

CHAPTER: 1

INTRODUCTION

1.1 DESCRIPTION OF PARTS

1.1.1 RASPBERRY PI

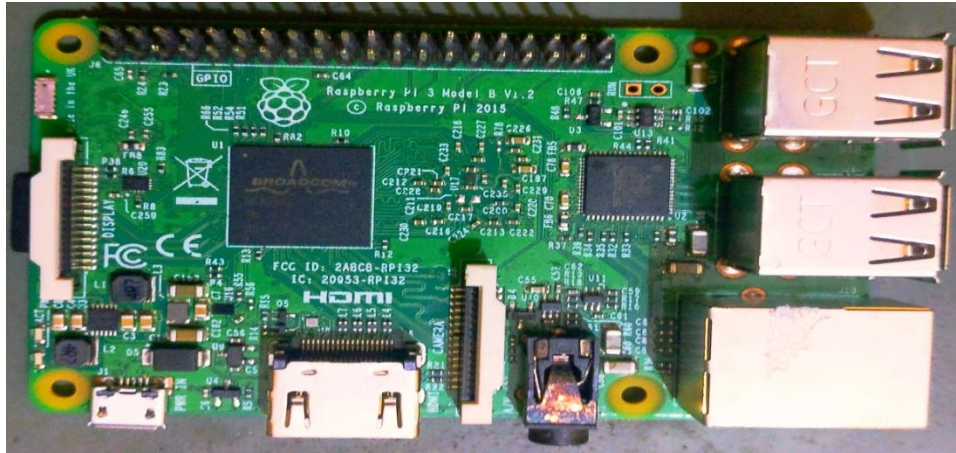


figure1.1: raspberry pi

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Compared to the Raspberry Pi 2 it has:

A 1.2GHz 64-bit quad-core ARMv8 CPU

802.11n Wireless LAN

Bluetooth 4.1

Bluetooth Low Energy (BLE)

Like the Pi 2, it also has:

4 USB ports

40 GPIO pins

Full HDMI port

Ethernet port

Combined 3.5mm audio jack and composite video

Camera interface (CSI)

Display interface (DSI)

Micro SD card slot (now push-pull rather than push-push)

Video Core IV 3D graphics core

The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2.

1.1.2 ARDUINO

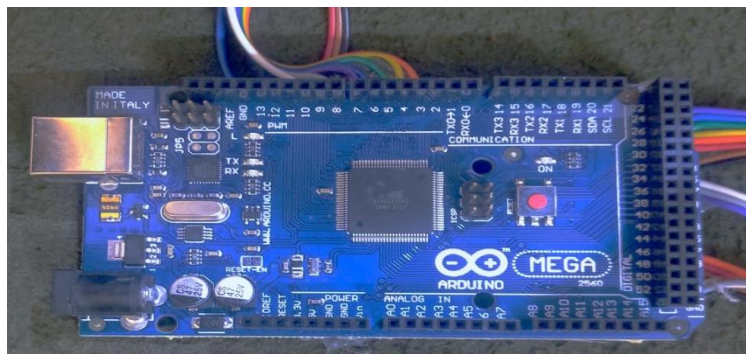


figure1.2: arduino

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Mega 2560 R3 also adds SDA and SCL pins next to the AREF. In addition, there are two new pins placed near the RESET pin. One is the IOREF that allow the shields to adapt to the voltage provided from the board. The other is a not connected and is reserved for future purposes. The Mega 2560 R3 works with all existing shields but can adapt to new shields which use these additional pins.

Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.

1.1.3 LDR SENSOR



figure1.3: ldr sensor

As its name implies, the Light Dependent Resistor (LDR) is made from a piece of exposed semiconductor material such as cadmium sulphide that changes its electrical resistance from several thousand Ohms in the dark to only a few hundred Ohms when light falls upon it by creating hole-electron pairs in the material.

The net effect is an improvement in its conductivity with a decrease in resistance for an increase in illumination. Also, photoresistive cells have a long response time requiring many seconds to respond to a change in the light intensity.

Materials used as the semiconductor substrate include, lead sulphide (PbS), lead selenide (PbSe), indium antimonide (InSb) which detect light in the infra-red range with the most commonly used of all photoresistive light sensors being Cadmium Sulphide (CdS).

A Light Sensor generates an output signal indicating the intensity of light by measuring the radiant energy that exists in a very narrow range of frequencies basically called “light”, and which ranges in frequency from “Infra-red” to “Visible” up to “Ultraviolet” light spectrum.

The light sensor is a passive devices that convert this “light energy” whether visible or in the infra-red parts of the spectrum into an electrical signal output. Light sensors are more commonly known as “Photoelectric Devices” or “Photo Sensors” because the convert light energy (photons) into electricity (electrons).

1.1.4 ULTRASONIC SENSOR



figure1.4: ultrasonic sensor

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

It is important to understand that some objects might not be detected by ultrasonic sensors. This is because some objects are shaped or positioned in such a way that the sound wave bounces off the object, but are deflected away from the Ultrasonic sensor. It is also possible for the object to be too small to reflect enough of the sound wave back to the sensor to be detected.

Other objects can absorb the sound wave all together (cloth, carpeting, etc), which means that there is no way for the sensor to detect them accurately. These are important factors to consider when designing and programming a robot using an ultrasonic sensor.

1.1.5 EM-18 MODULE READER

EM-18 RFID reader module uses a RFID reader that can read 125 KHz tags. So, it can be called as a low frequency RFID reader. It gives out a serial output and has a range of about 8-12 cm. There is a built-in antenna and it can be connected to the PC with the help of RS232.

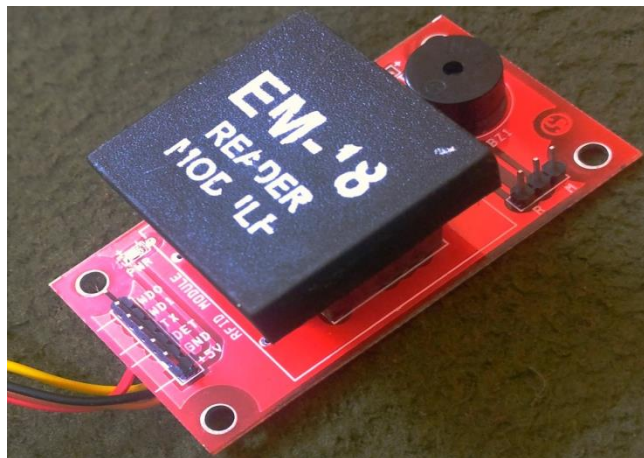


figure1.5: em-18 reader

Features

- Low cost.
- Low power.
- Form factor being small.
- Usability is easy.
- Direct interfacing with UART, PC is possible using RS232.
- Serial and TTL output.
- Excellent read performance without an external circuit.
- Two RFID cards.

- Cost-effective and compact size.

Specifications

- Read distance 10cm
- Current <50mA
- Operating frequency 125khz
- Parameter Value
- Operating Voltage 5v.

1.1.6 RFID

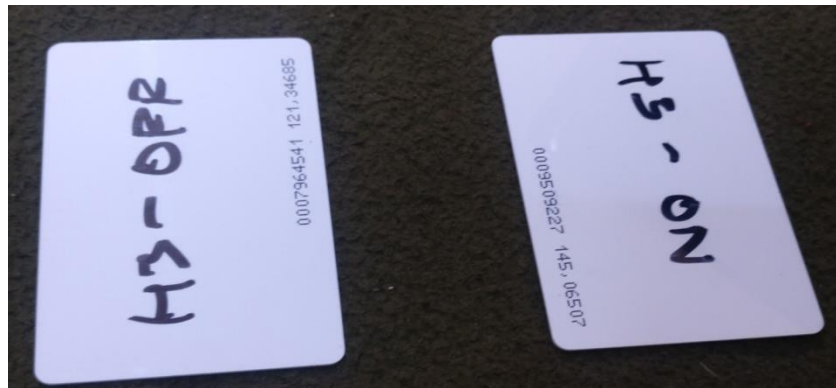


figure1.6: rfid card

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically stored information. Passive tags collect energy from a nearby RFID reader's interrogating radio waves. Active tags have a local power source (such as a battery) and may operate hundreds of meters from the RFID reader. Unlike a barcode, the tag need not be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method for Automatic Identification and Data Capture (AIDC).

A radio-frequency identification system uses tags, or labels attached to the objects to be identified. Two-way radio transmitter-receivers called interrogators or readers send a signal to the tag and read its response.

An RFID reader transmits an encoded radio signal to interrogate the tag. The RFID tag receives the message and then responds with its identification and other information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information. Since tags have individual serial numbers, the RFID system design can discriminate among several tags that might be within the range of the RFID reader and read them simultaneously.

1.1.7 LIGHT EMITTING DIODE

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p-n junction diode that emits light when activated. When a suitable current is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 mm²) and integrated optical components may be used to shape the radiation pattern.



figure1.7: led

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper and medical devices. They are also significantly more energy efficient and, arguably, have fewer environmental concerns linked to their disposal.

1.1.8 IR SENSORS

IR Sensors work by using a specific light sensor to detect a select light wavelength in the Infra-Red (IR) spectrum. By using an LED which produces light at the same wavelength as what the sensor is looking for, you can look at the intensity of the received light. When an object is close to the sensor, the light from the LED bounces off the object and into the light sensor. This results in a large jump in the intensity, which we already know can be detected using a threshold.



figure1.8: ir sensor

1.1.9 MOTOR DRIVE

A motor drive, in the field of photography, is a powered film transport mechanism. Historically, film loading, advancing, and rewinding were all manually driven functions. The desires of professional photographers for more efficient shooting, particularly in sports and wildlife photography, and the desires of amateur and novice photographers for easier to use cameras both drove the development of automatic film transport. Some early developments were made with clockwork drives, but most development in the field has been in the direction of electrically driven transport.

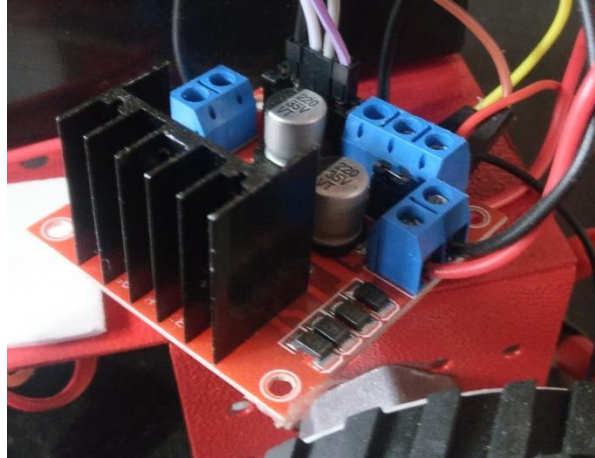


figure1.9: motor drive

1.2 TECHNIQUE FOR INNOVATION

In today's era, technology can enhance human life. Technology is evolving decade by decade. Automation was a science fiction earlier but not today. By combining latest technology with home, we can build an awesome home. With the Raspberry Pi and arduino, we can build a home automation system that is capable of operating home devices automatically.

We are doing a global home automation with Arduino and raspberry pi. The global home automation means we can control our home devices, which are connected to internet, can be access and control them by a single device like phone or a computer. Here devices such as air conditioner, refrigerator, lights, fans etc; which can be connected to a network can be made to be accessed by a phone or a computer. This is done by the help of Arduino and raspberry pi. Here we are also including an RFID cards so that we can on or off our whole homes electrical current by just one swipe. This allows us to be quick and assured that our homes devices are been shut down and we can save electricity as well as some money.

Now we can not only control a single building, but we can control all the houses in the lane and if expanded we can control a whole colony, city, country. This can be done only by raspberry pi, since it acts like a machine control unit and provides a medium for the devices to communicate with each other and effect each other's performance by the internet's help.

When we want to cut or obstruct a power supply for a home without following the existing method, we can just do it from our system. We have just block the card, which will ban the user from using the card further and is forced to pay the electricity bill immediately.

Here we are going to implement an automatic street light setup which automatically gets on in evenings and shuts itself down in the morning. This is been done by an LDR sensor which detects the light, so in day the sensor detects the light, so the street light is in off position. But as the light disappears the sensor detects no light and the streets light gets on and vice versa.

We are building a prototype of a theme park for kids so that they can access the automatic rides, with the help of an RFID card and can have some playtime.

We have an adding an automatic surveillance system which can be monitored and controlled by a single person at an office. This can be achieved by adding the cameras, which are connected to a raspberry pi.

We are implementing an automatic parking system for vehicles. This can be achieved by IR sensors placed on the vehicle and make sure that it is parked in the right place.

CHAPTER: 2

LITERATURE REVIEW

2.1 Home automation networks: A survey

Guilherme Mussi Toschi Leonardo Barreto Campos Carlos Eduardo Cugnasca

In this context, Machine-to-Machine (M2M) networks are emerging as an efficient means to provide automated communication among distributed ubiquitous devices on in a standardized manner, but none have been adopted universally. In an effort to present the technologies used in the M2M and home integration environment, this paper presents the home area network elements and definitions, and reviews the standards, architectures and initiatives created to enable M2M communication and integration in several different environments, especially at the smart home domain.

2.2 Intelligent homes' technologies to optimize the energy performance for the net zero energy home

Fadi Al Faris Adel Juaidi Francisco Manzano-Agugliaro.

The energy sector is one of the important domains to use the internet in the supply and demand sides. As the buildings consume two third of electricity globally, the real-time monitoring, by using the internet, is one of the key factors to manage the energy consumption and to optimize the buildings' performance. Accordingly, the buildings' systems have utilized the information technology's revolution in the last decade in several manners. One of the very interesting synergies is that developing relationship between the energy monitoring and the Internet of Things (IoT) for most of buildings' sensors, energy meters and building automation systems.

2.3 Smart City and IoT

Tai-hoon Kim^a Carlos Ramos^b Sabah Mohammed^c

The new Internet of Things (IoT) applications are enabling Smart City initiatives worldwide. It provides the ability to remotely monitor, manage and control devices, and to create new insights and actionable information from massive streams of real-time data. The main

features of a smart city include a high degree of information technology integration and a comprehensive application of information resources. The essential components of urban development for a smart city should include smart technology, smart industry, smart services, smart management and smart life. The Internet of Things is about installing sensors (RFID, IR, GPS, laser scanners, etc.) for everything, and connecting them to the internet through specific protocols for information exchange and communications, in order to achieve intelligent recognition, location, tracking, monitoring and management. With the technical support from IoT, smart city need to have three features of being instrumented, interconnected and intelligent

2.4 Intelligent Street Lights

Y M Jagadeesha , S Akilesa , S Karthika, Prasantha*

This newly developed concept will enable the street lights to adjust automatically based on the real time traffic conditions and change according to naturalistic condition (Full moon). This paper is concerned with the development and implementation of Low cost Sensor based Street Lights with dynamic which in turn reduces the energy consumption and CO2 emission. It consists of IR sensor, PIR sensors, low cost embedded controller and storage device. Electricity is one of the common demands for the people among the world, though it can be produced but it is not satisfying the exact demand. To overcome this, electricity during unused hours must be saved and it is very hard for the human to monitor and control.

2.5 Automatic surveillance in transportation hubs: No longer just about catching the bad guy

*SimonDenman^a ^bTristanKleinschmidt ^aDavidRyan^a ^bPaulBarnes ^aSridhaSridharan
^bClintonFookes^{ab}*

As critical infrastructure such as transportation hubs continue to grow in complexity, greater importance is placed on monitoring these facilities to ensure their secure and efficient operation. In order to achieve these goals, technology continues to evolve in response to the needs of various infrastructure. To date, however, the focus of technology for surveillance has been primarily concerned with security, and little attention has been placed on assisting operations and monitoring performance in real-time. Consequently, solutions have emerged to provide real-time measurements of queues and crowding in spaces, but have been installed as

system add-ons (rather than making better use of existing infrastructure), resulting in expensive infrastructure outlay for the owner/operator, and an overload of surveillance systems which in itself creates further complexity. Given many critical infrastructure already have camera networks installed, it is much more desirable to better utilise these networks to address operational monitoring as well as security needs.

2.6 Geometric Path Planning for Automatic Parallel Parking in Tiny Spots

Hélène Vorobieva * *Sébastien Glaser* * *Nicoleta Minoiu-Enache* * *Saïd Mammar* ***

This paper deals with path planning for car-like vehicle in parallel parking problems. The presented strategy consists in retrieving the vehicle from the parking spot and reversing the obtained path to park the vehicle. Two methods for parking in tiny spaces, where parking in one trial is not possible, are proposed. In these cases, the number of needed trials to park the vehicle can be calculated from a simple formula or from an iterative algorithm. The proposed planning methods are independent of the initial position and the orientation of the vehicle. Reference trajectories are generated so that the vehicle can park by following them. Simulations are provided for both methods. Since parking spots have become very narrow in big cities, drivers need to be experimented and very attentive when maneuvering the vehicle. This often leads to minor scratches on the car and increases traffic jam by multiple repositioning. Therefore, automatic parking is a solution to reduce stress and increase comfort and security of the driver. Here we consider parallel parking problem, which is particularly demanding for the driver.

2.7 DEFECTS AND ITS ENHANCEMENT OF THE ABOVE RESEARCH

Home automation networks: A survey

Defects:

- It is a local control.
- We cannot operate multiple devices at the same time.

Enhancement:

- We will be making it as a global control.
- We will operating multiple devices at the same time.

Intelligent homes' technologies to optimize the energy performance for the net zero energy home

Defects:

- It is a local control.

- Net zero energy home.

Enhancement:

- Making it a global control.
- Neglecting the net zero energy home.

Smart City and IoT

Defects:

- Limited up to Arduino.

Enhancement:

- Utilizing Arduino with raspberry pi.

Intelligent Street Lights

Defects:

- It mainly works on IR sensor.

Enhancement:

- Replacing the IR sensor with the LDR sensor.

Automatic surveillance in transportation hubs: No longer just about catching the bad guy

Defects:

- Not been able to be implemented to a city.

Enhancement:

- Making it as a city accessory.

Geometric Path Planning for Automatic Parallel Parking in Tiny Spots

Defects:

- It is an iterative algorithm method which is not much effective.

Enhancement:

- Using IR method.

CHAPTER: 3

DESIGN AND MODELLING

3.1 BLOCK DIAGRAM

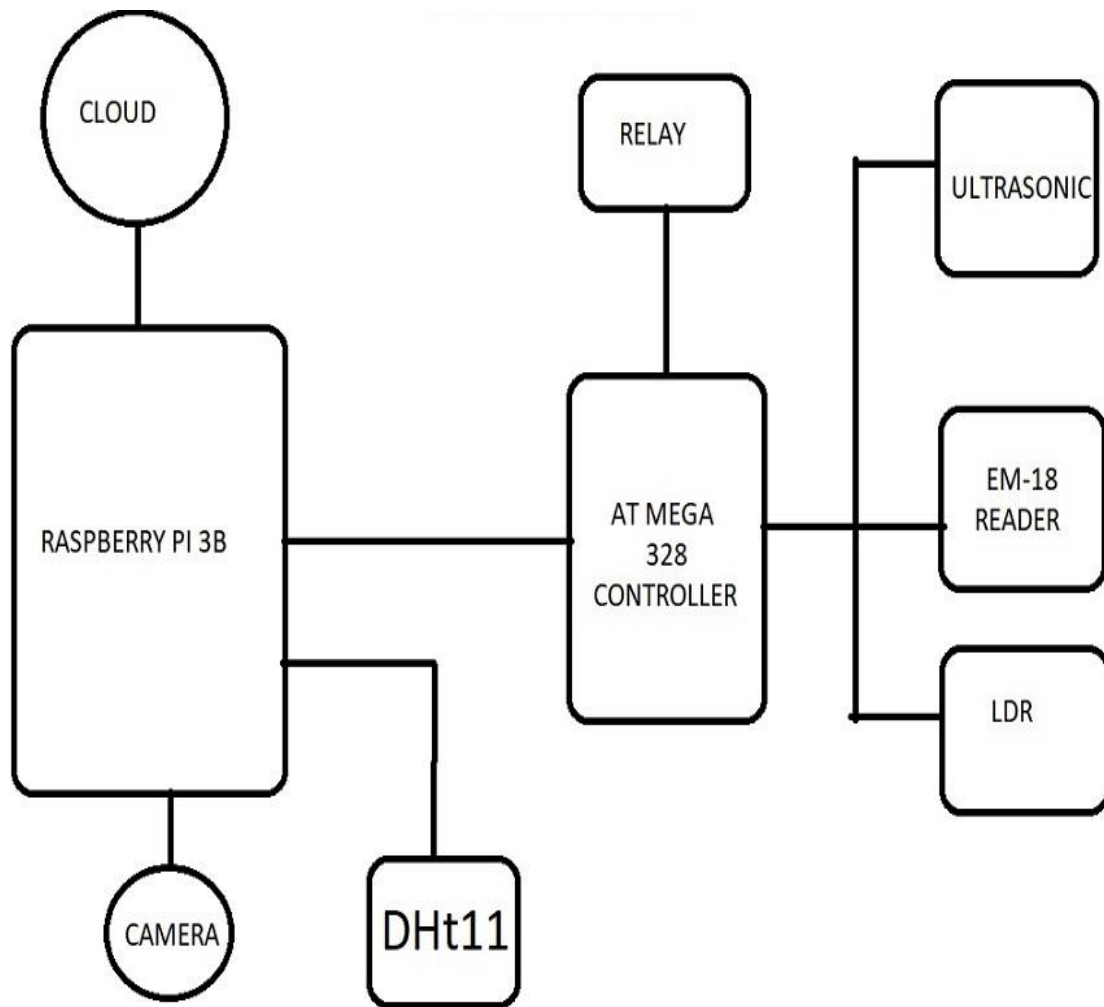


figure3.1: block diagram

3.2 GENERAL LAYOUT

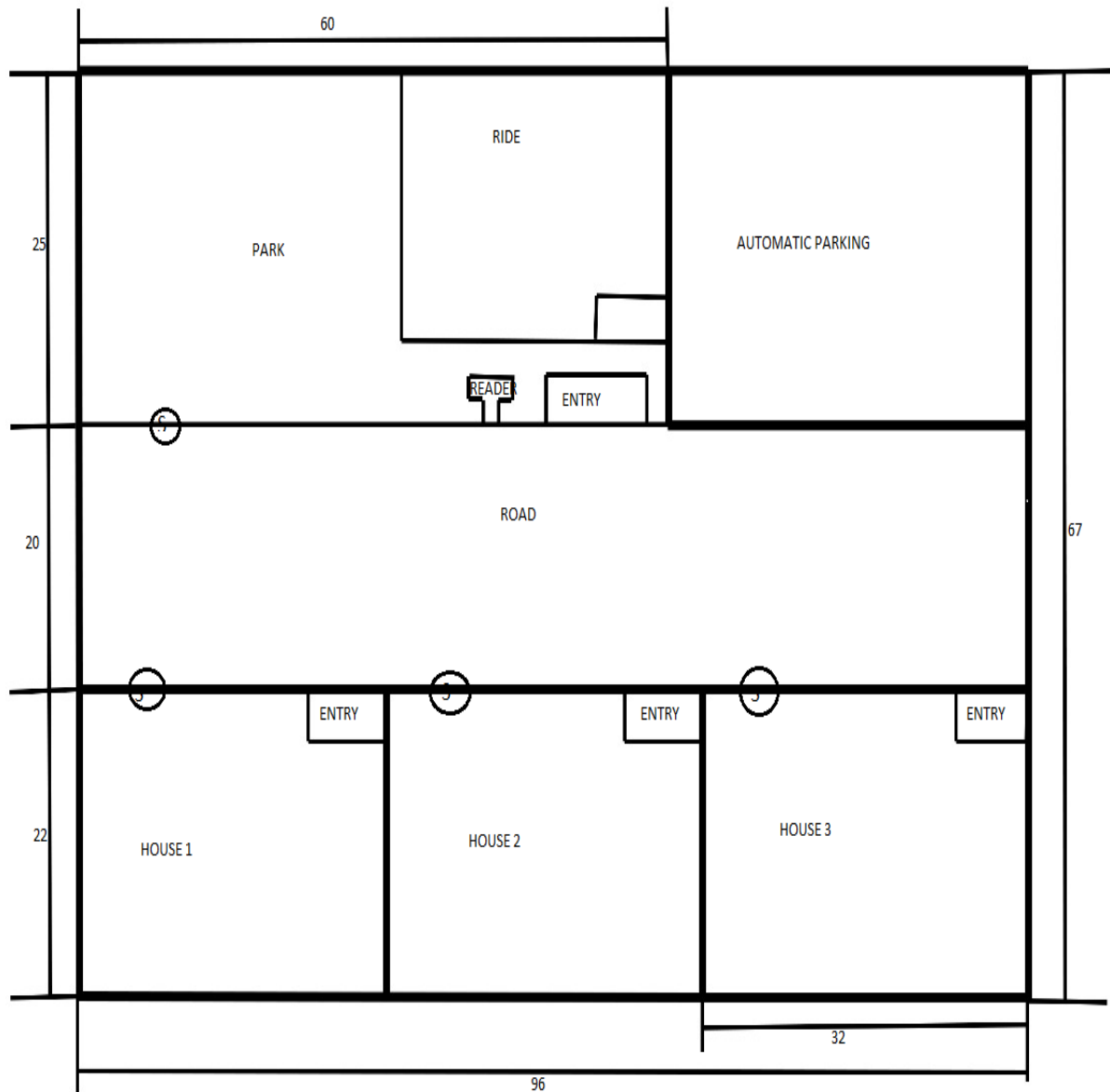


figure3.2: general layout

3.3 PRODUCT LAYOUT

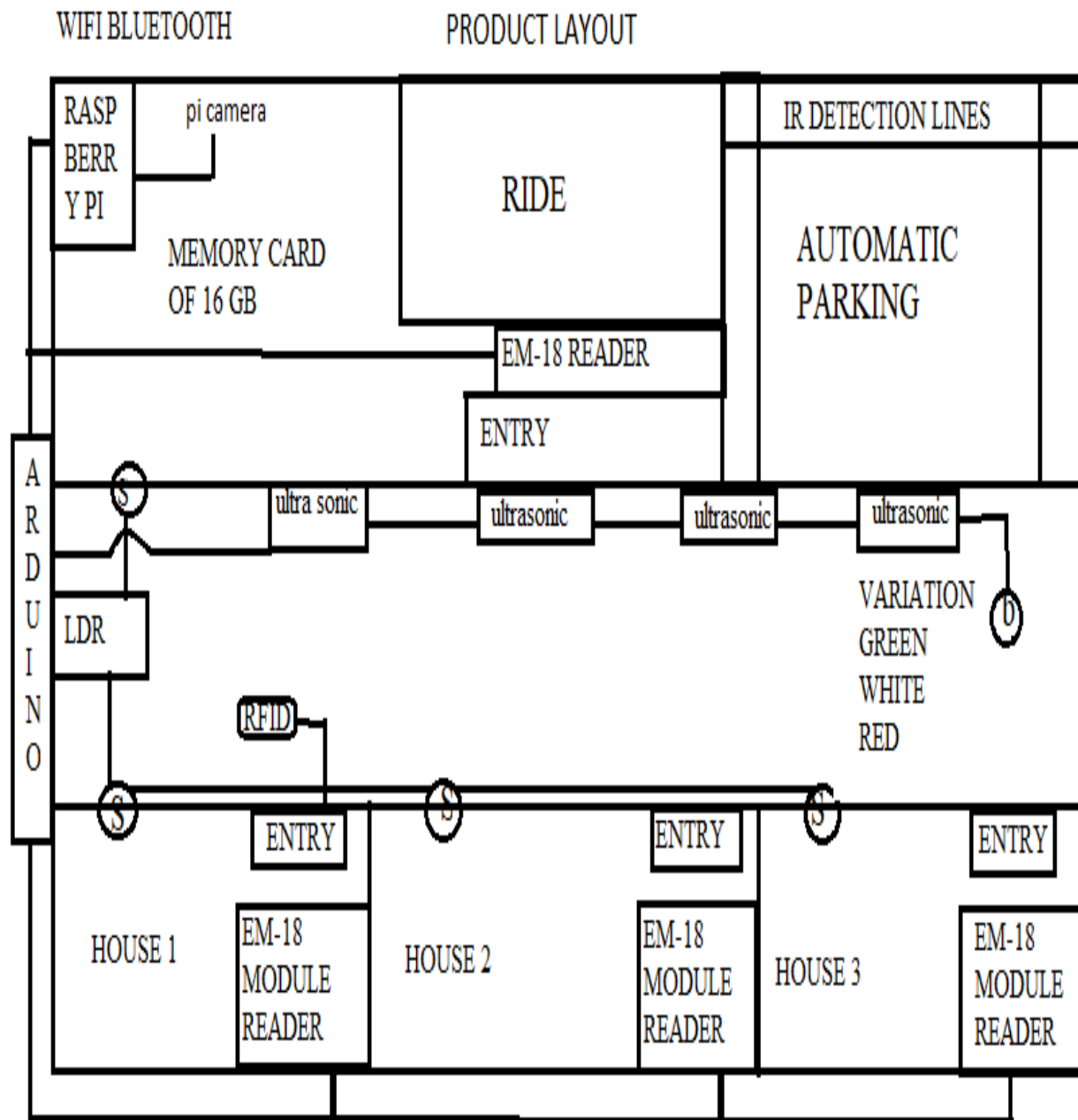


figure3.3: product layout

3.4 MODELLING

The below modelling is done in CREO software by assembling of designed individual parts.

3.4.1 Perception:1

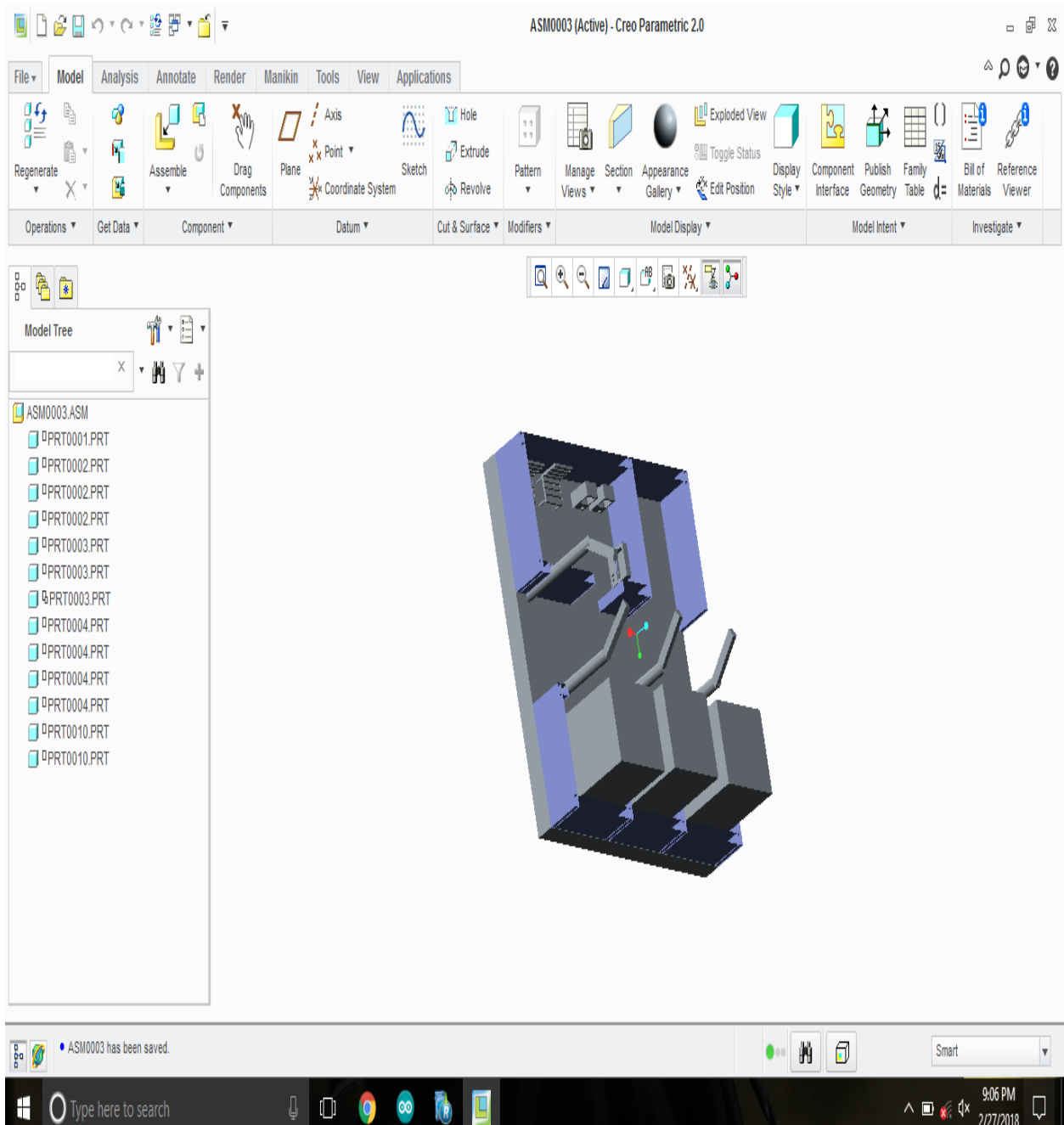


figure3.4: modelling view 1

3.4.2 Perception:2

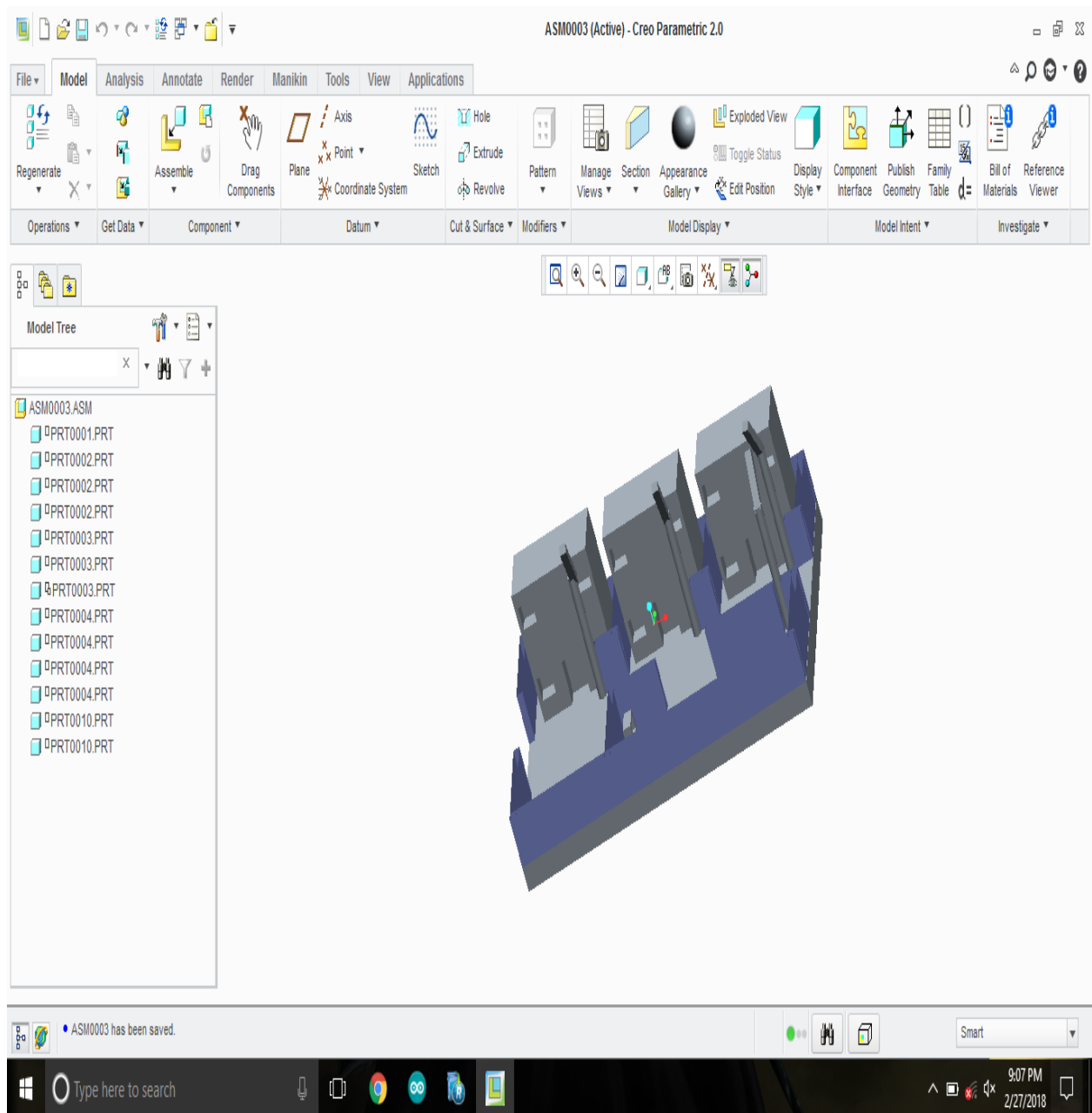


figure3.5: modelling view 2

3.4.3 Perception:3

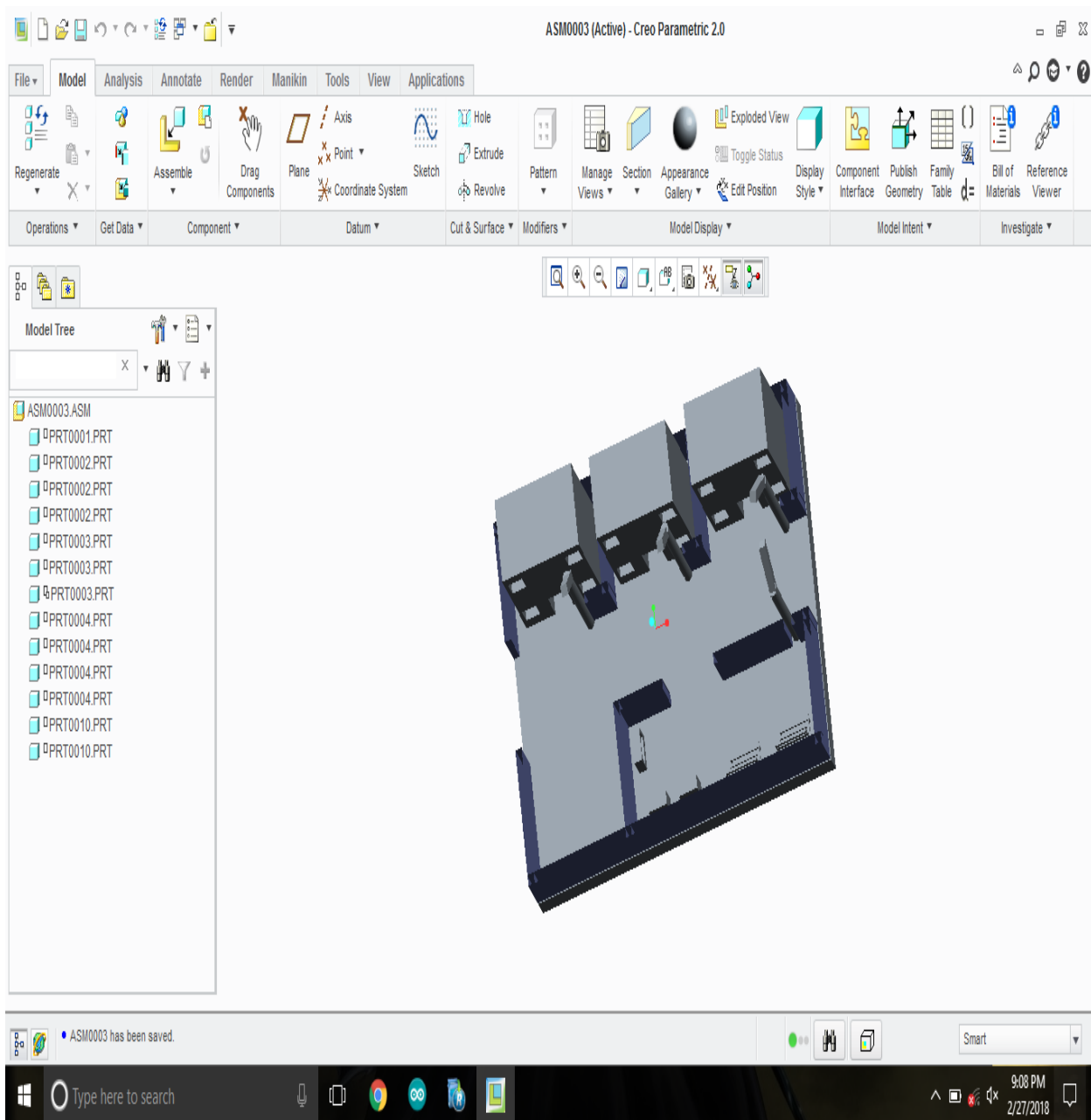


figure3.6: modelling view 3

CHAPTER: 4

HARDWARE DESCRIPTION

4.1 ARDUINO

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are :

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the microcontroller into a more accessible package.

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you

would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

4.1.1 Arduino Board Description

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

4.1.2 Technical Specifications

Microcontroller	<u>ATmega2560</u>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

table4.1: arduino technical specification

(source:www.tutorialspoint.com/arduino/arduino_overview.htm)

4.1.3 Warnings

The Mega 2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

4.1.4 Power

The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin. The input voltage to the 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 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.

4.1.5 Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

4.1.6 Input And Output

Each of the 54 digital pins on the Mega 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 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

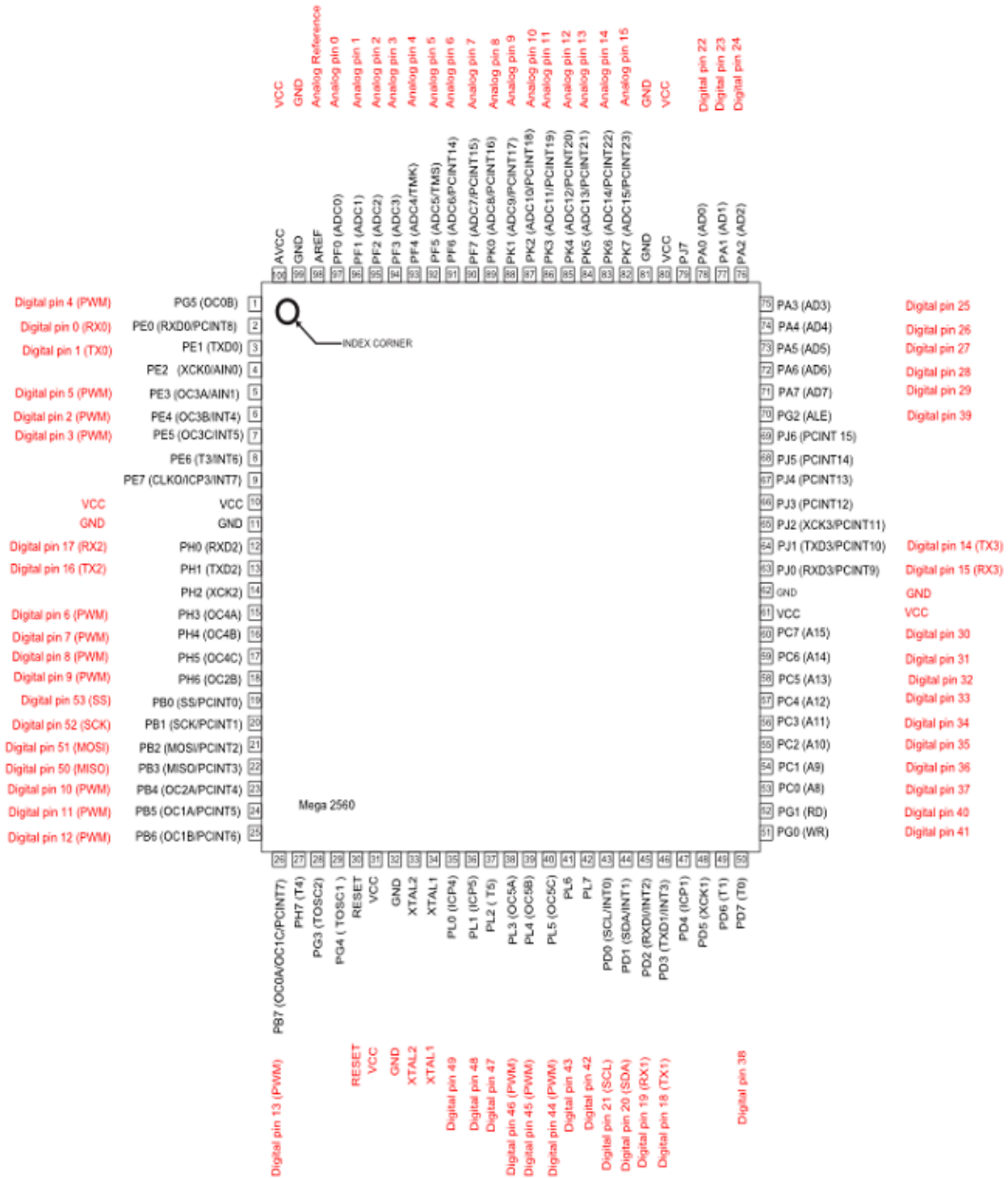


figure4.1: arduino pin description

(source: www.arduino.cc/en/hacking/pinmapping2560)

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the `attachInterrupt()` function for details.
- PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the `analogWrite()` function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.
- 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.
- TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.
- The Mega 2560 has 16 analog inputs, 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 `analogReference()` function.

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.

4.1.7 Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The Atmega2560 provides four hardware UARTs for TTL (5V) serial communication. An Atmega16U2 (Atmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the Atmega8U2/Atmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega 2560's digital pins.

The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

4.1.8 Physical Characteristics And Shield Compatibility

The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16”), not an even multiple of the 100 mil spacing of the other pins.

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

4.1.9 Arduino Mega 2560 Pin Mapping Table

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLK0/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)

23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	

48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28

73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0

98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

table4.2: pin mapping

(source:www.arduino.cc/en/hacking/pinmapping2560)

4.2 RASPBERRY PI

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.

This block diagram depicts Models A, B, A+, and B+. Model A, A+, and the Pi Zero lack the Ethernet and USB hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-point USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

4.2.1 Processor

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first modern generation smartphones (its CPU is an older ARMv6 architecture), which specifically includes a 700 MHz ARM1176JZF-S processor, VideoCoreIV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.

The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache. The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, the same SoC which is used on the Raspberry Pi 3, but under

clocked (by default) to the same 900 MHz CPU clock speed as the V1.1. The BCM2836 SoC is no longer in production (as of late 2016).

The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

4.2.2 Performance

The Raspberry Pi 3, with a quad-core ARM Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

Raspberry Pi 2 V1.1 included a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It was described as 4–6 times more powerful than its predecessor. The GPU was identical to the original. In parallelized benchmarks, the Raspberry Pi 2 V1.1 could be up to 14 times faster than a Raspberry Pi 1 Model B+.

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the performance of the Xbox of 2001.

The LINPACK single node compute benchmark results in a mean single precision performance of 0.065 GFLOPS and a mean double precision performance of 0.041 GFLOPS for one Raspberry Pi Model-B board. A cluster of 64 Raspberry Pi Model B computers, labelled "Iridis-pi", achieved a LINPACK HPL suite result of 1.14 GFLOPS (n=10240) at 216 watts for c. US\$4000.

4.2.3 Ram

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB

was for Linux only, with only a 1080p [framebuffer](#), and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC).¹ Comparatively the [Nokia 701](#) uses 128 MB for the Broadcom VideoCore IV.

For the later Model B with 512 MB RAM initially there were new standard memory split files released(`arm256_start.elf`, `arm384_start.elf`, `arm496_start.elf`) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). But a week or so later the RPF released a new version of `start.elf` that could read a new entry in `config.txt` (`gpu_mem=xx`) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single `start.elf` worked the same for 256 and 512 MB Raspberry Pis.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

4.2.4 Networking

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the Model B and B+, the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on Broadcom BCM43438 FullMAC chip with no official support for Monitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port. The Raspberry Pi 3B+ features dual-band IEEE 802.11b/g/n/ac WiFi, Bluetooth 4.2, and Gigabit Ethernet(limited to approximately 300 Mbit/s by the USB 2.0 bus between it and the SoC).

4.2.5 Peripherals

The Model 2B boards incorporate four USB ports for connecting peripherals.

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse . It may also be used with USB storage, USB to MIDI converters, and virtually any other device/component with USB capabilities.

Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.

4.2.5.1 General purpose input-output (GPIO) connector

Raspberry Pi 1 Models A+ and B+, Pi 2 Model B, Pi 3 Model B and B+, and Pi Zero and Zero W GPIO J8 have a 40-pin pinout. Raspberry Pi 1 Models A and B have only the first 26 pins.

GPIO#	2nd func.	Pin#	Pin#	2nd func.	GPIO#
	+3.3 V	1	2	+5 V	
2	SDA1 (I ² C)	3	4	+5 V	
3	SCL1 (I ² C)	5	6	GND	
4	GCLK	7	8	TXD0 (UART)	14
	GND	9	10	RXD0 (UART)	15
17	GEN0	11	12	GEN1	18
27	GEN2	13	14	GND	
22	GEN3	15	16	GEN4	23
	+3.3 V	17	18	GEN5	24
10	MOSI (SPI)	19	20	GND	
9	MISO (SPI)	21	22	GEN6	25
11	SCLK (SPI)	23	24	CE0_N (SPI)	8
	GND	25	26	CE1_N (SPI)	7

<i>(Pi 1 Models A and B stop here)</i>					
EEPROM	ID_SD	27	28	ID_SC	EEPROM
5	N/A	29	30	GND	
6	N/A	31	32		12
13	N/A	33	34	GND	
19	N/A	35	36	N/A	16
26	N/A	37	38	Digital IN	20
	GND	39	40	Digital OUT	21

table4.3: raspberry pi pin description

(source:www.raspberrypi.org/documentation/usage/gpio)

Models A and B provide GPIO access to the ACT status LED using GPIO 16. Models A+ and B+ provide GPIO access to the ACT status LED using GPIO 47, and the power status LED using GPIO 35.

4.3 IR SENSOR

Infrared Obstacle Sensor Module has builtin IR transmitter and IR receiver that sends out IR energy and looks for reflected IR energy to detect presence of any obstacle in front of the sensor module. The module has on board potentiometer that lets user adjust detection range. The sensor has very good and stable response even in ambient light or in complete darkness.

An IR sensor consists of an IR LED and an IR Photodiode; together they are called as Photo-Coupler or Opto-Coupler. As said before, the Infrared Obstacle Sensor has builtin IR transmitter and IR receiver. Infrared Transmitter is a light emitting diode (LED) which emits infrared radiations. Hence, they are called IR LED's. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye. Infrared receivers are also called as infrared sensors as they detect the radiation from an IR transmitter. IR receivers come

in the form of photodiodes and phototransistors. Infrared Photodiodes are different from normal photo diodes as they detect only infrared radiation. When the IR transmitter emits radiation, it reaches the object and some of the radiation reflects back to the IR receiver. Based on the intensity of the reception by the IR receiver, the output of the sensor is defined.

Hardware and Software Required

- IR Obstacle Sensor Module
- Arduino Uno
- Arduino IDE(1.0.6V)

Hardware Connections

- Vcc to 5V
- Gnd to Gnd
- Out to digital pin 7

4.4 EM 18 RFID READER MODULE

EM18 RFID Reader Module working at 125KHz suitable for Arduino, AVR, Raspberry Pi, 8051. Positron's EM-18 RFID Reader operated at 125kHz frequency and has a read range of 5cm-10cm. It works at 5V DC. It is an inexpensive solution for your RFID based application. The TTL output pins can be connected to interface with RFID Reader with any microcontroller development board such as Arduino, The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show your card within the reading distance and the card number is thrown at the output.

4.5 ULTRASONIC SENSORS

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1" to 13 feet. Its operation is not affected by sunlight or black material like sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

Features

Power Supply :+5V DC

Quiescent Current : <2mA

Working Current: 15mA

Effectual Angle: <15°

Ranging Distance : 2cm – 400 cm/1" – 13ft

Resolution : 0.3 cm

Measuring Angle: 30 degree

Trigger Input Pulse width: 10uS

Dimension: 45mm x 20mm x 15mm

The ultrasonic sensor uses sonar to determine the distance to an object. Here's what happens:

- The transmitter (trig pin) sends a signal: a high-frequency sound
- When the signal finds an object, it is reflected and
- The transmitter (echo pin) receives it.

CHAPTER: 5

SOFTWARE DESCRIPTION

5.1 ARDUINO IDE

A program for Arduino may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension *.ino*. Arduino Software (IDE) pre-1.0 saved sketches with the extension *.pde*.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

5.1.1 Installation and use

Step 1 – First you must have your Arduino board (you can choose your favorite board) and a USB cable. You use Arduino Mega 2560, you will need a standard USB cable (A plug to B plug), and the kind you would connect to a USB printer.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.



figure5.1: arduino cable

Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

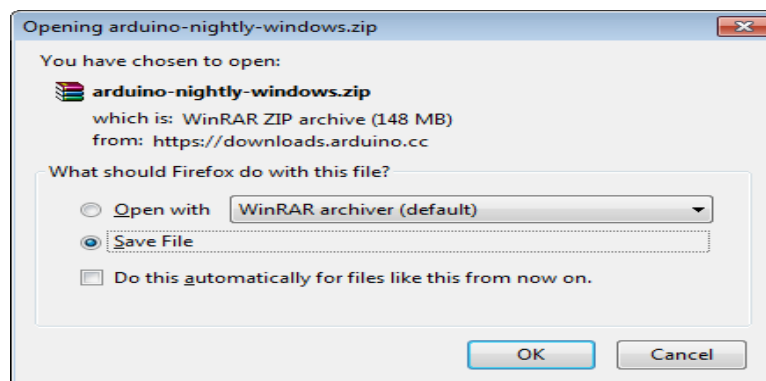


figure5.2: arduino zip file

(source:www.tutorialspoint.com/ardunio/index.htm)

Step 3 – Power up your board.

The Arduino Mega and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

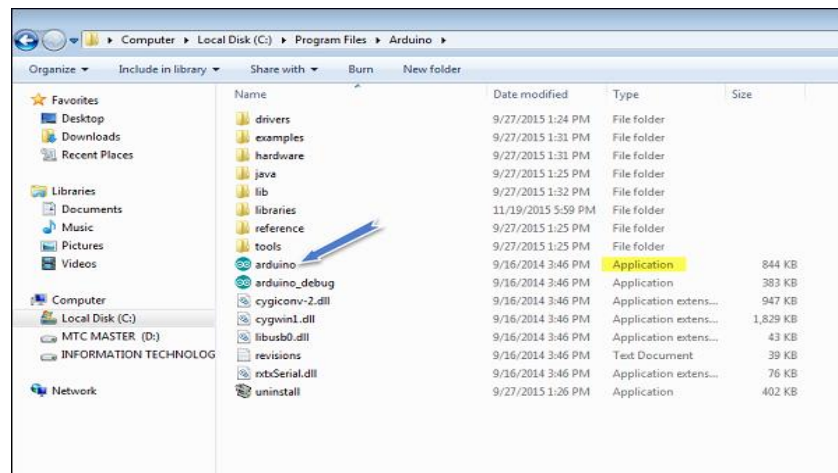


figure5.3: arduino launch dialog box

(source:www.tutorialspoint.com/ardunio/index.htm)

Step 5 – Open your first project.

Once the software starts, you have two options –

Create a new project.

Open an existing project example.

To create a new project, select File → New.

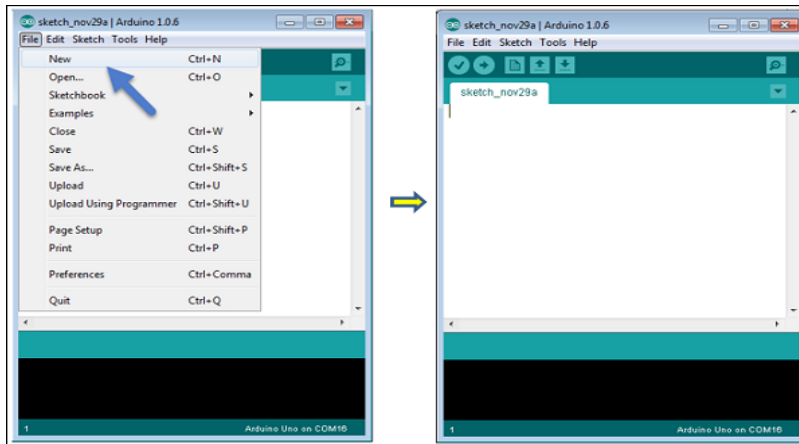


figure5.4: arduino sketch

(source: www.tutorialspoint.com/ardunio/index.htm)

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 – Select your Arduino board.

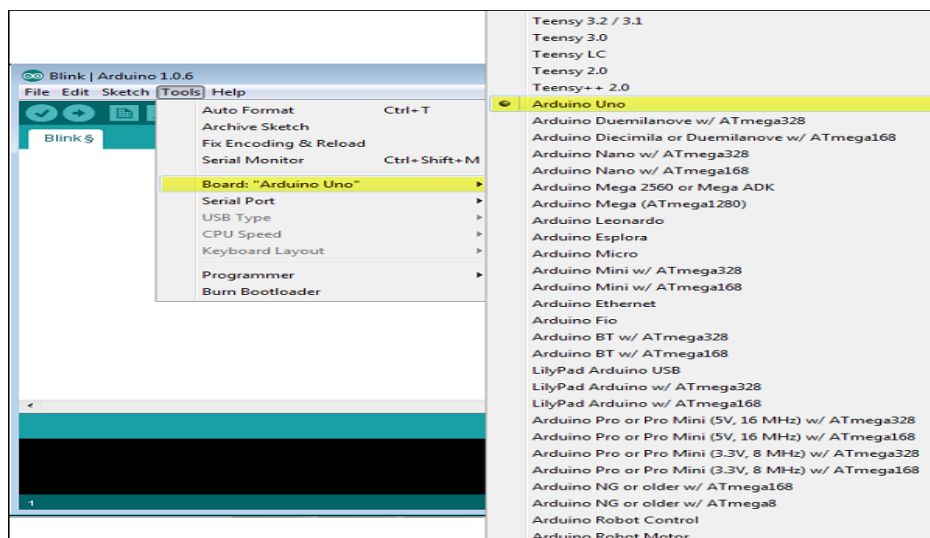


figure5.5: arduino board selection

(source: www.tutorialspoint.com/ardunio/index.htm)

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 – Select your serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

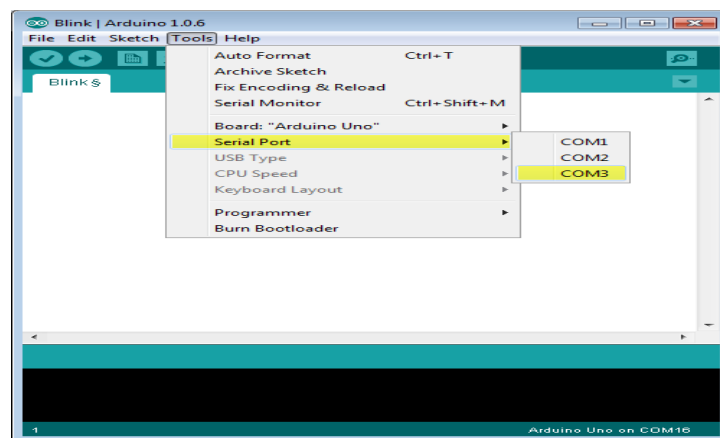


figure5.6: serial port selection

(source:www.tutorialspoint.com/ardunio/index.htm)

Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

A – Used to check if there is any compilation error.

B – Used to upload a program to the Arduino board.

C – Shortcut used to create a new sketch.

D – Used to directly open one of the example sketch.

E – Used to save your sketch.

F – Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

5.1.2 Arduino - Program Structure

In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch – The first new terminology is the Arduino program called “sketch”.

5.1.2.1 Structure

Arduino programs can be divided in three main parts: Structure, Values(variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions –

Setup() function

Loop() function

```
Void setup ( ) {
```

```
}
```

PURPOSE – The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

```
Void Loop ( ) {
```

```
}
```

PURPOSE – After creating a `setup()` function, which initializes and sets the initial values, the `loop()` function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

5.2 RASPBERRY PI

5.2.1 Installation Of Raspbian Os In Raspberry Pi With The Help Of The Noobs.

Downloading NOOBS

Using NOOBS is the easiest way to install Raspbian on your SD card. To get hold of a copy of NOOBS:

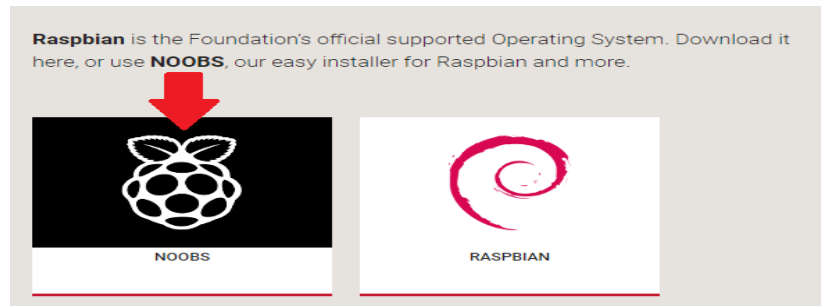


figure5.7: noobs

(source:www.raspberrypi.org/documentation)

5.2.2 Extracting NOOBS from the zip archive

Next, you will need to extract the files from the NOOBS zip archive you downloaded from the Raspberry Pi website. Go to your Downloads folder and find the zip file you downloaded. Extract the files and keep the resulting Explorer/Finder window open.

5.2.3 Copying the files

Now open another Explorer/Finder window and navigate to the SD card. It's best to position the two windows side by side. Select all the files from the NOOBS folder and drag them onto the SD card.

5.2.4 Booting from NOOBS

Once the files have been copied over, insert the micro SD Card into your Raspberry Pi, and plug the Pi into a power source. You will be offered a choice when the installer has loaded. You should check the box for Raspbian, and then click Install.

Click **Yes**, Raspbian will install.

5.2.5 Programming Pi

Now click on start icon and get into programming >Python 2 / Python 3

click upon your desired python IDE. use python 3.

when you open the python ide you will get the Python shell window. now to create new file go to File>New Window

a new python window will be in front of you. write down your code here.

After writing the code click on RUN option. RUN>RUN module

You will be prompted to save then file first. after saving it will be executed by ide and you can see the output in Python shell window. depending on your code you can control the GPIO pins of Raspberry pi, excess the internet for IoT application and much more.

As you can see here my code is actually creating the GUI using Python.

Now you can run this script by clicking on it or may just enter the command in terminal.

You have programmed the Raspberry pi successfully. Explore the world with giant processing power of ARM cpu with simplicity of pin controlling with GPIOs.

5.2.6 Raspberry Pi Camera Setup

Open up your Raspberry Pi Camera module. Be aware that the camera can be damaged by static electricity. Before removing the camera from its grey anti-static bag, make sure you have discharged yourself by touching an earthed object (e.g. a radiator or PC Chassis).

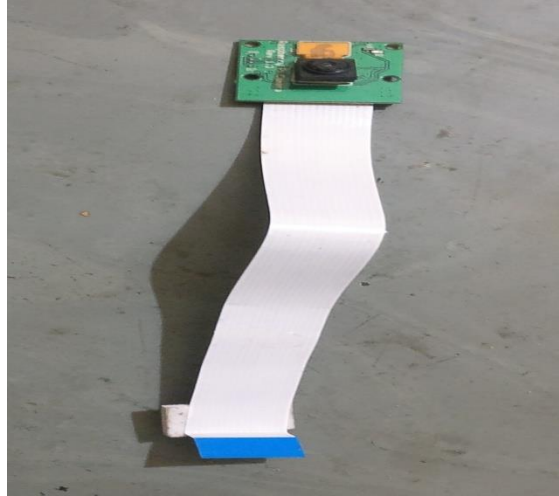


figure5.8: pi camera

5.2.6.1 Photo With Your Raspberry Pi Camera Module

1. "raspistill" is a command line application that allows you to capture images with your camera module. Below is an example of this command in use.

Record A Video With Your Raspberry Pi Camera Module

"raspivid" is a command line application that allows you to capture video with your camera module. Below is an example of this command in use.

2. To capture a 10 second video with your Raspberry Pi camera module, run "raspivid -o video.h264 -t 10000" at the prompt, where "video" is the name of your video and "10000" is the number of milliseconds

CHAPTER: 6

SMART CITY ANALYSIS

6.1 ARDUINO CONNECTIONS

The Arduino Mega been used here has three types of pins or ports, digital, analog and communication. So in this project to achieve the automation steps we use only the digital and communication pins, and no need of the analog, since we are having no readers or meters etc.;

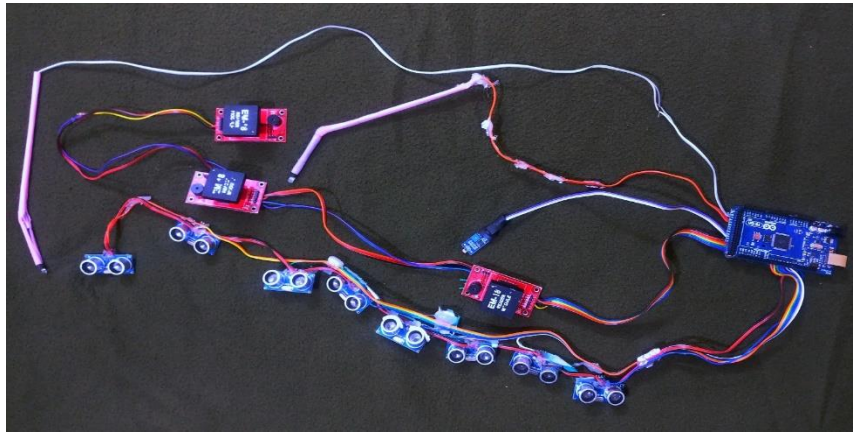


figure6.1: arduino connections

We have connected the parts to the Arduino according to the inputs and the outputs given by them. The ultrasonic sensor, which are in quantity of eight, are connected to the communication sector, since it gives the echo distance to the board in form of the voltage. It communicates with the board by giving the signal whenever triggered from the sensor and received an echo.

The digital side has connections of input and output giving things. A led is connected to this side as to show the ultrasonic sensor signal variations. That means it glows in different color variations in order to show the no. of ultrasonic receiving echo. This color variations are green means no traffic i.e.; no echo been received by the ultrasonic sensors, purple color means there are at least three to four ultrasonic sensors are working, the blue means there are at least five to six ultrasonic sensors are working. At last, the red color means all the sensors are in working. This implies a heavy traffic condition. The Arduino receives the distance value from the

ultrasonic sensors and based on the no. of ultrasonic sensors being active the board gives the RGB color intensities to the led, which in order glows accordingly.

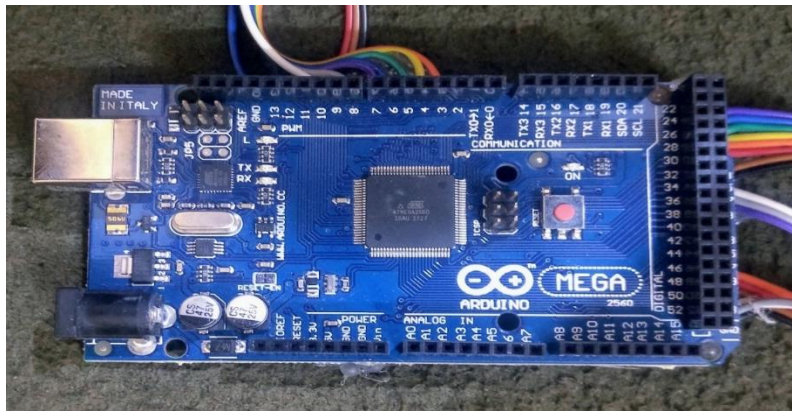


figure6.2 : arduino port connections

An LDR or photo resistor is made any semiconductor material with a high resistance. It has a high resistance because there are very few electrons that are free and able to move - the vast majority of the electrons are locked into the crystal lattice and unable to move. Therefore, in this state there is a high LDR resistance. As light falls on the semiconductor, the light photons are absorbed by the semiconductor lattice and some of their energy is transferred to the electrons. This gives some of them sufficient energy to break free from the crystal lattice so that they can then conduct electricity. This results in a lowering of the resistance of the semiconductor and hence the overall LDR resistance. The process is progressive, and as more light shines on the LDR semiconductor, so more electrons are released to conduct electricity and the resistance falls further. Now because of the resistance variations the current conduction varies, all the values of the LDR are taken as an input to the board and as the resistance reaches, its maximum limit the corresponding LED attached to the board on to the digital side glows.

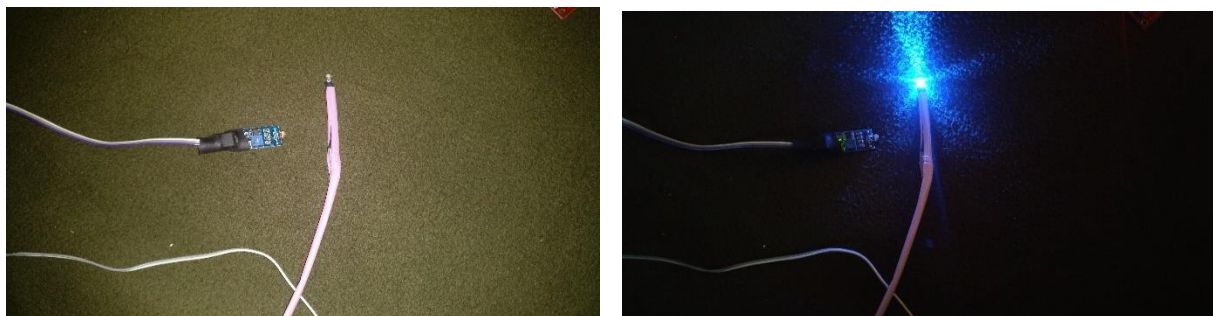


figure6.3: showing the working of ldr sensor

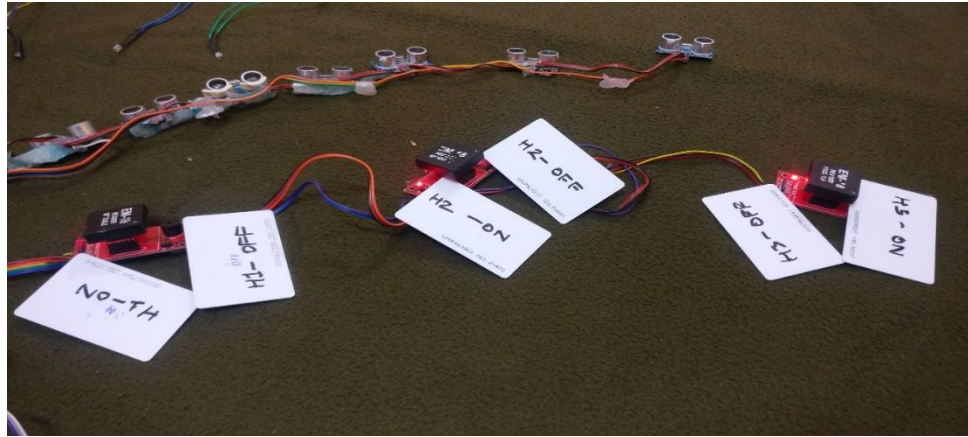


figure6.4: showing the rfid cards with their corresponding modules

Working of EM-18 RFID module: The module radiates 125 KHz through its coils and when a 125 KHz passive RFID tag is brought into this field, it will be energized from this field. These passive RFID tags mostly consist of CMOS IC EM4102, which can get enough power for its working from the field generated by the reader.

When a RFID Tag is bring in to the field of EM-18 RFID Reader, it will read its tag number and give output via TX terminal. The BEEP terminal will become LOW to indicate valid tag detection. The UART output will be 12 ASCII data, among these first 10 will be tag number and last 2 will be XOR result of the tag number which can be used for error testing.

As the card been read by the module, it gives the respective card voltage to the board and according to the received value; the board activates or deactivates the LED attached to the module accordingly. Here we are using three module with each having their own ON and OFF RFID cards. So whenever the card of the respected module is been detected the respective led which is been connected to the Raspberry PI is been activated accordingly. If ON card is been detected the LED of the respective house glows and if the OFF card is been detected the LED of the respective house is turned off.

6.2 RASPBERRY PI CONNECTIONS

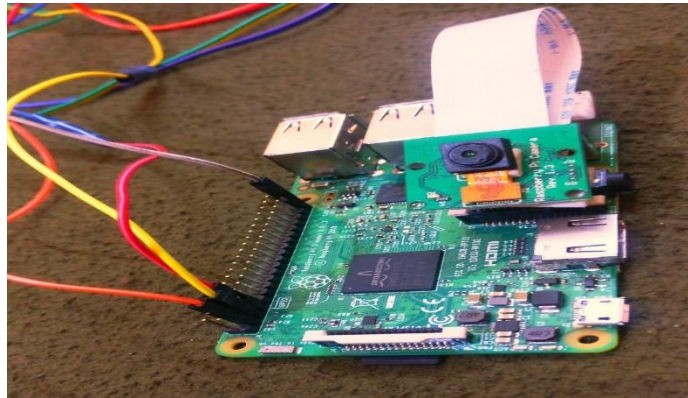


figure6.5: showing the connections of raspberry pi

Here to the Raspberry pi there are two things been connected, camera and the LEDs of the houses. The LEDs are connected to Raspberry pi in order to achieve the global automation. As the LEDs are connected to the board, they have indirectly established a connection to the internet. The Raspberry pi is been connected to the Arduino, the Arduino is programmed in such a way that the EM-18 module reader values are been given to the Raspberry pi and it is programmed in such a way that the reader value is uploaded to the internet and then with a lag of 5 seconds the LED is turned on. This is the normal manual procedure nothing to do with the internet; everything is done by the two broads.

Now, we made an application, which can be used to control the LEDs attached to the Raspberry pi.

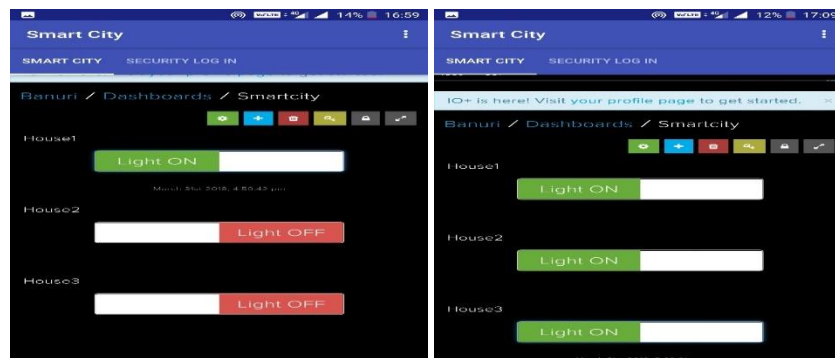


figure6.6: showing the screenshot of smart-city application

STEP1- At first we have to make sure that the Raspberry pi gets an internet connection through hotspot or a LAN cable.

STEP2- Open the application, make sure that the net is on, and then try to ON and OFF the respective lights of the houses.

Now, the above case the Raspberry pi receives the input given by us through the application, then runs the program accordingly in its stored memory, and then turns the LED on of the respective house. This is updated to the Arduino so that the reader is in same phase as the application.

PI CAMERA is been connected to the Raspberry pi can be accessed by the following steps:

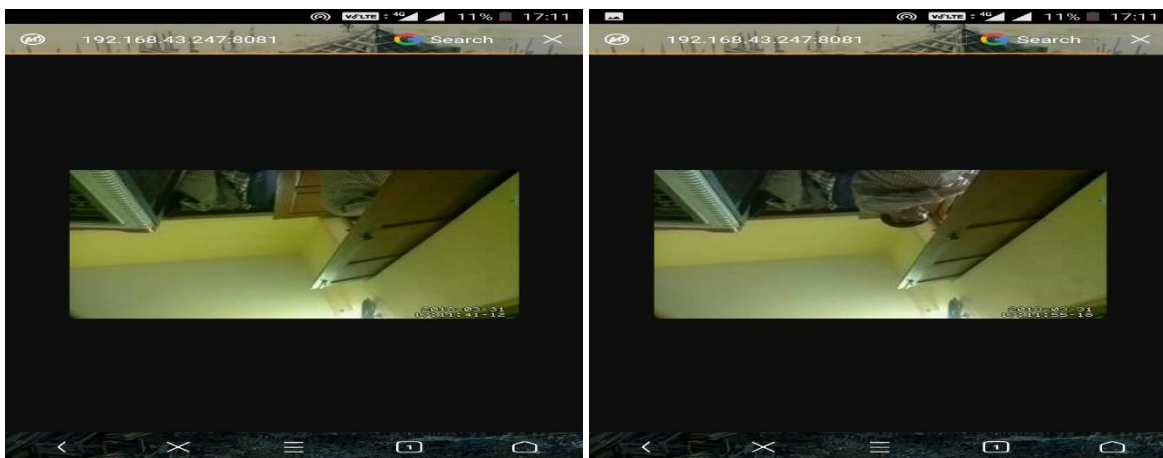


figure6.7: showing the working of camera surveillance

STEP1- make sure that the Raspberry pi is connected to the same hotspot or Wi-Fi as your phone or the pc.

STEP2- now type the following in the URL search, 192.168.43.247:8081. This would give us the camera feed live with the date and time. We can store the feed to any storage medium.

6.3 AUTOMATIC PARKING CAR

In this, we made a car, which has two motors of 45 rpm to turn the wheels. There is a motor drive connecting both the motors. This motor drive is used to control and change the speed accordingly. The motor drive is connected to the Arduino Nano, which is programmed to control

the car according to the INPUT given by the IR sensors. This IR sensors detect only the black lines i.e.; it is activated only when it senses the black color within its range. IR sensor whenever it senses the black color in its path it makes the Arduino board to get car running. When only one sensor detects the black line, the respective wheel rotates while the other is not rotating. This would cause the car to make a turn accordingly. This car is equipped with an ultrasonic sensor, which would stop the car if it senses any obstacle in its path. This is simply an obstacle detector here. When the black line finishes the both sensors stops working which ultimately stops the car, this would be our parking space.

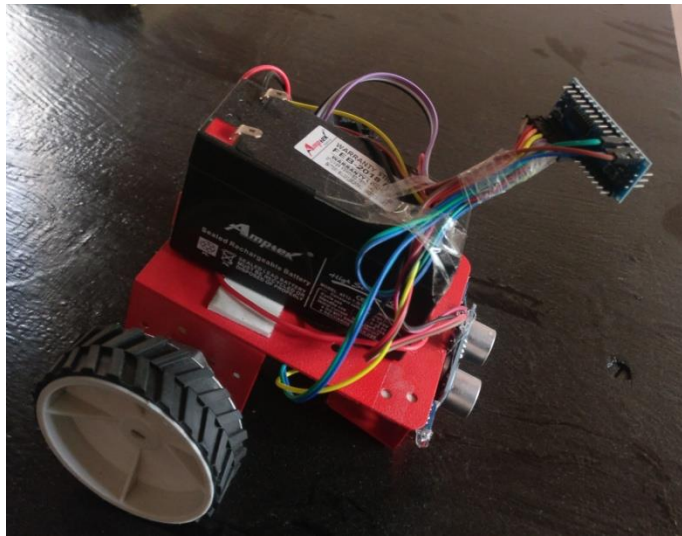


figure6.8: line follower

CHAPTER: 7

RESULT

The local and global automation of our prototype is been achieved by us using the complex parts like Raspberry pi and Arduino. The programming and setting up of the modules and the boards is done and application is been made in order to access the LEDs by internet. A sort of web link is also created which does the work if application fails. The RFID cards can be used for the ON and OFF the house lights by swiping them on the EM-18 module reader. The automatic parking is been done by using IRsensors, by placing them on a car. The traffic density is been determined by the ultrasonic sensors connected to the Arduino and the LED attached shows the density if traffic a head this is shown by RBG LED light by four-color variations for different traffic intensities. The automatic street light is been developed by using an LDr sensor, which is activated only in the nighttime, this would provide us with effective utilization of the power and time. A surveillance camera is been connected to the Raspberry pi, which gives us a video feed along with time and date. This video can be stored in a hard disk or can be uploaded to the internet.

CHAPTER: 8

FUTURE SCOPE AND CONCLUTION

Here in this global automation, we can develop the application to measure the units been consumed by the user per day/month. We can also upload our camera feed live to the channels of our choice. We can develop our setup to an extent of a load of a complete city automation. Taking the GOOGLE MAPS technique, we can enhance the automatic parking system. We can make an adjustment in our application in order to cut the current connection we can simply disable the RFID cards so that the user will be forced to pay his/her dues. We can maintain an count of devices running and not running, we can also get the number of units consumed by an single device and its working hours.

CHAPTER- 9

REFERENCE

1. "Arduino Nano". www.arduino.cc. Retrieved 20 February 2018.
2. "LilyPad Arduino Main Board". www.arduino.cc. Retrieved 20 February 2018.
3. "Arduino Robot". www.arduino.cc. Retrieved 20 February 2018.
4. "Arduino Esplora". www.arduino.cc. Retrieved 20 February 2018.
5. "Arduino Ethernet Rev3 without PoE". www.arduino.cc. Retrieved 20 February 2018.
6. "Arduino Yún". www.arduino.cc. Retrieved 20 February 2018.
7. "Arduino Due". www.arduino.cc. Retrieved 20 February 2018.
8. "Arduino - ArduinoShields". www.arduino.cc. Retrieved 2017-10-04.
9. "Arduino breadboard shield: \$10 & 10 mins". [todbot blog](http://todbot.com/blog/).
10. "Arduino Shields for Prototyping". tigoe.net.
11. Jonathan Oser. "Arduino Shield list". Retrieved 5 Nov 2013.
12. "Arduino Software Release Notes". Arduino Project. Retrieved April 9, 2017.
13. "Using Atmel Studio for Arduino development". Megunolink.com. Archived from the original on 2013-01-28. Retrieved 2013-01-18.
14. "Using AVR Studio for Arduino development". Engblaze.com. Retrieved 2013-01-18.
15. "Ch Arduino". Retrieved 2016-10-07.
16. "Programming Arduino Getting Started with Sketches". McGraw-Hill. Nov 8, 2011. Retrieved 2013-03-28.
17. "Arduino - BareMinimum". www.arduino.cc. Retrieved 20 February 2018.
18. Documentation for function setup (arduino.cc)
19. Documentation for function loop (arduino.cc)
20. "Blink Tutorial". Arduino.cc.
21. Documentation for function pinMode (arduino.cc)
22. Documentation for function digitalWrite (arduino.cc)
23. Documentation for function delay (arduino.cc)
24. <https://www.tutorialspoint.com/arduino/index.htm>
25. "Windows 10 for IoT". Raspberry Pi Foundation. 30 April 2015.
26. "Raspberry Pi 3 is out now! Specs, Benchmarks & More". The MagPi Magazine. 1 April 2016.
27. "Raspberry Pi FAQs – Frequently Asked Questions". Raspberry Pi. Retrieved 8 April 2017.
28. Cellan-Jones, Rory (5 May 2011). "A£15 computer to inspire young programmers". BBC News.
29. Price, Peter (3 June 2011). "Can a £15 computer solve the programming gap?". BBC Click. Retrieved 2 July 2011.
30. Bush, Steve (25 May 2011). "Dongle computer lets kids discover programming on a TV". Electronics Weekly. Retrieved 11 July 2011.

31. "Ten millionth Raspberry Pi, and a new kit – Raspberry Pi". 8 September 2016. Retrieved 9 September 2016. we've beaten our wildest dreams by three orders of magnitude
32. Gibbs, Samuel (18 February 2015). "Raspberry Pi becomes best selling British computer". The Guardian. Retrieved 28 December 2016.
33. The Mag Pi Magazine issue 53, Page 10, Raspberry Pi Foundation, On 25 November, it was confirmed that total sales of the Raspberry Pi have now topped 11 million.
34. "Ten millionth Raspberry Pi, and a new kit – Raspberry Pi". Raspberry Pi. 8 September 2016. Retrieved 1 February 2017.
35. <https://www.sciencedirect.com/science/article/abs/pii/S0920548916300654>
36. <https://www.sciencedirect.com/science/article/pii/S0378778817309477>
37. <https://www.sciencedirect.com/science/article/pii/S0167739X17305253>
38. <https://www.sciencedirect.com/science/article/pii/S2212017315002789>
39. <https://www.sciencedirect.com/science/article/pii/S0957417415005370>
40. <https://www.sciencedirect.com/science/article/pii/S1474667015347431>
41. Documentation for function pinMode (arduino.cc)
42. Documentation for function digitalWrite (arduino.cc)
43. Documentation for function delay (arduino.cc)
44. "Built a Motion Control Rig for Time-Lapse Photography - Make:". Retrieved 20 February 2018.
45. ArduinoPhone. Instructables.com (2013-07-17). Retrieved on 2013-08-04.
46. DIY Cellphone, MIT
47. Grossi, Marco; Riccò, Bruno (February 2017). "An automatic titration system for oil concentration measurement in metalworking fluids". Measurement. 97: 8–14. doi:10.1016/j.measurement.2016.11.014.
48. "MagPi, issue 40, Raspberry Pi Zero release article" (PDF). Raspberry Pi Foundation. 26 November 2015. Retrieved 26 November 2015.
49. "Raspberry Pi USB Serial Connection and power supply". Elinux.org. Retrieved 2 October 2015.
50. <http://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-B-Plus-V1.2-Schematics.pdf>
51. https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/RPI-3B-V1_2-SCHEMATIC-REDUCED.pdf
52. <https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-Rev-1.0-Model-AB-Schematics.pdf>
53. <https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-Rev-2.0-Model-AB-Schematics.pdf>
54. <https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-Rev-2.1-Model-AB-Schematics.pdf>
55. <http://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-Rev-2.1-Model-AB-Schematics.pdf>