IOT ROBOT

MAJOR PROJECT REPORT

By

ANAS

Department of Computer Engineering

April 2018

ABSTRACT

The purpose of this project is to control robot with an interface board of Arduino and the Raspberry Pi, sensors and software to full fill real time requirement. Controlling DC motors, different sensors, camera interfacing with raspberry Pi using USP. Live streaming, Command the robot easily, sends data of different sensors which works automatically or control from anywhere at any time. Design of the website and control page of Robot is done using HTML. This system works on IoT concept which is Internet of Things, where all the physical devices will connect with digital systems. This will enable raspberry pi to be used for more robotic applications and cut down the cost for building an IoT robot.

List of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

The research and development of raspberry pi controlled IoT based robot. It works as buddy or family member. IoT is Internet of thing where all the physical devices connects with digital systems, such as Refrigerator, TV, AC, Washing machine, Music system which can works automatically or control from anywhere. Data says by 2020 50 billions device will connect. Arduino and Raspberry Pi is a credit-card sized computer. It is connected with the Internet and robot can be control as per my command.

The Internet of Things is more than just a buzzword. In fact, the acronym “IoT” refers to a network of billions of physical devices around the globe. These devices are equipped with Internet-connected sensors that provide ongoing data collection and sharing. As such, IoT is transforming the way consumers and businesses learn about and connect with one another.

There is no shortage of IoT devices and applications. Many consumers are reaping the benefits of smart light bulbs, activity trackers, hands-free speakers, and other IoT devices. Meanwhile, businesses are using industrial IoT (IIoT) devices to track equipment maintenance and repairs, monitor employee performance, improve product quality and safety, and much more.

Some cities are even using IoT devices to set up real-time data collection and alerts to notify municipal services when trash bins need to be emptied, improve parking space availability, and manage pollution. Clearly, there’s a lot to do around the Internet of Things, and IIoT provides growth opportunities for organizations of all sizes and across all vertical markets, including healthcare. IIoT offers many benefits, such as:

Unprecedented connectivity: IIoT data and insights from connected applications and devices empower organizations with the ability to deliver innovative new products and services faster than their competitors.

Increased efficiency: IIoT networks of smart and intelligent devices provide real-time data to arm employees with the information they need to optimize their day-to-day efficiency and productivity.

Cost savings: IIoT devices provide accurate data collection and automated workflows to help organizations reduce their operating costs and minimize errors.

Time savings: Connected smart devices can help organizations enhance the performance of systems and processes to save time.

The global IIoT market is projected to grow significantly. Grand View Research has projected that the global IIoT market will be worth nearly $934 billion by 2025. The market research firm noted the rising demand for cloud computing, big data analytics, and other technologies are among the prime factors that could accelerate this market’s growth.

Many consumers and businesses want to get the most out of their connected systems, but most fail to understand the opportunities IoT presents. Web and mobile app development firm Seamgen is trying to take the guesswork out of IoT. Check out Seamgen’s IoT infographic below to learn the basics about how IoT works and its potential benefits.

1.1 Background/Motivation

User can see live streaming from computer device as website or phone application as camera is attached. Different buttons are there such as Forward, Reverse, Left, Right and Stop to control the Robot. Different sensors are attached with the device such as Ultrasonic sensor, PIR sensors to detect obstacle and distance and generate notifications and sends data to user. In smart home concepts it can add value in it.

The Internet of Things (IoT) has not been around for very long. However, there have been visions of machines communicating with one another since the early 1800s. Machines have been providing direct communications since the telegraph (the first landline) was developed in the 1830s and 1840s. Described as “wireless telegraphy,” the first radio voice transmission took place on June 3, 1900, providing another necessary component for developing the Internet of Things. The development of computers began in the 1950s.

The Internet, itself a significant component of the IoT, started out as part of DARPA (Defense Advanced Research Projects Agency) in 1962, and evolved into ARPANET in 1969. In the 1980s, commercial service providers began supporting public use of ARPANET, allowing it to evolve into our modern Internet. Global Positioning Satellites (GPS) became a reality in early 1993, with the Department of Defense providing a stable, highly functional system of 24 satellites. This was quickly followed by privately owned, commercial satellites being placed in orbit. Satellites and landlines provide basic communications for much of the IoT.

One additional and important component in developing a functional IoT was IPV6's remarkably intelligent decision to increase address space. Steve Leibson, of the Computer History Museum, states, "The address space expansion means that we could assign an IPV6 address to every atom on the surface of the earth, and still have enough addresses left to do another 100+ earths." Put another way, we are not going to run out of internet addresses anytime soon.

Realizing the Concept

The Internet of Things, as a concept, wasn't officially named until 1999. One of the first examples of an Internet of Things is from the early 1980s, and was a Coca Cola machine, located at the Carnegie Melon University. Local programmers would connect by Internet to the refrigerated appliance, and check to see if there was a drink available, and if it was cold, before making the trip.

By the year 2013, the Internet of Things had evolved into to a system using multiple technologies, ranging from the Internet to wireless communication and from micro-electromechanical systems (MEMS) to embedded systems. The traditional fields of automation (including the automation of buildings and homes), wireless sensor networks, GPS, control systems, and others, all support the IoT.

Simply stated, the Internet of Things consists of any device with an on/off switch connected to the Internet. This includes almost anything you can think of, ranging from cellphones to building maintenance to the jet engine of an airplane. Medical devices, such as a heart monitor implant or a biochip transponder in a farm animal, can transfer data over a network and are members the IoT. If it has an off/on switch, then it can, theoretically, be part of the system. The IoT consists of a gigantic network of internet connected "things" and devices. **Ring**, a doorbell that links to your smart phone, provides an excellent example of a recent addition to the Internet of Things. Ring signals you when the doorbell is pressed, and lets you see who it is and to speak with them.

Kevin Ashton, the Executive Director of Auto-ID Labs at MIT, was the first to describe the Internet of Things, while making a presentation for Procter & Gamble. During his 1999 speech, Mr. Ashton stated:

"Today computers, and, therefore, the Internet, are almost wholly dependent on human beings for information. Nearly all of the roughly 50 petabytes (a petabyte is 1,024 terabytes) of data available on the Internet were first captured and created by human beings by typing, pressing a record button, taking a digital picture or scanning a bar code. The problem is, people have limited time, attention, and accuracy. All of which means they are not very good at capturing data about things in the real world. If we had computers that knew everything there was to know about things, using data they gathered without any help from us, we would be able to track and count everything and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling and whether they were fresh or past their best."

Kevin Ashton believed Radio Frequency Identification (RFID) was a prerequisite for the Internet of Things. He concluded if all devices were "tagged," computers could manage, track, and inventory them. To some extent, the tagging of things has been achieved through technologies such as digital watermarking, barcodes, and QR codes. Inventory control is one of the more obvious advantages of the Internet of Things.

Connecting Devices in New Ways

When thinking of the IoT, consider the idea, "any device capable, can be interconnected with other devices." The IoT is ripe for new and creative ideas to add to the tasks already in use. Imagine an alarm waking you at 6 AM in the morning, and then simultaneously signaling your coffee maker to turn on and start brewing coffee. Imagine your printer knowing when you are running low on paper, and automatically ordering more. Imagine the watch on your wrist telling you "where" you have been the most productive, while at work. The IoT can be used to organize such things as transportation networks. "Smart cities" can use it to reduce waste and maximize the efficient use of energy.

In truth, the IoT provides a nearly endless supply of opportunities to interconnect our devices and equipment. In terms of creativity, this field is wide open, with an infinite number of ways to "interconnect the devices." It can be an exciting time for innovative individuals, in part, because we don't fully understand the impact of these interconnections. The IoT offers both opportunities and potential security problems. At present, the Internet of Things is best viewed with an open mind, for purposes of creativity, and a defensive posture for purposes of privacy and security.

Customer Privacy

As sensors and video cameras become more common place, especially in public spaces, consumers have less and less knowledge about the **information being collected**, and no way to avoid it.

Many people are uncomfortable with the idea of companies collecting information about them, and even more uncomfortable having that information sold to anyone and everyone. Generally speaking, older people dislike having information about themselves collected more than younger people, but according to one survey, about 45% of “all” respondents did not trust companies to use the data they collected to protect their privacy.

Currently, choices regarding privacy are very black and white, or on/off. The customer is forced to give up all privacy, (often in an agreement so convoluted people don’t bother to plow through it) or the customer simply cannot access the service. This has led to continuing discussions about consumer privacy and how to best educate consumers regarding privacy and the accessibility of data.

Security

While there are steps to take to help **ensure security**, it should come as no surprise this issue has become a significant concern with the growth of the IoT. Literally billions of devices are being interconnected together, making it possible (eventually) for someone to hack into your coffee maker, and then access your entire network. The Internet of Things also makes businesses all around the world more open to security threats. Additionally, data sharing and privacy becomes issues when using the Internet of Things. Consider how concerns will grow when billions of devices are interconnected. Some businesses will be faced with storing the massive amounts of information these devices will be producing. They will need to find a method of securely storing the data, while still being able to access, track, and analyze the huge amounts of it being generated.

James Lewis, who is a cybersecurity researcher for the Center for Strategic and International Studies, has written a report describing how the Internet of Things’ interconnections will allow computer hackers to wreak havoc through interconnected devices. The threat is so real, even the Federal Trade Commission has gotten involved, wanting to know how to guarantee privacy, and how security safeguards are being installed in new Internet-connected devices. For example, new cars can now be hijacked by way of their Wi-Fi connections. Consider the threat of hackers when automated driving becomes popular. Security and **risk management** should not be taken lightly when creating new ways to use the Internet of Things.

Cisco has been driving the term Internet of Everything (IoE). Intel initially called it the "embedded internet". Other terms that have been proposed but don't mean exactly all the same are:

- M2M (Machine to machine) communication
- Web of Things
- Industry 4.0
- Industrial internet (of Things)
- Smart systems
- Pervasive computing
- Intelligent systems

How the concepts differ from each other

M2M

The term Machine to Machine (M2M) has been in use for more than a decade, and is well-known in the Telecoms sector. M2M communication had initially been a one-to-one connection, linking one machine to another. But today's explosion of mobile connectivity means that data can now be more easily transmitted, via a system of IP networks, to a much wider range of devices.

Industrial Internet (of Things)

The term industrial internet is strongly pushed by GE. It goes beyond M2M since it not only focuses on connections between machines but also includes human interfaces.

Internet of Things (IoT)

IoT has yet a wider reach as it also includes connections beyond the industrial context such as wearable devices on people.

Internet (as we know it)

In the above graph, the internet is a fairly small box. In its core it connects only people.

Web of Things

The Web of Things is much narrower in scope as the other concepts as it solely focuses on software architecture.

Internet of Everything (IoE)

Still a rather vague concept, IoE aims to include all sorts of connections that one can envision. The concept has thus the highest reach.

Industry 4.0

The term Industry 4.0 that is strongly pushed by the German government is as limited as the industrial internet in reach as it only focuses on manufacturing environments. However, it has the largest scope of all the concepts. Industry 4.0 describes a set of concepts to drive the next industrial revolution. That includes all kinds of connectivity concepts in the industrial context. However, it goes further and includes real changes to the physical world around us such as 3D-printing technologies or the introduction of new augmented reality hardware.

Problem Formulation

This project will develop a remotely controlled robot using a raspberry pi board, website, and camera. The robot would act as a substitute for humans in hazardous environments, high risk areas or places not reachable by humans. This would prevent the loss of lives in hazardous situations and would make rescue operations safer and more effective. Such situations may include discovering people who may be trapped in a fire, pipe inspection, and forest and volcano research.

1.2 Aims & Objectives

Compared with the traditional information networks, IoT has three new goals, i.e., more extensive interconnection, more intensive information perception, and more comprehensive intelligent service. They are elaborated as follows.

- **More Extensive Interconnection**

IoT extends the interconnection among the information equipments, such as computer and mobile phone, to the interconnection of all intelligent or non-intelligent physical objects. It has the following outstanding characteristics:

1) Extensiveness in the quantity of devices. The amount of the connected devices will sharply rise from several billions to over hundreds of billions, including a multitude of equipments, sensors, actuators, vehicles, and devices attached with RFID.

2) Extensiveness in the type of devices. Networking devices (networking elements) may be owered by the electronic power directly or by batteries; the computation and communication capacity may be greatly different, e.g., some devices even may not have any computational ability.

3) Extensiveness in the connection mode. The devices may be connected in a wired or wireless mode; the communication could be a single hop or multiple hops; the connection can be strong state routing or statistical weak state routing.

- **More Intensive Information Perception**

IoT extends the paradigm of traditional single sensors that sense the local environment independently to the new paradigm of collaboration of multi-sensors to achieve the global environment awareness. Sensing information from each single sensor may contain uncertainties in the following aspects:

- Non-uniformity. Data formats for temperature, humidity, audio, video, and other information are different from each other.
- Inconsistency. There is inconsistent information due to the distortion of space-time mapping.
- Inaccuracy. A range of information inaccuracies are often caused by the variety of sampling methods and different capabilities of the sensors.
- Discontinuities. Intermittent information availability is often caused by the dynamic network transmission capacity.

- **More Comprehensive Intelligent Service**

  Based on the extensive interconnection of ordinary physical objects and the intensive perception of the physical world, IoT can provide comprehensive intelligent services, where physical objects are actively involved in the service process.

To develop an IoT technology based Robot can be controlled by a mobile devices/ Laptops over the Wi-Fi from anywhere at any time.

The core objectives are:

- Gather system requirements
- Evaluate and study the platform required for the system
- Evaluate and study suitable development language, technologies and tools
- Evaluate Methods of Interface
- Program Arduino
- Program Raspberry Pi
- Interface board for dc motors
- Program Website & Control Page
- Evaluate and test the system
- Maintain system

1.3 Organization

This project consists of six chapters mainly as follows:

Chapter 1: Introduction

In this section it gives a general idea of the project and also about the Aims and objectives of this project.

Chapter 2: Design Methodology

This section contains information of the block diagram of the project and described each component in the detail part.

Chapter 3: Industry Applications Using IoT

This chapter contains information about the application of using IoT in the industry

Chapter 4: Software Implementation

In this the hardware tools and the software tools required for the carrying out of the project is been specified also the interface designs for the project are also discussed.

Chapter 5: Results

This chapter contains the test results of what has been done and what more of the objectives could have been achieved.

Chapter 6: Discussion / Conclusions / Future Work

This chapter contains the discussion, conclusion and the future work of the project.

Chapter 2 Design Methodology

The design consists more on actual planning of hardware part than the code to be created. A number of software and hardware implementation techniques were used to design and develop the system. Fig. 1 shows the block diagram of system. This section can be divided into many parts: Raspberry pi controller, Router, Raspberry cam, Ultrasonic sensor, Usb Mic ,and Motor control design. The Block diagram is shows as below:

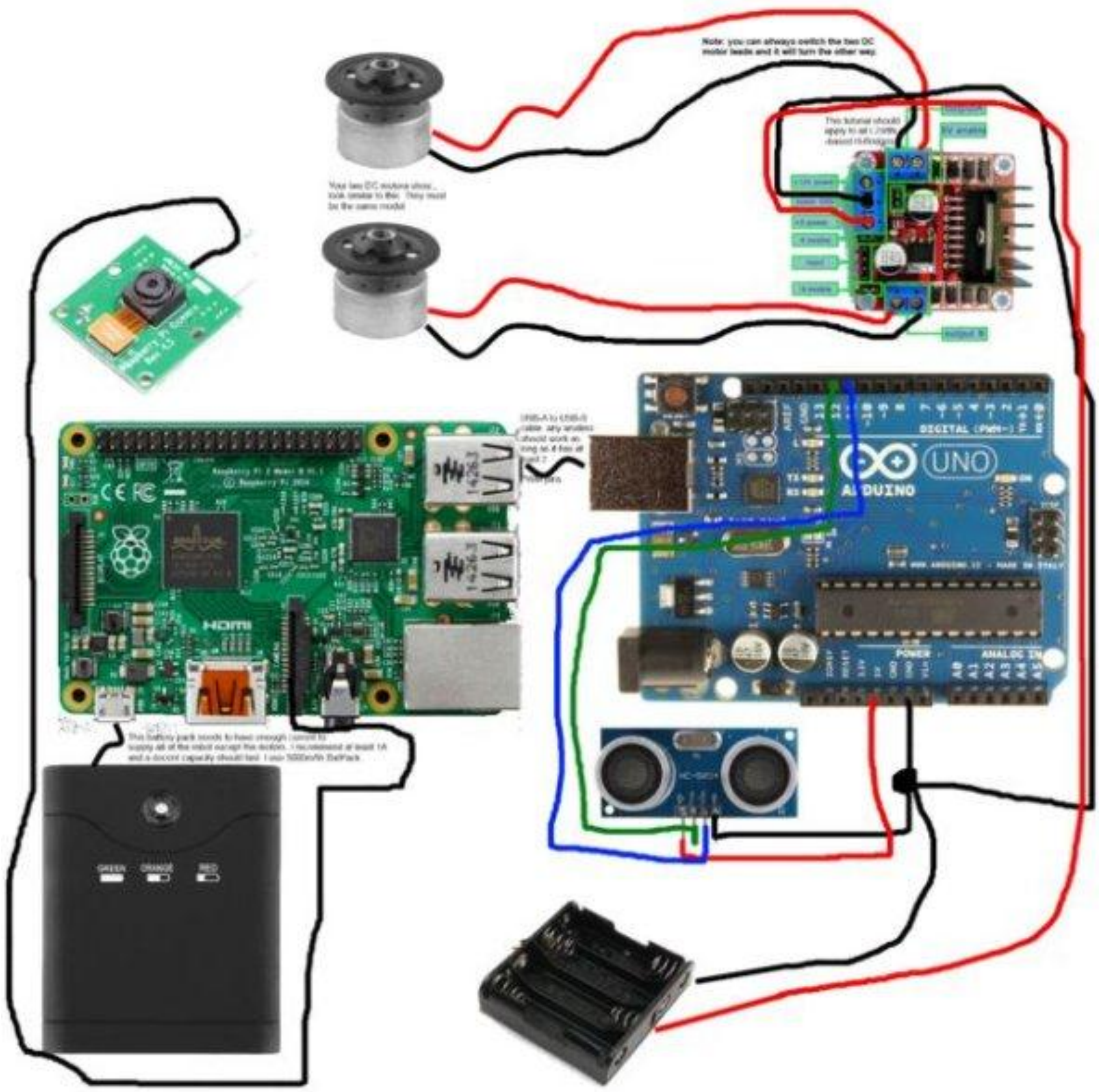

Figure 2-1The initia block diagram for IOT Robot

Table 2-1 No & Item Included of Block Diagram

No.	Item	No.	Item
1	**Raspberry Pi 2**	7	**PIR Sensors**
2	**Router**	8	**Left Side DC Motor**
3	**Rpi Camera**	9	**Right Side DC Motor**
4	**5V PowerBank**	10	**L293D Motor Driver Board**
5	**Ultrasonic Sensor**	11	**12V Battery**
6	**Servo motor**	12	**Microphone**

Components Details:

Different components of block diagram are described in details below.

2.1 Raspberry pi 3

Raspberry Pi is a single board computer (SBC) that can be quickly mounted to any TV or other output channels to create different projects. You do not necessarily need a TV unless you want to play games or do some coding. You can use these boards A+ and B+ in robots and in Internet of Things projects. There are several DIY projects posted on the Internet for you to try out. This article checks out the differences between Raspberry Pi A+ and Raspberry Pi B+.

The main feature of Raspberry Pi A+ is that the power consumption is pretty low compared to Raspberry Pi B+. You can use it to create small robots that can run for a longer time without having to change batteries very frequently. The cost too is a little less compared to Raspberry Pi Raspberry Pi B+. The Raspberry Pi B+ comes at USD 35 while the A+ comes at USD 20. The size of Raspberry A+ is smaller compared to B+ and is just 65mm whereas the Raspberry Pi B+ is 85mm. Thus, it can easily be used to create projects of small sizes

The A+ has forty General Purpose Input and Output (GPIO) pins. Though B+ too has the same number of B+, the latter is used mostly with TVs and Gaming. The number of ports is higher in Raspberry Pi B+ but then, the cost of B+ is higher for the reason. As said earlier, B+ is good if you intend to connect to TV. Likewise, A+ is best when you wish to create projects that are small in size but nevertheless offer better flexibility over B+.

Raspberry Pi B+ has double the memory compared to A+. The Raspberry Pi B+ memory is 512 SDRAM and A+ is just 256 MB SDRAM. The GPU processor of B+ is dual core VideoCore Multimedia Co Processer. It provides Open GL ES 2.0 which means 1080p definition. The GPU processor in A+ is same as that of B+. It too offers up to 1080p definition.

The Raspberry A+ has 700 MHz Low Power ARM1176JZFS Applications Processor. The Raspberry B+ too runs the same processor. The only great difference between specs is the memory and the number of ports they hold. The A+ has one USB port while the B+ has 4 USB ports of 2.0 generation. The A+ has no Ethernet port so it cannot connect to the wired Internet. B+ has a 10/100 base T Ethernet port that allows you to connect it to other computers or directly to the Internet. The B+ also has a 15 pin camera connector in addition to DSI display connector. Both Raspberry Pi A+ and Raspberry Pi B+ have Micro Card of Slider type so that SD cards can be slipped in easily. Note that the Raspberry Pi have their OS on SD cards that can be removed and replaced to replace the entire operating system. You can get more information on the Rasberry Pi website.

To sum up, Raspberry B+ is better compared to Raspberry A+. The latter is useful when you need a tiny board with plenty of battery life. Otherwise, given the number of ports and memory of B+, it makes a good Single Computer Board for many tasks.

What is System on Chip?

A complex IC that integrates the major functional elements into a single chip or chipset.

- Programmable processor
- On-chip memory
- Accelerating function hardware (e.g. GPU)
- Both hardware and software •analog components
- Benefits of SoC –Reduce overall system cost –Increase performance –Lower power consumption –Reduce size

Writing Raspbianto the SD card

a) Plug your SD card into your PC

b) In the folder you made in step 3(b), run the file named Win32DiskImager.exe

c) If the SD card (Device) you are using isn't found automatically then click on the drop down box and select it

d) In the Image File box, choose the Raspbian.imgfile that you downloaded

e) Click Write

f) After a few minutes you will have an SD card that you can use in your Raspberry Pi

Booting your Raspberry Pi for the first time

- On first boot you will come to the Raspi-configwindow
- Change settings such as timezoneand locale if you want
- Finally, select the second choice: expand_rootfsand say 'yes' to a reboot
- The Raspberry Pi will reboot and you will see raspberrypi login: •Username: pi, password: raspberry
- Start the desktop by typing: startx
- The desktop environment is known as the Lightweight X11 Desktop Environment (LXDE)

Re-mapping Keyboard:

- sudovi /etc/default/keyboard XKBLAYOUT=”gb” Change “gb” to “us”
- (This assumes you want a us mapping, if not replace the gbwith the two letter code for your country)
- Install and Start SSH
- Update apt-get package index files: –sudoapt-get update
- Install SSH: –sudoapt-get install ssh
- Start SSH server: –sudo/etc/init.d/sshstart
- To start the SSH server every time the Pi boots up: –sudoupdate-rc.dsshdefaults

- SSH client for Windows: –PuTTY –http://www.putty.org/
- SSH Secure File Transfer –http://www.utexas.edu/learn/upload/ssh_client.h tml

Install Java

- JDK 8 (with JavaFX) for ARM Early Access http://jdk8.java.net/fxarmpreview/ –Download from Raspberry pi –Download from your own PC and copy it (scp) to Raspberry pi
- Extract the JDK tar.gz file –tar –zxvffileToExtract.tar.gz –You will get a folder "jdk1.8.0" Set Java PATH
- If you put the folder "jdk1.8.0" in the home directory (i.e. /home/pi), you will see the java executables (e.g. javac, java, appletviewer) in the directory: /home/pi/jdk1.8.0/bin
- open /etc/profile add: PATH=$PATH:/home/pi/jdk1.8.0/bin export PATH
- Reboot: sudoreboot

The Micro SD card is used for installing OS and the complete project will be done with python coding.

The board has specification:

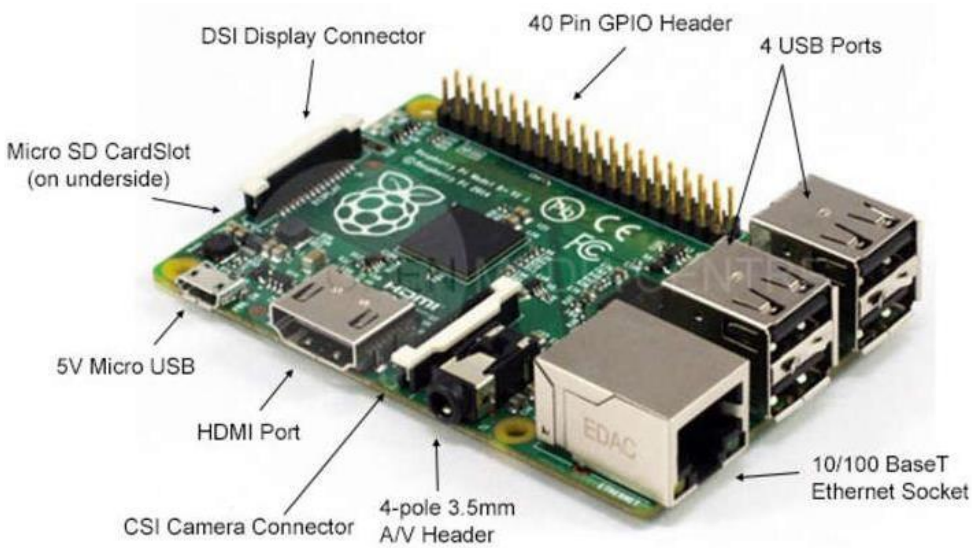

Figure 2-2 Raspberry Pi B+ model

The Raspberry Pi B+ model Specifications:

- A 900MHz quad-core ARM Cortex-A7 CPU
- 4 USB ports
- 1GB RAM
- Full HDMI port
- 40 GPIO pins
- Display Interface
- Ethernet Port
- Micro SD card Slot
- Combined 3.5mm audio jack and composite video
- Video core IV 3D graphics core
- Camera Interface

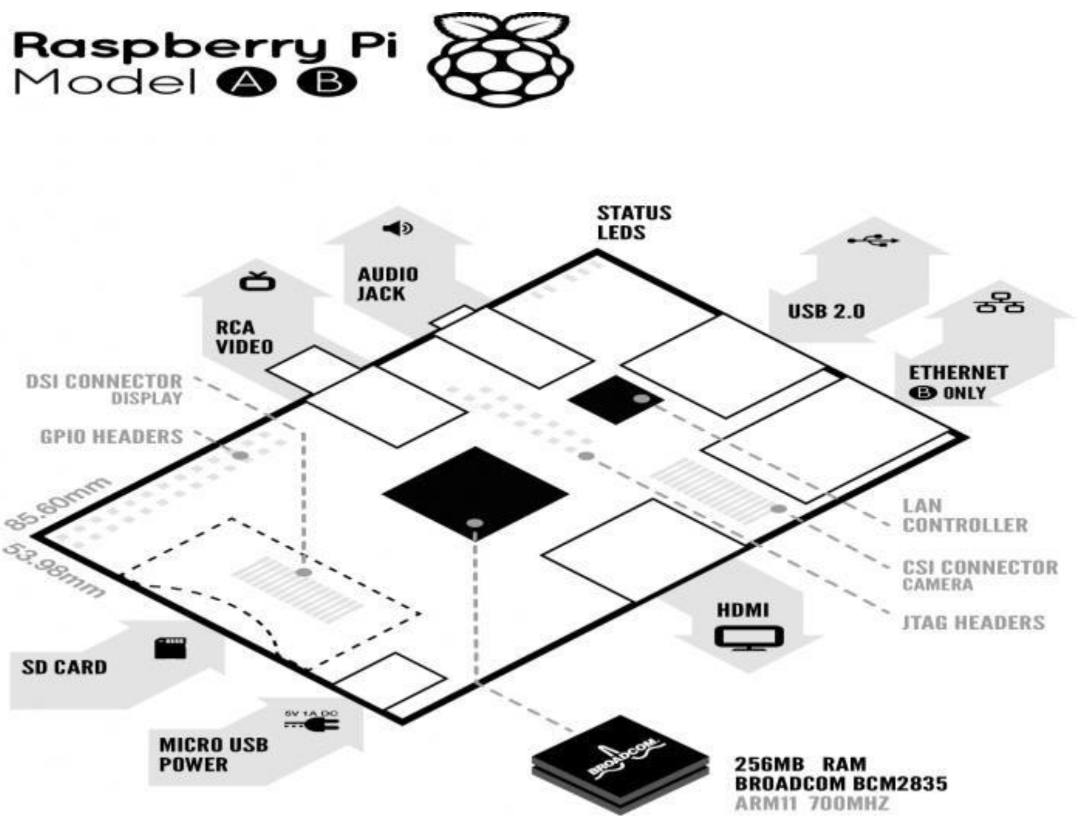

Figure 2-3 Structure of Raspberry Pi B+ model

A brief description of the components on the Raspberry Pi.

2.1.1 Processor / SoC (System on Chip)

The Raspberry Pi has a Broadcom BCM2835 System on Chip module. It has an ARM1176JZF-S processor. The Broadcom SoC used in the Raspberry Pi is equivalent to a chip used in an old smartphone (Android or iPhone). The Raspberry Pi chip operating at 700 MHz by default, will not become hot enough to need a heat sink or special cooling.

2.1.2 Power source

The Pi is a device which consumes 700mA or 3W or power. It is powered by a Micro USB charger or the GPIO header. Any good smartphone charger will do the work of powering the Pi.

2.1.3 SD Card

The Raspberry Pi does not have any onboard storage available. The operating system is loaded on a SD card which is inserted on the SD card slot on the Raspberry Pi. The operating system can be loaded on the card using a card reader on any computer.

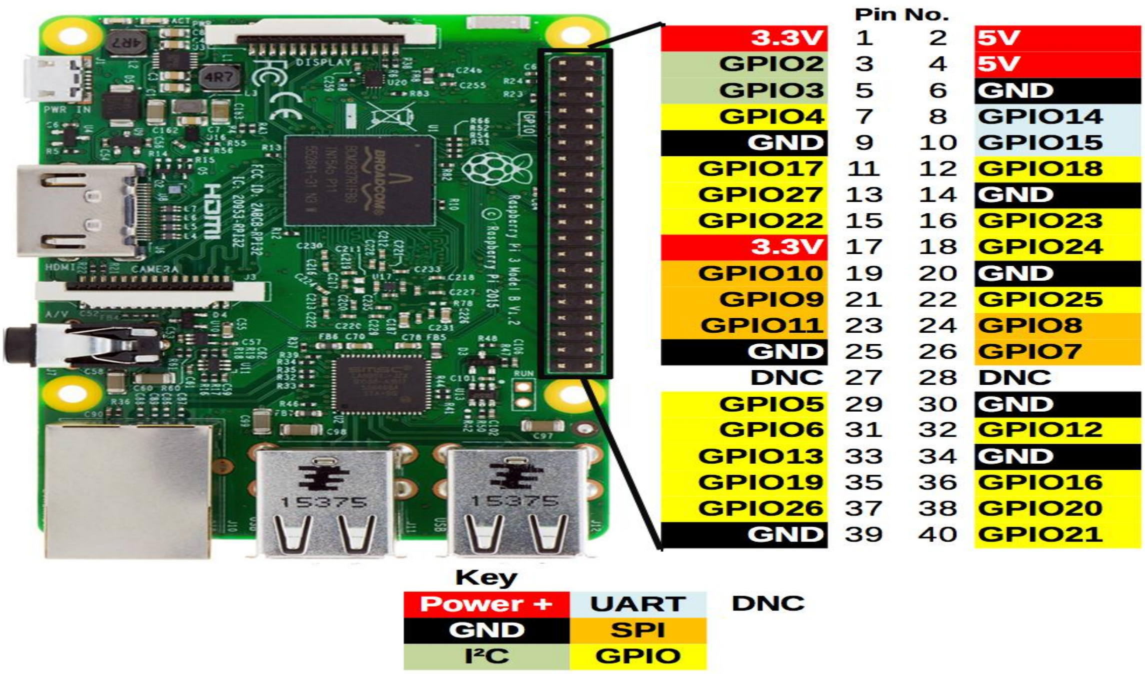

Figure 2-4 Raspberry Pi board

2.1.4 GPIO – General Purpose Input Output

General-purpose input/output (GPIO) is a generic pin on an integrated circuit whose behavior, including whether it is an input or output pin, can be controlled by the user at run time.

GPIO capabilities may include:

- GPIO pins can be configured to be input or output
- GPIO pins can be enabled/disabled
- Input values are readable (typically high=1, low=0)
- Output values are writable/readable
- Input values can often be used as IRQs (typically for wakeup events)

The production Raspberry Pi board has a 26-pin 2.54 mm (100 mil) expansion header, marked as P1, arranged in a 2x13 strip. They provide 8 GPIO pins plus access to I^2C, SPI, UART), as well as +3.3 V, +5 V and GND supply lines. Pin one is the pin in the first column and on the bottom row.

2.1.5 DSI Connector

The Display Serial Interface (DSI) is a specification by the Mobile Industry Processor Interface (MIPI) Alliance aimed at reducing the cost of display controllers in a mobile device. A DSI compatible LCD screen can be connected through the DSI connector, although it may require additional drivers to drive the display.

2.1.6 RCA Video

RCA Video outputs (PAL and NTSC) are available on all models of Raspberry Pi. Any television or screen with a RCA jack can be connected with the RPi.

Figure 2-5 RCA Video Connector

2.1.7 Audio Jack

A standard 3.5 mm TRS connector is available on the RPi for stereo audio output. Any headphone or 3.5mm audio cable can be connected directly. Although this jack cannot be used for taking audio input, USB mics or USB sound cards can be used.

2.1.8 Status LEDs

There are 5 status LEDs on the RPi that show the status of various activities as follows:

“OK” - SD Card Access (via GPIO16) - labelled as "OK" on Model B Rev1.0 boards and "ACT" on Model B Rev2.0 and Model A boards

“POWER” - 3.3 V Power - labelled as "PWR" on all boards

“FDX” - Full Duplex (LAN) (Model B) - labelled as "FDX" on all boards

“LNK” - Link/Activity (LAN) (Model B) - labelled as "LNK" on all boards

“10M/100” - 10/100Mbit (LAN) (Model B) - labelled (incorrectly) as "10M" on Model B Rev1.0 boards and "100" on Model B Rev2.0 and Model A boards.

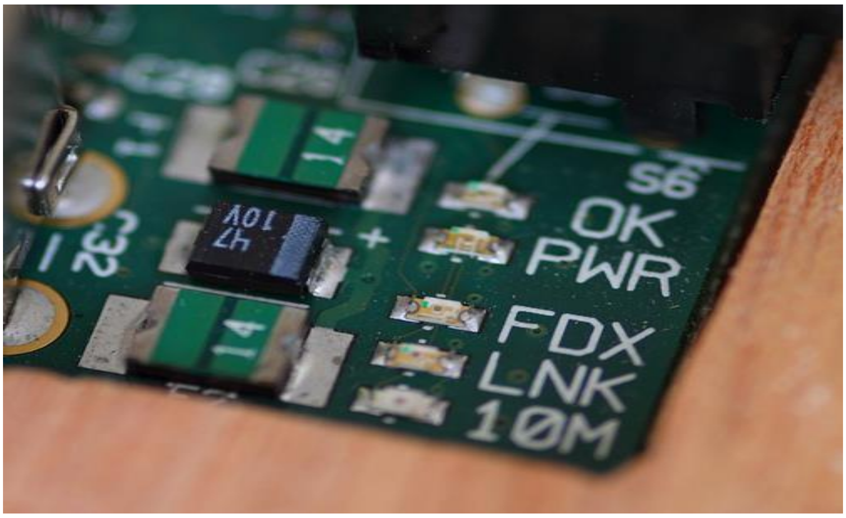

Figure 2-6 Status LEDs

2.1.9 USB 2.0 Port

USB 2.0 ports are the means to connect accessories such as mouse or keyboard to the Raspberry Pi. There is 1 port on Model A, 2 on Model B and 4 on Model B+. The number of ports can be increased by using an external powered USB hub which is available as a standard Pi accessory.

2.1.10 Ethernet

Ethernet port is available on Model B and B+. It can be connected to a network or internet using a standard LAN cable on the Ethernet port. The Ethernet ports are controlled by Microchip LAN9512 LAN controller chip.

2.1.11 CSI connector

CSI – Camera Serial Interface is a serial interface designed by MIPI (Mobile Industry Processor Interface) alliance aimed at interfacing digital cameras with a mobile processor. The RPi foundation provides a camera specially made for the Pi which can be connected with the Pi using the CSI connector.

2.1.12 JTAG headers

JTAG is an acronym for 'Joint Test Action Group', an organization that started back in the mid 1980's to address test point access issues on PCB with surface mount devices. The organization devised a Many thousands of devices now include this standardized port as a feature to allow test and design engineers to access pins.

2.1.13 HDMI – High Definition Multimedia Interface

HDMI 1.3 a type a port is provided on the RPi to connect with HDMI screens.

2.2 WiFi Dongle

Wireless-N USB adapter is used to connect Raspberry Pi with Internet. Standards are IEEE 802.11n, 802.11g, 802.11b and frequency range is 2.4 GHz. Its 150Mbps Nano USB adapter. Support the systems as XP, Vista, WIN 7, WIN 8, and Linux.

Figure 2-7 WiFi Dongle

2.3 Raspberry Pi Camera

The Raspberry Pi camera board contains a 5 MPixel sensor, and connects via a ribbon cable to the CSI connector on the Raspberry Pi. In Raspbian support can be enabled by the installing or upgrading to the latest version of the OS and then running Raspi-config and selecting the camera option. The cost of the camera module is 1600Rs. In India (10 May 2015) and supports 1080p, 720p, 640x480p video. The footprint dimensions are 25 mm x 20 mm x 9 mm. Since Raspberry Pi has a ready-to-use socket for camera cable, no extra cables or power supplies are needed.

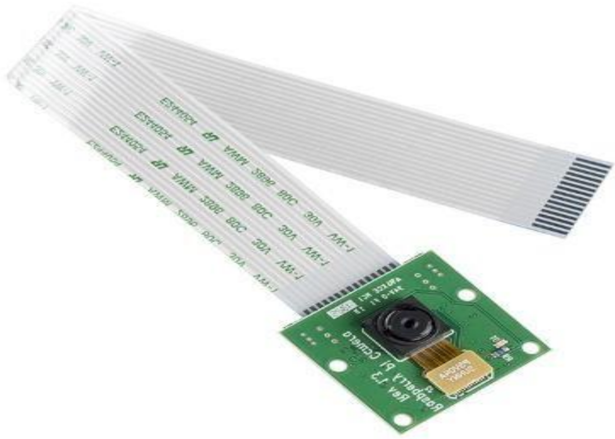

Figure 2-8 Raspberry Pi Camera

The camera board attaches to the Raspberry Pi via a 15-way ribbon cable. There are only two connections to make: the ribbon cable needs to be attached to the camera PCB, and to the Raspberry Pi itself. You need to get the cable the right way round, or the camera will not work. On the camera PCB, the blue backing on the cable should face away from the PCB, and on the Raspberry Pi it should face towards the Ethernet connection (or where the Ethernet connector would be if you're using a model A).

Although the connectors on the PCB and the Pi are different, they work in a similar way. On the Raspberry Pi itself, pull up the tabs on each end of the connector. It should slide up easily, and be able to pivot around slightly. Fully insert the ribbon cable into the slot, ensuring it is set straight, then gently press down the tabs to clip it into place. The camera PCB connector also requires you to pull the tabs away from the board, gently insert the cable, then push the tabs back. The PCB connector can be a little more awkward than the one on the Pi itself.

2.4 5V Adapter Power Supply

Raspberry Pi require power source to turn it on. 5V adapter Power supply is enough to power up. In project I connect Power bank with raspberry Pi, so that it can be put easily in structure.

Figure 2-9 5V Power Supply adapter

2.5 Ultrasonic Sensor

The 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.

Figure 2-10 Ultrasonic Module

This sensor is a high performance ultrasonic range finder. It is compact and measures an amazingly wide range from 2cm to 4m. This ranger is a perfect for any robotic application, or any other projects requiring accurate ranging information. This sensor can be connected directly to the digital I/O lines of your microcontroller and distance can be measured in time required for travelling of sound signal using simple formula as below. The module works on 5VDC input and also gives an output signal directly for detection of any obstacle up to 4M. As soon as the signals are transmitted the “Echo” pin goes to high level and remains in high level until the same sound waves are received by the receiver. If the received sound waves are same as what the same sensor transmitted then the Echo pin goes to low level. If no object is detected within 5M after 30ms the Echo signal will automatically go to low level.

2.6 PIR Sensor

PIRs are basically made of a pyroelectric sensor (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

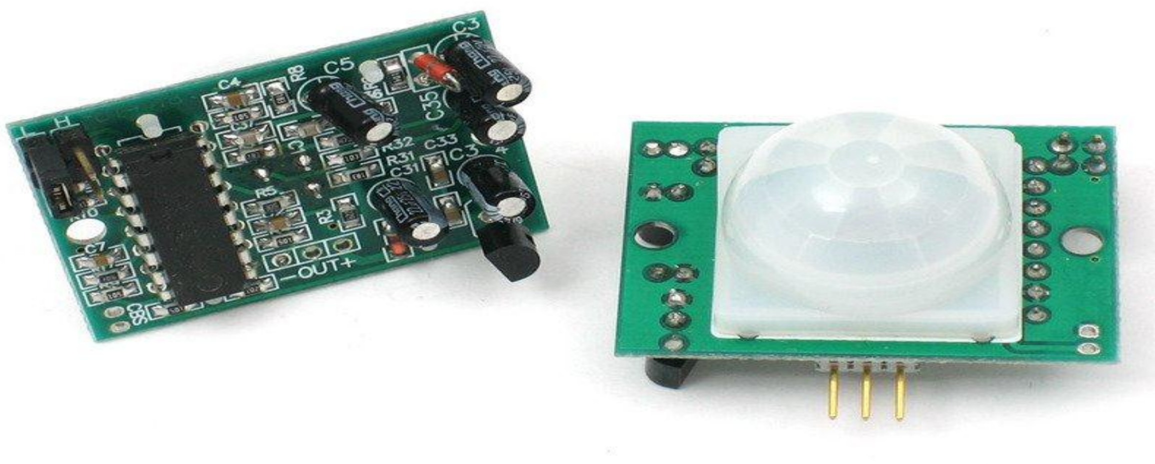

Figure 2-11 PIR sensor

The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.

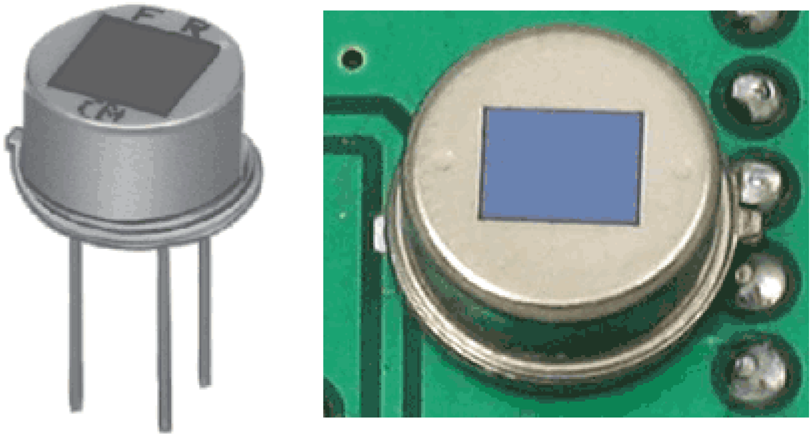

Figure 2-12 pyroelectric sensor

This image shows the internal schematic. There is actually a JFET inside (a type of transistor) which is very low-noise and buffers the extremely high impedence of the sensors into something a low-cost chip (like the BIS0001) can sense.

PIR sensors are rather generic and for the most part vary only in price and sensitivity. Most of the real magic happens with the optics. This is a pretty good idea for manufacturing: the PIR sensor and circuitry is fixed and costs a few dollars. The lens costs only a few cents and can change the breadth, range, sensing pattern, very easily.

In the diagram up top, the lens is just a piece of plastic, but that means that the detection area is just two rectangles. Usually we'd like to have a detection area that is much larger. To do that, we use a simple lens such as those found in a camera: they condenses a large area (such as a landscape) into a small one (on film or a CCD sensor). For reasons that will be apparent soon, we would like to make the PIR lenses small and thin and moldable from cheap plastic, even though it may add distortion. For this reason the sensors are actually Fresnel lenses

2.7 Servo Motor

Servo motor is a **rotary actuator** or **linear actuator** that allows for precise control of angular or linear position, velocity and acceleration.**[1]** It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a **closed-loop control**system. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

Figure 2-13 Servo Motor

A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped). This type of servomotor is not widely used in industrial motion control, but it forms the basis of the simple and cheap servosused for radio-controlled models.

2.8 DC Motor

Almost every mechanical movement that we see around us is accomplished by an electric motor. Electric machines are means of converting conventional energy. Motors take electrical energy and produce mechanical energy. Electric motor is used to power hundreds of devices we use in everyday life.

Figure 2-14 DC Motor

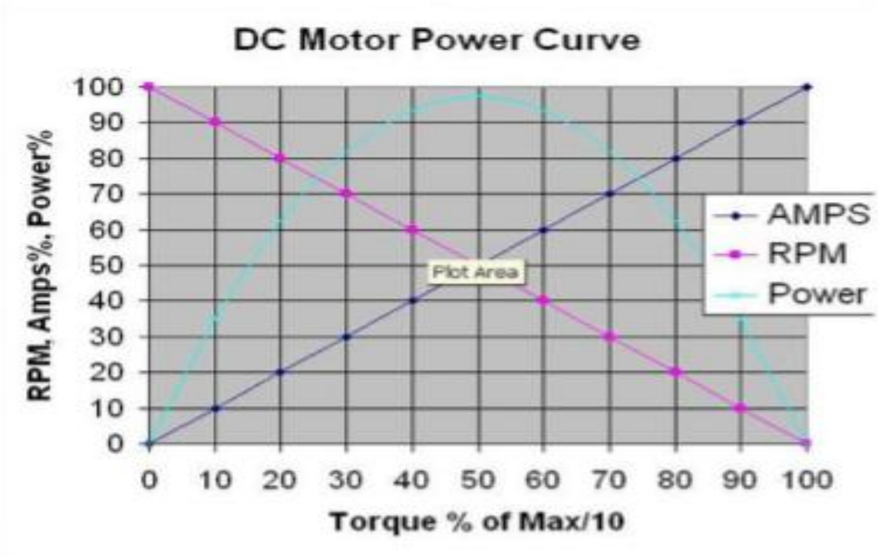

Figure 2-15 DC Motor Power Curve

2.9 H-bridge Motor Driver

The Dual H-Bridge Motor Driver module Board, using ST's L298N chip can directly drive two 3-30V DC motor. You can easily control the DC motor speed and direction, you can also control the 2-phase stepper motor, smart car essential.

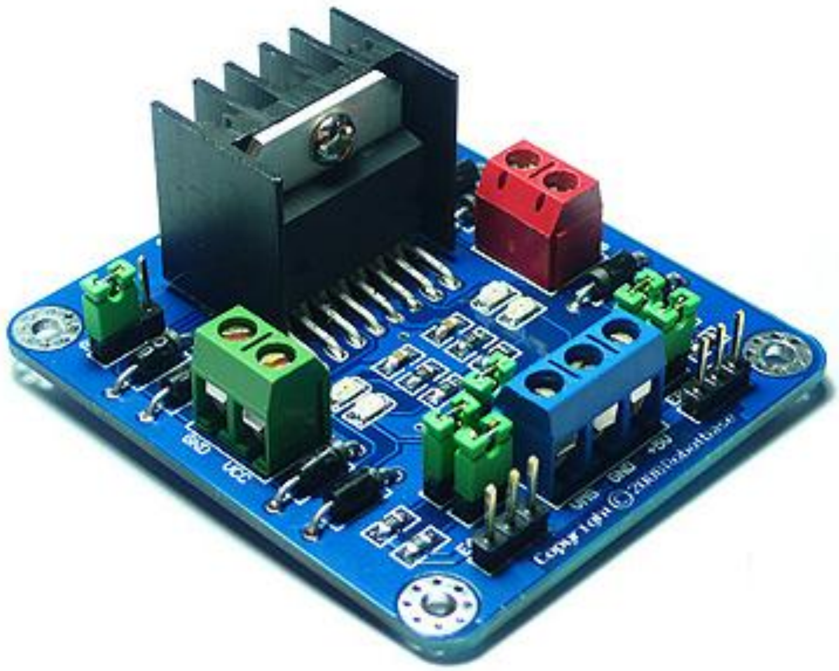

Figure 2-16 H-bridge Motor Driver

The switching elements (Q1..Q4) are usually bi-polar or FET transistors, in some high-voltage applications IGBTs. Integrated solutions also exist but whether the switching elements are integrated with their control circuits or not is not relevant for the most part for this discussion. The diodes (D1..D4) are called catch diodes and are usually of a Schottky type.

The top-end of the bridge is connected to a power supply (battery for example) and the bottom-end is grounded. In general all four switching elements can be turned on and off independently, though there are some obvious restrictions. Though the load can in theory be anything you want, by far the most pervasive application if H-bridges is with a brushed DC or bipolar stepper motor (steppers need two H-bridges per motor) load. In the following I will concentrate on applications as a brushed DC motor driver.

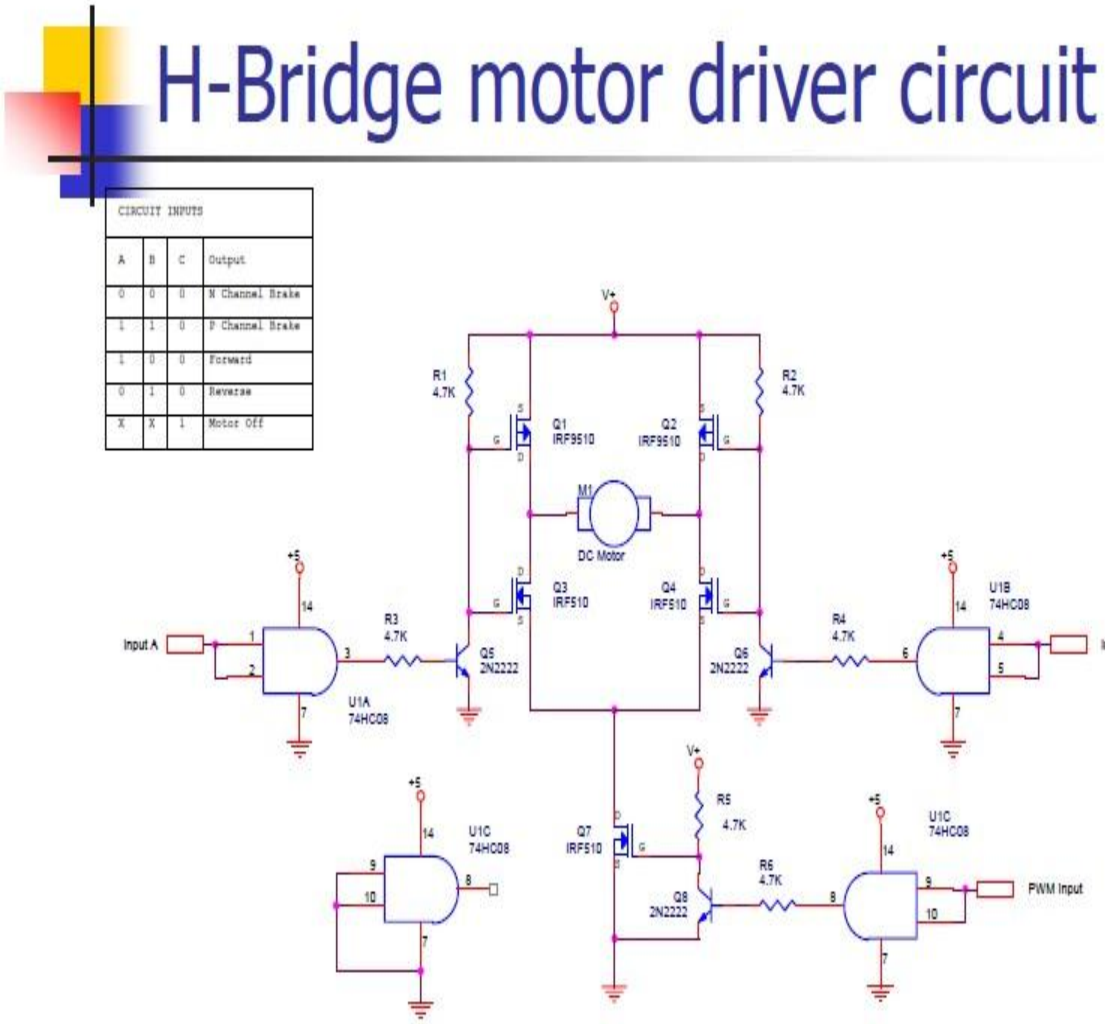

Figure 2-17 Screenshot of DC motor and H-bridge interfacing circuit

The basic operating mode of an H-bridge is fairly simple: if Q1 and Q4 are turned on, the left lead of the motor will be connected to the power supply, while the right lead is connected to ground. Current starts flowing through the motor which energizes the motor in (let’s say) the forward direction and the motor shaft starts spinning.

If Q2 and Q3 are turned on, the reverse will happen, the motor gets energized in the reverse direction, and the shaft will start spinning backwards. In a bridge, you should never ever close both Q1 and Q2 (or Q3 and Q4) at the same time. If you did that, you just have created a really low-resistance path between power and GND, effectively short-circuiting your power supply. This condition is called ‘shoot-through’ and is an almost guaranteed way to quickly destroy your bridge, or something else in your circuit.

2.10 Specifications of Different Raspberry PI Model

Comparison of available Raspberry Pi Model with different parameters including its Price, Wight, Size and Power Source, etc.

Table 2-2 Specifications of Raspberry Pi different model

	Model A	Model B	Model B+
Target price:	**US$25**	**US$35**	
SoC:	**Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, and single USB port)**		
CPU:	**700 MHz ARM1176JZF-S core**		
GPU:	**Broadcom Video Core IV @ 250 MHz**		
Memory (SDRAM):	**256 MB**	**512 MB**	
USB 2.0 ports:	**1**	**2**	**4**
Video input:	**15-pin MIPI camera interface (CSI) connector**		
Video outputs:	**3.5 mm jack, HDMI ,raw LCD Panels**		
Audio outputs:	**3.5 mm jack, HDMI**		
Onboard storage:	**SD / MMC / SDIO card slot**		**MicroSD**
Onboard network:	**None**	**10/100 Mbit/s Ethernet USB adapter**	
Low-level peripherals:	**8× GPIO**		**17× GPIO**
Power ratings:	**300 mA (1.5 W)**	**700 mA (3.5 W)**	**600 mA (3.0 W)**
Power source:	**5 V via MicroUSB or GPIO header**		
Size:	**85.60 mm × 56 mm**		
Weight:	**45 g**		

2.11 Comparison of Raspberry with the competitors

The chief competitors of the Raspberry Pi are the Arduino and the Beagleboard. Both are single board computers and have applications similar to the Raspberry Pi. A brief comparison of the three of them is shown below:

Table 2-3 Comparison of RPi with chief competitors

Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	$29.95	$35	$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog

2.12 Advantages and disadvantages

Advantages and disadvantages of Raspberry Pi from different aspect are described below:

Advantages of the Raspberry Pi

- This microcomputer is useful for small or home based businesses that run on a smaller budget than bigger companies for you are not required to purchase any special licenses from the Raspberry Pi Foundation to use their product or if you invent new technology that embeds the product.
- The product does not require the user to have extensive programming experience since it is aimed for the younger generation to learn about programming. Python, the programming language that the Pi uses, is less complex than other languages available.
- The SD cards on the board can be easily switched, which allows you to change the functions of the device without spending a lot of time re-installing the software.
- The Raspberry Pi is perfect for adaptive technology: it is able to display images or play videos at 1080p high definition resolution. This product makes it possible to build complex and effective products at a cheaper price.
- This small credit card sized product makes it easy to recycle and does not release as much carbon dioxide emissions into the environment, unlike big servers that require lots of energy and extensive cooling systems.

Disadvantages

- It does not replace your computer, since the Ethernet is only a 10/100 and the processor is not as fast, it is time consuming to download and install software and is unable to do any complex multitasking.
- Not compatible with other operating systems such as Windows (There are currently 1.3 billion Windows users around the world.)
- This product will not be useful for bigger businesses that already have big servers, which would already do everything that the Raspberry Pi does, so it would not be worth it to take the time to get someone to put it together.

2.13 Arduino

Arduino Overview

Figure 2-18 Arduino UNO

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy,[2] aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

Hardware Specifications :

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended):7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins:6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

The Many Flavors of Arduino

- **Arduino Uno**
- **Arduino Leonardo**
- **Arduino lalypad**
- **Arduino Mega**
- **Arduino Nano**
- **Arduino Mini**
- **Arduino Mini Pro**
- **Arduino BT**

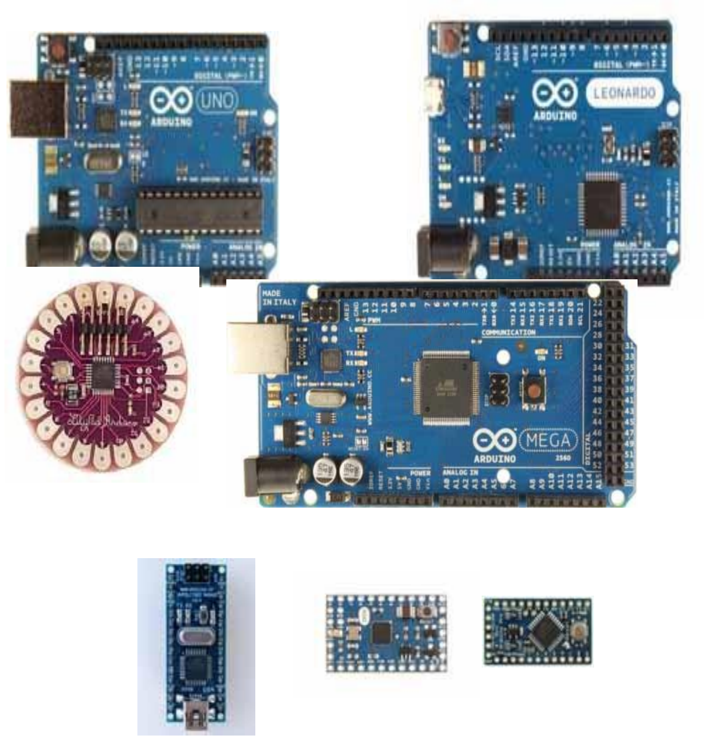

Figure 2-19 Different Arduino Modules

Getting to know the Arduino: Electrical Inputs and Outputs

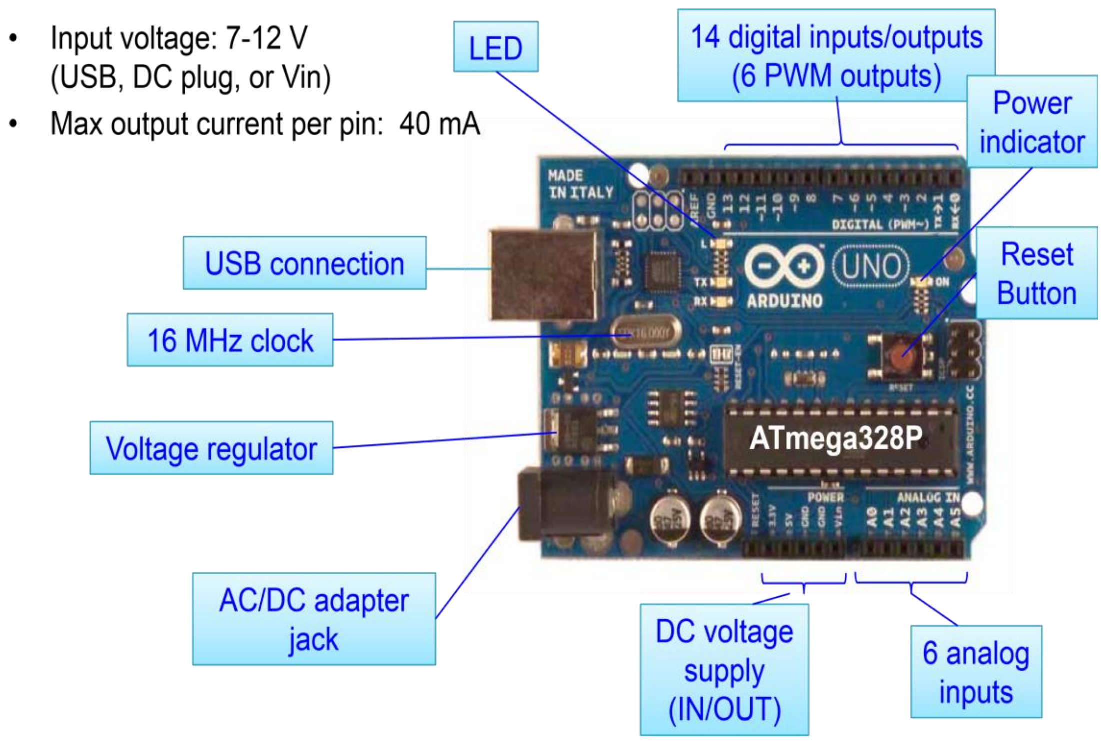

Figure 2-20 Arduino I/P and O/P Ports

Download and Install

- Download Arduino compiler and development environment from: http://arduino.cc/en/Main/Software • Current version: 1.0.1
- Available for:
  - Windows
  - MacOX
  - Linux
- No installer needed... just unzip to a convenient location
- *Before running Arduino*, plug in your board using USB cable (external power is not necessary)
- When USB device is not recognized, navigate to and select the appopriate driver from the installation directory
- Run Arduino

Software:

- **IDE**

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is 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. The source code for the IDE is released under the GNU General Public License, version 2

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.

Sketch

A program written with the Arduino IDE 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.

A minimal Arduino C/C++ program consist of only two functions:

setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

loop(): After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Blink example

Power LED (red) and User LED (green) attached to pin 13 on an Arduino compatible board

Most Arduino boards contain a light-emitting diode (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions. A typical program for a beginning Arduino programmer blinks a LED repeatedly. This program uses the functions pinMode(), digitalWrite(), and delay(), which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacturer.

```
#define LED_PIN 13              // Pin number attached to LED.
void setup() {
   pinMode(LED_PIN, OUTPUT);      // Configure pin 13 to be a
digital output.
}
void loop() {
   digitalWrite(LED_PIN, HIGH);    // Turn on the LED.
   delay(1000);                // Wait 1 second (1000 milliseconds).
   digitalWrite(LED_PIN, LOW);     // Turn off the LED.
```

Specifications of Different Arduino Model

Comparison of available Arduino Model with different parameters including its Wight, Size and Power Source, etc.

Table 2-4 Arduino Modules with different parameters

Boards	Microcontroller	Operating Voltage/s (V)	Digital I/O Pins	PWM Enabled Pins	Analog I/O Pins	DC per I/O (mA)	Flash Memory (KB)	SRAM (KB)	EEPROM (KB)	Clock (MHz)	Length (mm)	Width (mm)	Cable	Native Network Support
Uno	ATmega328	5	14	6	6	20	32	2	1	16	68.6	53.4	USB A-B	None
Leonardo	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	micro-USB	None
Micro	ATmega32u4	5	20	7	12	40	32	2.5	1	16	48	18	micro-USB	None
Nano	ATmega328	5	22	6	8	40	32	1	0.51	16	45	18	mini-B USB	None
Mini	ATmega328	5	14		6	20	32	2	1	16	30	18	USB-Serial	None
Due	Atmel SAM3X8E ARM Cortex-M3 CPU	3.3	54	12	12	800	512	96	X	84	102	53.3	micro-USB	None
Mega	ATmega2560	5	54	15	16	20	256	8	4	16	102	53.3	USB A-B	None
M0	Atmel SAMD21	3.3	20	12	6	7	256	32	X	48	68.6	53.3	micro-USB	None
Yun Mini	ATmega32u4	3.3	20	7	12	40	32	2.5	1	400	71.1	23	micro-USB	Ethernet/Wifi
Uno Ethernet	ATmega328p	5	20	4	6	20	32	2	1	16	68.6	53.4	Ethernet	Ethernet
Tian	Atmel SAMD21	5	20	12	0	7	16000	64000	X	560	68.5	53	micro-USB	Ethernet/Wifi
Mega ADK	ATmega2560	5	54	15	16	40	256	8	4	16	102	53.3	USB A-B	None
M0 Pro	Atmel SAMD21	3.3	20	12	6	7	256	32	X	48	68.6	53.3	micro-USB	None
Industrial 101	ATmega32u4	5	7	2	4	40	16000	64000	1	400	51	42	micro-USB	Ethernet/Wifi
Uno Wifi	ATmega328	5	20	6	6	20	32	2	1	16	68.6	53.4	USB A-B	Wifi
Leonardo Ethernet	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	USB A-B	Ethernet
MKR1000	Atmel SAMD21	3.3	8	12	7	7	256	32	X	48	64.6	25	micro-USB	Wifi

Advantages:

1- Ready to Use:

The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5V regulator, a burner, an oscillator, a micro-controller, serial communication interface, LED and headers for the connections. You don't have to think about programmer connections for programming or any other interface. Just plug it into USB port of your computer and that's it. Your revolutionary idea is going to change the world after just few words of coding.

2- Examples of codes:

Another big advantage of Arduino is its library of examples present inside the software of Arduino. I'll explain this advantage using an example of **voltage measurement**. For example if you want to measure voltage using ATmega8 micro-controller and want to display the output on computer screen then you have to go through the whole process. The process will start from learning the ADC's of micro-controller for measurement, went through the learning of serial communication for display and will end at **USB - Serial converters**. If you want to check this whole process click on the link below.

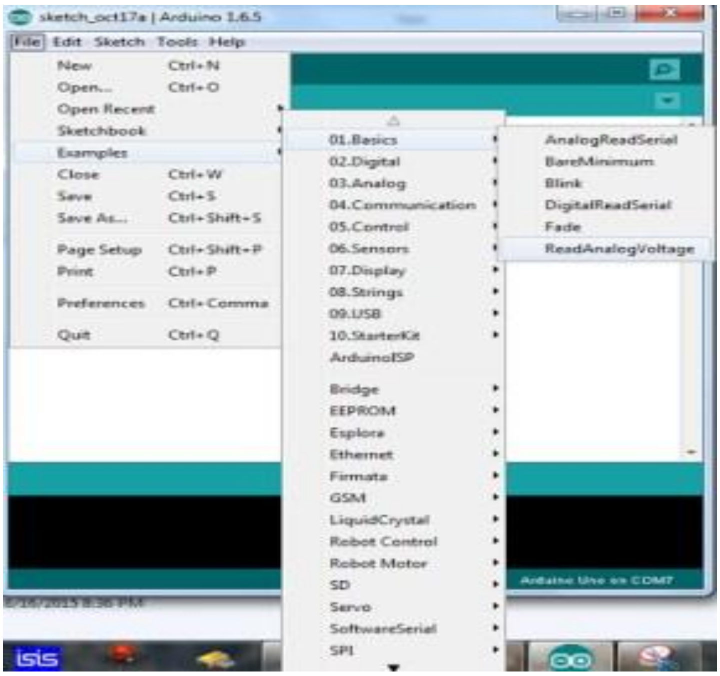

Figure 2-21 Read Analog Voltage example

DC voltage measurement using Atmel AVR micro-controller.

On the other hand, if you want to measure the voltage using Arduino. Just plug in your Arduino and open the Read Analog Voltage example as shown in the figure 1.

The project is ready after putting some reasonable resistors and zener diode. You can easily see the voltage on the Serial terminal of Arduino.

3- Effortless functions:

During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability. You can say that during debugging you don't have to worry about the units conversions.

Just use your all force on the main parts of your projects. You don't have to worry about side problems.

4- Large community:

There are many forums present on the internet in which people are talking about the Arduino. Engineers, hobbyists and professionals are making their projects through Arduino. You can easily find help about everything. Moreover the Arduino website itself explains each and every functions of Arduino.

So, We should conclude the advantage of Arduino by saying that during working on different projects you just have to worry about your innovative idea. The remaining will handle by Arduino itself.

Disadvantages:

1- Structure:

Yes, the structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible. But with the big structures of Arduino we have to stick with big sized PCB’s. If you are working on a small micro-controller like ATmega8 you can easily make your PCB as small as possible.

2- Cost:

The most important factor which you cannot deny is cost. This is the problem which every hobbyist, Engineer or Professional has to face. Now, we must consider that the Arduino is cost effective or not.

Some years' ago I was working on a project in which I had to build three smart energy meters. Now, for three smart energy meters present at some distance connected with different loads must have their own processors. So, I estimated my expenditures with and without the Arduino which you can see in the block diagram present below.

Note: I took cost of all products from Amazon with shipping charges. There may be some difference of cost in your area.

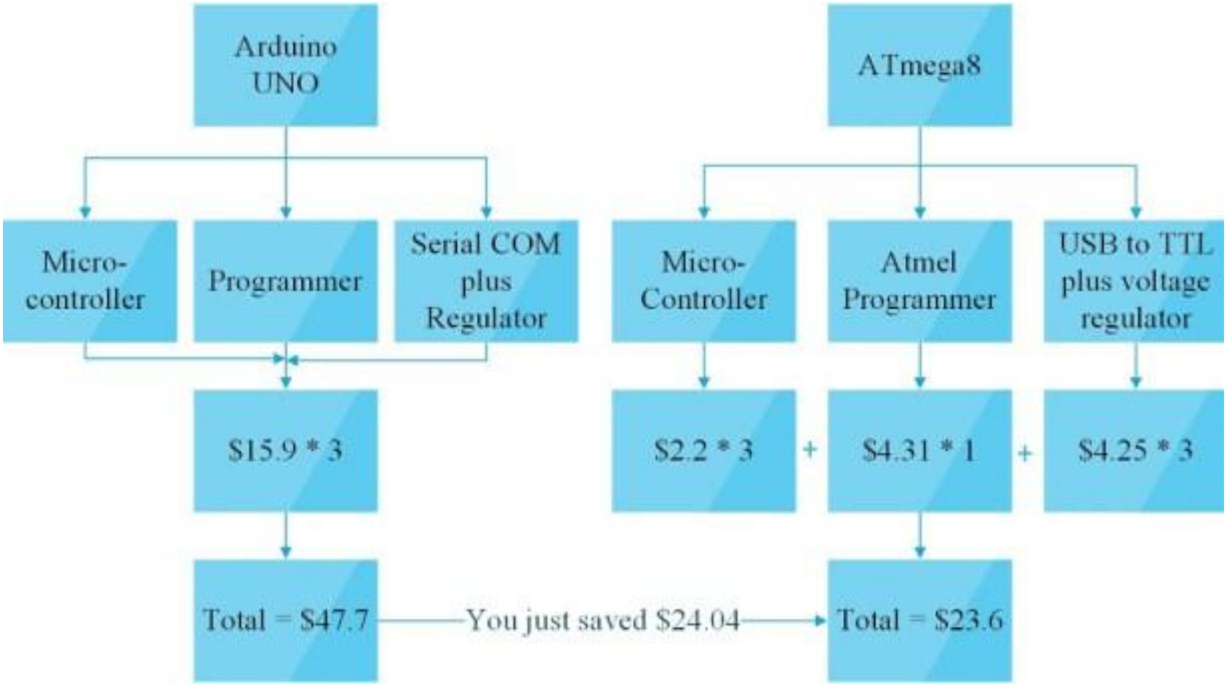

Figure 2-22 Cost Comparison of Arduino and ATmega

The thing must be noted that I multiplied Atmel Programmer with 1 because we don't need many programmers for all the micro- controllers. Only one programmer is enough. The difference between the costs is mainly due to this programmer reason. Still if you need one package then the cost difference will be as less as nearly $5 and it will rise when you have to use many packages.

3- Easy to use:

In my opinion, if you started your journey of micro-controllers with Arduino then it will be very difficult for you to make the complex intelligent circuitries in future. The easy to use hardware/software of Arduino unable a person to learn the basics of many things likes Serial communication, ADC, I2C etc.

2.14 Arduino Ethernet shield

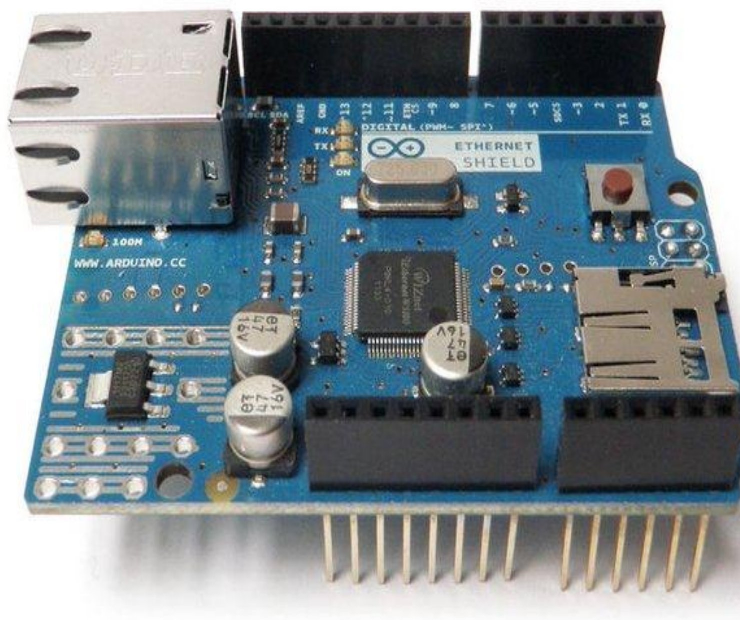

Figure 2-23 Arduino Ethernet shield

The Arduino Ethernet Shield allows you to easily connect your Arduino to the internet. This shield enables your Arduino to send and receive data from anywhere in the world with an internet connection. You can use it to do fun stuff like control robots remotely from a website, or ring a bell every time you get a new twitter message. This shield opens up endless amounts of possibility by allowing you to connect your project to the internet in no-time flat.

Setup:

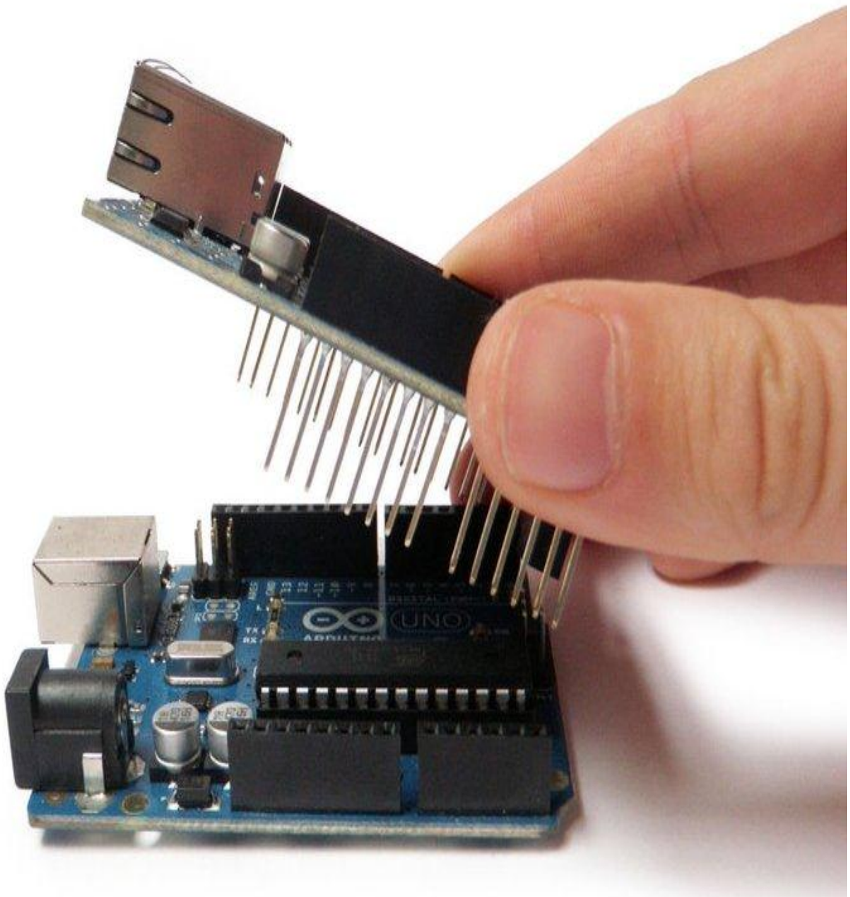

Figure 2-24 plugging the header pins

Setting it up is as simple as plugging the header pins from the shield into your Arduino.

Shield Features:

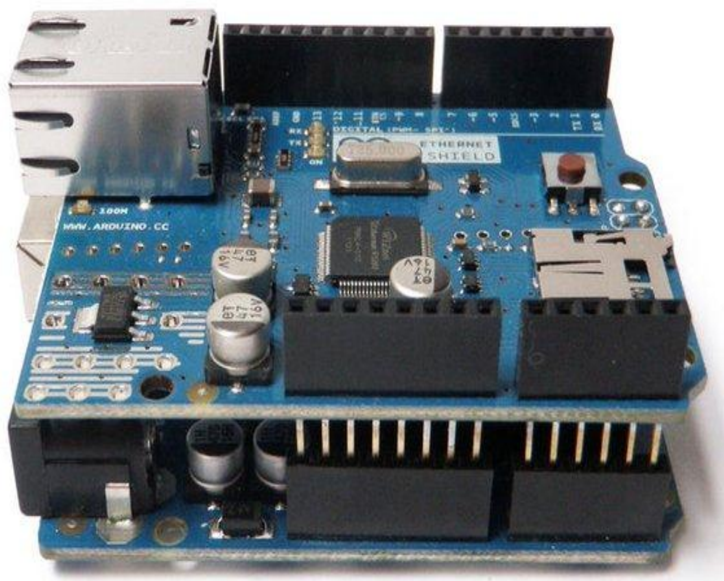

Figure 2-25 Shield Features

The Ethernet Shield is based upon the W51000 chip, which has an internal 16K buffer. It has a connection speed of up to 10/100Mb. This is not the fastest connection around, but is also nothing to turn your nose up at.

It relies on the Arduino Ethernet library, which comes bundled with the development environment.

There is also an on-board micro SD slot which enables you to store a heck-of-a-lot of data, and serve up entire websites using just your Arduino. This requires the use of an external SD library, which does not come bundled with the software.

The board also has space for the addition of a Power over Ethernet (PoE) module, which allows you to power your Arduino over an Ethernet connection.

Get Started

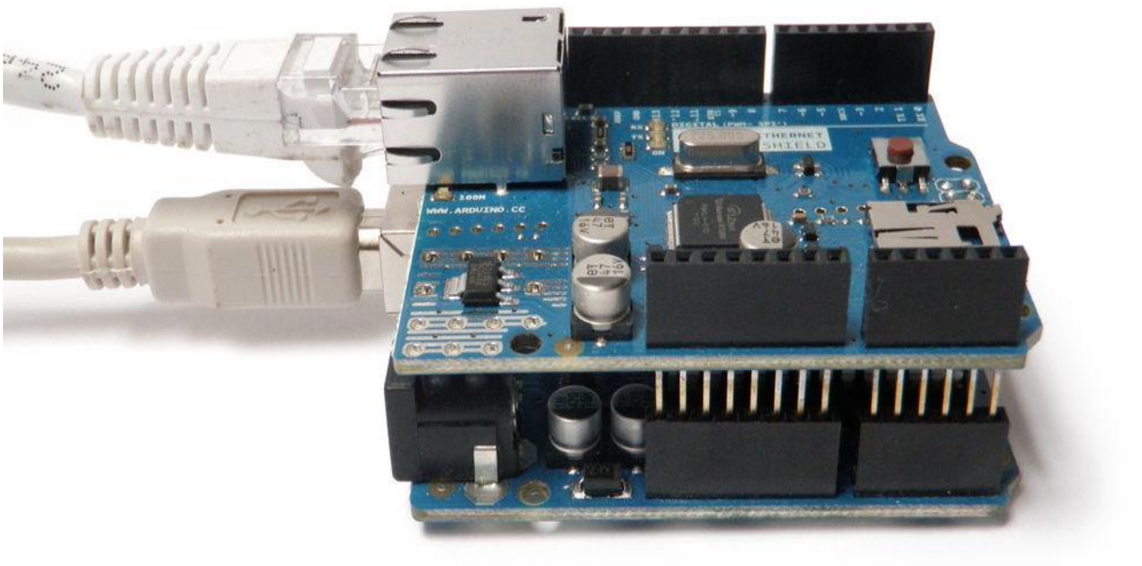

Figure 2-26 Ethernet shield

Plug the Arduino into your computer's USB port, and the Ethernet shield into your router (or direct internet connection).

Next, open the Arduino development environment. I highly recommend upgrading to Arduino 1.0 or later (if you have not done so already). This version of the software has built in DHCP support, and does not require manually configuring an IP address.

To figure out what IP address has been assigned to your board, open the DhcpAddressPrinter sketch. This can be found at:

File --> Examples --> Ethernet --> DhcpAddressPrinter

Once open, you may need to change the Mac address. On newer versions of the Ethernet shield, you should see this address on a sticker attached to the board. If you are missing a sticker, simply making up a unique mac address should work. If you are using multiple shields, make sure each has a unique mac address.

Once the mac address is properly configured, upload the sketch to your Arduino, and open the serial monitor. It should print out the IP address in use.

Chapter 3 Industry Applications Using IoT

3.1 Digital/connected factory:

IoT enabled machinery can transmit operational information to the partners like original equipment manufacturers and to field engineers. This will enable operation managers and factory heads to remotely manage the factory units and take advantage of process automation and optimization. Along with this, a digitally connected unit will establish a better line of commands and help identify key result areas (KRAs) for managers.

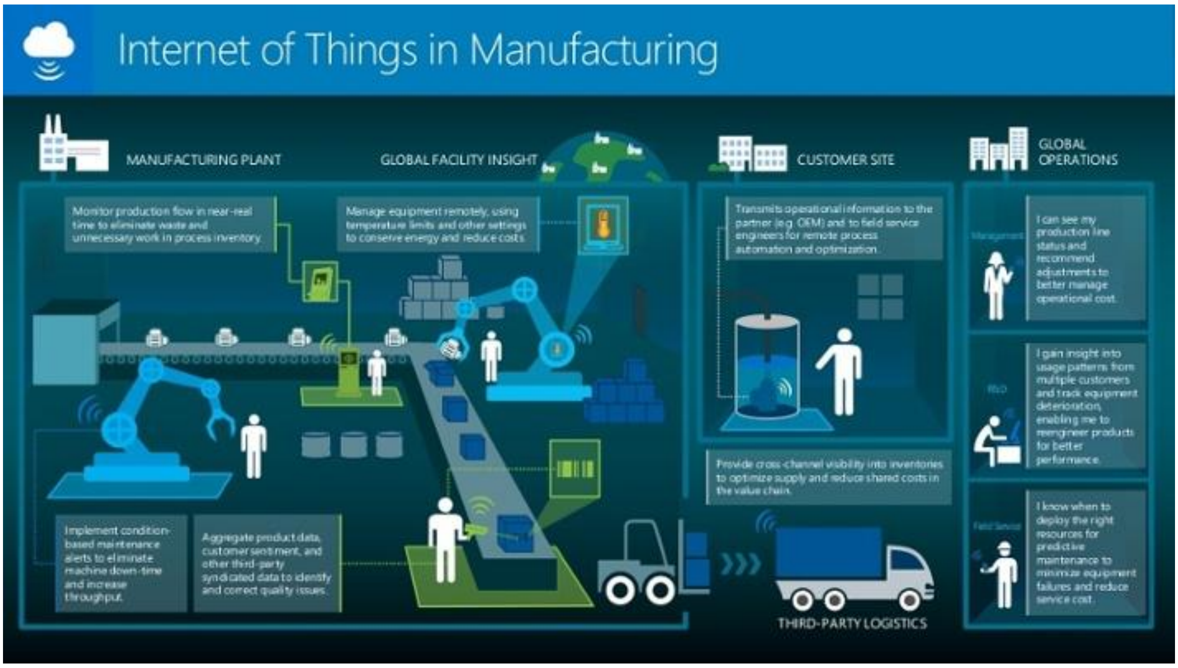

Figure 3-1 Digital/connected factory

3.2 Facility management:

The use of IoT sensors in manufacturing equipment enables condition-based maintenance alerts. There are many critical machine tools that are designed to function within certain temperature and vibration ranges. IoT Sensors can actively monitor machines and send an alert when the equipment deviates from its prescribed parameters. By ensuring the prescribed working environment for machinery, manufacturers can conserve energy, reduce costs, eliminate machine downtime and increase operational efficiency.

3.3 Production flow monitoring:

IoT in manufacturing can enable the monitoring of production lines starting from the refining process down to the packaging of final products. This complete monitoring of the process in (near) real-time provides scope to recommend adjustments in operations for better management of operational cost. Moreover, the close monitoring highlights lags in production thus eliminating wastes and unnecessary work in progress inventory.

3.4 Inventory management:

Figure 3-2 Inventory management

IoT applications permit the monitoring of events across a supply chain. Using these systems, the inventory is tracked and traced globally on a line-item level and the users are notified of any significant deviations from the plans. This provides cross-channel visibility into inventories and managers are provided with realistic estimates of the available material, work in progress and estimated the arrival time of new materials. Ultimately this optimizes supply and reduces shared costs in the value chain.

3. 5 Plant Safety and Security:

IoT combined big data analysis can improve the overall workers' safety and security in the plant. By monitoring the Key Performance Indicators (KPIs) of health and safety, like the number of injuries and illness rates, near-misses, short- and long-term absences, vehicle incidents and property damage or loss during daily operations. Thus, effective monitoring ensures better safety. Lagging indicators, if any, can be addressed thus ensuring proper redressal health, safety, and environment (HSE) issues.

3. 6. Quality control:

Figure 3-3 Quality control

IoT sensors collect aggregate product data and other third-party syndicated data from various stages of a product cycle. This data relates to the composition of raw materials used, temperature and working environment, wastes, the impact of transportation etc. on the final products. Moreover, if used in the final product, the IoT device can provide data about the customer sentiments on using the product. All of these inputs can later be analyzed to identify and correct quality issues.

3.7. Packaging Optimization:

By using IoT sensors in products and/or packaging, manufacturers can gain insights into the usage patterns and handling of product from multiple customers. Smart tracking mechanisms can also trace product deterioration during transit and impact of weather, road and other environment variables on the product. This will offer insights that can be used to re-engineer products and packaging for better performance in both customer experience and cost of packaging.

3.8. Logistics and Supply Chain Optimization:

The Industrial IoT (IIoT) can provide access to real-time supply chain information by tracking materials, equipment, and products as they move through the supply chain. Effective reporting enables manufacturers to collect and feed delivery information into ERP, PLM and other systems. By connecting plants to suppliers, all the parties concerned with the supply chain can trace interdependencies, material flow and manufacturing cycle times. This data will help manufacturers predict issues, reduces inventory and potentially reduces capital requirements.

NewGenApps specializes in building IoT solutions for varying markets. We are a strong believer in innovating with the trends and have built deep expertise in many technologies.

3.9 Application of Using a regulator and Protection circuit for Motor driver

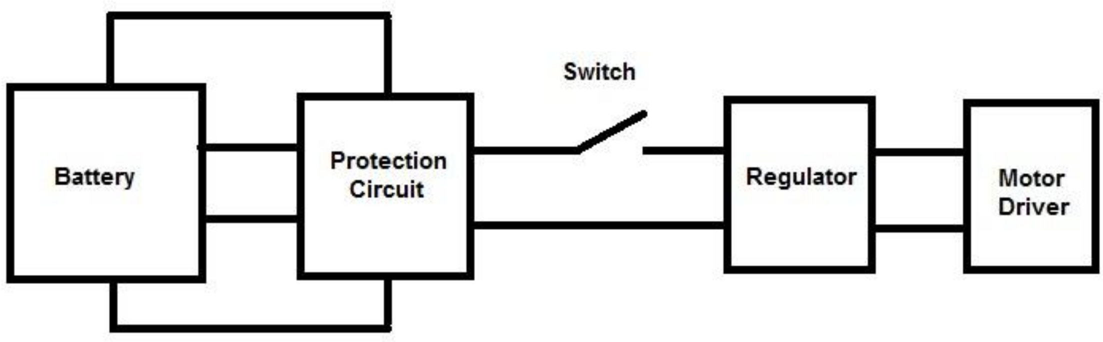

Figure 3-4 Using a regulator and Protection circuit for Motor driver

Chapter 4 System overview

Motion:

- **Motor driver:**

The aim of this to conjunction with the robot motors, and to assemble a list of commands that could be used to control the robots direction and speed.

- **Speed test:**

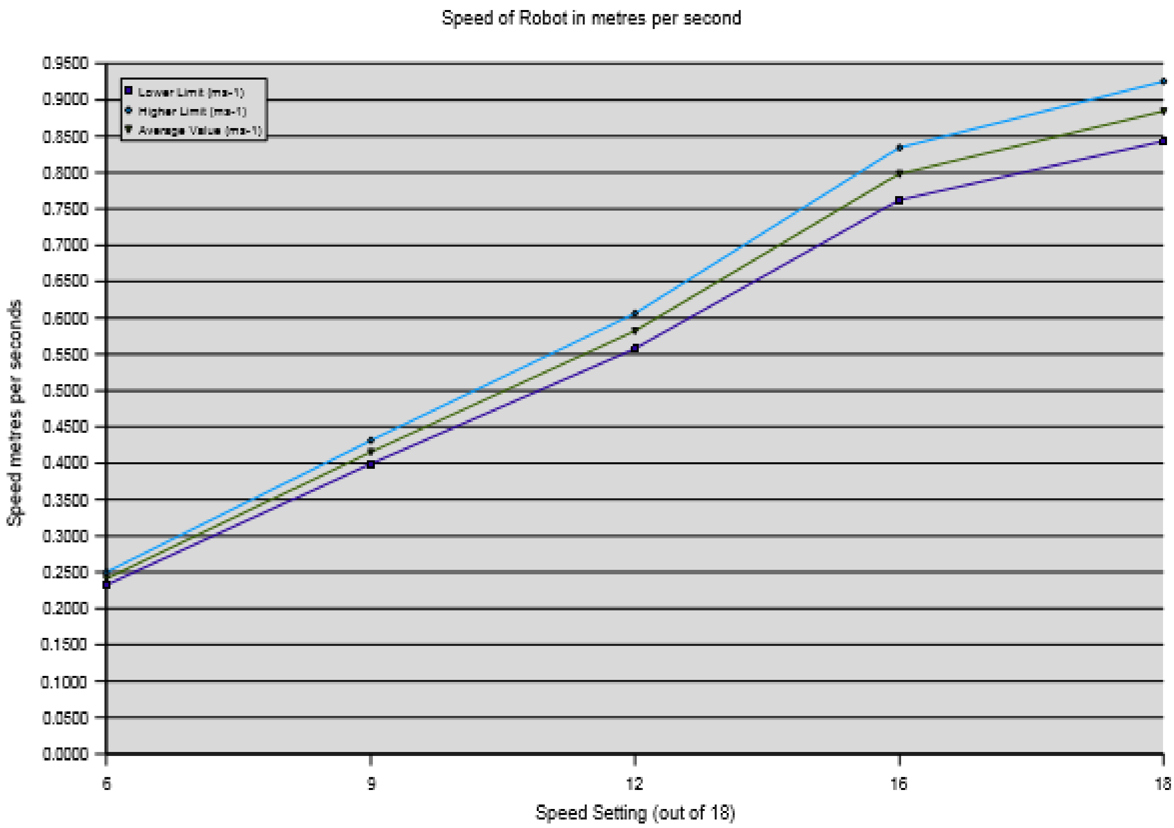

Figure 4-1 Speed of Robot in meters per second

- **Auto navigation :**

Robot anticipated designing a navigation system as part of the control system. This navigation system would consist of inertial reckoning Ultra sonic sensors.

Stream:

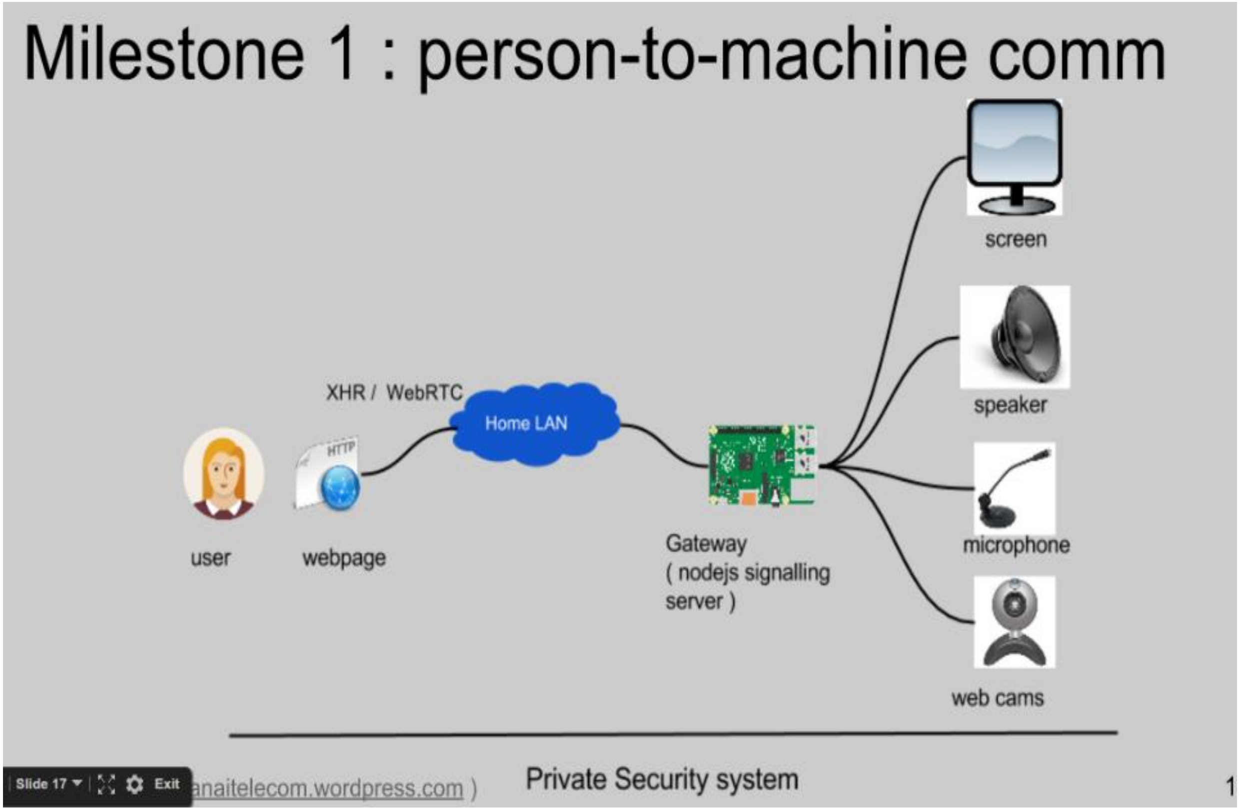

Figure 4-2 Person to machine communication

- **Video Streaming :**

video streaming system in order to better determine the communication bandwidth needed. which uses the Uv4l system to capture video from a RspberryPiCam. For initial testing and development, used the built in webcam on his laptop and programmed in Raspian. wrote code to compress a raw image to JPEG format and convert the image data into a String to be sent over Ethernet. resulted in a frame rate of 3-4 frames per second with a 2-3 second delay. This puts the computational requirements on the Computer for reading the video feed over the network directly from the camera

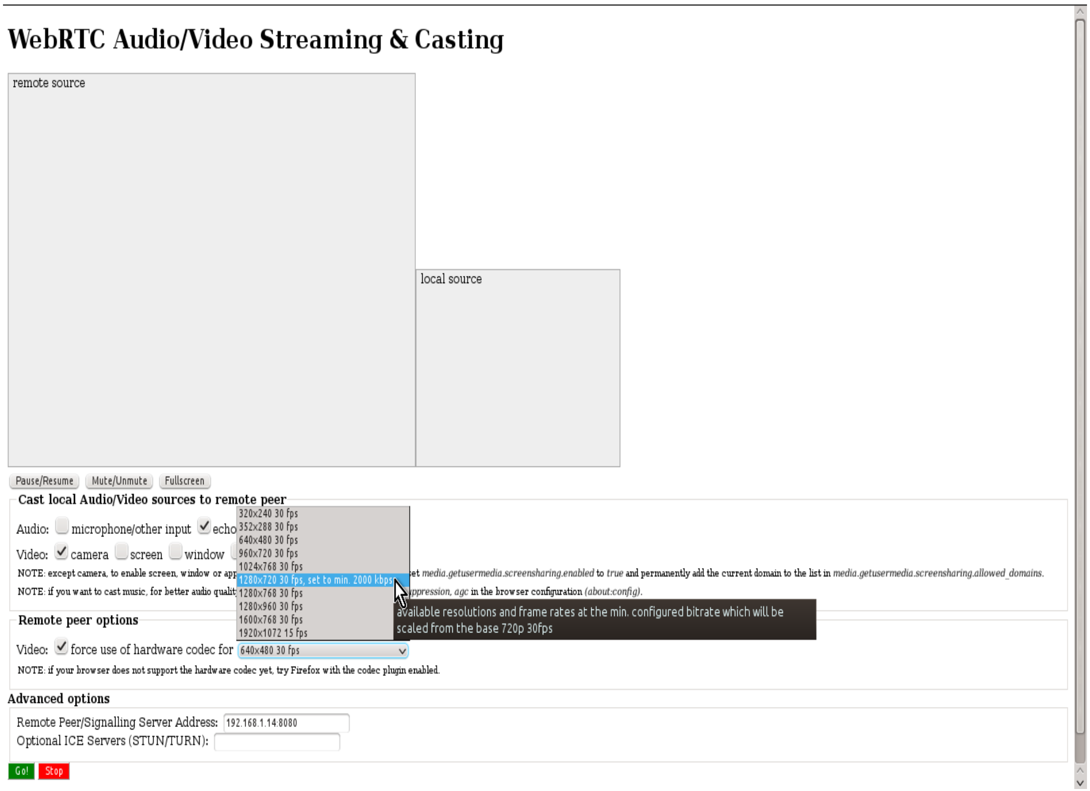

Figure 4-3 Audio streaming

- Audio streaming:

Audio streaming system in order to better determine the communication bandwidth needed. which uses the Uv4l system to stream from a USB speaker. This puts the computational requirements on the Computer for reading the Audio feed over the network directly from the speaker

RADAR:

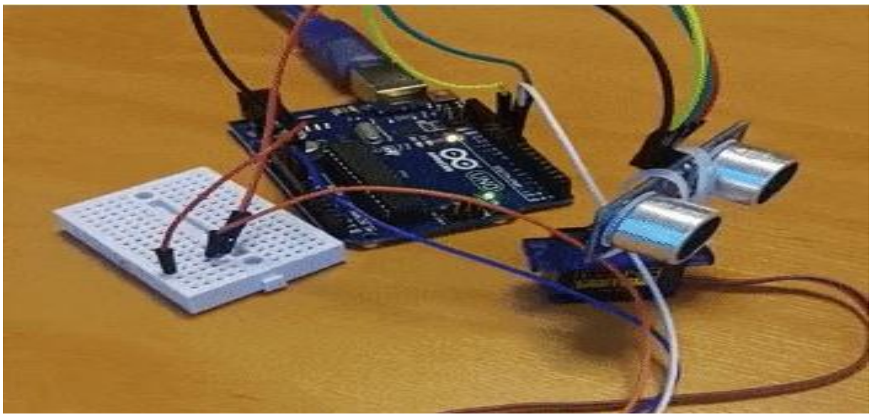

Figure 4-4 Radar circuit using Ultrsonic sensor

will create a simple system to monitor the location coordinates of the objects by manufacturing a digital radar using the HC-SR04 ultrasonic sensor device, the Arduino panel as a system controller

PIR :

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

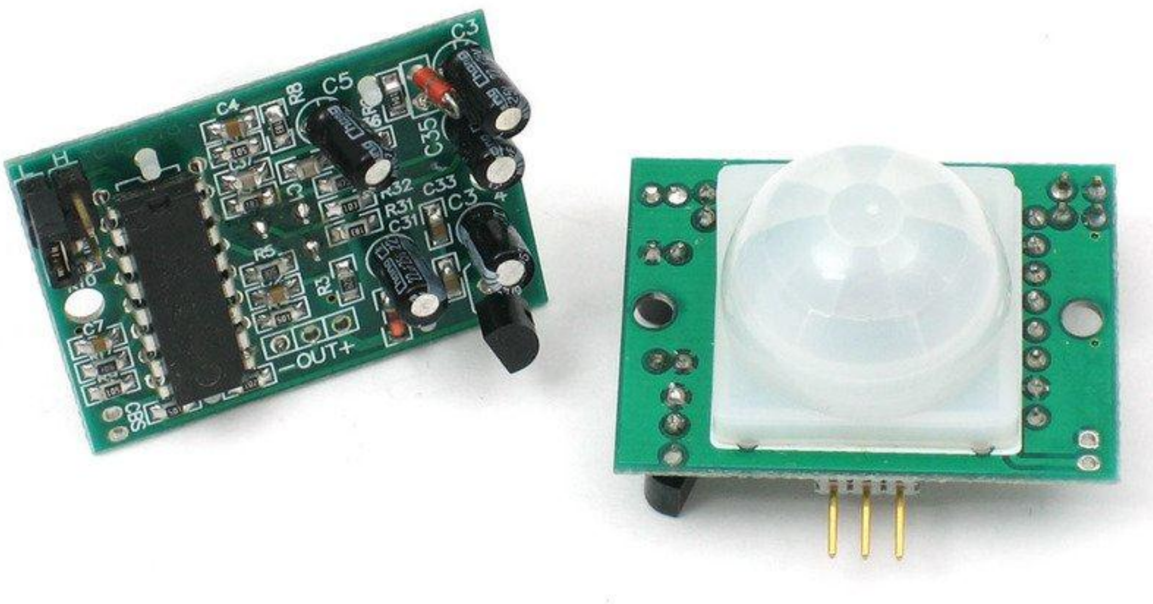

Figure 4-5 PIR Sensor

PIRs are basically made of a pyroelectric sensor (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

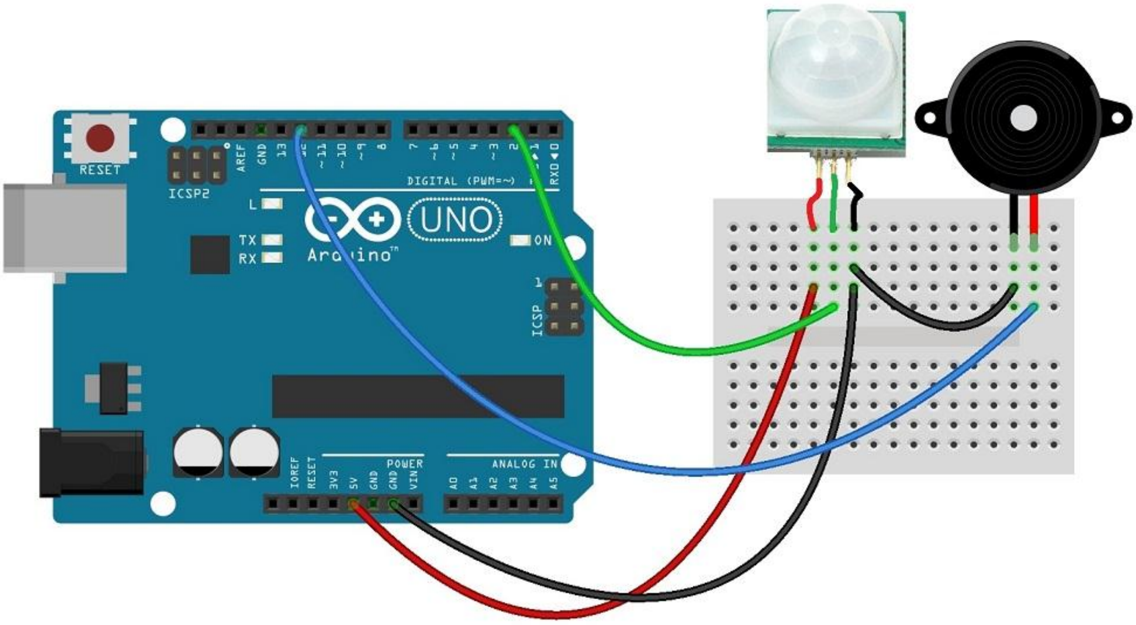

Figure 4-6 Connection of PIR sensor with Arduino

PIR sensors are more complicated than many of the other sensors explained in these tutorials (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output. To begin explaining how a basic sensor works, we'll use this rather nice diagram The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR.

The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.

Detection of motion using motion sensor – Application in project- Whenever a person is entered in the room, his entry is detected by the motion sensor placed at the door. Motion sensor detects the motion whenever a person move in front of it (within a range of sensor),then the motion signal is being read by Arduino

Chapter 5 Software Implementation

It required number of Programming Tools & Languages to build a project. Eclipse Kepler, HTML/CSS, SERVLET/JSP, Java Script, JQuery, Ajax, MySQL database. Tomcat Web Server,
MobaXtreme, WinSCP. and Putty.

5.1 Eclipse Kepler

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug in system for customizing the environment. I used Eclipse Kepler and version Figure 3.1 Eclipse 4.3.2.

5.2 HTML/CSS

Hypertext Mark-up Language, commonly abbreviated as HTML, is the standard mark-up language used to create web pages. Along with CSS. and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a language. Although most often used to set the visual style of web pages and user interfaces written in HTML.

I used HTML and CSS both for the design of website and web page to control the Robot.

5.3 JSP/Servlet

Java Server Pages (JSP) technology enables web developers and designers to rapidly develop and easily maintain, information-rich, dynamic Web pages that leverage existing business systems.

Servlets are most often used to process or store a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes which respond to requests.

5.4 Java Script

JavaScript is a high-level, dynamic, untipped, and interpreted programming language. JavaScript's typing is dynamic. JavaScript is loaded as human-readable source code. JavaScript's are prototype-based.

MySQL

MySQL is an open-source relational database management system. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open-source web application software stack. LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python".

Debian Raspberry PI Language

Debian is a Unix-like computer operating system that is composed entirely of free software, most of which is under the GNU General Public License, and packaged by a group of individuals known as the Debian Project. Raspbian Jessi Lite version 4.1 is installed in Raspberry Pi 2.

Apache Tomcat

Tomcat implements several Java EE specifications including Java Servlet, Java Server Pages (JSP), Java EL, and Web Socket, and provides a "pure Java" HTTP web server environment in which Java code can run. User has to install Apache in Eclipse and Raspberry Pi to run the java code. If the version is different and it will cause error to run different version Java code.

WinSCP (*Windows Secure Copy*) is a free and source SFTP, FTP, WebDAV and SCP client for Microsoft Windows. Its main function is secure file transfer between a local and a remote computer. Beyond this, WinSCP offers basic manager and file synchronization functionality. For secure transfers, it uses Secure Shell (SSH) and supports the SCP protocol in addition to SFTP. Raspberry Pi to Laptop file transfer WinSCP is used in project.

MobaXterm

MobaXterm is your ultimate toolbox for remote computing. In a single Windows application, it provides loads of functions. Raspberry Pi is open in the MobaXterm application and it's easier to open Remote Desktop or code in the Raspberry Pi.

Putty is an SSH and telnet client, developed originally by Simon Tat ham for the Windows platform. Putty is open source software that is available with source code and is developed and supported by a group of volunteers. IP address of Raspberry Pi have to enter and raspberry pi is ready for programming.

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

Linux System Administration

Kernel and Distribution:

Although only the kernel itself is rightly called Linux, the term is often used to refer to a collection of different opensource projects from a variety of companies. These collections come together to form different flavors of Linux, known as distributions

File System Logical Layout :

boot: This contains Linux kernel and other packages needed to start the Pi

bin: OS-related binary files, like those required to run the GUI, are stored here.

dev : Virtual directory, which doesn't actually exist on the SD card. All devices connected to the system can be accessed from here.

etc: This stores miscellaneous configuration files, including the list of users and their encrypted passwords

home: Each user gets a subdirectory beneath this directory to store all their personal files

lib: This is a storage space for libraries, which are shared bits of code required by different applications.

lost+found: A special directory where file fragments are stored if the system crashes.

media: A special directory for removable storage devices, like USB memory sticks or external CD drives**.**

mnt: This folder is used to manually mount storage devices, such as external hard drives.

opt: This stores optional software that is not part of the OS itself. If you install new software to your Pi, it will usually go here.

proc: Another virtual directory, containing information about running programs which are known in Linux as processes.

selinux: Files related to Security Enhanced Linux, a suite of security utilitesoriginally developed by the US National Security Agency.

sbin: Stores special binary files, primarily used by the root account for system maintenance**.**

sys: This directory is where special OS files are stored**.**

tmp: Temporary files are stored here automatically**.**

usr: This directory provides storage for user accessible programs**.**

var: This is virtual directory that programs use to store changing values or variables**.**

Software:

LXTerminaland Root Terminal: use the Linux command line in a window without leaving the GUI.

Midori & NetSurf: Lightweight web browser

IDLE and IDLE 3: IDE for Python 2.7 and 3

Task Manager: Checks the available memory, processor workload, closes crashed or unresponsive programs

Music player at the console: moc

OpenOffice.org: sudoapt-get install openoffice.org

Image Editing: Gimp

LAMP (Linux, Apache, MySQL and PHP) stack Sudoapt-get install apache2 php5 php5-mysql mysql-server.

Installing, Uninstalling and Updating Software:

•Package manager in Debian: apt

•GUI for apt, Synaptic Package Manager doesn't work well on Pi due to the lack of memory

•Make sure that the apt cache is up to date:

•apt-get update

•Finding software:

•apt-cache search emacs

•Installing software and dependencies:

•sudoapt-get install emacs

•Uninstalling software:

•sudoapt-get remove emacs

•sudoapt-get purge emacs(removes everything including configurations)

•Upgrading software:

•Sudoapt-get upgrade

•Sudoapt-get install emacs

Troubleshooting:

Keyboard and Mouse Diagnostics

Power Diagnostics

Display Diagnostics

Network Diagnostics

Emergency Kernel

Wired Networking Configuration:

sudonano/etc/ntework/interfaces

ifaceeth0 inetstatic [tab] address 192.168.0.10 [tab] netmask255.255.255.0 [tab] gateway 192.168.0.254

sudo/etc/init.d/networking restart

sudonano/etc/reslov.conf

nameserver8.8.8.8 nameserver8.8.4.4

sudo/etc/init.d/networking restart

ping –c 1 www.raspberrypi.org

Wireless Networking Configuration

•USB Wi-Fi adapters are very power-hungry. Connect a powered USB hub to the Pi, and then insert the Wi-Fi adapter into that.

•Print out the entire kernel ring buffer and find out the company that makes the actual chip: mesg| grep^usb

- Atmel-firmware
- Firmware-atheros
- Firmware-brcm80211

- Firmeware-intelwimax
- Firmware-ipw2x00
- Firmware-iwlwifi
- Firmware-ralink
- Firmware-realteck

•Check the current status of the network: iwconfig

Configuration of Raspberry Pi:

RPi doesn’t have a BIOS menu. It relies on text files containing configuration strings that are loaded by the chip when powers on.

- Hardware settings: config.txt
- Memory Partitioning: start.elf
- Software Settings: cmdline.txt

Arduino Coding:

```
#include <SPI.h>
#include <Ethernet.h>
#include <Client.h>
```

Libraries used

```
int x=0;
int sw;
int time = 570;
const int ledPin =1;
// pir
#define LDR 30
// buzzer
#define SPEAKER 31
//for motor driver 1
int enA = A0;
int IN1 = 6;
int IN2 = 7;
int enB = A5;
int IN3 = 8;
int IN4 = 9;
//for ultrasonic
const int trigPinFront = A1;
const int echoPinFront = A2;
```

Initialize Pins

```
//motor driver1 pins
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(enA,OUTPUT);
pinMode(enB,OUTPUT);

//LDR
pinMode(LDR, INPUT);
pinMode(SPEAKER, OUTPUT);

//Ultrasonic
pinMode(trigPinFront, OUTPUT);
pinMode(echoPinFront, INPUT);
```

Mac address of Ethernet shield

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 0, 9); // IP address, may need to change depending on network
EthernetServer server(80);  // create a server at port 80
```

Initialize Ethernet Device

```
Ethernet.begin(mac, ip);
server.begin();
Serial.begin(9600);
```

Try to get client

```
EthernetClient client = server.available();
if (client) {
boolean currentLineIsBlank = true;
while (client.connected()) {
if (client.available()) {
char c = client.read();
if (webClickRequest.length() < 100) {
webClickRequest += c;  }
}
```

The WebPage sent

```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
// send web page
client.println("<!DOCTYPE html>");
client.println("<HTML>");
client.println("<HEAD>");
client.println("<TITLE>Internet Controlled RC Car</TITLE>");
client.println("<STYLE>");
client.println("body{margin:50px 0px; padding:0px; text-align:center;}");
client.println("h1{text-align: center; font-family:\"Trebuchet MS\",Arial, Helvetica, sans-serif; font-size:24px;}");
client.println("a{text-decoration:none; width:75px; height:50px; border-color:black; font-family:\"Trebuchet MS\",Arial, Helvetica, sans-serif; padding:6px; background-color:#aaaaaa; text-align:center; border-radius:10px 10px 10px; font-size:24px;}");
client.println("a:link {color:white;}");
client.println("a:visited {color:white;}");
client.println("a:hover {color:red;}");
client.println("a:active {color:white;}");
client.println("</STYLE>");
client.println("</HEAD>");
client.println("<BODY>");
client.println("<H1>Internet Controlled Car</H1>");
client.println("<br />");
client.println("<br />");
client.println("<a href=\"/?left\"\">LEFT</a>");
client.println(" ");
client.println("<a href=\"/?forward\"\">FORWARD</a>");
```

```
client.println("<a href=\"/?right\"\">RIGHT</a>");
client.println(" ");
client.println("<a href=\"/?back\"\">BACK</a>");
client.println(" ");
client.println("<a href=\"/?STOP\"\">STOP</a>");
client.println("<br />");
client.println("<br />");
client.println("<br />");
client.println("<a href=\"/?sw_On\"\">Auto Driver ON</a>");
client.println(" ");
client.println(" ");
client.println("<a href=\"/?sw_Off\"\">Auto Driver OFF</a>");
client.println(" ");
client.println(" ");
client.println("<a href=\"/?ledon\"\">Turn led on</a>");
client.println(" ");
client.println(" ");
client.println("<a href=\"/?ledoff\"\">Turn led off</a>");
client.println(" ");
client.println("<a href=HTTPS://192.168.0.100:8080>Streaming Video</a>");
client.println("<br />");
client.println("<br />");
client.println("<br />");
client.println("<br />");
client.println("</BODY>");
client.println("</HTML>");
delay(1);      // give the web browser time to receive the data
client.stop(); // close the connection
```

Commands from WebPage to Arduino

```
if(webClickRequest.indexOf("?left") > 0){
Serial.println("hello");
left();
delay(100);
brake();

}
else if(webClickRequest.indexOf("?forward") >0){
forward();
delay(600);
brake();
}
else if(webClickRequest.indexOf("?right") >0){
right();
delay(100);
brake();
}
else if(webClickRequest.indexOf("?back") >0){
reverse();
delay(600);
brake();
}
else if(webClickRequest.indexOf("?STOP") >0){
brake();
delay(10000);
brake();
}
```

```
else if(webClickRequest.indexOf("?ledon") >0){
        digitalWrite(ledPin,1) ;
    }
else if(webClickRequest.indexOf("?ledoff") >0){
        digitalWrite(ledPin,0) ;
    }
else if(webClickRequest.indexOf("?sw_On") >0){
        x=1;
    }
else if(webClickRequest.indexOf("?sw_Off") >0){
        x=0;
        brake();
    }
```

Motion Function

```
void forward(){
analogWrite(enA,128);
analogWrite(enB,128);
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
}
void reverse(){
digitalWrite(IN3,HIGH);
```

```
digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
}
void left(){
analogWrite(enA,128);
analogWrite(enB,128);
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
}
void right(){
analogWrite(enA,128);
analogWrite(enB,128);
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
}
void brake(){
analogWrite(enA,128);
analogWrite(enB,128);
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);
}
```

Rasperry Pi Programming:

Building Stream Audio and Video With Webrtc

Initial Raspberry Pi setup

- **sudo apt-get update**
- **sudo apt-get install libmariadbclient18 libpq5 libavcodec57 libavformat57 libavutil55 libswscale4**

Setup Uv4l Software

```
• $ curl http://www.linux-projects.org/listing/uv4l_repo/lrkey.asc | sudo apt-key add -
```

add the following line to the file ***/etc/apt/sources.list*****:**

```
• deb http://www.linux-projects.org/listing/uv4l_repo/raspbian/stretch stretch main
```

Finally, ready to update the system and to fetch and install the packages:

```
• $ sudo apt-get update

• $ sudo apt-get install uv4l uv4l-raspicam
```

The above two commands will upgrade UV4L to the most recent version, if it's already installed.

If you want the driver to be loaded at boot, also install this optional package:

- `$ sudo apt-get install uv4l-raspicam-extras`

As a convenience, the above package will install a service script for starting, stopping or restarting the driver at any time, for example:

- `$ sudo service uv4l_raspicam restart`

For detailed informations, options, etc… about the modules installed type accordingly:

- `$ man uv4l`
- `$ man uv4l-raspicam`

Install the driver for the Raspberry Pi Camera Board, the following Streaming Server front-end and drivers can be optionally installed:

- `$ sudo apt-get install uv4l-server uv4l-uvc uv4l-xscreen uv4l-mjpegstream uv4l-dummy uv4l-raspidisp`

for which the manual pages are available:

- $ man uv4l-server
- $ man uv4l-uvc
- $ man uv4l-xscreen
- $ man uv4l-mjpegstream

The *WebRTC* extension for the Streaming Server is:

- $ sudo apt-get install uv4l-webrtc

As the Streaming Server is able to serve and run any custom web applications, an optional package containing the source code of some of the demos mentioned in the examples is available. The files will be installed in */usr/share/uv4l/demos/*:

- $ sudo apt-get install uv4l-demos

Note that some browsers may no longer exploit many of the WebRTC functionalities over HTTP for security reasons.will need to configure secure *HTTPS* in the Streaming Server instead. To do this, must provide a **password-less private key** and a valid **certificate** via the *–ssl-private-key-file* and the *–ssl-certificate-file* server options. A private key and a self-signed certificate can be generated as follows:

```
• $ openssl genrsa -out selfsign.key 2048 && openssl req -new -x509 -key selfsign.key -out selfsign.crt -sha256
```

Once you have installed and eventually configured the HTTP(S) Streaming Server module as shown above, make sure to reload *uv4l* for it to notice and start the server. Afterwards you can access the server with the browser at the default address and port *https://raspberryip:8080/* (where *raspberry* has to be replaced with the actual hostname or IP address of your RaspberryPi and the protocol can be either *http* or *https*).

Chapter 6 Results

The aim of the project is to develop a Robot on IoT based concept.It is working as buddy or family Member because you have to command it and control from anywhere at any time. If a personal wants to find something he/she has to command it from live steaming can see the actual scenario at that place and easily find out that object. It works as to take care for children's; pet at home, too.

6.1 Flow Chart

The flow chart describe the necessary steps to execute once in the beginning, then when the Raspberry Pi turn on all the compilation files and video streaming file execute at its own using startup.sh file

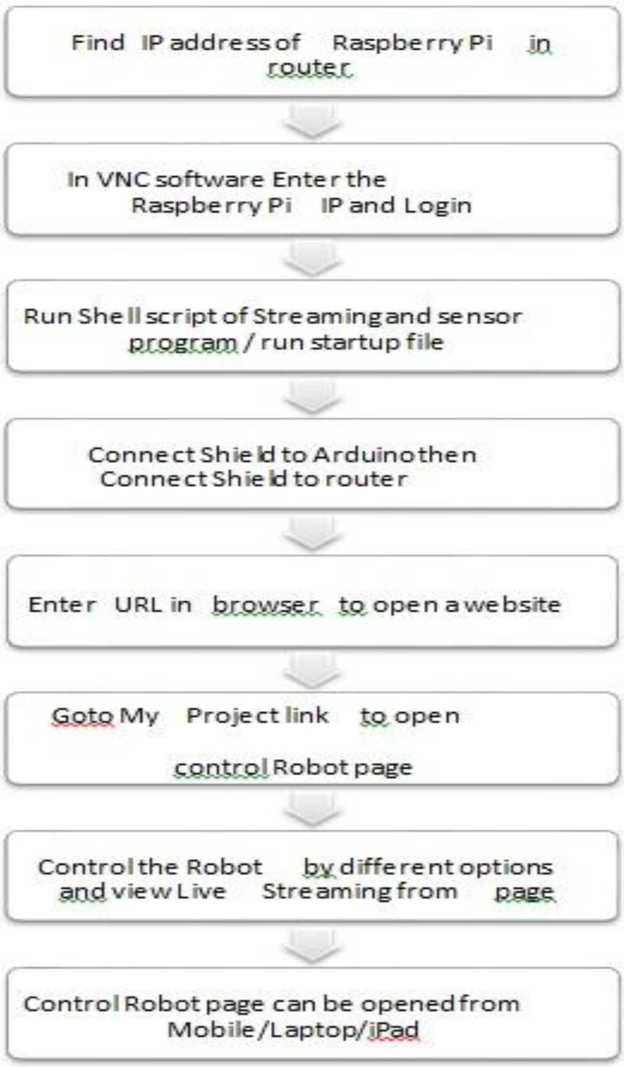

Figure 6-1 Flow chart

Carrying out unit test is kind of a pre-release of the system. Performing unit test in presence of supervisor ensures that final product had met all requirements. Unit test results listed described in below showing table:

Table 6-1 Result Set

No.	Test case description	Test Result
1	**Webcam image display**	**Accepted**
2	**Move Forward**	**Accepted**
3	**Move Reverse**	**Accepted**
4	**Turn Left**	**Accepted**
5	**Turn Right**	**Accepted**
6	**Stop**	**Accepted**
7	**Ultrasonic sensor reading**	**Accepted**
8	**PIR Sensor reading**	**Accepted**
9	**Auto Driving ON**	**Accepted**
10	**Auto Driving OFF**	**Accepted**
11	**Microphone Audio**	**Accepted**
12	**LED**	**Accepted**

Chapter 7 Discussion / Conclusion / Future Work

7.1 Discussion

During the whole period of the project I gained of lot of knowledge on the raspberry pi , arduino, motors and programming in C. If we talk about the achievements out of the project when starting to do the project it was to control the motors using Arduino on a robot and transmit that data via any wireless technology to another device and able to collect the data and control the robot or raspberry pi in a real time instance. Out of which all the work was completed. The main achievements that I gained out of the project were that I got to learn programming in C and could learn how to program a user interface web page. Another main achievement was I could learn and understand the raspberry pi technology, the wide applications of raspberry pi and IoT. There are lots of many other areas where the raspberry pi could be used for robotic applications and that are the reasons for me to choose this project.

7.2 Conclusions

To get to the aim of a project there will be always a set of objectives, to achieve that objectives we need to know how where and with what resource is the step towards completing the objectives taken. Now in this project too to get to the aim of the project there was a set of objectives, which gradually changed as the project research was completed and then while testing a certain technology the objectives again changed due to the failure of the method. Now the first thing of the project is a good research, I had to do a wide and a strong research before I started to put my objectives as this technology was new in market.

The research for the project was done using Advanced Google search and also from the search engines available in the student portal like tutorials, pi4j, w3school and raspberry pi.

The Google advanced search is the one that was more widely used as it is a new technology and there are very less articles or journals published regarding the raspberry pi technology and IoT.

Each stage of the project was tested after every part of it was completed and then moved on to the next one. During the course of the project I gained knowledge of Java I also gained knowledge of the raspberry pi technology and what the small computer is capable of. After knowing the capabilities of raspberry pi and the applications it could have in the field of robotics, and IoT it actually has made me to think of doing more research work on the raspberry pi for the robotic and IoT applications.

The challenges that I faced during the course of the project were that of the time constrain, as I had to learn about the raspberry pi and then learn programming in Java and HTML. Then during the programming of the server client interfaces the problems of calling functions with a button press. One of the main challenges that No output comes when some functions are called from software side. Other than the small problem the buddy robot works fine and meets all its purpose.

If given an opportunity to work again on the same technology i.e. the raspberry pi and IoT technology or on a project like this where the raspberry pi is used for any kind off application I would be happy to take it up.

7.3 Future Work

In the future this raspberry pi technology can be used in various different fields of work. The buddy robot can be made autonomous with the help of more sensor, gyroscope, compass and a GPS. So that it can be set to a target or a specific area where in can monitor. The robot can also be developed into an advanced robot toy for young people. Others future works described below:

- Face recognition: All the family members face images are stored in controller when an unknown person will come at door, it will create alert and click the image and send it to user.
- In changing the Mechanical design work using the same concept, different functions as Open the door, Turn on/off switch, bring newspaper for user, etc work can be done.
- Adding the Pneumatics design in Mechanical design robot can walk, go up and down and it will be control from anywhere at any time.

References

[1] **About Raspberry Pi, http://www.raspberrypi.org/about/**

[2] **Early prototype of Raspberry Pi, http://www.raspberrypi.org/raspberry-pi-2006-edition/**

[3] **Raspberry Pi Hardware information, http://elinux.org/RPi_Hardware**

[4] **How SoC works, http://www.androidauthority.com/how-it-works-systems-on-a-chip-soc-93587/**

[5] **Boot process, http://thekandyancode.wordpress.com/2013/09/21/how-the-raspberry-pi-boots-up/**

[6] **List of available OS, http://elinux.org/RPi_Distributions**

[7] **Home automation project, http://www.instructables.com/id/Raspberry-Pi-GPIO-home-automation**

[8] **Arcade gaming machine project, http://www.cnet.com/how-to/25-fun-things-to-do-with-a-raspberry-pi/**

[9] **Robot project, http://www.wired.com/2013/01/raspberry-pi-r2d2/**

[10] **Landline telephone project, https://www.twilio.com/blog/2013/03/build-a-twilio-hard-phone-with-sipfrom-twilio-raspberry-pi-asterisk-freepbx-and-the-obihai-obi100.html**

[11] **Decoration lights project, http://www.element14.com/community/community/raspberrypi/raspberrypi_projects/blog/2013/12/12/raspberry-pi-enabled-christmas-lights**

[12] **Advantages-Disadvantages, https://sites.google.com/site/mis237groupprojectraspberrypi/home/what-israspberry-pi/pros-and-cons-of-the-raspberry-pi**

[13] **HTML for webpage, http://www.tutorialspoint.com/html/**

[14] **MobaXteme, http://mobaxterm.mobatek.net/**

[15] **Design and build a Raspberry Pi robot By Stewart Watkiss, (PenguinTutor)**

[16] **Rasp-Pi based Remote Controlled Smart Advertising of Still and Moving Images, International Journal of Engineering and Computer Science, Volume 4 Issue 9 Sep 2015.**

[17] **Build and Interface Internet Mobile Robot using Raspberry Pi and Arduino by Prof. Dr. Nabeel Kadim**

Abid Al-Sahi, Innovative Systems Design and Engineering Vol.6, No.1, 2015

[18] "Arduino - Introduction". arduino.cc.

[19] David Kushner (2011-10-26)

[20] "The Making of Arduino". IEEE Spectrum.

[21] Justin Lahart (27 November 2009).

[22] "Taking an Open-Source Approach to Hardware". The Wall Street Journal. Retrieved 2014-09-07.

[23] Hernando Barragán (2016-01-01)

[24] . "The Untold History of Arduino". arduinohistory.github.io. Retrieved 2016-03-06.

[25] "Rhizome - Interview with Casey Reas and Ben Fry". 2009-09-23. Retrieved 2014-08-23.

[26] "How many Arduinos are "in the wild?" About 300,000". Adafruit Industries. May 15, 2011. Retrieved 2013-05-26.

[27] "Arduino FAQ – With David Cuartielles". Malmö University. April 5, 2013. Retrieved 2014-03-24.

[28] "Arduino's New CEO, Federico Musto, May Have Fabricated His Academic Record". WIRED. Retrieved 2017-12-22.

[29] "Free Arduino | Make:". Make: DIY Projects and Ideas for Makers. 2017-06-09. Retrieved 2017-12-22.

[30] "The Arduino Foundation: What's Up?". Hackaday. 2017-06-19. Retrieved 2017-12-22.

[31] . "CEO controversy mars Arduino's open future". TechCrunch. Retrieved 2017-12-22.

[32] "Arduino reborn partners with ARM". Electronics Weekly. 2017-10-06. Retrieved 2017-11-03.

[33] "Business Entity Summary for Arduino LLC". Mass.gov. State of Massachusetts.Allan, Alasdair (6 March2015).

[34] "Arduino Wars: Group Splits, Competing Products Revealed?". makezine.com. Maker Media, Inc. Retrieved 21 April 2015.

[35] "Massimo Banzi: Fighting for Arduino". makezine.com. Maker Media,

[36] Williams, EllIoT (28 March 2015).

[37] "Arduino SRL to Distributors: "We're the Real Arduino"". Hackaday.com. Hackaday.com. Retrieved 21 April 2015.

[38] "Arduino LLC vs Arduino SRL lawsuit; United States Courts Archive". Retrieved 20 February 2018.

Problems:

1. There were some problems in the H-bridge driver in terms of the Power.
2. Install a free wheel instead of front motors but found it would be difficult to control during the rotation**.**
3. There were some problems in the code in terms of programming Motion through the web page
   (Conroling from the web page).
4. There were problems with auto driving and that was the problem of Ultrasonic on the edges of the robot.
5. Radar can't be read distance in Meter.
6. Speed of the DC motors only 60% percentage from the total speed.
7. problem that the Auto button driving in the web page does not work**.**
8. There was a problem in the movement of the robot left and right where he was fullcycle around itself.
9. There was a problem installing the radar where he does not read the distance between the robot and the closest object for him on a web page
10. Problem that Rasperry pi software (Rasbian Stretch) with Streaming Audio and Video.
11. Using Usb cam with buit in mic.
12. Stream Audio from the robot to the web server when using (MJPEG software) .
13. Stream Video from the robot to the web server when using (FFMPG software) .
14. Problem with Web server IP from the Router.

Solutions:

1. Solve the H-bridge driver by 18650 battaries

   the Battary18650 has less resistance battery gives more power and torque is greater

   that the problem with AA batteries have high resistance gives current least this gives less torque

   also use boost converter 3 batteries 3.7 gives 11.1 and boost covertor increases to 12 using 3 batteries 18650 3.7 volts .... 3 batteries 11.1 laeken relgalator modify the voltage to give the 12-volt power outlet in the sense of more than 0.9 volts

also use protection circuit where makes the battery and go less than 3 volt damage dc motor arrived 3 volt Cuts Power.

2. Solve the free wheel problem by replacing to dc motor other front we connect the same H-bridge

   driver with rear motors.

3. Solve the controlling motion made some amendments in the Code so that the correct movement through the function right & left.
4. Solve the problem of the full cycle when using left or right button in web server by reducing the delay time of the move to the right and left in the Code.
5. Solve the problem of reading radar distance by insert the function that convert from ‘FT’ to ‘Cm’ (distanceCm= distanceft/58).
6. Solve the problem of Rasperry pi by update and upgrade the software of raspian stretch to the last version.
7. Solve the problem of Usb cam by replace to Raspicam or Usb cam support MAC.
8. Solve Streaming problem by using “UV4L” Software and using (WEBRTC) to stream audio and video at the same time without any delay and support many diveces also.
9. Solve the problem of Webserver IP by change the range of the IPV4 in the router to the maximum.

Photos of the project:

Figure 7-1 Project-overview

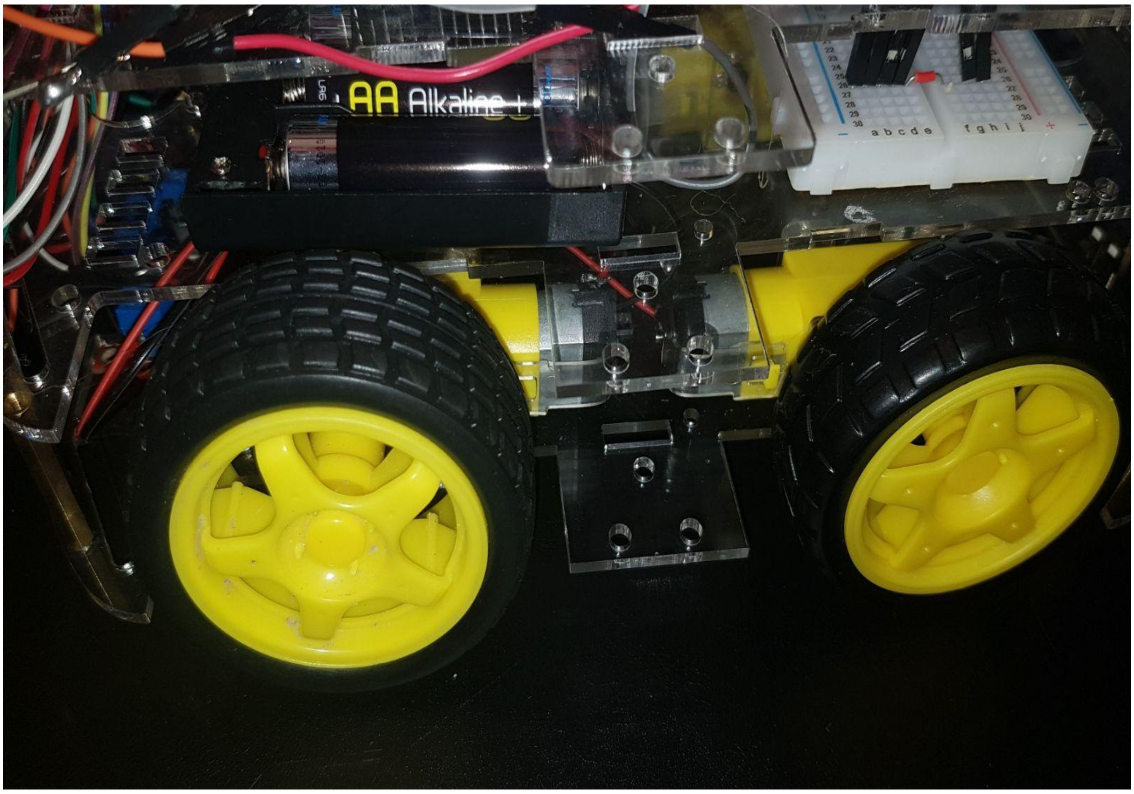

Figure 7-2 Side view

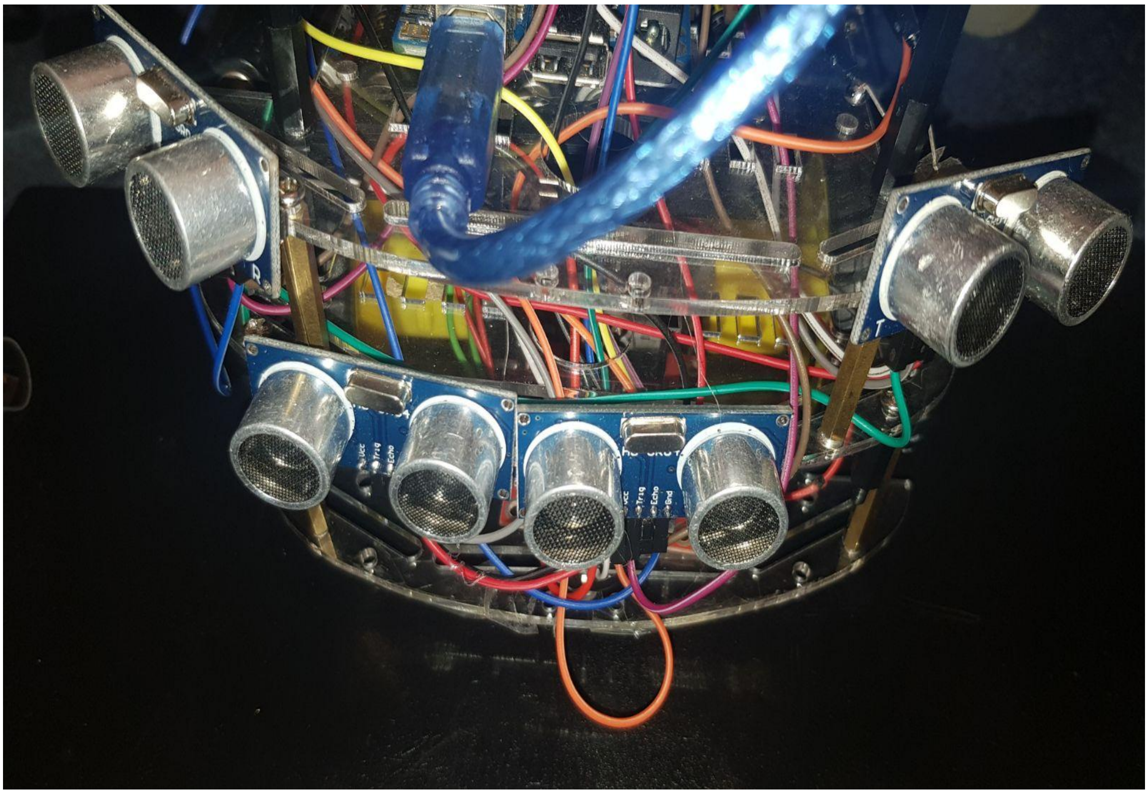

Figure 7-3 Front view

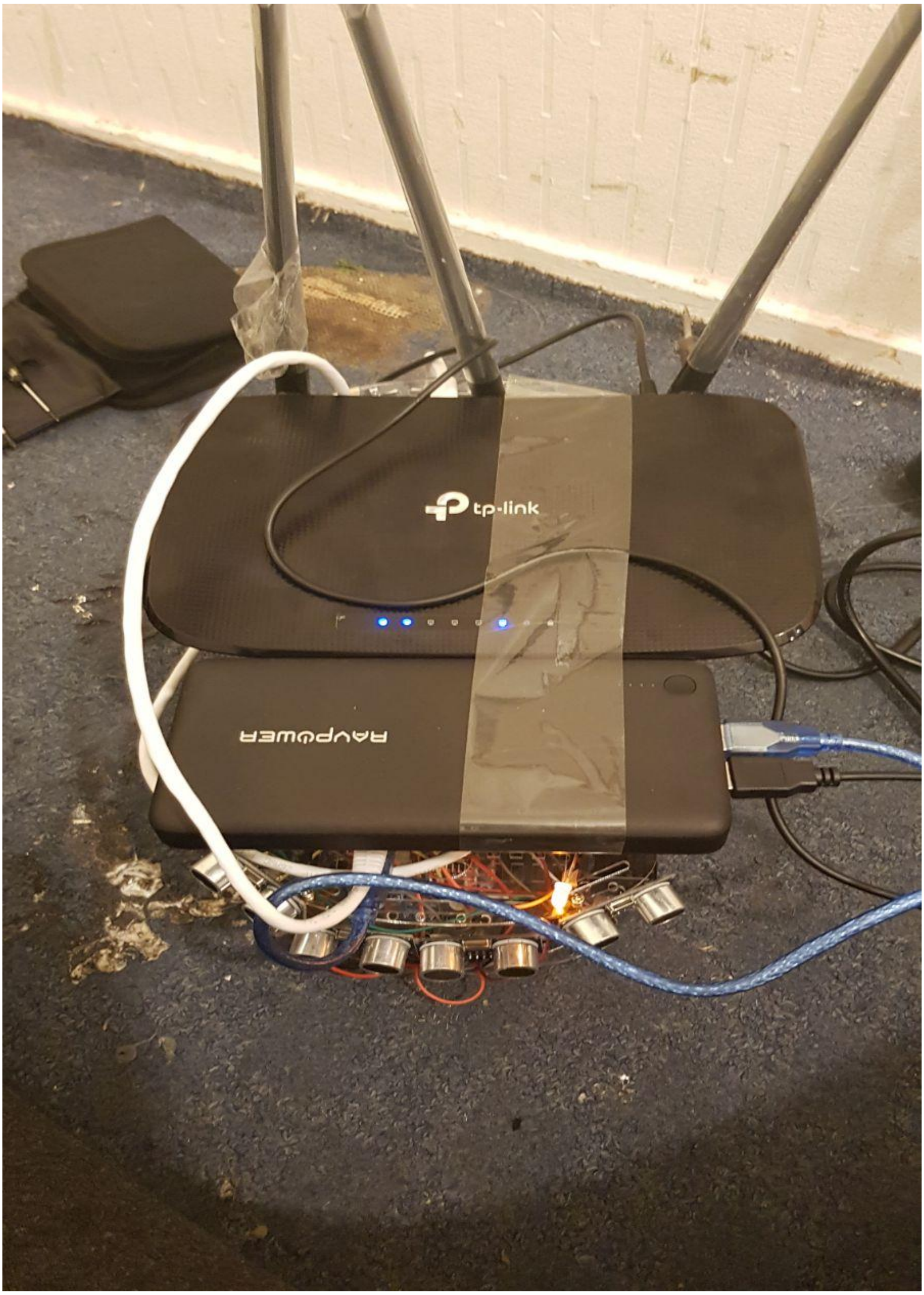

Figure 7-4 Top View

Control Page of Robot :

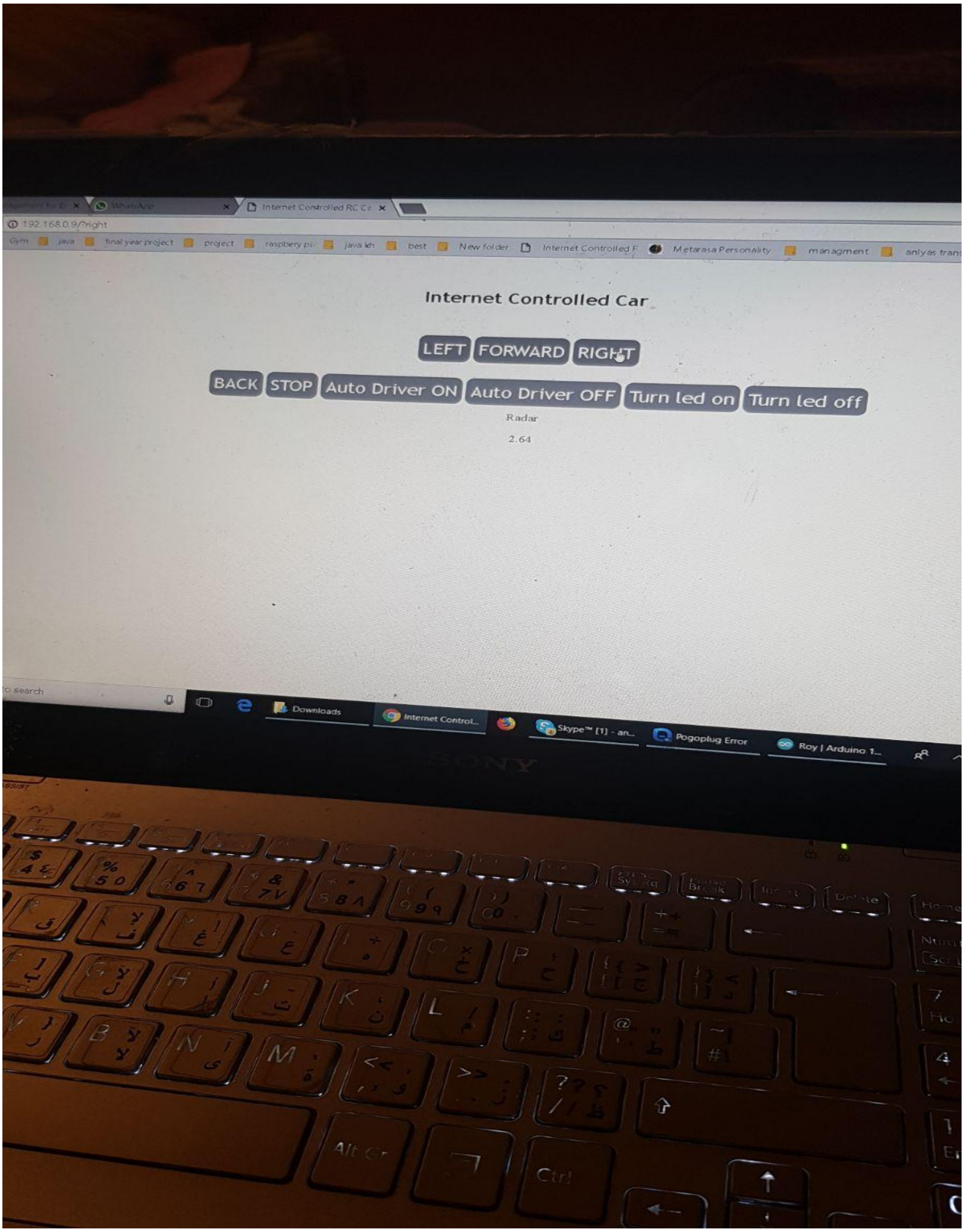

Figure 7-5 Control Page of Robo

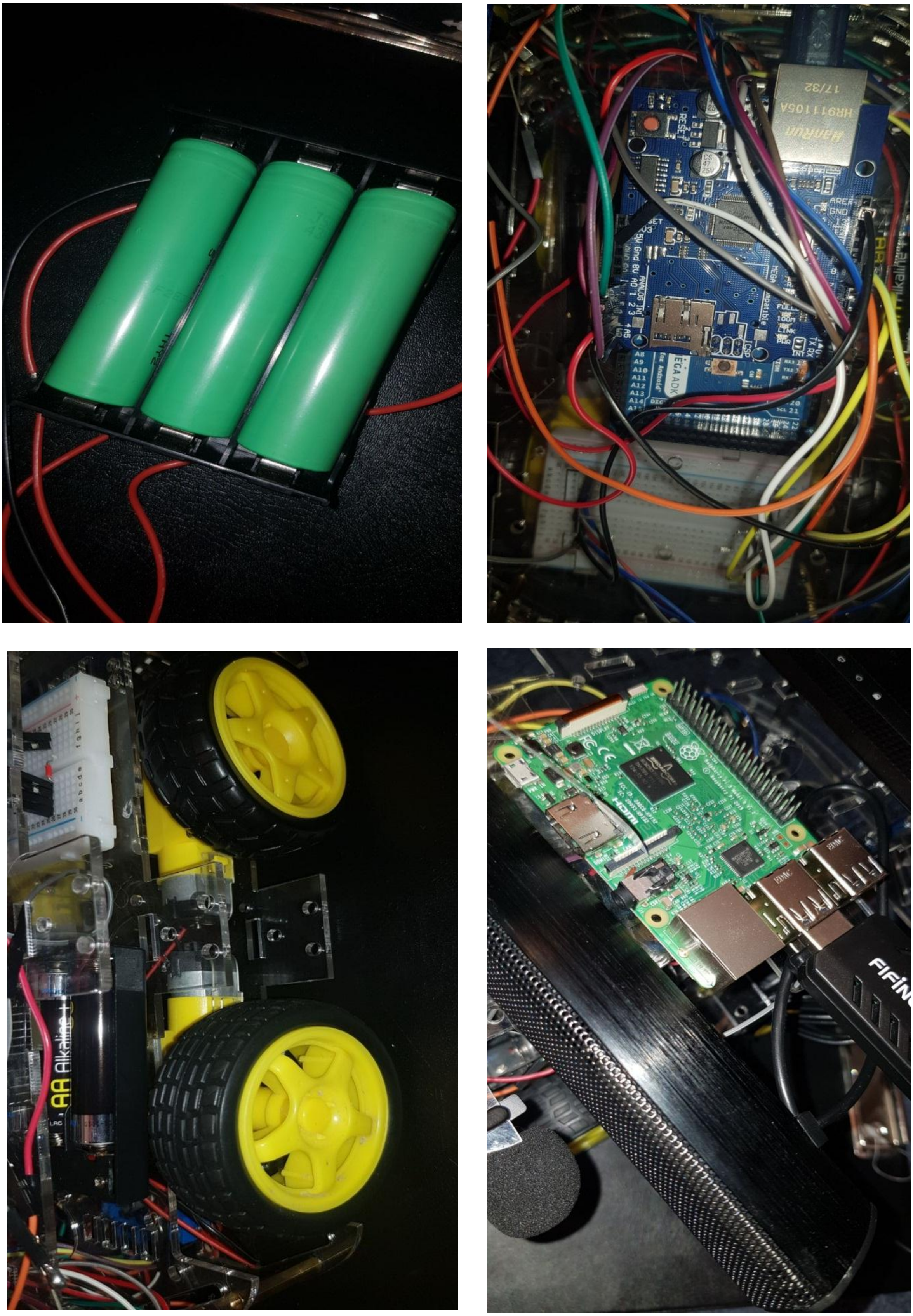

Figure 7-6 Some parts of the project