run in the software IDE so that both Arduino and LED matrix module kit can communicate well.

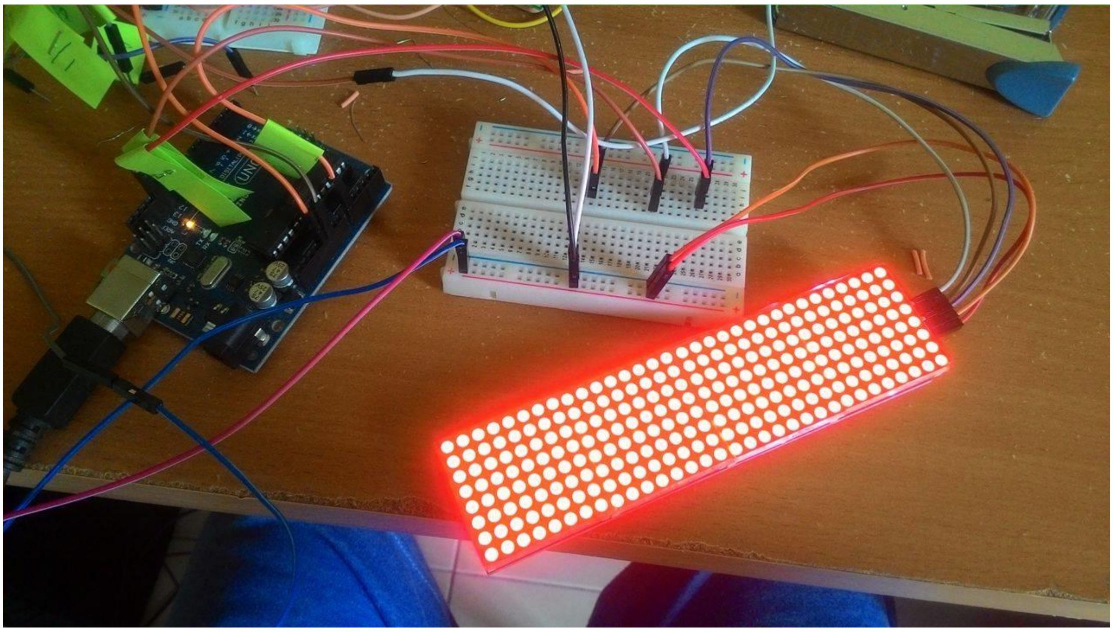


Figure 5-1-F3 LED matrix module on

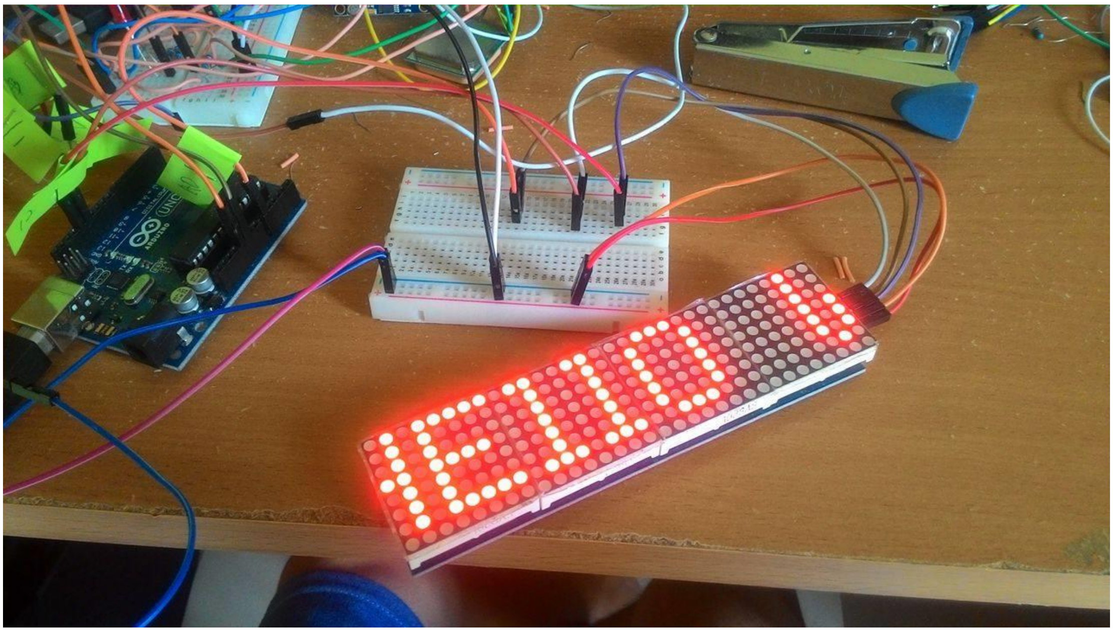


Figure 5-1-F4 LED matrix module display

Coding:

```
#include <pgmspace.h>
#include <LedControlMS.h>

const int numDevices = 4;      // number of MAX7219s used
const long scrollDelay = 75;   // adjust scrolling speed
int lastButtonState = HIGH;

unsigned long bufferLong [14] = {0};

LedControl lc=LedControl(13,12,11,numDevices);

const unsigned char scrollText[] PROGMEM ={
   "HEllO WORLD\0"};

void setup(){

   for (int x=0; x<numDevices; x++){
      lc.shutdown(x,false);       //The MAX72XX is in power-saving mode on startup
      lc.setIntensity(x,8);       // Set the brightness to default value
      lc.clearDisplay(x);         // and clear the display
   }
}

void loop(){

    scrollMessage(scrollText);
    scrollFont();
}
```

Explanation:

It is necessary to include both header file of pgmspace.h and LedControlMS.h that provide by the developer in library of Arduino or can get it on github website.

Then need to define how many MAX7219 that are using. Most important is assign the correct pin 13, 12 and 11 that CLK, CS, DIS in the panel which connecting to Arduino UNO.

5.2 Model LA16Y-11 no latch with 3A/250VAC

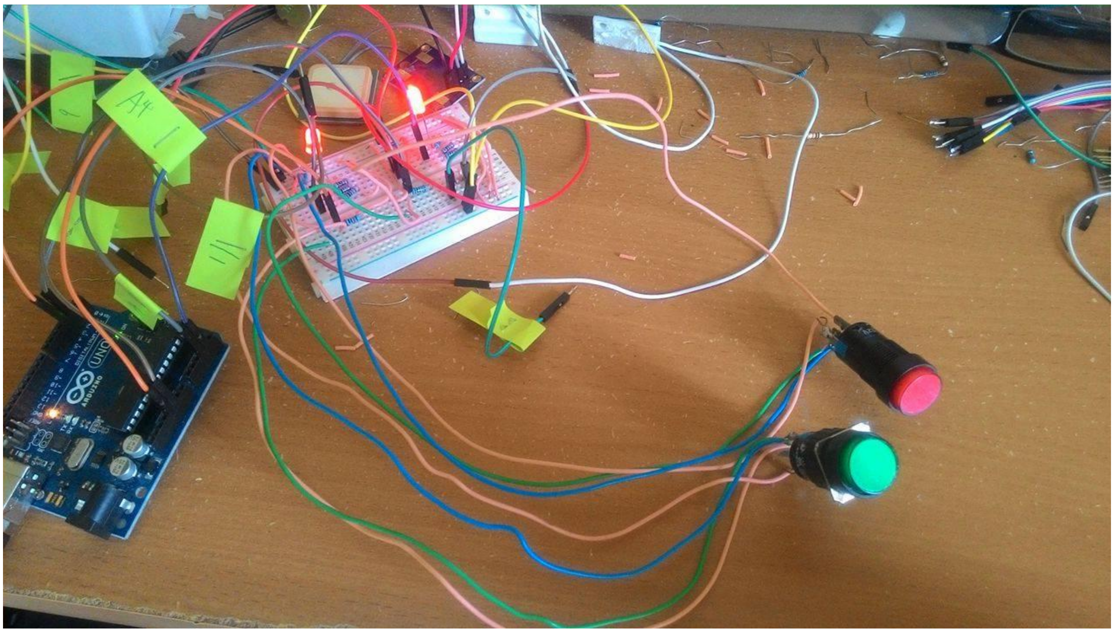


Figure 5-2-F1 Push button on

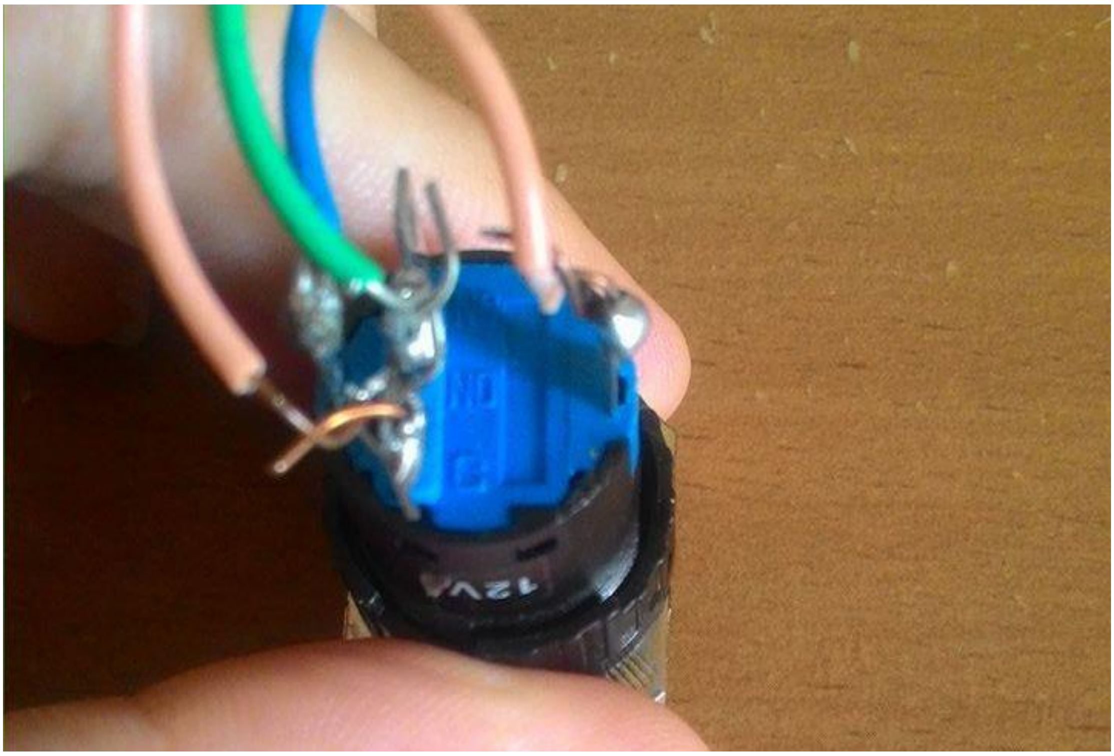


Figure 5-2-F2 Push button back

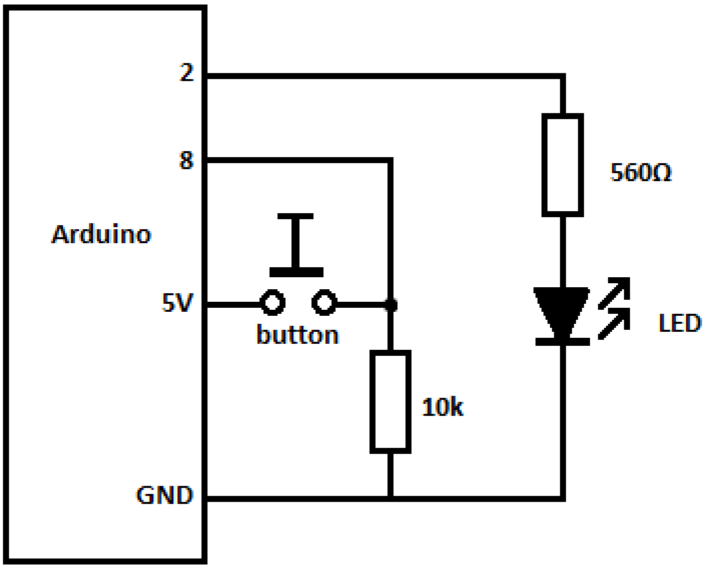


Figure 5-2-F3 Push button with Arduino

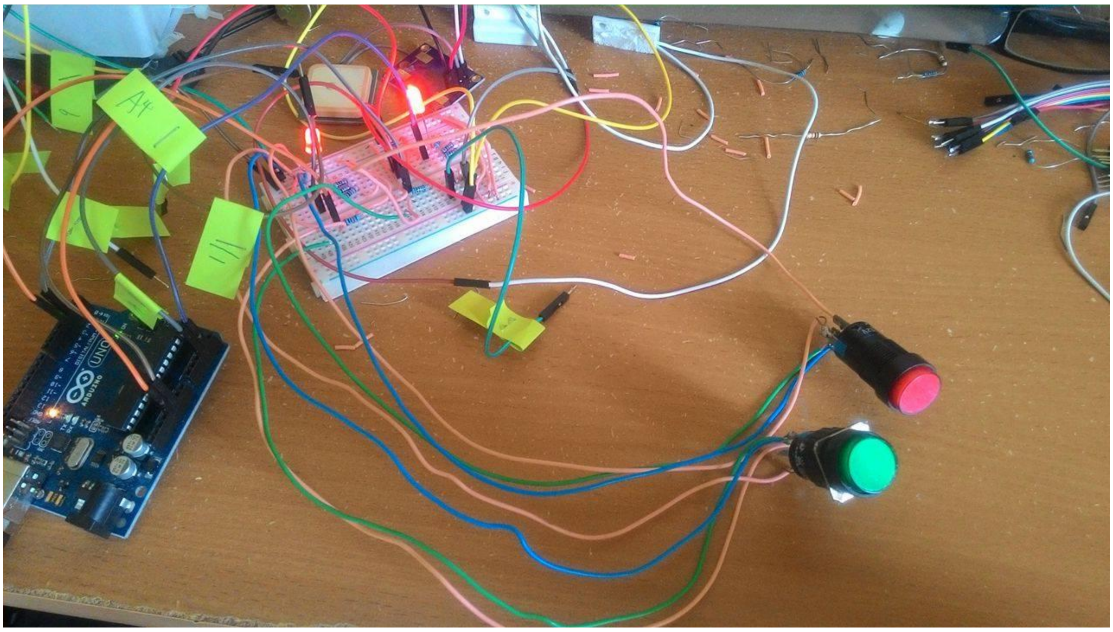


Figure 5-2-F4 Push button with Arduino circuit

In this project will need two of this button, one is use to turn on the led matrix panel to show out left side signal and second is to show out right side signal when second button is push and for the signal light in led matrix panel will only turn off when the second time of the button is push. It is a model LA16Y-11 no latch with 3A/250VAC current and rated voltage. Its contact resistance is less than or equal to 50 mohm and insulation resistance is more than or equal to 100 Mohm. When connect these two button in the circuit they need 10 kohm resistor to resist current to high that will spoil the button. Therefore, I used 10 kohm resistor to connect it with Arduino UNO so that it would not burn or damage the button. In this button, it contains two circuits. First circuit is to turn on/off of the light of button. Second is the circuit that generates the button signal and transmit ‘1’ out to receiver, so that receiver can take any action of they want.

Besides that, 2 red LED light had been used to test the signal from button. Due to not damage the LED light, a 560ohm of resistor is necessary for complete this circuit. Furthermore, due to lab of UTAR did not enough 560ohm resistor, I used three 220 ohm to alternate with 560 ohm resistors. In the other hands, for the software part, I using interrupt option in Arduino UNO to bring out the signal generate from on/off button and transmit from it to receive by the digital interrupt pin 2 and 3 that had been fixed by Arduino UNO.

Coding:

```
int ledPin = A3;
int ledPin2 = A4;
int buttonPin = 3;
int buttonPin2 = 2;
int lastButtonState = HIGH;
int lastButtonState2 = HIGH;
int ledState = HIGH;
int ledState2 = HIGH;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
   pinMode(ledPin2, OUTPUT);
  pinMode(buttonPin2, INPUT);
}

void loop()
{
  // read from the button pin
  int buttonState = digitalRead(buttonPin);
   int buttonState2 = digitalRead(buttonPin2);
  // if the button is not in the same state as the last reading
  if (buttonState==LOW && buttonState!=lastButtonState)
  {
    // change the LED state
    if (ledState==HIGH)
    {
      ledState = LOW;
    }
    else
    {
      ledState = HIGH;
    }
  }
   if (buttonState2==LOW && buttonState2!=lastButtonState2)
  {
    // change the LED state
     if (ledState2==HIGH)
    {
      ledState2 = LOW;
    }
    else
    {
      ledState2 = HIGH;
    }
  }
  digitalWrite(ledPin, ledState);
  digitalWrite(ledPin2, ledState2);
  // store the current button state
  lastButtonState = buttonState;
  lastButtonState2 = buttonState2;
  // add a delay to avoid multiple presses being registered
  delay(20);
}
```

Explanation:

At the beginning, I assigned analog output for LED light and digital pin 2 and 3 for the input of the interrupt from button. Then, initialize state for the button and the LED light so that it store the ‘HIGH’ or ‘LOW’ value when button is being press and the value of LED light is either on or off . Its theory is when the button is press and button state not equal to before button state,, then turn on LED light, else do nothing. Due to button press and release is one cycle, it need to add delay to avoid multiple press. Furthermore, after press is on LED light then it need to press one more time to close the LED light.

5.3 Arduino Uno

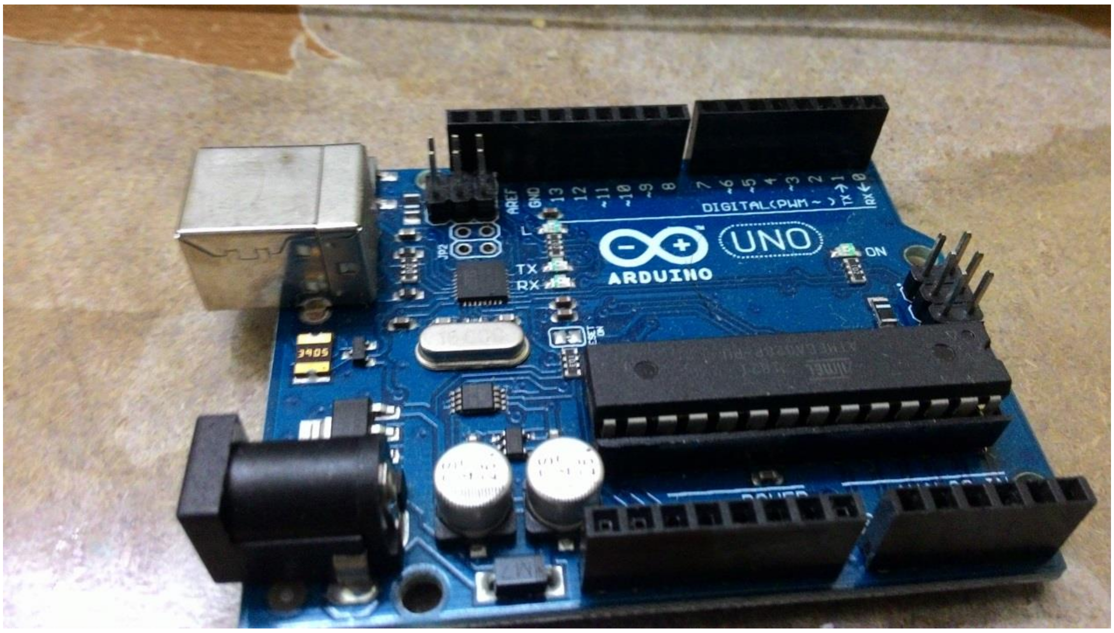


Figure 5-3-F1 Arduino Uno

The Arduino uno is the best board to be start with electronic and coding. It is a microcontroller board based on the ATmega328P. It used a USB connection to connect with compatible computer. Besides, it has a power jack, a reset button, a 16 MHz quartz crystal, 14 digital inputs which 6 can be by PWM as outputs and 6 analog inputs. Arduino Uno can be easily start with power it with AC-to DC adapter or battery or connect it to a computer with a USB cable. This UNO can be replace the chip for a few dollars and start over again, therefore, it can be tinker to the Uno without worring too much. The Uno board is used with Arduino Software (IDE).

ARDUINO MICROCONTROLLER	
Microcontroller	ATmega328
Architecture	AVR

Operating Voltage	5 V
Flash memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	6
EEPROM	1 KB
DC Current per I/O Pins	40 mA on I/O Pins; 50 mA on 3,3 V Pin
GENERAL	
Input Voltage	7-12 V
Digital I/O Pins	20 (of wich 6 provide PWM output)
PWM Output	6
PCB Size	53.4 x 68.6 mm
Weight	25 g
Product Code	A000066 (TH); A000073 (SMD)

Technical Specification

The power source of Arduino Uno is selected automatically either via the USB connection or with an external power supply. For the power supply can be battery or AC to DC adapter (wall-wart) in external non-USB power. The board can support from 6 up to 20 volts in external supply to operate itself. If supply not enough, the board may be unstable.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:
- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

5.4 Door sensor

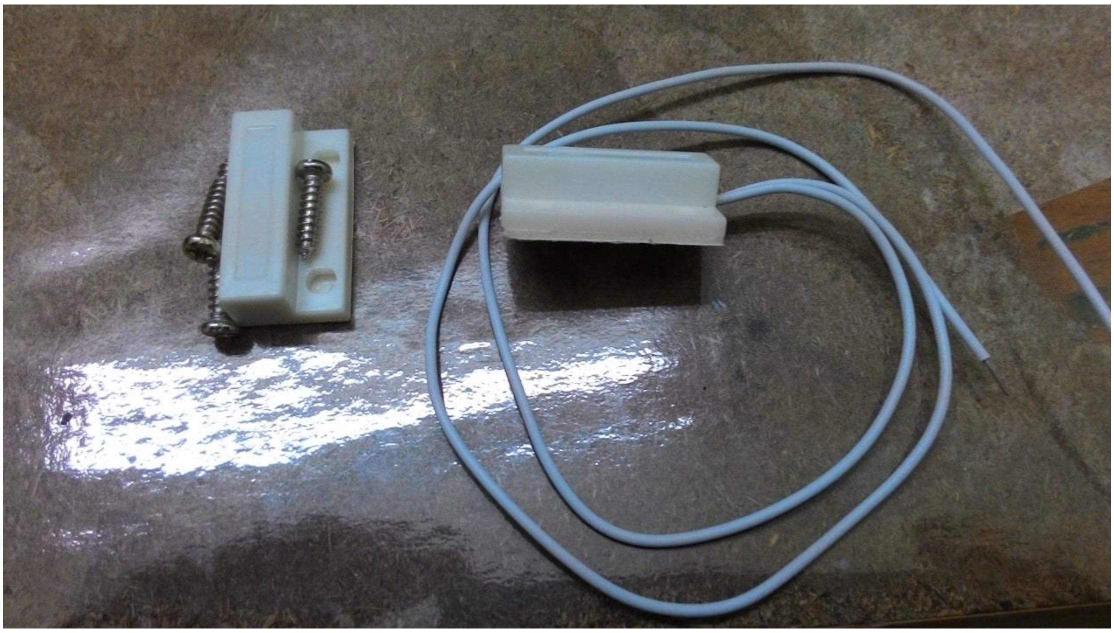


Figure 5-4-F1 Door sensor

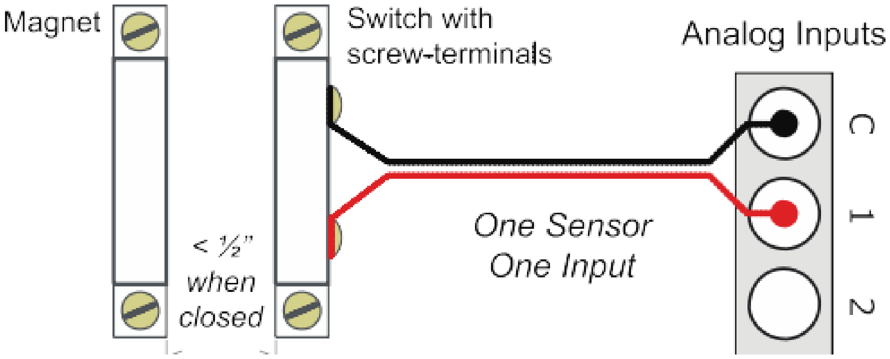


Figure 5-4-F2 Door sensor circuit

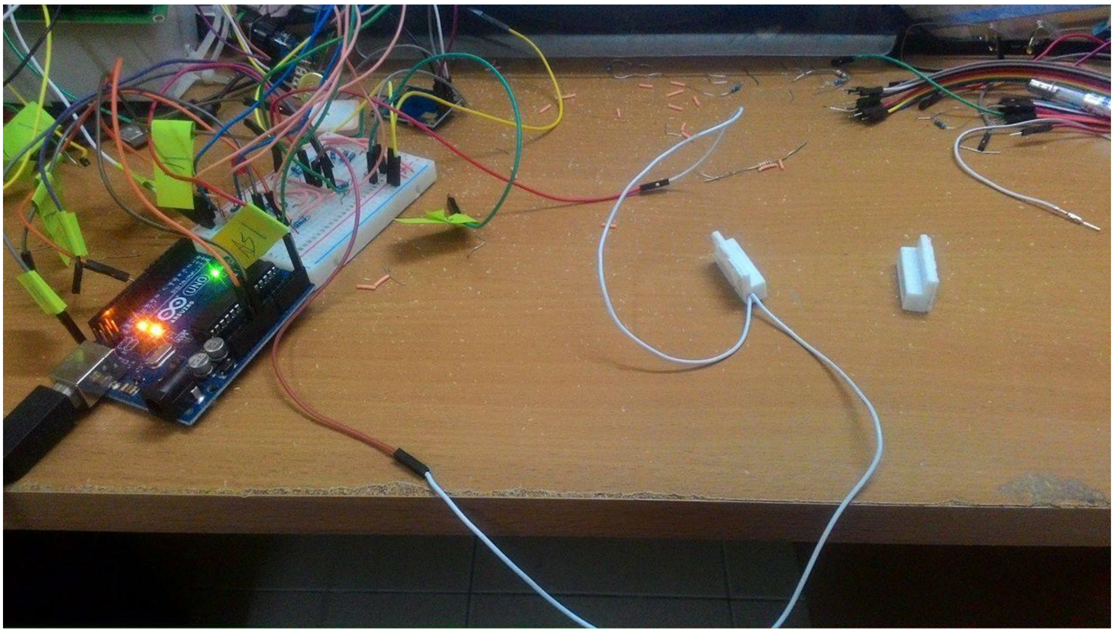


Figure 5-4-F3 Door sensor open circuit

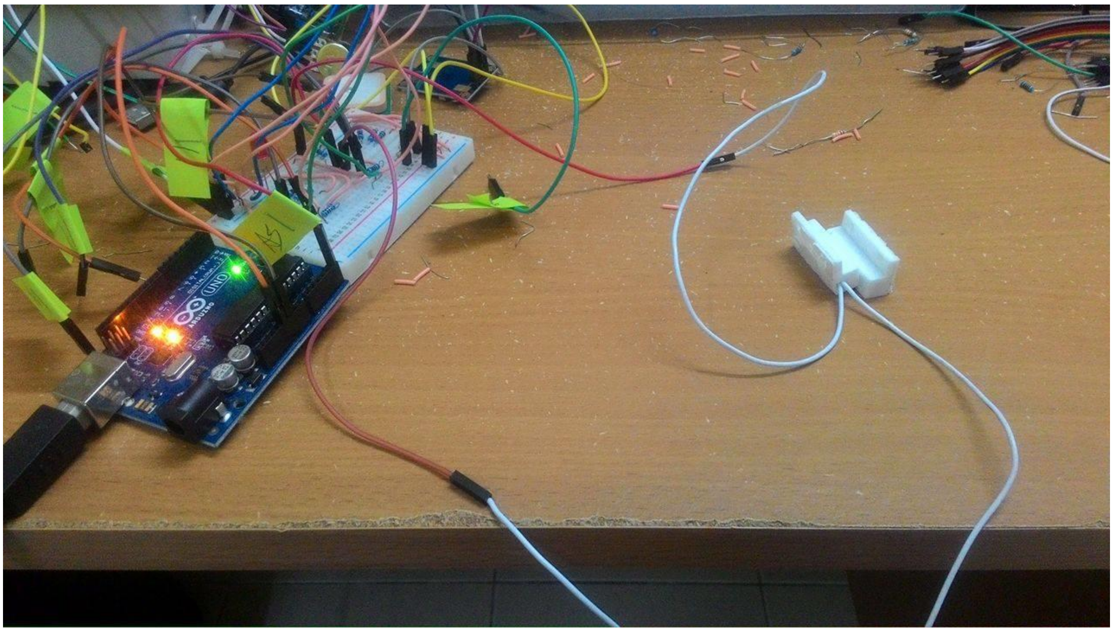
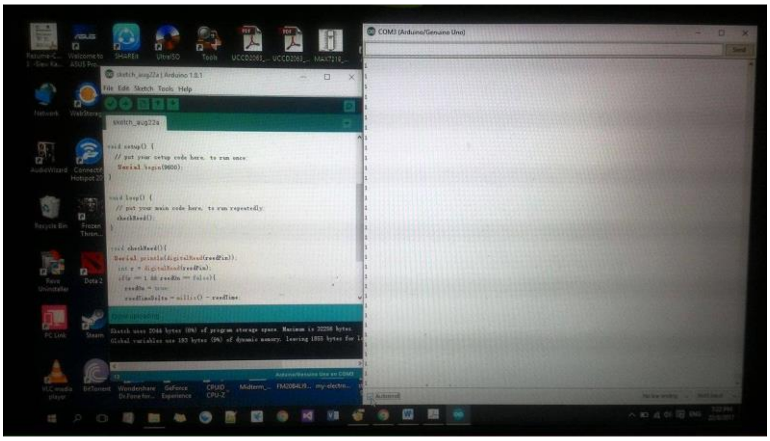


Figure 5-4-F4 Door sensor close circuit

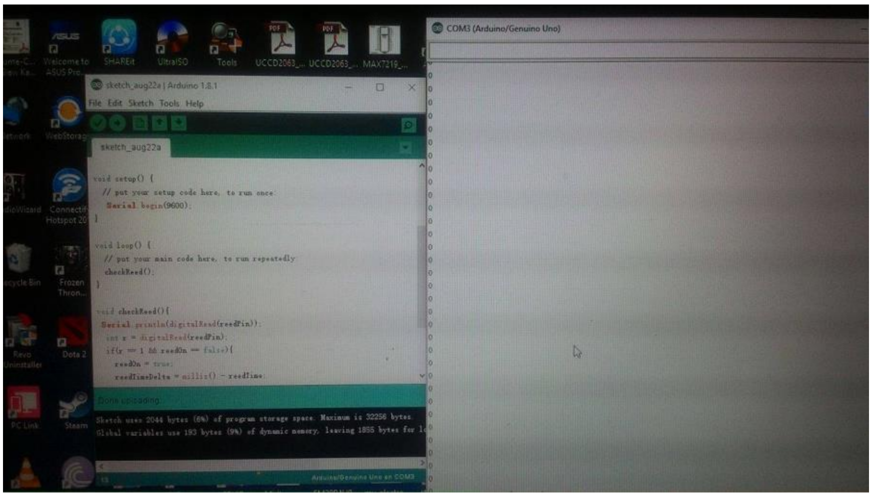


sketch_aug22a | Arduino 1.8.1
File Edit Sketch Tools Help
sketch_aug22a
COM3 (Arduino/Genuino Uno)

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  checkReed();
}

void checkReed(){
 Serial.println(digitalRead(reedPin));
  int r = digitalRead(reedPin);
  if(r == 1 && reedOn == false){
    reedOn = true;
    reedTimeDelta = millis() - reedTime;
```

```
Done uploading.
Sketch uses 2044 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 193 bytes (9%) of dynamic memory, leaving 1855 bytes for l
```

Figure 5-4-F5 Door sensor close circuit result



sketch_aug22a | Arduino 1.8.1
File Edit Sketch Tools Help
sketch_aug22a
COM3 (Arduino/Genuino Uno)

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  checkReed();
}

void checkReed(){
 Serial.println(digitalRead(reedPin));
  int r = digitalRead(reedPin);
  if(r == 1 && reedOn == false){
    reedOn = true;
    reedTimeDelta = millis() - reedTime;
```

```
Done uploading.
Sketch uses 2044 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 193 bytes (9%) of dynamic memory, leaving 1855 bytes for l
```

Figure 5-4-F6 Door sensor open circuit result

This door sensor is used on the bike tyre ring to detect the duration for one cycle so that it can calculate out the bike speed. Besides, it connects to the Arduino Uno at pin analog 5. It only can transmit 1 or 0 to the receiver.

Coding:

const int reedPin = A5;

Serial.println(digitalRead(reedPin));

Explanation:

This two simple code is to test the door sensor is working or not. By get the pin number and print it out in the serial monitor of the Arduino IDE.

5.5 LCD 16x2 green display

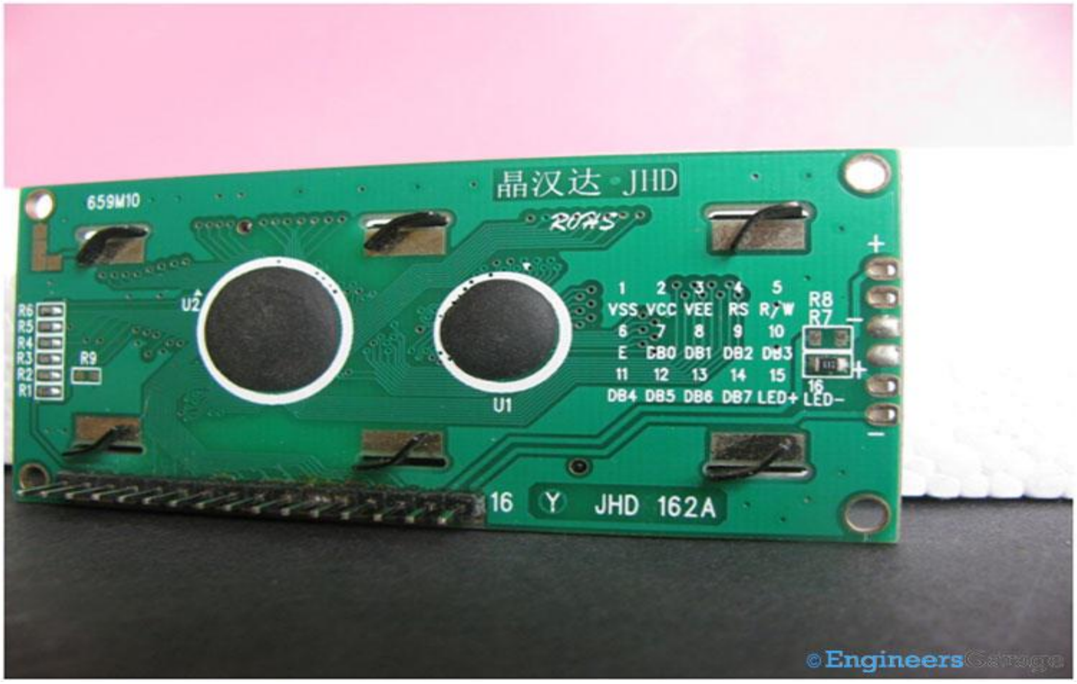


Figure 5-5-F1 LCD backside

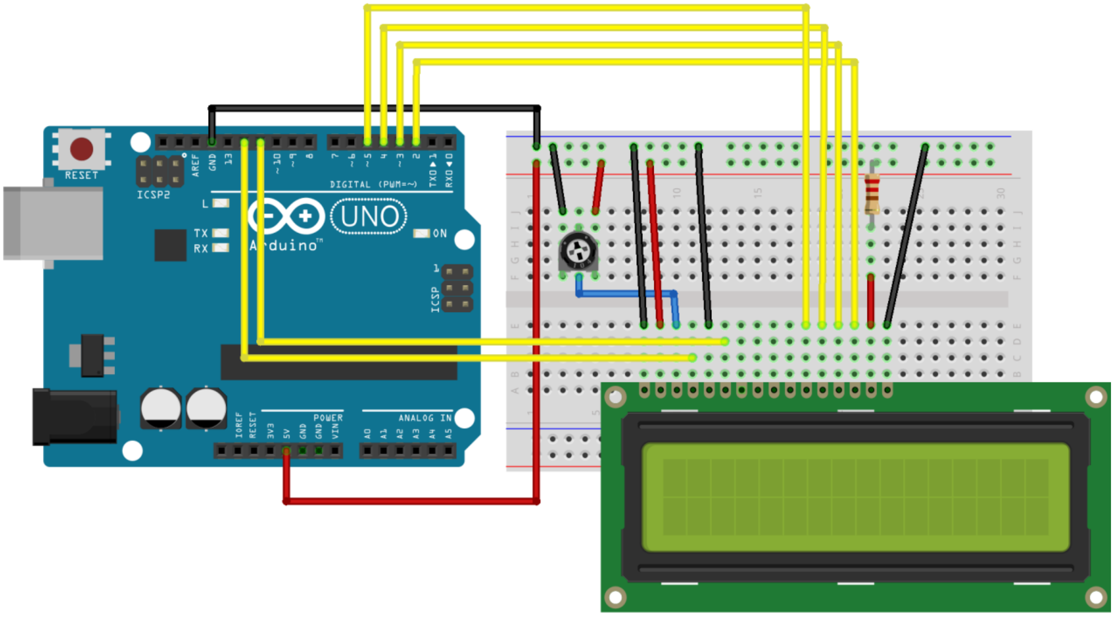


Figure 5-5-F2 LCD with Arduino circuit

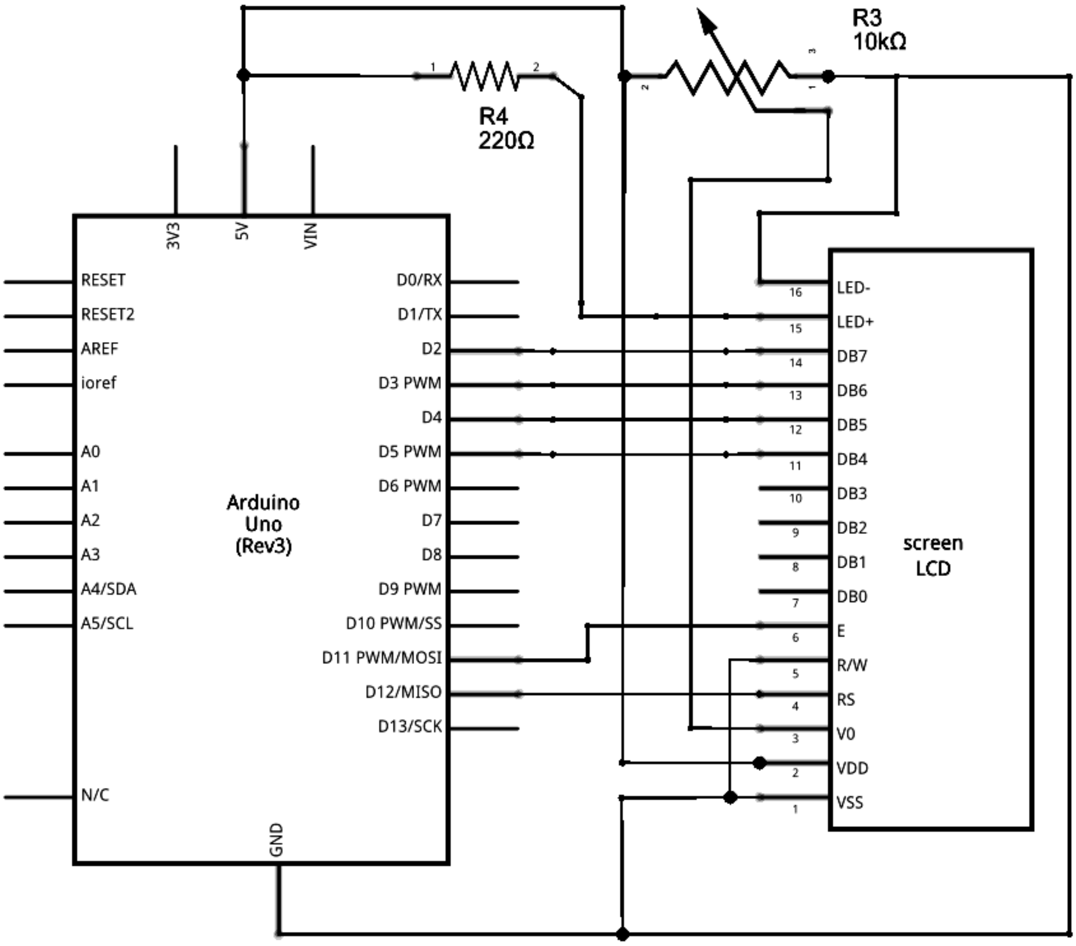


Figure 5-5-F3 LCD with Arduino schematic

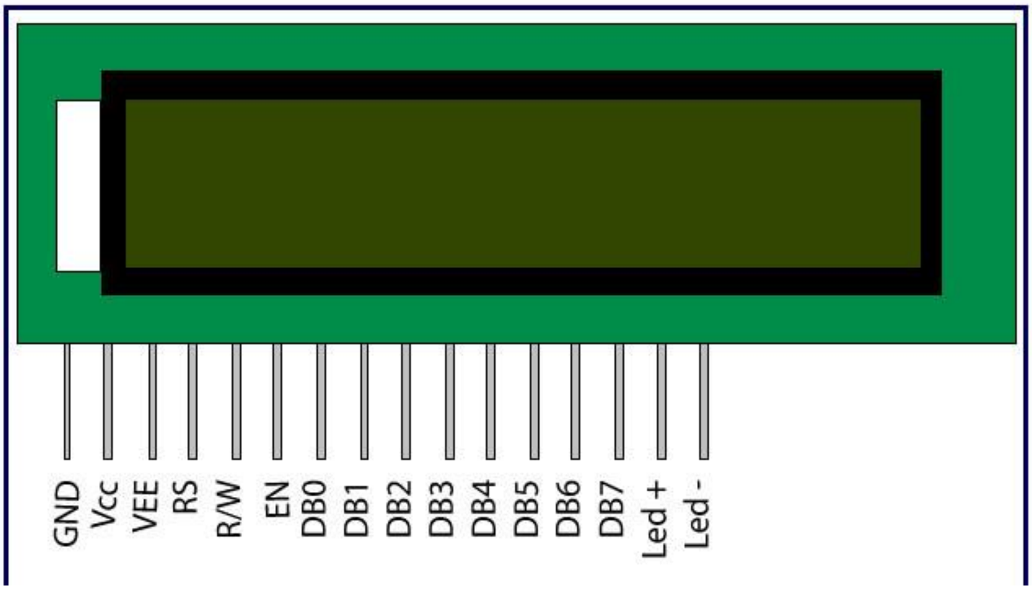


Figure 5-5-F4 LCD with Arduino pin layout

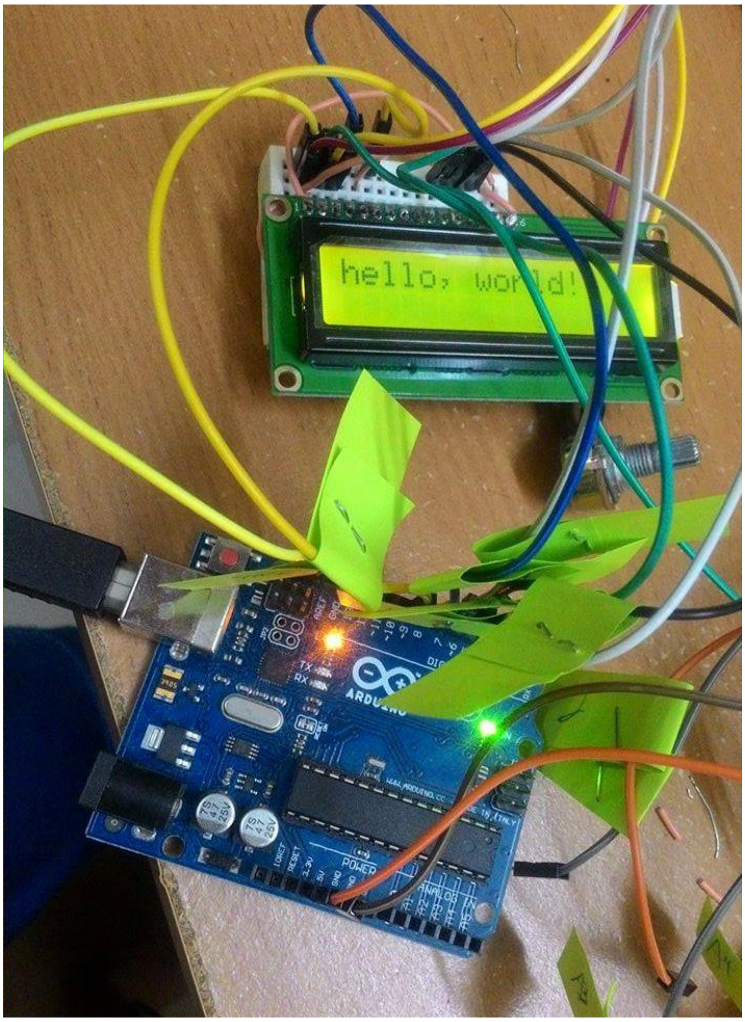


Figure 5-5-F5 LCD with Arduino circuit

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Table 5-T1 LCD pin description

Liquid Crystal Display screen find a wide range of application and is an electronic display module. A module like 16x2 LCD screen is very basic and is very commonly used in various devices and circuits. Compare to seven segments and other multi segment LEDs, these modules LCD display are preferred over them. This is because LCDs are economical and easy programmable. Besides, it have no limitation of display special, even custom characters and animation and so on also one the reasons. Unlike seven segment display, some of these characters it cannot display.

A 16X2 LCD screen means it can display 16 characters per line and these are two such lines. In this LCD screen, each character is displayed in 5x7 pixel matrix. LCD's are available in various shapes and sizes depending on the configurations. This

LCD screen has two registers, it is Command and Data. The command register stores the command instruction given to the LCD screen.

A command is an instruction given to LCD screen to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD screen. The data is the ASCII value of the character to be displayed on the LCD screen. It is capable to display any character with ASCII values ranging from 0 to 255. In the backside of LCD, a PCB is attached which contains the required circuitry to process the signals.

Besides, when connect to Arduino Uno to have a testing, it need 7 pins to connect with Arduino. First two pins are 5v pin and ground pin. Remaining are rs, en, d4 pin to d7 pin. rs pin mean register select and en pin mean enable which connect to Arduino digital pin o 8,9,4, 5,6 and 7 respectively. Furthermore, LCD screen cannot adjust the contrast of black screen light display in it with software, it need to adjust in the hardware. Therefore, it needs a potentiometer to adjust the voltage, thus add the voltage to display black screen light with more contrast or lower down the voltage to display black screen light with low contrast.

Coding:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);



void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}



void loop() {
  // Turn off the display:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
}
```

Explanation:

First of all, it need to include necessary liquid crystal header from the library. Second initialize pin respect to the figure as show above, which connect en, rs, 4, 5, 6 and 7 to 8, 9, 4, 5, 6 and 7 of Arduino Uno digital pin. After that, set up the LCD's number of columns and rows and print a message to the LCD. Then make it blinking as can test the LCD screen.

5.6 Slide Potentiometers

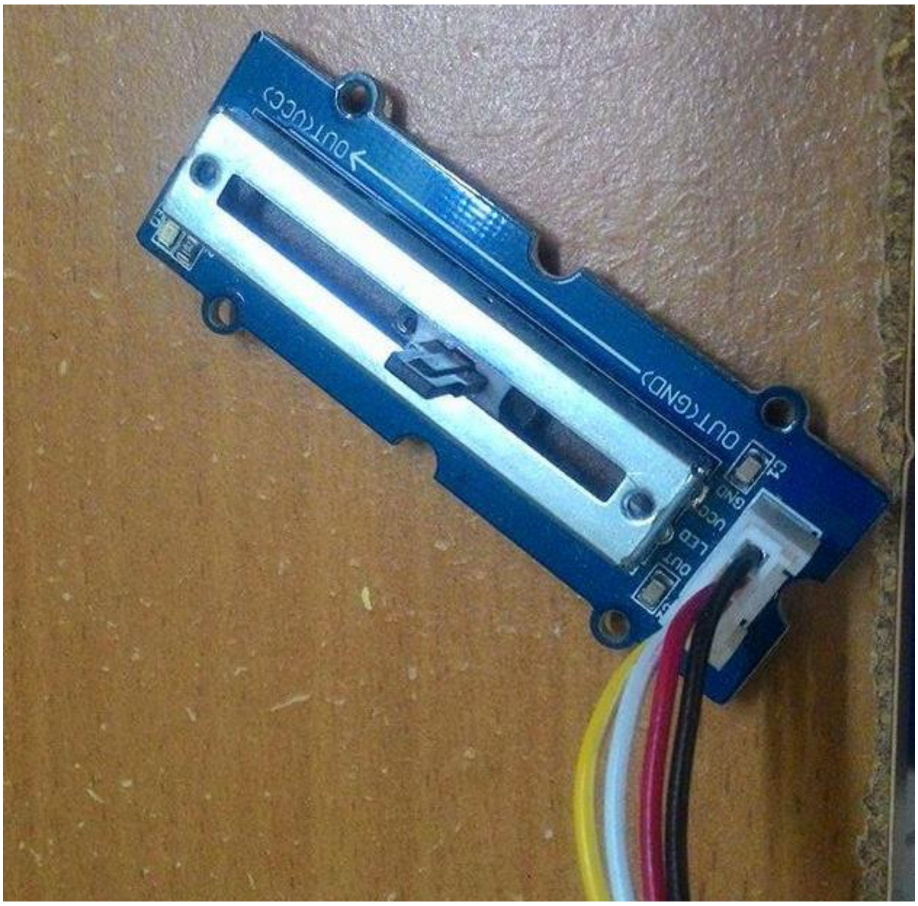


Figure 5-6-F1 Slide potentiometer

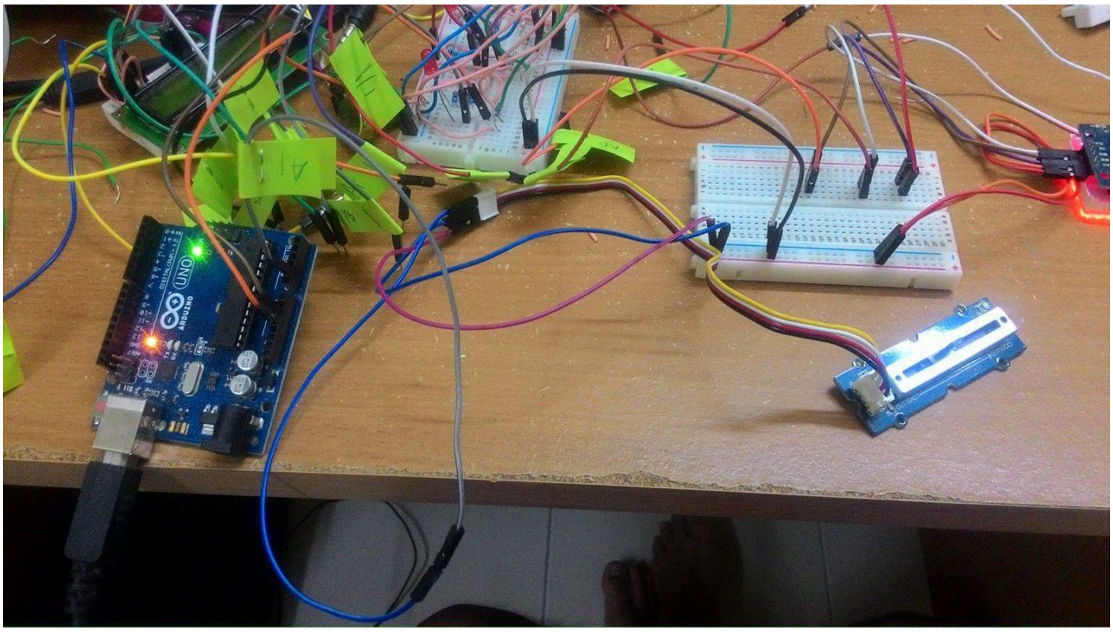


Figure 5-6-F2 Slide potentiometer with Arduino

Item	Min	Typical	Max
Voltage (DC)	3.3V	5.0V	30V
Current	-	-	30mA
Dimension	24mm x60mm		
Net Weight	8.6g		
Rotational life	>15,000 cycles		
Total resistance	10KΩ		
Stroke length	30mm		
Total resistance tolerance	+/- 20%		

Table 5-6-T1 Slide potentiometer details

This slide potentiometer is a linear variable resistor with a total resistance of 10k. When you move the lever from one side to the other, its output voltage will range between 0 V to the VCC you apply. Three of four Grove pins are connected to VCC, GND and the ADC IN on the slide, while the remaining pin is connected to a green indicator LED. You can use the indicator LED to visually display the change on the potentiometer. It is 30mm long slide length and in linear resistance taper. Furthermore, it is grove compatible and 10K ohm of linear resistor.

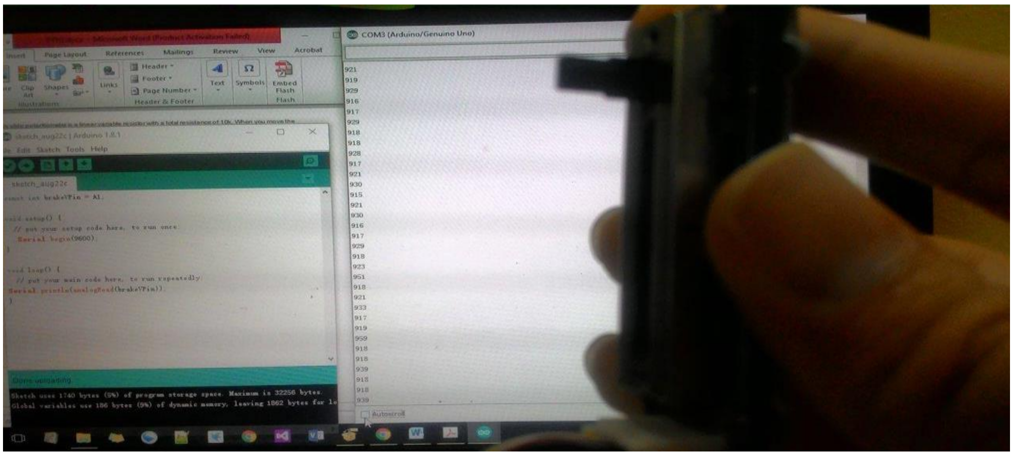
In the other hands, while connect to Arduino Uno, it only had 4 pin to connect in order to test or use it. First is VCC pin for the input voltage, second is the ground pin to connect ground pin of slide potentiometer. Third is the LED light that get to know whether this slide potentiometer hardware function or not. This pin need connect to a 3V pin of Arduino Uno. The last pin is the output pin that need connect to the analog pin of Arduino Uno.

Coding:

```
const int brakeVPin = A1;

void setup() {
  // put your setup code here, to run once:
   Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
Serial.println(analogRead(brakeVPin));
}
```

Explanation:

At first, initialize the analog pin A1 in Arduino Uno where the output pin of slide potentiometer is connecting. Then start the serial monitor with respect baud rate which is 9600. Last but not least, print out the output of the slide potentiometer.



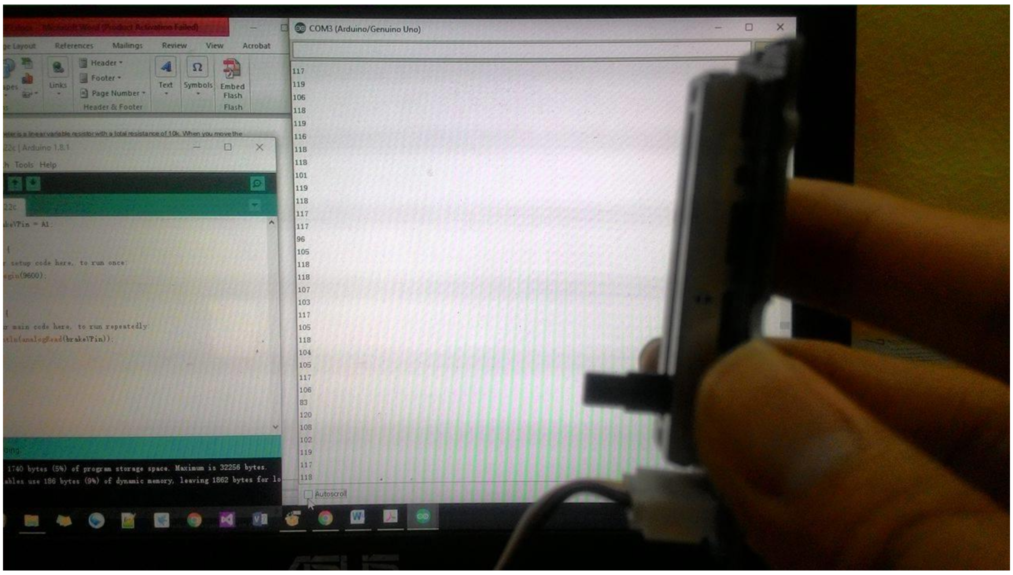
```
const int brakeVPin = A1;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
Serial.println(analogRead(brakeVPin));
}
```

Done uploading.

```
Sketch uses 1740 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 186 bytes (9%) of dynamic memory, leaving 1862 bytes for lo
```

```
921
919
929
916
917
929
918
918
928
917
921
930
915
921
930
916
917
929
918
923
951
918
921
933
917
919
959
918
918
939
918
918
939
```

Figure 5-6-F3 Slide potentiometer results when open



```
akeVPin = A1;

 {
r setup code here, to run once:
egin(9600);

 {
r main code here, to run repeatedly:
ntln(analogRead(brakeVPin));
```

```
 1740 bytes (5%) of program storage space. Maximum is 32256 bytes.
ables use 186 bytes (9%) of dynamic memory, leaving 1862 bytes for lo
```

```
117
119
106
118
119
116
118
118
101
119
118
117
117
96
105
118
118
107
103
117
105
118
104
105
117
106
83
120
108
102
119
117
118
```

Figure 5-6-F4 Slide potentiometer results when close

5.7 GY-NEO6MV2 GPS module for Arduino

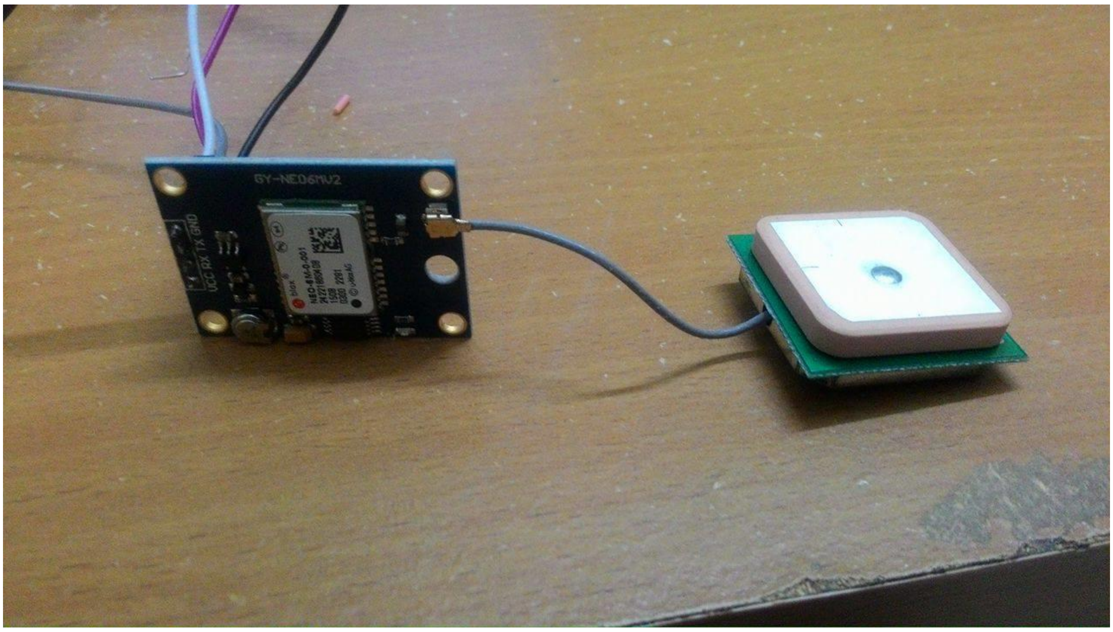
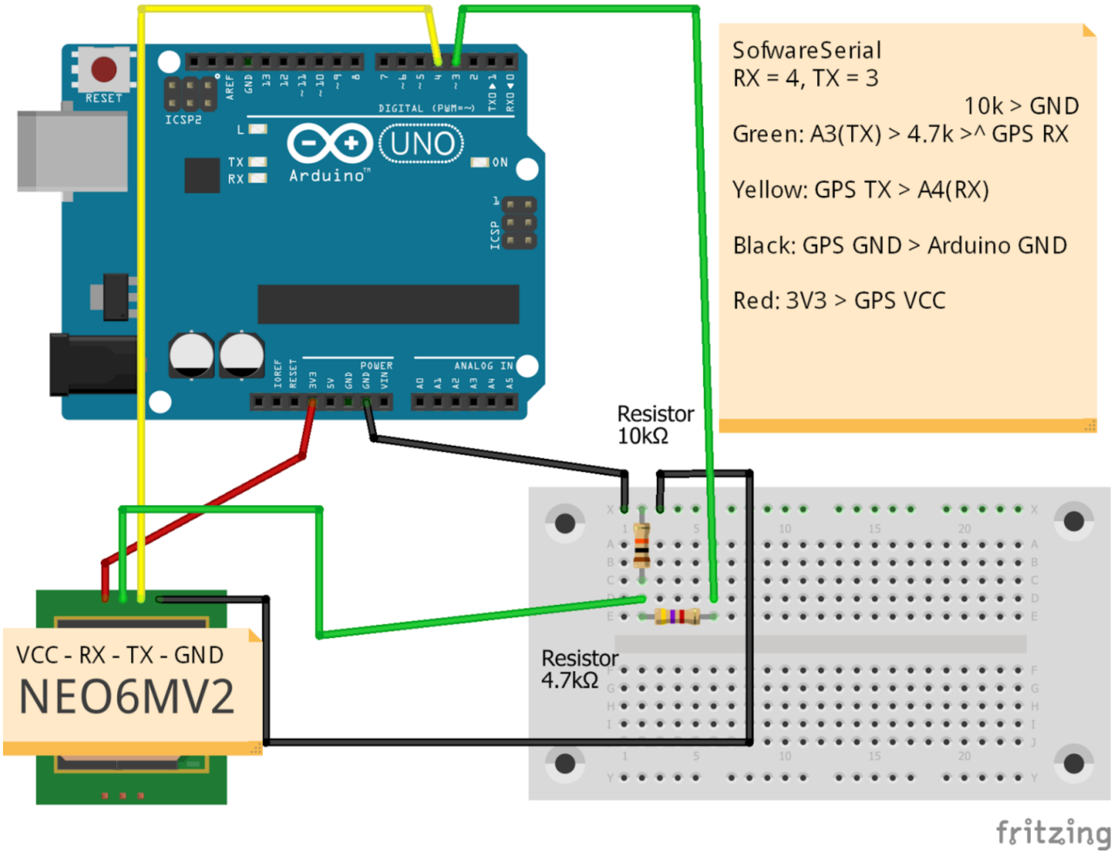


Figure 5-7-F1 GPS module


SofwareSerial
RX = 4, TX = 3
10k > GND
Green: A3(TX) > 4.7k >^ GPS RX

Yellow: GPS TX > A4(RX)

Black: GPS GND > Arduino GND

Red: 3V3 > GPS VCC

Figure 5-7-F2 GPS module with Arduino

This board features the u-blox NEO-6M GPS module with antenna and built-in EEPROM. It is compatible with various flight controller boards designed to work with a GPS module, but sometime is also named as GY-GPS6MV2. Besides, it includes ceramic antenna and EEPROM for saving the configuration data when powered off. In the other hands, it contains backup battery and its operating voltage is from 3.3 to 5Vdc.

Furthermore, its communication interface is UART and in 9600bps of baud rate. It also had a LED signal indicator to let user know whether it is function or not. When connect to Arduino Uno, its VCC pin can handle from 3.3 to 5V and it connects to 5V pin in Arduino Uno. Besides, RX pin is data receive in, TX is data transmit out and GND is ground.

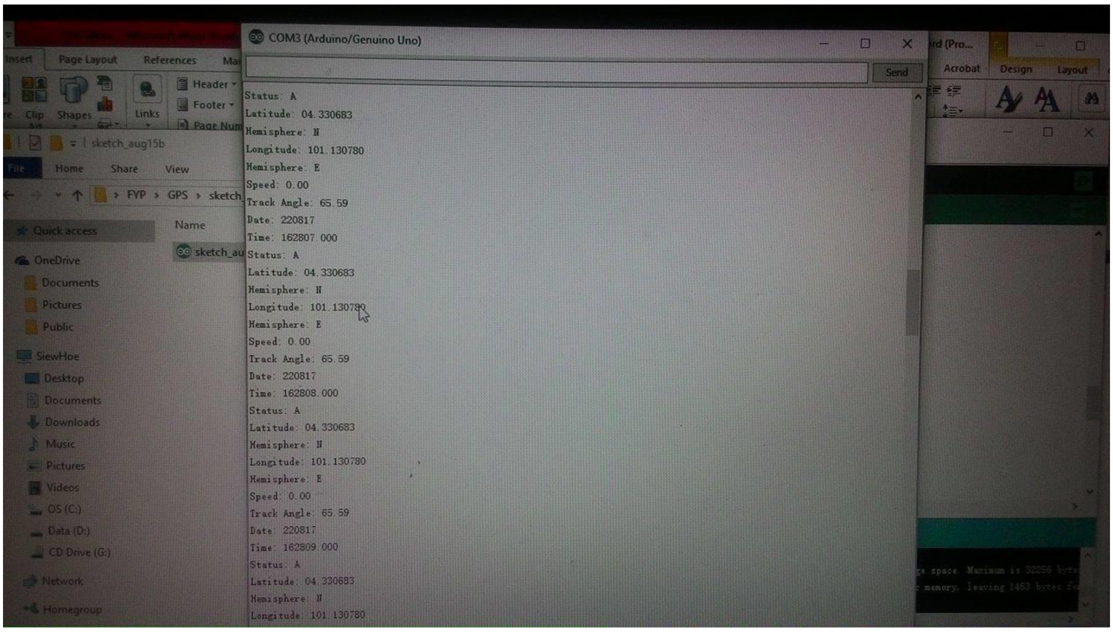
Due to GY-NEO6MV2 GPS module is very sensitive product. Therefore, it need a 10k ohm and a 4.7k ohm of resistor while connect to the Arduino Uno. It only can receive the location data when it used the resistor connect with Arduino Uno. Otherwise, the gps module cannot work well. The connection between gps module and Arduino Uno are showed as below.

Coding:

```
#include <SoftwareSerial.h>
SoftwareSerial GPSModule(11, 10); // RX, TX

void loop() {
  Serial.flush();
  GPSModule.flush();
  while (GPSModule.available() > 0)
  {
    GPSModule.read();

  }

void loop() {
  Serial.flush();
  GPSModule.flush();
  while (GPSModule.available() > 0)
  {
    GPSModule.read();

  }

    Serial.print(labels[i]);
      Serial.print(nmea[i]);
      Serial.println("");
}

String lngfirst;
  float lngsecond;
  for (int i = 0; i < nmea[4].length(); i++) {
    if (nmea[4].substring(i, i + 1) == ".") {
      lngfirst = nmea[4].substring(0, i - 2);
      //Serial.println(lngfirst);
      lngsecond = nmea[4].substring(i - 2).toFloat();
      //Serial.println(lngsecond);

    }
  }
  lngsecond = lngsecond / 60;
  String CalcLng = "";
  char charVal[9];
  dtostrf(lngsecond, 4, 6, charVal);
  for (int i = 0; i < sizeof(charVal); i++)
  {
    CalcLng += charVal[i];
  }
  lngfirst += CalcLng.substring(1);
  lngfirst = posneg += lngfirst;
  return lngfirst;
}
```

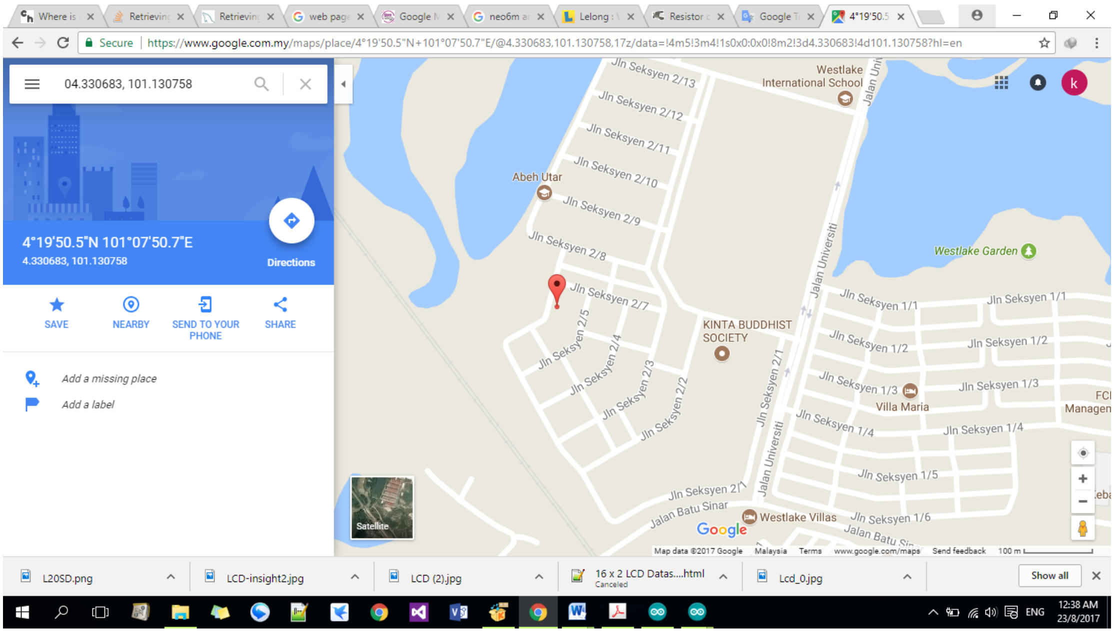
Explanation:

First I need to include the software serial header file from Arduino IDE library. Then I assign a new gps module with pin in it. After that, I run the serial monitor and gps module in the same baud rate which is 9600, otherwise it won’t work. Then open the serial monitor to check the result of the data that gps receiver received. Which are latitude and longitude.



```
Status: A
Latitude: 04.330683
Hemisphere: N
Longitude: 101.130780
Hemisphere: E
Speed: 0.00
Track Angle: 65.59
Date: 220817
Time: 162807.000
Status: A
Latitude: 04.330683
Hemisphere: N
Longitude: 101.130780
Hemisphere: E
Speed: 0.00
Track Angle: 65.59
Date: 220817
Time: 162808.000
Status: A
Latitude: 04.330683
Hemisphere: N
Longitude: 101.130780
Hemisphere: E
Speed: 0.00
Track Angle: 65.59
Date: 220817
Time: 162809.000
Status: A
Latitude: 04.330683
Hemisphere: N
Longitude: 101.130780
```

Figure 5-7-F3 GPS module results



https://www.google.com.my/maps/place/4°19'50.5"N+101°07'50.7"E/@4.330683,101.130758,17z/data=!4m5!3m4!1s0x0:0x0!8m2!3d4.330683!4d101.130758?hl=en
04.330683, 101.130758
4°19'50.5"N 101°07'50.7"E
4.330683, 101.130758
Directions
SAVE
NEARBY
SEND TO YOUR PHONE
SHARE
Add a missing place
Add a label
Westlake International School
Abeh Utar
KINTA BUDDHIST SOCIETY
Westlake Garden
Villa Maria
Westlake Villas
Satellite

Figure 5-7-F4 GPS module results in map

Chapter 6: Software development

6.1 Raspberry Pi 3

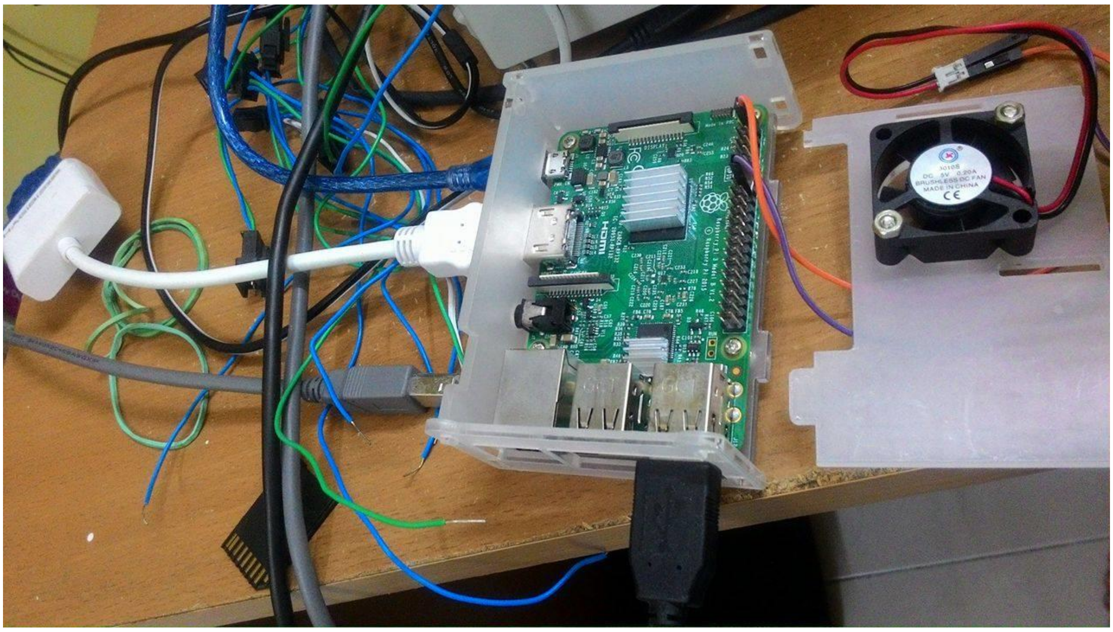


Figure 6-1-F1 Raspberry Pi 3

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles.

According to the Raspberry Pi Foundation, over 5 million Raspberry Pis have been sold before February 2015, making it the best-selling British computer. By November 2016 they had sold 11 million units.

Built on the latest Broadcom 2837 ARMv8 64bit processor the Raspberry Pi 3 Model B is faster and more powerful than its predecessors.

It has improved power management to support more powerful external USB devices and now comes with built-in wireless and Bluetooth connectivity. To take full advantage of the improved power management on the Raspberry Pi 3 and provide even support for even more powerful devices on the USB ports, a 2.5A adapter is recommended.

Raspberry Pi 3 has broadcom BCM2837 64bit ARMv8 QUAD core 64bit processor powered Single Board and let Computer running at 1.2GHz with 1GB RAM. It includes BCM43143 WiFi on board and Bluetooth Low Energy (BLE) on board too. Besides, it has 40pin extended GPIO, 4 x USB2 ports, 4 pole stereo output and composite video port, full size HDMI. Besides, it also includes CSI camera port for connecting the Raspberry Pi camera and DSI display port for connecting the Raspberry Pi touch screen display. Furthermore, MicroSD port for loading your operating system and storing data and upgraded switched Micro USB power source (now supports up to 2.5 Amps)

Key Benefits

- Low cost
- Consistent board format
- 10x faster processing
- Added connectivity

Key Applications

- Low cost PC/tablet/laptop
- IoT applications
- Media centre
- Robotics
- Industrial/Home automation
- Server/cloud server
- Print server • Security monitoring
- Web camera • Gaming
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)

First of all is setup Raspberry Pi 3:

1) A Raspberry Pi 3
2) An HDMI to VGA converter.
3) A USB keyboard and mouse
4) An 8GB MicroSD card and card reader
5) A power supply

Simple process to setup :

1. Put SD card into computer or SD card reader.
2. Download NOOBs. Pick the “offline and network install” option. This version includes Raspbian in the download itself.
3. It may need to format SD card as FAT.
4. Extract the ZIP file and copy the entire contents of the folder to SD card. Once it’s complete, eject SD card and insert it into the Raspberry Pi.

Connecting all your devices to the Raspberry Pi is very easy, but you want to do it in a specific order so it can recognize all your devices when it boots up. First, connect your HDMI cable to your Raspberry Pi and your monitor, then connect your USB devices. Raspberry Pi 3 had Wi-Fi build in, therefore it no need any ethernet cable to online also can use the internet.

Finally, once everything is connected, go ahead and plug in your power adapter. The Raspberry Pi does not have a power switch, so once you connect the power adapter, it’ll turn on all by itself.

When you first boot up NOOBs, it’ll take a couple of minutes to format the SD card and set up a few things, so let it do its thing. Eventually, you’ll see a screen asking you to install an operating system. This process is super easy:

1. At the bottom of the screen, choose your language and keyboard layout for your region.
2. Click the checkbox next to Raspbian, then click Install.

Now, let NOOBs run the installation process, which can take 10 or 20 minutes. When it's complete, it will restart and send you straight into the Raspbian desktop, where you can configure everything else.

6.2 Python Grabserial installation

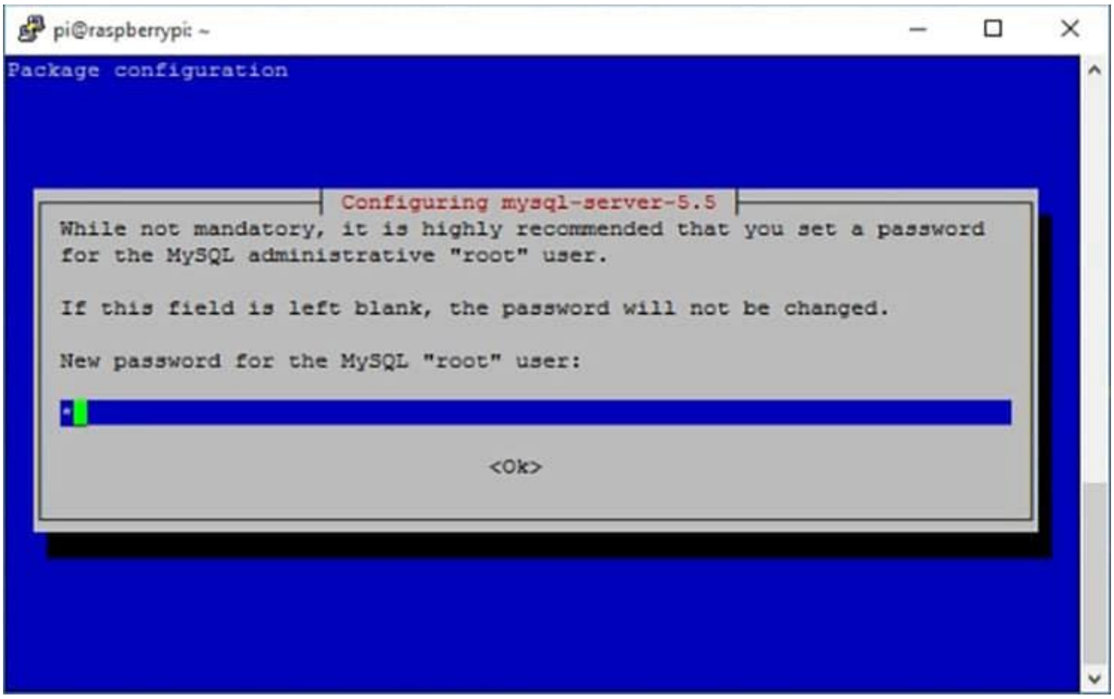
1. By using command sudo apt-get install python grabserial
2. then sudo nano grabserial.py file
3. By changing the baud rate to 9600
4. By changing the port to ACMO
5. run the python file by using command: python grabserial.py

6.3 MySQL and PHPMyAdmin setup

1) Firstly install the mysql server onto the Raspberry Pi.

   sudo apt-get install mysql-server

2) It will be prompted to enter a password for the root user. A password is required to access the MYSQL server and connect PHPMyAdmin to it.



```
Package configuration

Configuring mysql-server-5.5
While not mandatory, it is highly recommended that you set a password
for the MySQL administrative "root" user.

If this field is left blank, the password will not be changed.

New password for the MySQL "root" user:

<Ok>
```

Figure 6-2-F1 Mysql database server

mysql -u root -p

It will now prompt to enter the password we just created.
Now can enter MYSQL commands to create, alter, delete databases.
Can leave the command line by simply entering quit.

If want to be able to interact with MYSQL in Python I need to install the Python bindings as well.
sudo apt-get install python-mysqldb

6.4 Installing Raspberry Pi PHPMyAdmin

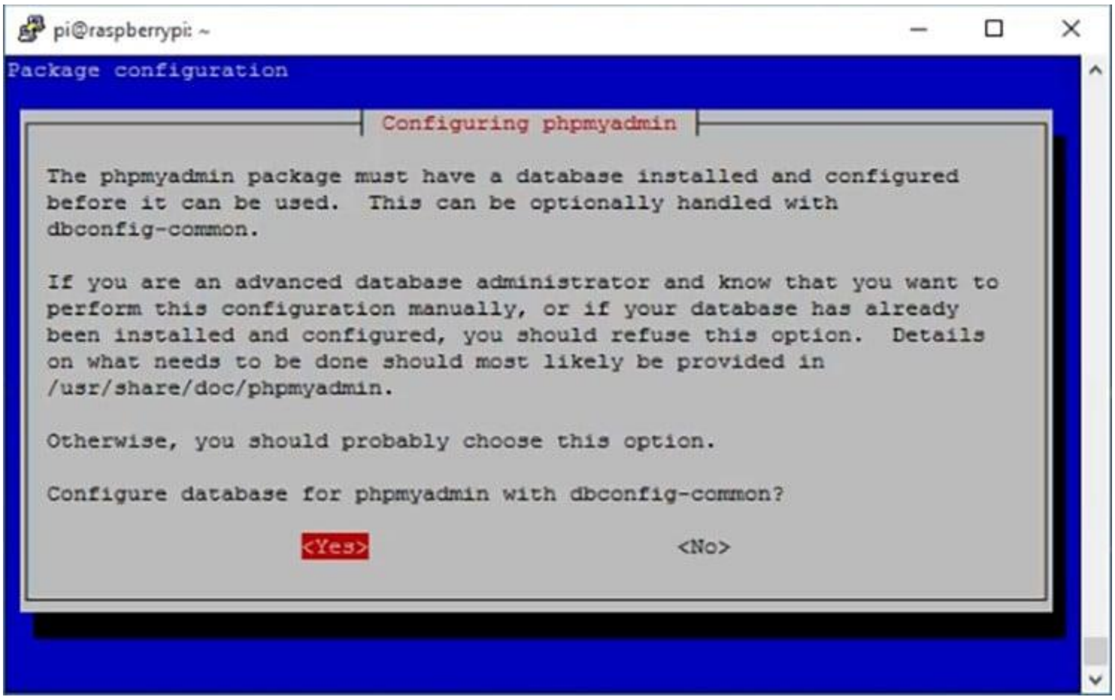
sudo apt-get install apache2 php5 libapache2-mod-php5

Now let’s install the PHPMyAdmin package, you can do this by entering the following:

sudo apt-get install phpmyadmin

It will now begin to install. It will be presented with a screen asking the type of web server you want it to run off. Select apache2 as this is the server I just installed.

4. Next will need to configure PHPMyAdmin to connect a database.



```
The phpmyadmin package must have a database installed and configured
before it can be used.  This can be optionally handled with
dbconfig-common.

If you are an advanced database administrator and know that you want to
perform this configuration manually, or if your database has already
been installed and configured, you should refuse this option.  Details
on what needs to be done should most likely be provided in
/usr/share/doc/phpmyadmin.

Otherwise, you should probably choose this option.

Configure database for phpmyadmin with dbconfig-common?

                  <Yes>                          <No>
```

Figure 6-3-F1 Phpmyadmin setup

It will now ask for a password, I enter the one set previously when I set up MYSQL.

Next it will ask me to set a password for PHPMyAdmin. I keep it the same as the password to the MYSQL database or something separate.

I also need to setup Apache to include our PHPMyAdmin installation

sudo nano /etc/apache2/apache2.conf

Now at the bottom of this file enter the following line:

Include /etc/phpmyadmin/apache.conf

sudo /etc/init.d/apache2 restart

By typing localhost:5000 it can access the database by phpmyadmin.

6.5 Flask

Flask is a simple and easy microframework for Python based. It is easy to setup and micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed.

The latest stable version of Flask is 0.12 as of December 2016. Applications that use the Flask framework include Pinterest, LinkedIn and the community web page for Flask itself.

Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

By using a simple command to install falsk

```
sudo apt-get install python-pip

sudo pip install flask
```

First of all, create a directory that name as Web_app. Secondly, inside it create a simple python file call hello.py. Then write the following codes.

```
from flask import Flask

app = Flask(__name__)



@app.route('/')

def hello_world():

   return 'Hello World’
```

```
if __name__ == '__main__':

   app.run()
```

Now by typing localhost:5000, flask web page will be prompt out on the web browser. By adding a static folder to store the css and javascript file in it for the use of web page.

Then add one more folder call templates to store the html file inside it and call it by the command of render_templates in the python file.

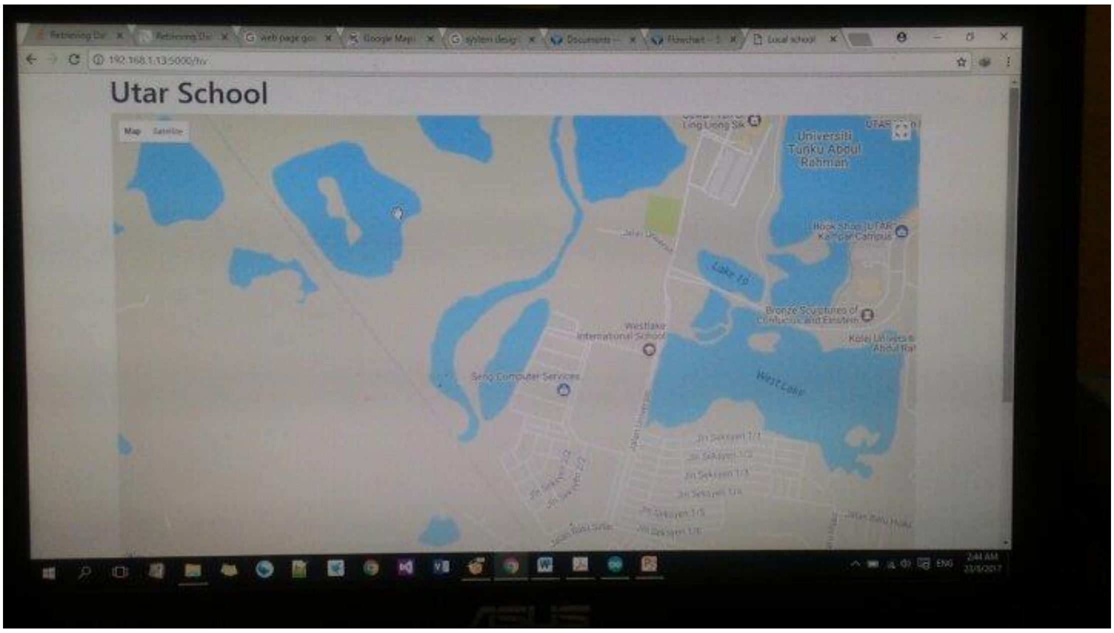


Figure 6-4-F1 Google map result

Chapter 7: Requirement and Specification

7.1 In this project the several hardware materials are:

1) Arduino Uno Board
2) Raspverry Pi 3
3) LED matrix panel
4) Slide potentiometer
5) GPS module
6) On/off push button
7) Door sensor
8) LCD display screen

7.2 Several hardware components are:

1) Breadboard
2) Potentiometer
3) 10k ohm resistor
4) 4.7k ohm resistor
5) 660 ohm resistor
6) LED light
7) Male to male jumper wire
8) Female to female jumper wire
9) Male to female jumper wire
10) Waterproof casing box
11) DC current adapter
12) Wire
13) HDMI to VGA cable

7.3 Several hardware equipments:

1) Monitor
2) USB keyboard and mouse
3) Power bank
4) Battery

5) Road Bike
6) Tape

7.4 Analysis, Design and Verification Plan

Design of this project is to assist cyclist when they are using bicycle on the road. It has to be able show signal light and brake light. Besides, it will let user know the speed and distance that they travel. Furthermore, they can trace back where they went before after they had finished their ride. Therefore, it need a Wi-Fi board in chip board like raspberry pi to transfer data to databases and store in as server and can retrieve the data in web page.

7.5 Verification Plan

First of all, when both Uno board and raspberry pi board, the whole circuit run successfully.

Secondly, when button push signal light in LED matrix panel on.

Third, when slide potentiometer slide, signal light in LED matrix panel on.

Fourth, LCD display screen show the speed and distance by the door sensor.

Fifth, Uno board send data to raspberry pi and Pi 3 sotred into database and upload to web page.

7.6 Testing

Sensor	Output
Left button push	Left signal light display on in LED matrix module
Left button push	Left signal light display off in LED matrix module
Right button push	Right signal light display on in LED matrix module
Right button push	Right signal light display off in LED

	matrix module
Slide potentiometer slide	Brake signal light display on in the LED matrix module
Slide back potentiometer slide	Brake signal light display off in the LED matrix module
Bicycle cycling	LCD display speed and distances
Power on	GPS led light blinking
Power on	Arduino transfer data to Raspberry Pi 3

Table 7-6-T1 Testing plan

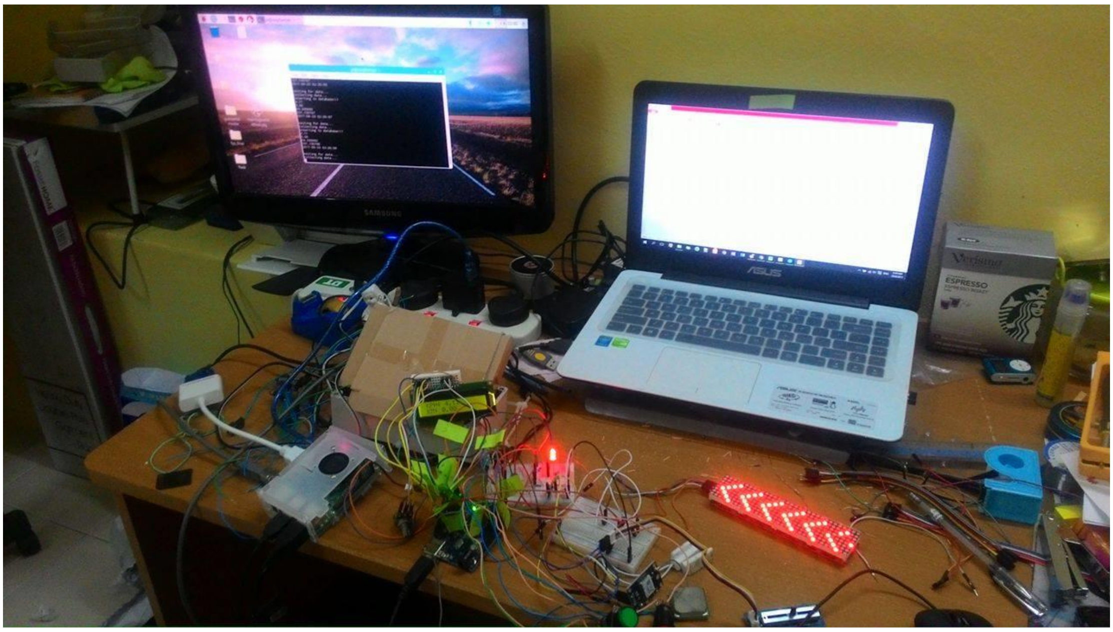


Figure 7-5-F1 Verification plan 1

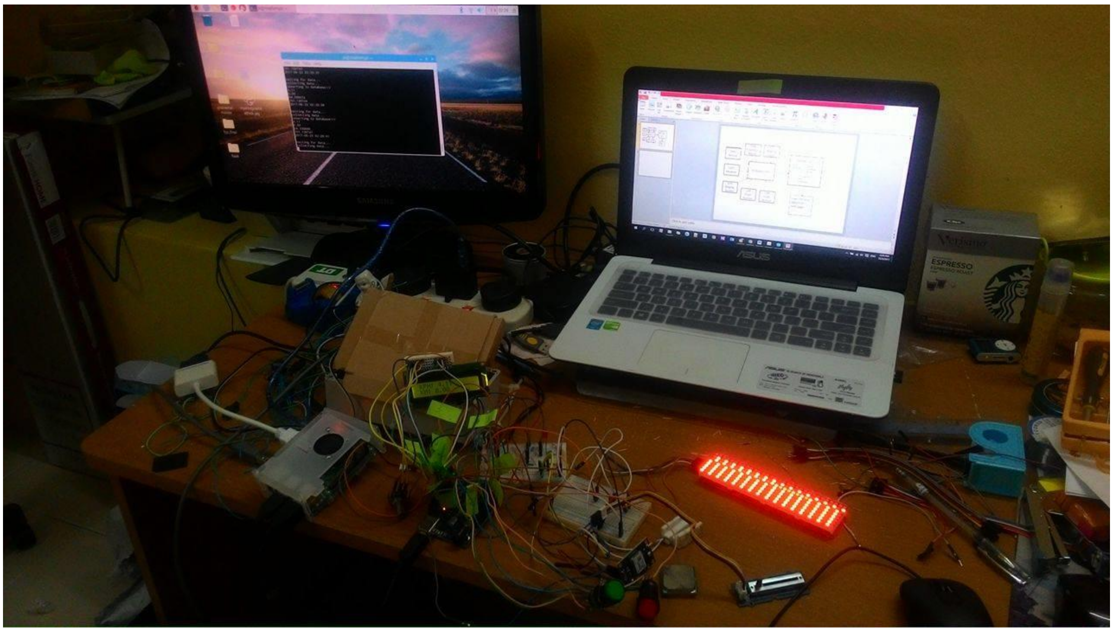


Figure 7-5-F2 Verification plan 2

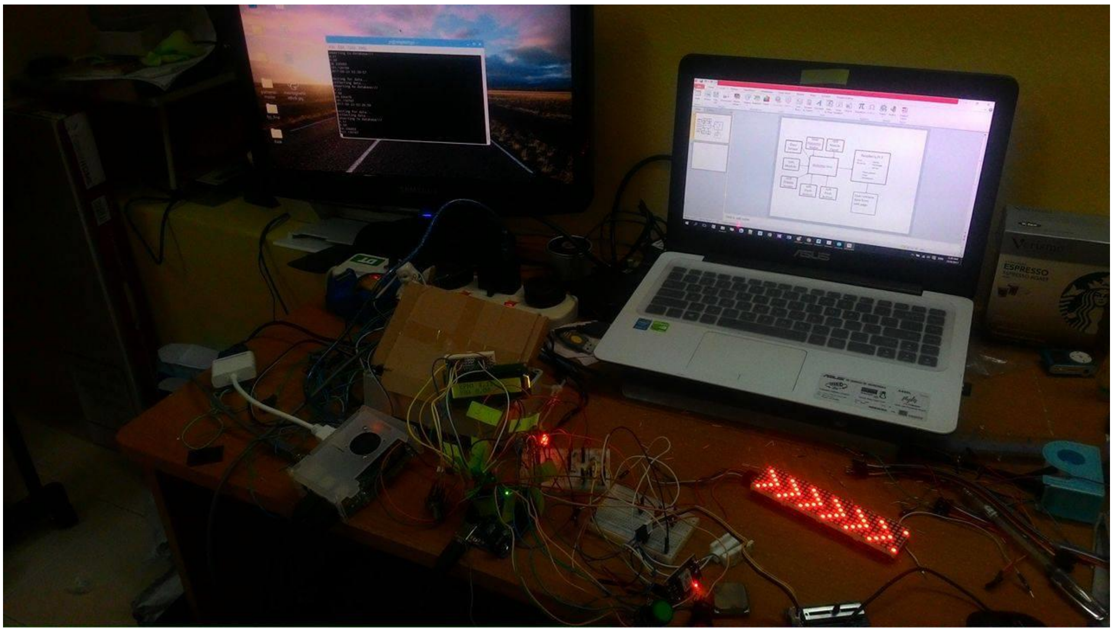
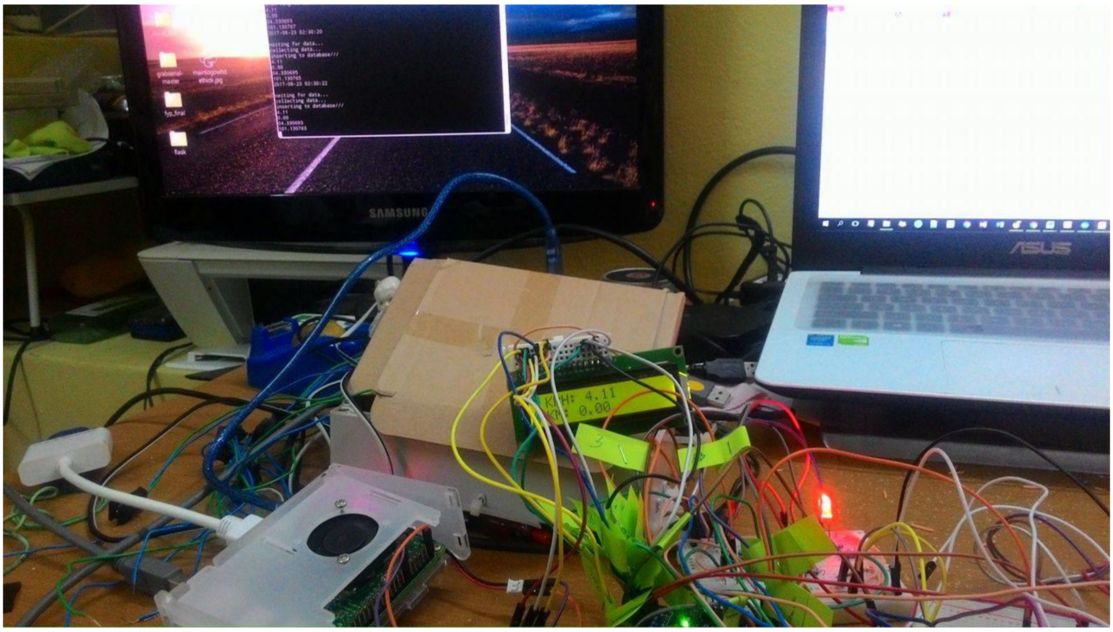


Figure 7-5-F3 Verification plan 3



```
4.11
0.00
4.330693
101.130767
2017-08-23 02:30:20

aiting for data...
ollecting data...
nserting to database///
4.11
0.00
4.330695
101.130765
2017-08-23 02:30:22

aiting for data...
ollecting data...
nserting to database///
4.11
0.00
4.330693
101.130763
```

Figure 7-5-F4 Verification plan 4

Chapter 8: Conclusion

Multifunctional bike system for cyclist is a project that use to solve the dangerous of cyclist when they are on the road. Due to they can’t show out information to others road users therefore others user not concern about their safety. Therefore, this project help cyclist to show their signal via through when a button is click and signal LED light will show out through a LED matrix panel.

Besides, the bike system not only can show out the led signal light, it also can show the speed of the cyclist and store the speed of the cyclist onto the cloud via internet or WiFi. Furthermore, this project also in water proof condition.

The reason why this project will design out is because most of the bike didn’t have any suitable system to assist cyclist while they are cycling on the road. Thus, it leads some inconvenience and will cause cyclist into dangerous situation. Therefore, the main objective of this project is to assist cyclist to cycle safety and improve their performance in cycling too.

In the other hands, the hardware to develop this system also carries important role to ensure the system can be successful build out. The hardware of this system include Arguino Uno board and other sensors, such as, LED matrix bicycle turn signal and brake light and buttons, speedometer, GPS module and other components like USB cable and so on. The software that will be use is Arguino IDE to communicate with the Arguino Uno board and database that can store the cycling data and retrieve the cycling data of the cyclist in the raspberry pi as server to the web page.

Furthermore, during this project a lot of problem had been encountered. Such as hardware components not function due to too high voltage power supply like GPS module, cannot get LCD display due to pin not solder well, cannot send through serial port due to baud rate and serial port name not the same, cannot insert into database due to format not same and cannot draw out google map after retrieve data from database. All these problems have been solved and further improved.

In my opinion, when need to start up a project that we don’t have much experience we need to start it early and read all the instruction of the resources before start to avoid from dangerous situation or damage hardware. Besides, when faced problem that cannot solve, we should be find another way to solve it or ask suggestion from supervisor but not give up easily.

Future improvement of this project can be use android app to manage the system and can add sensor to turn on a front sport light when it is dark.

Chapter 9: References / Bibliography

1 1 Adam, G 1990, *Signal Helmet*. Available from: <https://www.google.com/patents/US4891736>. [15 August 2016]

2 Chris K & Frauke B 2015, 'Smart e-bike monitoring system: real-time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles', *Journal of IET Intelligent Transport Systems*, vol 10, pp. 79-88.

3 Izzo, 1994, *Turn signal and horn assembly for a bicycle*. Available from: <https://www.google.com/patents/US5617303 >. [14 August 2016]

4 Kovarix, Z 1974, *Bicycle signal system*. Available from: <https://www.google.com/patents/US3878387>. [20 August 2016]

5 Mary, O'C 2015, *Leveraging IoT Technology to Make Driving and Cycling Safer and More Efficient*. Available from: <http://www.iotjournal.com/articles/view?12805>. [21 August 2016]

6 Von, L 1994, *Bicycle light signal*. Available from: <https://www.google.com/patents/US5276593?dq=ininventor:%22Von+Lighthill%22&hl=en&sa=X&ved=0ahUKEwi0zcby2dPOAhXCuo8KHSlbAwQQ6AEIHjAA>. [12 August 2016]