**SMART GRASS CUTTER**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

BY

SHUBHAM CHAVAN

ACHAL KALWAR

ROHAN MATHUR

UNDER THE GUIDANCE OF

**PROF. MEENA UGALE**

(Department of Information Technology)



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**XAVIER INSTITUTE OF ENGINEERING**

**UNIVERSITY OF MUMBAI**

**2019 – 2020**

**XAVIER INSTITUTE OF ENGINEERING**

**MAHIM CAUSEWAY, MAHIM, MUMBAI - 400016.**

**CERTIFICATE**

This to certify that

SHUBHAM CHAVAN (XIEIT171805)

ACHAL KALWAR (XIEIT171820)

ROHAN MATHUR (XIEIT171831)

Have satisfactorily carried out the IOT MINI-PROJECT work titled **“SMART GRASS CUTTER”** in partial fulfillment of the degree of Bachelor of Engineering as laid down by the University of Mumbai during the academic year 2019-2020.

**Prof. Meena Ugale**

**Internal Examiner External Examiner**

**Date:**

**Place: MAHIM, MUMBAI**

**DECLARATION**

I declare that this written submission represents my ideas in my own words and where others’

Ideas or words have been included, I have adequately cited and referenced the original sources.

I also declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which thus have not been properly cited or from whom proper permission have not been taken when needed.

SHUBHAM CHAVAN (XIEIT171805) -------------------------------

ACHAL KALWAR (XIEIT171820) -------------------------------

ROHAN MATHUR (XIEIT171831) -------------------------------

Date:

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **TOPIC** | **PAGE NO.** |
| A. | LIST OF FIGURES | i |
| B. | LIST OF TABLES | iii |
| C. | ACKNOWLEDGEMENT | iv |
| D. | LAB OUTCOMES | v |
| 1. | INTRODUCTION TO IOT (LO1)  INTRODUCTION  1.1 IDENTIFY THE REQUIREMENTS FOR THE REAL WORLD PROBLEMS.  1.2 APPLICATIONS OF IOT | 1  2  6 |
| 2. | INTRODUCTION TO SMART GRASS CUTTER (LO2)  2.1 PROBLEM DEFINITION  2.2 AIMS AND OBJECTIVES   * 1. SCOPE OF THE PROJECT   2.4 FEATURES OF THE PROJECT | 9  10  11  11  11 |
| 3. | REVIEW OF LITERATURE (LO2) | 12 |
| 4. | SYSTEM DESCRIPTION (LO3, LO4, LO5)  4.1 SYSTEM DESIGN  4.2REQUIREMENT OF HARDWARE &SOFTWARE  4.3 IMPLEMENTATION METHODOLOGY  4.4 HARDWARE CIRCUIT DIAGRAM  4.5 CODE  4.6 FINAL PROTOTYPE  4.7 CONCLUSION & FUTURE SCOPE OF THE PROJECT  4.8 CONSTRAINTS FOR REAL TIME DEPLOYMENT | 15  16  17  21  22  22  25  26  26 |
|  | REFERENCES | 27 |
|  | APPENDIX |  |
|  | A1: ASSIGNMENT 1- EMBEDDED SYSTEM CORES | 28 |
|  | A2: ASSIGNMENT 2- EMBEDDED SYSTEM TOOLS | 36 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **SR NO.** | **FIGURE CAPTION** | **PAGE NO.** |
| 1.1 | Introduction to IoT | 2 |
| 4.1 | Block diagram of Smart Grass Cutter | 16 |
| 4.2 | Flowchart of Smart Grass Cutter | 16 |
| 4.3 | Arduino Mega | 17 |
| 4.4 | L298N Motor Driver | 18 |
| 4.5 | DC Motor | 18 |
| 4.6 | Ultrasonic Sensor | 19 |
| 4.7 | Breadboard | 19 |
| 4.8 | Wheels | 19 |
| 4.9 | Jumper wires | 20 |
| 4.10 | USB cable | 20 |
| 4.11 | SPST Rocker Switch | 20 |
| 4.12 | Circuit Diagram of Smart Grass Cutter | 22 |
| 4.13 | Front view of Prototype | 25 |
| 4.14 | Top view of Prototype | 25 |
| A1.1 | CISC | 30 |
| A1.2 | RISC | 32 |
| A1.3 | ARM | 33 |

i

|  |  |  |
| --- | --- | --- |
| A1.4 | DSP | 34 |
| A1.5 | SOC | 35 |
| A2.1 | Arduino | 38 |
| A2.2 | Raspberry Pi | 39 |
| A2.3 | ARM Cortex | 41 |
| A2.4 | Intel Galileo | 43 |
| A2.5 | Temperature sensor | 44 |
| A2.6 | Proximity sensor | 45 |
| A2.7 | Infrared sensor | 45 |
| A2.8 | Pressure sensor | 46 |
| A2.9 | Light sensor | 46 |
| A2.10 | Ultrasonic sensor | 47 |
| A2.11 | Color sensor | 47 |
| A2.12 | Humidity sensor | 47 |

ii

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **SR NO.** | **TABLE TITLE** | **PAGE NO.** |
| 4.1 | Hardware, Software Requirements for Smart Grass Cutter | 21 |
| A2.1 | Types of Arduino | 38 |

iii

**Acknowledgement**

We would like to thank Fr. (Dr.) John Rose S.J. (Director of XIE) for providing us with such an environment so as to achieve goals of our project and supporting us constantly.

We express our sincere gratitude to our Honorable Principal Mr. Y.D.Venkatesh for encouragement and facilities provided to us.

We would like to place on record our deep sense of gratitude to Prof. Chhaya Narvekar, Head of Dept Of Information Technology, Xavier Institute of Engineering, Mahim, Mumbai, for her generous guidance help and useful suggestions.

With deep sense of gratitude we acknowledge the guidance of our project guide Prof. Meena Ugale. The time-to-time assistance and encouragement by her has played an important role in the development of our project.

We would also like to thank our entire Information Technology staff who have willingly cooperated with us in resolving our queries and providing us all the required facilities on time.

SHUBHAM CHAVAN -----------------------------

ACHAL KALWAR -----------------------------

ROHAN MATHUR -----------------------------

iv

**Subject: Internet of Things Mini-projects Lab**

**Lab Outcomes**

|  |  |  |
| --- | --- | --- |
| **At the end of the course student will be able to:** | | **Bloom Level** |
| ITL 504-1  (LO1) | Identify the requirements for the real world problems. | Understand |
| ITL 504-2  (LO2) | Conduct a survey of several available literatures in the preferred field of study. | Understand |
| ITL 504-3  (LO3) | Study software/ hardware skills and build the project successfully by hardware requirements, coding, emulating and testing. | Apply, Implement |
| ITL 504-4  (LO4) | To report and present the findings of the study conducted in the preferred domain | Apply, Implement |
| ITL 504-5  (LO5) | Demonstrate an ability to work in teams and manage the conduct of the research study. | Apply |

v

**XAVIER INSTITUTE OF ENGINEERING**

**Department of Information Technology**

**Class/ Sem/ A.Y: TE IT/ V/ 2019-20**

**Course Name: IoT Mini-ProjectLab**

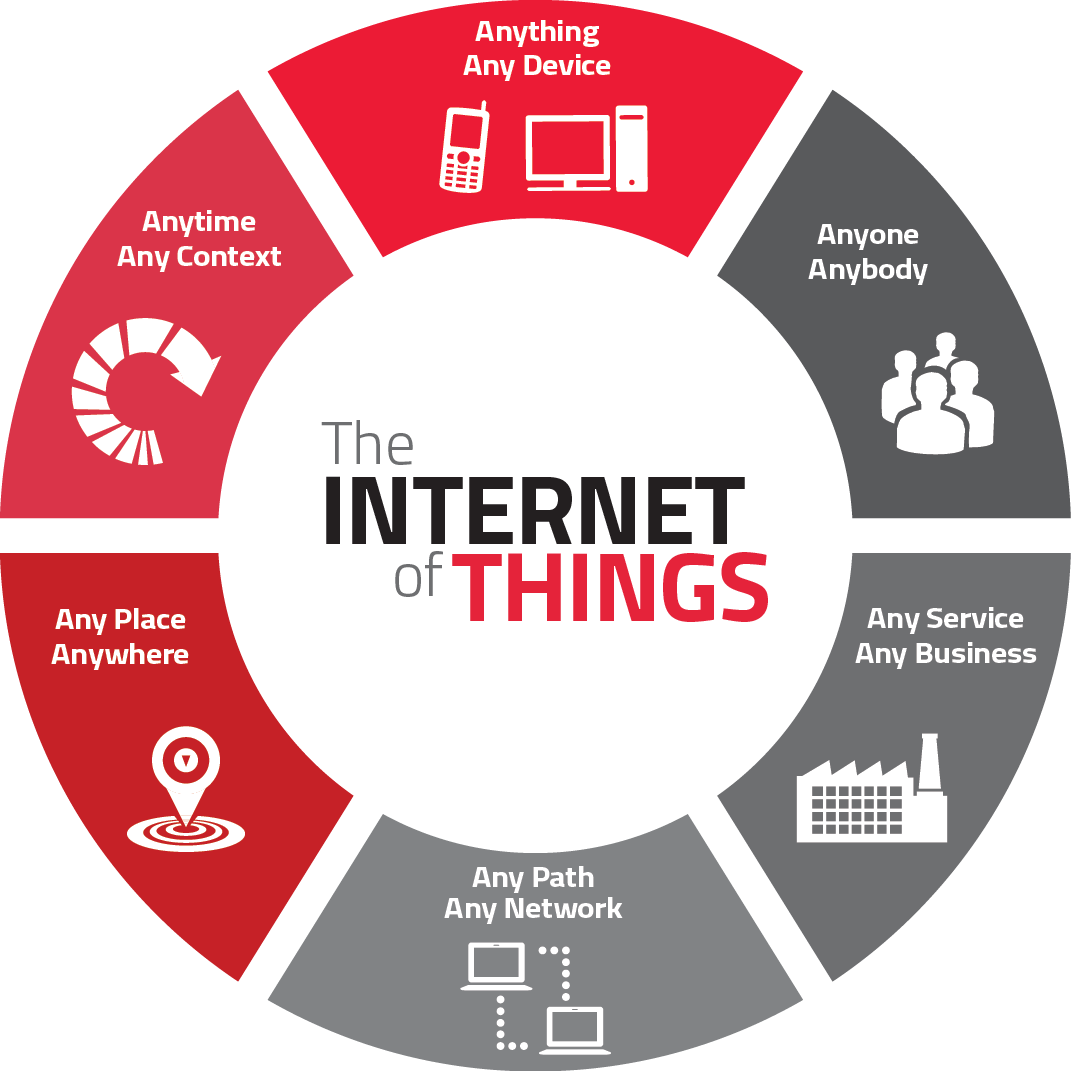
Group No/ Roll No: 17

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chapter 1: Introduction to IoT** | | | | | | |
| **LO1:Identify the requirements for the real world problems.** | | | | | | |
| **Rubrics For Laboratory Work** | | | | | | |
| **Roll No.** | **Name of the Student** | **Knowledge/ Understanding**  **(05)** | **Contents**  **(04)** | **Presentation (04)** | **Punctuality&**  **lab ethics**  **(02)** | **Total (15)** |
| XIEIT171805 | SHUBHAM CHAVAN |  |  |  |  |  |
| XIEIT171820 | ACHAL KALWAR |  |  |  |  |  |
| XIEIT171831 | ROHAN MATHUR |  |  |  |  |  |

Ms. Meena Ugale

**CHAPTER 1. INTRODUCTION TO IoT**

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.



**Figure 1.1: Introduction to IoT**

**1.1 REQUIREMENTS FOR THE REAL WORLD PROBLEMS**

**1. Ability to monitor security** of substations, as well as real-time data on electrical use, and report exceptions so they can be addressed in a timely manner. Ability to solve problems across silos in a utility where IT, Operations and Security don’t talk to each other. Facilitate communications about what’s important.

**2**. **Any information you need from a device** in order to perform a particular action. Examples provided:

* Re-closer on the distribution side of a power line that gets hit by lightening can be closed, checked and reopened by machine after reading the information on the site.
* Measure temperature and flow of a pipeline to ensure everything is working as expected or be notified is variances outside the norm are seen.
* Solar back-up to devices that may have power or battery issues.

**3. Ability to know how to fix your garage door** because the sensor can tell the company what’s wrong and they can tell you how to fix it. B2B example - a biolab is not aware of the volume of enzymes their clients still had on hand. If client ran out, they’d have to put a project on hold for a week or two while more enzymes we made and shipped. They now put one of our devices on every enzyme container so they know which scientist is using which enzyme and when supply is running low. Real-time stock updates enable New biolab to optimize the supply chain for their clients.

**4. Communication, collaboration, cohesion and unification of various objects.**Becoming more convenient and connected with the devices we use everyday. How to collaborate with devices and work through technology.

**5. Streamlines efficiency and communication of information.**Sensors provide data all the time. Influences decisions by giving you real-time data. Sensors in stores and in manufacturing environments tell you exactly what’s going on and if something is out of the acceptable range you can correct it quickly. Emergence of smaller computers are enhancing communications. We’re taking commodity hardware and optimizing with sensors.

6**.** All technology starts as a novelty. Becomes a convenience. That’s where IoT is now. **Making it more convenient to control and monitor the 3D printer.** Not yet a pure utility (the end stage for a technology). Ultimately, going forward, things will be built with IoT as a core element, not a special feature. For us, the end point is the printer which can source content from the internet.

**7. Industrial internet** - **digitization can be applied to the decision making process.**Consumer and healthcare companies will have new apps with connected devices to help save lives and mitigate disasters (e.g., floods, earthquakes).

**8. All IoT solutions solve some problem** - some are smaller, some are bigger. It’s easy to make a lock to connect to the internet. We use cryptography, website and code so the lock doesn’t have to be connected to the internet. Our IoT is not connected to the internet, just the website. As such, the lock cannot be hacked.

**9. Simpler, less expensive home health monitoring** (e.g., scales, blood pressure) to prevent post-surgical events that require return visits to the hospital. Increase adoption and adherence to medication protocols. Opportunity to use data to predict what’s going to happen. Preventive and predictive healthcare.

**10**. **A** **connection between humans and computers.** Use Amazon Echo to get all his songs from Amazon Prime and play on demand. Links home management like garage door, lights and HVAC.

**11. Energy saving.** A lot of devices are left on overnight, or longer. Interact with buildings and homes to save energy.

**12. Health devices connected to Smartphone’s diagnose health conditions quickly.** You can take pictures and obtain diagnostics to share with health professionals around the world. Enables the collection and sharing of data in an affordable way. Allows inventors to think about use cases. Digitizes the power grid. Play with how energy is being served. Every device in your house will give you an energy profile. Enterprises will benefit from the digitization of devices and enable the next wave of digitization.

**13.** Know people that are in the building and **have visibility into what’s going on**. We monitor several thousand conference rooms at Microsoft’s campus to determine if they’re occupied, if A/V is working, what devices people are using, scheduling, booking. We collect data in the cloud to analyze uptime and failure rates. We proactively monitor to see what’s going on. We have statistics about room use and occupancy that will inform and influence the design of the conference rooms on Microsoft’s new campus. In homes, our hub connects all light switches, thermostats, keypads, security system and provides statistics to the cloud so the homeowner can view a dashboard to see how the home is being used. Occupancy use data. We’ll be able to use predictive analytics to make suggestions on how to change the real-time lighting, temp, etc. for your home. Many more touchpoints - switches, mobile phone, devices, reporting to the cloud versus a single thermostat (Nest) - provide more data for analysis.

**14. Changed the brand cycle.**It used to be 18 to 24 months. Now you must be monitoring social networks to hear what customers are saying and address their concerns or leverage what they are seeing as most beneficial. As John Chambers says, 40% of companies won’t exist in 10 years if they’re not listening to, and responding to the needs of, their customers.

**15. Asset management** - how to engage information to run control systems. Understand the health of the asset producing the work. Know the health and diagnostics of the machine to reduce down time and proactively provide maintenance. Ability to tie the supply chain into the process and provide information back to manufacturing thus reducing costs and expense.

**16. Manufacturers using crowd sourcing to build out their manufacturing floor.** Consumer wireless routers are only secure for a couple of years. Consumer products have a short life expectancy with consumers. Whereas industrial companies need to have an ongoing relationship with their customers since they have service contract and the products often need ongoing service. In healthcare alone IoT has already made incredible contributions saving lives, giving doctors the ability to see a spectrum of health conditions across a large number of people. It will enable more self-care by patients. Clinical trials are now being based on data received from IoT devices thus accelerating time to market. Industrial is incredibly influential because of the buy in from so many big players like IBM, Cisco and GE.

**17. Enables people to try a new approach.** Automate and control things remotely in ways you couldn’t before. Opportunities differ by industry but every industry has many opportunities.

**18.** We’re at the very early stages but making progress every day. **Getting basic, real-time visibility into places where we haven’t had it before.** For example, we can see a pipeline every half mile and look at KPIs for variances rather than have a human out driving the line and taking measurements. We’ve figured out how to put predictive diagnostics in place. We’re creating a digital twin on the product based on historical performance so we can identify potential needs. IoT provides visibility and reliability where we’ve never had it in the past.

**19. Mundane things.** We have a school district in Florida with wi-fi clocks. We save them time and money, and ensure the clocks are accurate, after power outages and daylight saving time changes. Student athletes at a university are calling the help desk regarding how to get their personal devices to connect with the scales at the athletic center so the trainers can track biometrics. Also in universities, next generation professors need to connect to the Apple TVs for their presentations and improving basic efficiencies.

**20. Specific use cases:**

* Implantable healthcare devices implanted subcutaneously to monitor vital signs for diagnostic follow-up. Communicate vitals to base station to app in the cloud. Depending on the situation, the correct people are notified of the vital signs. using communications logic, secure data following HIPPA laws escalating notifications as dictated by the data or situation.
* Dam holding water has myriad sensors monitoring the water level and the aperture of the gates. Preset threshold communications logic kicks in. Looks at the schedule to see who’s on staff. If the right people are on staff, they respond to the action required. If not, the communications are escalated based on the communications policy logic and sensor information. Scale and extreme impact of the decision that the logic needs to make. Do we need to evacuate? Zones of evacuation are identified based on what the sensors are saying and home can receive phone calls at 2:00 am if evacuation is called for.
* Innocuous business decisions - making dumb devices smarter. Badge for new employee or visitor provides access control, an active directory and a begins a set of initiatives to (e.g., assignment of office, desk, computer, insurance) and business security. Profile of the person is continually updated based on the last use of their badge so we know where they are in the event you need to communicate with them because you need their expertise in an adjacent area, or in the event of a building emergency.
* Assembly lines with complex machinery coming down the line. Numerous tests are run with data indicating whether an item/action passes or fails. If it fails, here are the people who must be notified to deal with it. Depending on the failure, the communication may, or may not, need to be escalated.

**1.2 APPLICATIONS OF INTERNET OF THINGS**

### Smart Home:

Smart Home clearly stands out, ranking as highest Internet of Things application on all measured channels. More than 60,000 people currently search for the term “Smart Home” each month. This is not a surprise. The IoT Analytics company database for Smart Home includes 256 companies and startups. More companies are active in smart home than any other application in the field of IoT. The total amount of funding for Smart Home startups currently exceeds $2.5bn. This list includes prominent startup names such as Nest or AlertMe as well as a number of multinational corporations like Philips, Haier, or Belkin.

### 2.   Wearables:

Wearable remains a hot topic too. As consumers await the release of Apple’s new smart watch in April 2015, there are plenty of other wearable innovations to be excited about: like the Sony Smart B Trainer, Look See bracelet. Of all the IoT startups, wearable maker Jawbone is probably the one with the biggest funding to date. It stands at more than half a billion dollars!

### 3.   Smart City:

Smart city spans a wide variety of use cases, from traffic management to water distribution, to waste management, urban security and environmental monitoring. Its popularity is fueled by the fact that many Smart City solutions promise to alleviate real pains of people living in cities these days. IoT solutions in the area of Smart City solve traffic congestion problems, reduce noise and pollution and help make cities safer.

### 4.     Smart grids:

Smart grids is a special one. A future smart grid promises to use information about the behaviors of electricity suppliers and consumers in an automated fashion to improve the efficiency, reliability, and economics of electricity. 41,000 monthly Google searches highlights the concept’s popularity. However, the lack of tweets (Just 100 per month) shows

### 5.    Industrial internet:

The industrial internet is also one of the special Internet of Things applications. While many market researches such as Gartner or Cisco see the industrial internet as the IoT concept with the highest overall potential, its popularity currently doesn’t reach the masses like smart home or wearable do. The industrial internet however has a lot going for it. The industrial internet gets the biggest push of people on Twitter (~1,700 tweets per month) compared to other non-consumer-oriented IoT concepts.

### 6.   Connected car:

The connected car is coming up slowly. Owing to the fact that the development cycles in the automotive industry typically take 2-4 years, we haven’t seen much buzz around the connected car yet. But it seems we are getting there. Most large auto makers as well as some brave startups are working on connected car solutions.  And if the BMWs and Fords of this world don’t present the next generation internet connected car soon, other well-known giants will: Google, Microsoft, and Apple have all announced connected car platforms.

### 7.   Connected Health (Digital health/Tele-health/Telemedicine):

Connected health remains the sleeping giant of the Internet of Things applications. The concept of a connected health care system and smart medical devices bears enormous potential, not just for companies also for the well-being of people in general. Yet, Connected Health has not reached the masses yet. Prominent use cases and large-scale startup successes are still to be seen. Might 2015 bring the breakthrough?

### 8.   Smart retail:

Proximity-based advertising as a subset of smart retail is starting to take off. But the popularity ranking shows that it is still a niche segment. One LinkedIn post per month is nothing compared to 430 for smart home.

### 9.    Smart supply chain:

Supply chains have been getting smarter for some years already. Solutions for tracking goods while they are on the road, or getting suppliers to exchange inventory information have been on the market for years. So while it is perfectly logic that the topic will get a new push with the Internet of Things, it seems that so far its popularity remains limited.

### 10.   Smart farming:

Smart farming is an often overlooked business-case for the internet of Things because it does not really fit into the well-known categories such as health, mobility, or industrial. However, due to the remoteness of farming operations and the large number of livestock that could be monitored the Internet of Things could revolutionize the way farmers work. But this idea has not yet reached large-scale attention. Nevertheless, one of the Internet of Things applications that should not be underestimated. Smart farming will become the important application field in the predominantly agricultural-product exporting countries.

**XAVIER INSTITUTE OF ENGINEERING**

**Department of Information Technology**

**Class/ Sem/ A.Y: TE IT/ V/ 2019-20**

**Course Name: IoT Mini-Project Lab**

Group No/ Roll No: 17

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chapter 2: Introduction to your Mini-Project Topic** | | | | | | |
| **LO2: Conduct a survey of several available literatures in the preferred field of study.** | | | | | | |
| **Rubrics For Laboratory Work** | | | | | | |
| **Roll No.** | **Name of the Student** | **Knowledge/ Understanding**  **(05)** | **Contents**  **(04)** | **Presentation (04)** | **Punctuality&**  **lab ethics**  **(02)** | **Total (15)** |
| XIEIT171805 | SHUBHAM CHAVAN |  |  |  |  |  |
| XIEIT171820 | ACHAL KALWAR |  |  |  |  |  |
| XIEIT171831 | ROHAN MATHUR |  |  |  |  |  |

Ms. Meena Ugale

**CHAPTER 2. INTRODUCTION TO SMART GRASS CUTTER**

**2.1 PROBLEM DEFINITION**

A Smart Grass Cutter is a machine that uses a revolving blade or blades to cut a lawn. Lawn mowers employing a blade that rotates about a vertical axis are known as rotary mowers (blade rotates horizontally).Many designs have been made, each suited to a particular purpose. The smallest types, pushed by a human, are suitable for small residential lawns and gardens, while larger, self-contained, ride-on mowers are suitable for large lawns, and the largest, multi-gang mowers pulled behind a tractor, are designed for large expanses of grass such as golf courses and for parks.

* **The problems with available grass cutter robots are as follows:**
* Power consumption: The available grass cutter are electrical powered which will consume large amount of conventional energy source.
* Human effort: The mowing work always needs to get control with a worker for the proper mowing.
* Time consumption: For mowing the land in different patterns and design it takes larger time and human effort
* Safety.
* Classical grass cutters with heavy engines create noise pollution and local air pollution due to the combustion in the engine.
* Fuel powered engines require time to time maintenance such as changing the engine oils etc.
* If the electric grass cutter is cord type, to use it could prove to be problematic and dangerous.
* Moving the grass cutters with a standard motor is inconvenient, and no one takes pleasure in it.

**2.2 AIMS AND OBJECTIVES**

The objective of the proposed work is to design and construct the smart grass cutter. It is a fully automated grass cutting robotic vehicle. It also avoids obstacles without the need of any human interaction. The system uses batteries to power the vehicle movement motors as well as the grass cutter motor.

* To reduce power consumption and time consumption.
* To reduce human effort.
* To reduce noise and air pollution.
* The self- powered objective is to come up with a cutter that is portable, durable, easy to operate and maintain.
* It also aims to design a self- powered cutter of electrical source, a cordless electric grass mower.
* To implement agricultural automation.
* To enhance safety.

**2.3 SCOPE OF THE PROJECT**

The grass cutter and vehicle motors are interfaced to Arduino that controls the working of all the motors. It is also interfaced to an Ultrasonic sensor for object detection. The Arduino controller moves the vehicle motors in forward direction in case no obstacle is detected. On obstacle detection ultrasonic sensor monitors it and the controller thus rotates the grass cutter motor so as to avoid any damage to the object/human/animal. Controller then moves the grass cutter in forward direction again. This project aims to make a daily purpose robot which is able to cut the grasses in lawn. The system will have some automation work for guidance and other obstacle detection.

**2.4 FEATURES OF THE PROJECT**

* Easy to move from one place to another place.
* Compact size and portable.
* Operating principle is simple.
* Non-skilled person also can operate this machine.

**XAVIER INSTITUTE OF ENGINEERING**

**Department of Information Technology**

**Class/ Sem/ A.Y: TE IT/ V/ 2019-20**

**Course Name:IoT Mini-Project Lab**

Group No/ Roll No: 17

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chapter 3: Review of Literature** | | | | | | |
| **LO2: Conduct a survey of several available literatures in the preferred field of study.** | | | | | | |
| **Rubrics For Laboratory Work** | | | | | | |
| **Roll No.** | **Name of the Student** | **Knowledge/ Understanding**  **(05)** | **Contents**  **(04)** | **Presentation (04)** | **Punctuality&**  **lab ethics**  **(02)** | **Total (15)** |
| XIEIT171805 | SHUBHAM CHAVAN |  |  |  |  |  |
| XIEIT171820 | ACHAL KALWAR |  |  |  |  |  |
| XIEIT171831 | ROHAN MATHUR |  |  |  |  |  |

Ms. Meena Ugale

**CHAPTER 3. REVIEW OF LITERATURE**

The first lawn mower was invented by Edwin Budding in 1830 in Thrupp, just outside Stroud, in Gloucestershire, England. Budding's mower was designed primarily to cut the grass on sports grounds and extensive gardens, as a superior alternative to the scythe, and was granted a British patent on August 31, 1830.In 1995, the first fully solar powered robotic mower became available. The mower can find its charging station via radio frequency emissions, by following a boundary wire, or by following an optional guide wire [5].

Automation is rapidly growing in the present technology. So automation plays a vital role in the agricultural field which is helpful for the farmers. In the earlier days, the grass cutters used were manually handheld devices. Because of this, there was pollution and loss of energy as they used gas and petrol engines. So the old grass cutters need to be replaced by automated ones[1].

Grass cutter machines have become very popular today. Most of the times, grass cutter machines are used for soft grass furnishing. In a time where technology is merging with environmental awareness, consumers are looking for ways to contribute to the relief of their own carbon footprints.Herein, we propose a model of the automatic grass cutting machine. Automatic grass cutting machine is a machine which is going to perform the grass cutting operation on its own. This model reduces both environment and noise pollution [2].

Smart Grass Cutter is a fully automated grass cutting robotic vehicle that also avoids obstacles and is capable of cutting grass without the need of any human interaction. So the traditional grass cutters are to be replaced by daily purpose robot which will be capable of cutting the grass in lawn without human intervention[3].

It is also interfaced to an ultrasonic sensor for object detection. The microcontroller moves the vehicle motors in forward direction in case no obstacle is detected. On obstacle detection, ultrasonic sensor monitors it and the microcontroller thus stops the grass cuter motor so as to avoid any damage to the object/human/animal. Microcontroller then turns the robotic vehicle off until it gets clear of the object and then moves the grass cutter in forward direction again [6].

The Ultrasonic distance sensor provides precise, non-contact distance measurements from about 2cm to 3meters. It is very easy to connect to Micro Controllers, propeller chip, or arduino, requiring only one i/o pin. The sensor has amale3-pinheader used to supply ground, power and signal[7].

The system uses 12V batteries to power the vehicle movement motors as well as the grass cutter motor. The grass cutter and vehicle motors are interfaced to an 8051 family microcontroller that controls the working of all the motors[4].

**XAVIER INSTITUTE OF ENGINEERING**

**Department of Information Technology**

**Class/ Sem/ A.Y: TE IT/ V/ 2019-20**

**Course Name: IoT Mini-Project Lab**

Group No/ Roll No: 17

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chapter 4: System Description** | | | | | | |
| **LO3: Study software/ hardware skills and build the project successfully by hardware requirements, coding, emulating and testing.**  **LO4: To report and present the findings of the study conducted in the preferred domain**  **LO5: Demonstrate an ability to work in teams and manage the conduct of the research study.** | | | | | | |
| **Rubrics For Laboratory Work** | | | | | | |
| **Roll No.** | **Name of the Student** | **Problem Statement**  **(05)** | **Creativity &**  **Quality of Work done**  **(05)** | **Punctuality&**  **lab ethics**  **(02)** | **Performance/**  **Presentation (03)** | **Total (15)** |
| XIEIT171805 | SHUBHAM CHAVAN |  |  |  |  |  |
| XIEIT171820 | ACHAL KALWAR |  |  |  |  |  |
| XIEIT171831 | ROHAN MATHUR |  |  |  |  |  |

Ms. Meena Ugale

**CHAPTER 4. SYSTEM DESCRIPTION**

**4.1 DESIGN**

**BLOCK DIAGRAM OF SMART GRASS CUTTER:**

**Figure 4.1: Block diagram of Smart Grass Cutter.**

**FLOWCHART OF SMART GRASS CUTTER:**

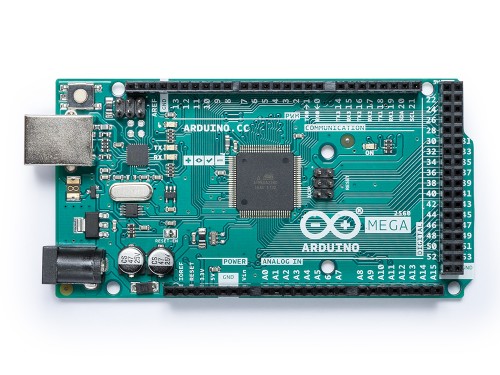
**Figure 4.2: Flowchart of Smart Grass Cutter.**

**4.2 HARDWARE AND SOFTWARE SPECIFICATIONS**

* **HARDWARE:**

1. **Arduino Mega 2560 (1)**

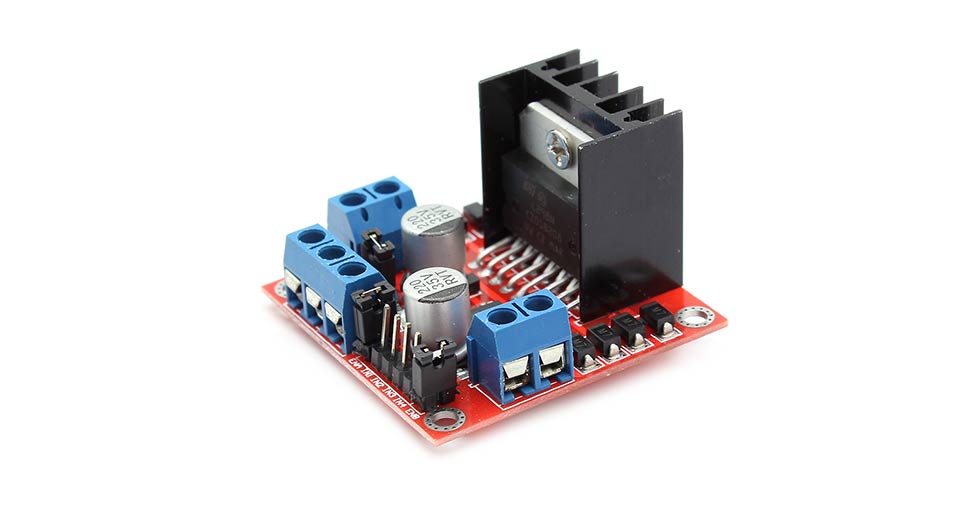
The ArduinoMega2560 is a microcontroller board based on the [Atmega2560](http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.



**Figure 4.3: Arduino Mega**

1. **L298N Motor Driver (1)**

The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage , high current dual full-bridge driver de-signed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals .The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor. An additional Supply input is provided so that the logic works at a lower voltage.



**Figure 4.4: L298N Motor Driver**

1. **DC Motors (12V) (3)**

The speed controller works on the fundamental by varying the average voltage sent to the motor. It could do this by simply adjusting the voltage sent to the motor, but this an inefficient method. A better way is to switch the motor supply on and off very quickly. If the switching is fast enough, the motor functioning does not get affected , it only notices the average effect.



**Figure 4.5: DC Motor**

1. **Ultrasonic Sensor (HC-SR04) (1)**

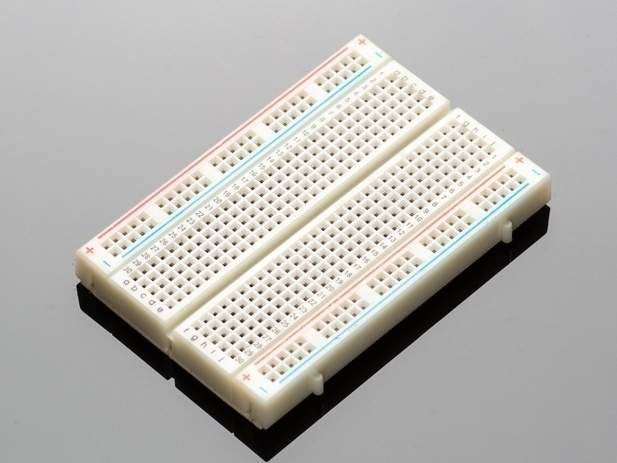
The Ultrasonic distance sensor provides precise, non-contact distance measurements from about 2cm to 3meters. It is very easy to connect to Micro Controllers, propeller chip, or arduino, requiring only one i/o pin. The sensor has amale3-pinheaderused to supply ground, power and signal. The header may be plugged into a directly into Solder less Bread board, or in to a Standard 3- Wire Extension Cable. The sensor detects objects bye matting a short ultra sonic burst and then" listening" for the echo. Under control of a host micro controller, the sensor emits a short 40 KHz burst. This burst travels through the air, hits an object and then bounces back to the sensor. The sensor provides an output pulse to the host that will terminate when the echo is detected hence the width of this pulse corresponds to the distance to the target.



**Figure 4.6: Ultrasonic Sensor**

1. **Breadboard (1)**

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.



**Figure 4.7: Breadboard**

1. **Wheels (3)**

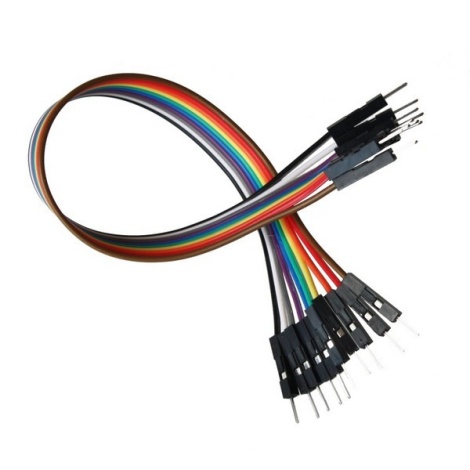
Wheels are used to move forward, backward and left or right in case of any obstacle is detected.



**Figure 4.8: Wheels**

1. **Jumper Wires (20-30)**

Jumper wires are used to connect the sensors, motors, LCD display and all the other components to the Arduino board.



**Figure 4.9: Jumper Wires**

1. **USB Cable (1)**

USB cable is used to give power supply to Arduino board.



**Figure 4.10: USB Cable**

1. **SPST Rocker Switch (1)**

**SPST**rocker switches make or break the connection of a single conductor in a single branch circuit. This switch type typically has two terminals and is referred to as a single-pole switch.



**Figure 4.11: SPST Rocker Switch**

* **SOFTWARE:**

1. **Arduino IDE**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.This software can be used with any Arduino board.

1. **C Programming Language**

C is a general-purpose programming language that is extremely popular, simple and flexible.

It is machine-independent, structured programming language.

**Table No 4.1 Hardware, Software Requirements for Smart Grass Cutter.**

|  |  |
| --- | --- |
| **HARDWARE REQUIREMENTS** | **SOFTWARE REQUIREMENTS** |
| Arduino Mega 2560 | Arduino IDE |
| L293N Motor Driver | C Programming Language |
| DC Motors |  |
| SPST Rocker Switch |  |
| Ultrasonic Sensor (HC-SR04) |  |
| Breadboard |  |
| Wheels |  |
| Jumper Wires and USB cable |  |

**4.3 IMPLEMENTATION METHODOLOGY**

STEP 1: Analyzed project requirements and collected various components required to build the project.

STEP 2: Studied the specification of each component along with the pin configuration of the Ultrasonic sensor, L298N motor driver, DC motors and Arduino Mega board.

STEP 3: Tested each component individually, checked the working of sensor and motor driver with motors.  
STEP 4: Ultrasonic sensor interfaced with arduino.

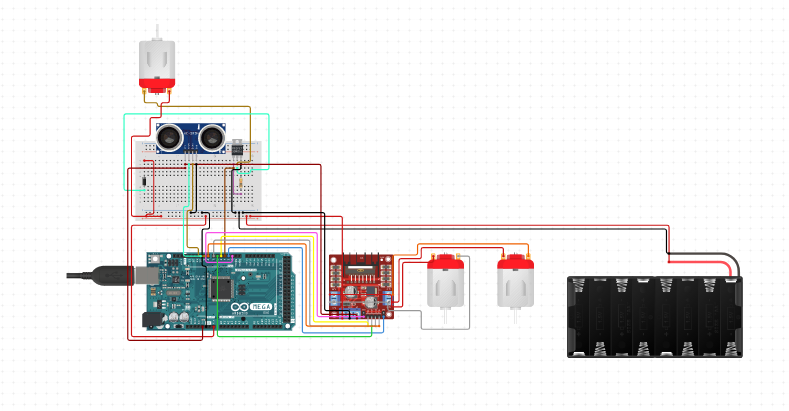
STEP 5: DC motors interfaced with motor driver and the arduino.

STEP 6: Developed the model with plywood and angles (dc motor holder).

STEP7: Connected the sensor and motor driver with arduino board and merged the code for all the components that are used.

STEP 8: Placed all the components on to the plywood model.

**4.4 HARDWARE CIRCUIT DIAGRAM**



**Figure 4.12: Circuit Diagram of Smart Grass Cutter.**

**4.5 CODE**

#include <NewPing.h>

const int LeftMotorForward = 7;

const int LeftMotorBackward = 6;

const int RightMotorForward = 5;

const int RightMotorBackward = 4;

//sensor pins

#define trig\_pin A1 //analog input 1

#define echo\_pin A2 //analog input 2

#define maximum\_distance 200

boolean goesForward = false;

int distance = 100;

NewPing sonar(trig\_pin, echo\_pin, maximum\_distance); //sensor function

void setup(){

pinMode(RightMotorForward, OUTPUT);

pinMode(LeftMotorForward, OUTPUT);

pinMode(LeftMotorBackward, OUTPUT);

pinMode(RightMotorBackward, OUTPUT);

}

void loop(){

int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20){

Serial.println(distance);

moveStop();

delay(300);

moveBackward();

delay(400);

moveStop();

delay(300);

if (distance >= distanceLeft){

turnRight();

moveStop();

}

else{

turnLeft();

moveStop();

}

}

else{

moveForward();

}

distance = readPing();

}

int readPing(){

delay(70);

int cm = sonar.ping\_cm();

if (cm==0){

cm=250;

}

return cm;

}

void moveStop(){

digitalWrite(RightMotorForward, LOW);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

digitalWrite(LeftMotorBackward, LOW);

}

void moveForward(){

if(!goesForward){

goesForward=true;

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}}

void moveBackward(){

goesForward=false;

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

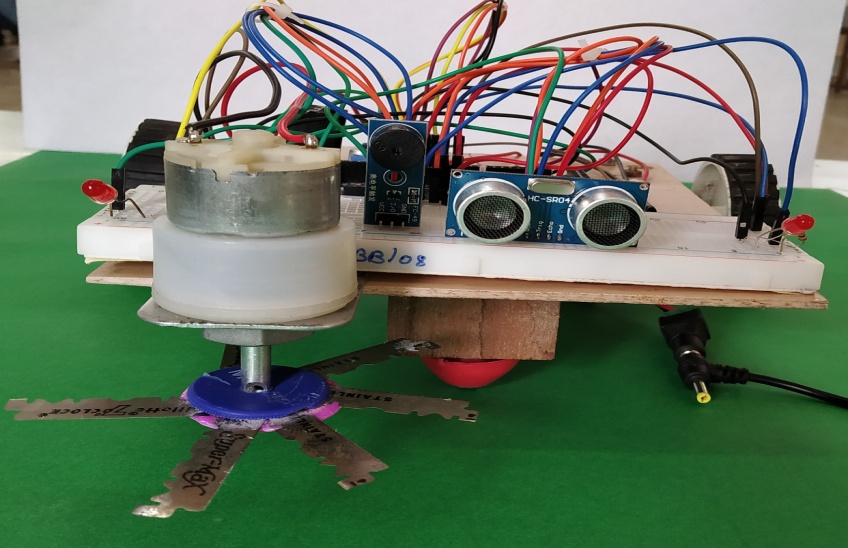
digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

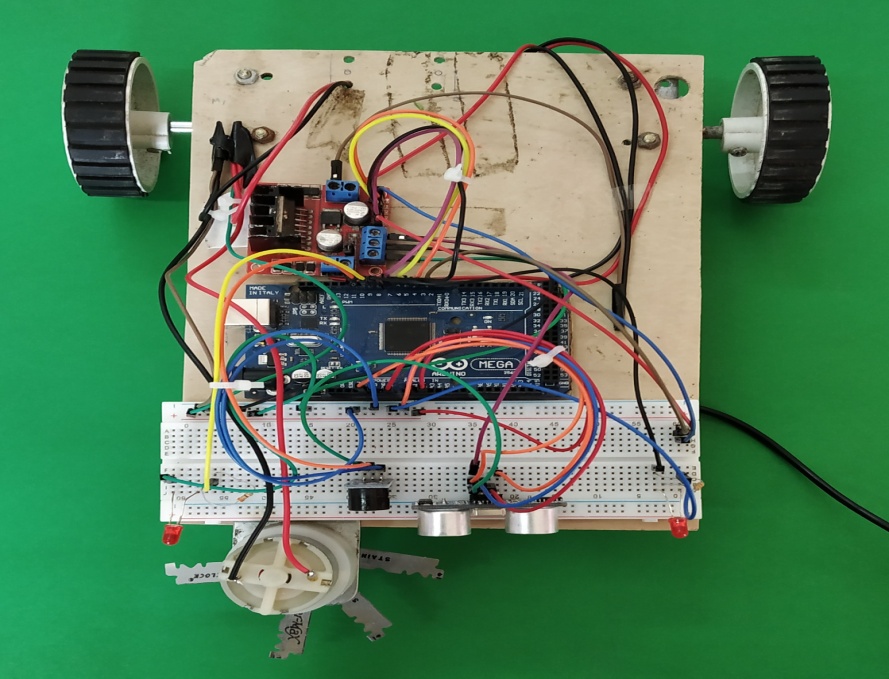
digitalWrite(RightMotorBackward, LOW);

}

**4.6 FINAL PROTOTYPE**

****

**Figure 4.13: Front view of Prototype.**

****

**Figure 4.14: Top view of Prototype.**

**4.7 CONCLUSION AND FUTURE SCOPE OF THE PROJECT**

* **CONCLUSION:**

The previous bot systems were studied and a suitable design was made. The schematic for the same was made on which prototyping will take place. The components have been chosen based on design requirement and based on a few other parameters. The prototype is mounted on a bakelite chassis and the detection is done using ultrasonic sensor and the output was obtained. Since grassing cutting is a mundane task requiring a lot of time, it is believed that human time should not be wasted on such tasks or at least reduced to the bare minimum. The cost effectiveness and the ease provided makes the bot to be a necessity instead of a luxury.

* **FUTURE SCOPE:**

Size can be reduced to make it compact. Efficiency can be amended by incrementing the battery capacity. More sensors can be incorporated for precise results. Bluetooth or Wi-Fi module can be used to operate the robot from anywhere through mobile phone. Solar panels can be used to store power in order to charge the battery. Programming can be enhanced to make the device perform different operations.

**4.8 CONSTRAINTS FOR REAL TIME DEPLOYMENT**

* The battery consumption of the robot is high, hence frequent charging of the battery is required.
* If the height of the grass is too high, the ultrasonic sensor will detect it as an obstacle and won’t cut the grass.
* The prototype is difficult to operate during rainy season.
* High speed cutter motor required to cut the grass smoothly.

**REFERENCES**

1. Neha, Syeda Asra, “ Automated Grass Cutter Robot Based on IoT”, IJTSRD, Volume- 2 Issue – 5, Jul-Aug 2018
2. Prof.S.M.Patil, Bhandirge Prajakta, Kumbhar Snehal, Patil Dhanashri, “Smart Solar Grass Cutter With Lawn Coverage”, IRJET, Volume: 05 Issue: 03 , Mar-2018
3. Tushar Baingane, Sweta Nagrale, Suraksha Gumgaonkar, Shaila Ramteke, Girish Langade, Prof. V. M. Dhumal – IJARIIE, Volume-4 Issue-2, 2018
4. Shekhar Kumar, Anshika Sharma, Rishabh Sharma, Shubham Kesarwani -

“A Review Paper on Grass Cutter Device Using Bluetooth”, Journal of Electronics and Electromagnetic Technology, Volume-1 Issue-1, 2018.

1. Ms. Yadav Rutuja A., Ms. Chavan Nayana V., Ms. Patil Monika B., Mr. V. A. Mane –“AUTOMATED SOLAR GRASS CUTTER”, IJSDR , Volume 2, Issue 2 , February 2017.
2. <https://nevonprojects.com/fully-automated-solar-grass-cutter/> accessed on 1st August 2019, 9:38 AM.
3. G. Manoj Kumar, G. Sravanthi, J. Ankamma Chowdary, D. Aparna4, V. Ajay Kumar- Solar Grass Cutter Robot with Obstacle Avoidance , IJESC, Volume-7 Issue-4, April 2017.

**XAVIER INSTITUTE OF ENGINEERING**

**Department of Information Technology**

**Class/ Sem/ A.Y: TE IT/ V/ 2019-20**

**Course Name:IoT Mini-Project Lab**

Group No/ Roll No: 17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assignment No. 1: Embedded System Cores** | | | | |
| **LO1:Identify the requirements for the real world problems.**  **LO2: Conduct a survey of several available literatures in the preferred field of study.**  **LO3: Study software/ hardware skills and build the project successfully by hardware requirements, coding, emulating and testing.**  **LO4: To report and present the findings of the study conducted in the preferred domain**  **LO5: Demonstrate an ability to work in teams and manage the conduct of the research study.** | | | | |
| **Rubrics For Assignment Work** | | | | |
| **Roll No.** | **Name of the Student** | **Knowledge & Content**  **(03)** | **Neatness and Timeline**  **(02)** | **Total (05)** |
| XIEIT171805 | SHUBHAM CHAVAN |  |  |  |
| XIEIT171820 | ACHAL KALWAR |  |  |  |
| XIEIT171831 | ROHAN MATHUR |  |  |  |

Ms. Meena Ugale

**ASSIGNMENT NO. 01**

**INTRODUCTION TO EMBEDDED MICROCONTROLLER CORES**

## CISC (Complex Instruction Set Computer) Architecture:

A **complex instruction set computer** is a computer in which single [instructions](https://en.wikipedia.org/wiki/Instruction_set_architecture) can execute several low-level operations (such as a load from [memory](https://en.wikipedia.org/wiki/Memory_(computers)), an [arithmetic](https://en.wikipedia.org/wiki/Arithmetic) [operation](https://en.wikipedia.org/wiki/Operator_(programming)), and a [memory store](https://en.wikipedia.org/wiki/Memory_(computers))) or are capable of multi-step operations or [addressing modes](https://en.wikipedia.org/wiki/Addressing_mode) within single instructions. The term "CISC" (complex instruction set computer or computing) refers to computers designed with a full set of computer instructions that were intended to provide needed capabilities in the most efficient way. Intel's [Pentium](https://whatis.techtarget.com/definition/Pentium) microprocessors are CISC microprocessors. The main intend of the CISC processor architecture is to complete task by using less number of assembly lines. For this purpose, the processor is built to execute a series of operations. Complex instruction is also termed as MULT, which operates [memory banks](https://www.elprocus.com/stack-memory-allocation-and-register-set-in-8051-microcontroller/)of a computer directly without making the compiler to perform storing and loading functions.

**Features of CISC Architecture**

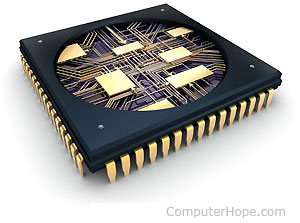
* To simplify the computer architecture, CISC supports microprogramming.
* CISC have more number of predefined instructions which makes high level languages easy to design and implement.
* CISC consists of less number of registers and more number of addressing modes, generally 5 to 20.
* CISC processor takes varying cycle time for execution of instructions – multi-clock cycles.
* Because of the complex instruction set of the CISC, the pipelining technique is very difficult.
* CISC consists of more number of instructions, generally from 100 to 250.
* Special instructions are used very rarely.
* Operands in memory are manipulated by instructions.

**Advantages of CISC architecture**

* Each machine language instruction is grouped into a microcode instruction and executed accordingly, and then are stored inbuilt in the memory of the main processor, termed as microcode implementation.
* As the microcode memory is faster than the main memory, the microcode instruction set can be implemented without considerable speed reduction over hard wired implementation.
* Entire new instruction set can be handled by modifying the micro program design.
* CISC, the number of instructions required to implement a program can be reduced by building rich instruction sets and can also be made to use slow main memory more efficiently.
* Because of the superset of instructions that consists of all earlier instructions, this makes micro coding easy.

**Drawbacks of CISC**

* The amount of clock time taken by different instructions will be different – due to this – the performance of the machine slows down.
* The instruction set complexity and the chip hardware increases as every new version of the processor consists of a subset of earlier generations.
* Only 20% of the existing instructions are used in a typical programming event, even though there are many specialized instructions in existence which are not even used frequently.
* The conditional codes are set by the CISC instructions as a side effect of each instruction which takes time for this setting – and, as the subsequent instruction changes the condition code bits – so, the compiler has to examine the condition code bits before this happens.



**Figure A1.1: CISC**

## RISC (Reduced Instruction Set Computer) Architecture:

The [microcontroller architecture](https://www.elprocus.com/8051-microcontroller-architecture-and-applications/) that utilizes small and highly optimized set of instructions is termed as the Reduced Instruction Set Computer or simply called as RISC. It is also called as LOAD/STORE architecture. In the late 1970s and early 1980s, RISC projects were primarily developed from Stanford, UC-Berkley and IBM. The John Coke of IBM research team developed RISC by reducing the number of instructions required for processing computations faster than the CISC. The RISC architecture is faster and the chips required for the manufacture of RISC architecture is also less expensive compared to the CISC architecture.

**Typical Features of RISC Architecture**

* Pipelining technique of RISC, executes multiple parts or stages of instructions simultaneously such that every instruction on the CPU is optimized. Hence, the RISC processors have Clock per Instruction of one cycle, and this is called as One Cycle Execution.
* It optimizes the [usage of register](https://www.elprocus.com/know-about-types-of-registers-in-8051-microcontroller/) with more number of registers in the RISC and more number of interactions within the memory can be prevented.
* Simple addressing modes, even complex addressing can be done by using arithmetic [AND/ OR logical operations](https://www.elprocus.com/different-types-of-digital-logic-circuits/).
* It simplifies the compiler design by using identical general purpose registers which allows any register to be used in any context.
* For efficient usage of the registers and optimization of the pipelining uses, reduced instruction set is required.
* The number of bits used for the opcode is reduced.
* In general there are 32 or more registers in the RISC.

**Advantages of RISC processor architecture**

* Because of the small set of instructions of RISC, high-level language compilers can produce more efficient code.
* RISC allows freedom of using the space on [microprocessors](https://www.elprocus.com/embedded-microprocessor-importance-and-its-real-time-applications/)because of its simplicity.
* Instead of using Stack, many RISC processors use the registers for passing arguments and holding the local variables.
* RISC functions uses only a few parameters, and the RISC processors cannot use the call instructions, and therefore, use a fixed length instructions which are easy to pipeline.
* The speed of the operation can be maximized and the execution time can be minimized.
* Very less number of instruction formats (less than four), a few number of instructions (around 150) and a few addressing modes (less than four) are needed.

**Drawbacks of RISC processor architecture**

* With the increase in length of the instructions, the complexity increases for the RISC processors to execute due to its character cycle per instruction.
* The performance of the RISC processors depends mostly on the compiler or programmer as the knowledge of the compiler plays a major role while converting the CISC code to a RISC code; hence, the quality of the generated code depends on the compiler.
* While rescheduling the CISC code to a RISC code, termed as a code expansion, will increase the size. And, the quality of this code expansion will again depend on the compiler, and also on the machine’s instruction set.
* The first level cache of the RISC processors is also a disadvantage of the RISC, in which these processors have large memory caches on the chip itself. For feeding the instructions, they require very [fast memory systems](https://www.elprocus.com/different-types-of-memory-modules-used-embedded-system/).



**Figure A1.2: RISC**

* **ARM (Advanced RISC Machines):-**

An ARM processor is one of a family of [CPUs](https://whatis.techtarget.com/definition/processor) based on the [RISC](https://search400.techtarget.com/definition/RISC) (reduced instruction set computer) architecture developed by Advanced RISC Machines (ARM). ARM makes 32-bit and [64-bit](https://searchdatacenter.techtarget.com/definition/64-bit-processor) RISC [multi-core processors](https://searchdatacenter.techtarget.com/definition/multi-core-processor). RISC [processors](https://whatis.techtarget.com/definition/microprocessor-logic-chip) are designed to perform a smaller number of types of computer [instructions](https://whatis.techtarget.com/definition/instruction) so that they can operate at a higher speed, performing more millions of instructions per second ([MIPS](https://searchitoperations.techtarget.com/definition/MIPS-million-instructions-per-second)).  By stripping out unneeded instructions and optimizing pathways, RISC processors provide outstanding performance at a fraction of the power demand of [CISC](https://whatis.techtarget.com/definition/CISC-complex-instruction-set-computer-or-computing) (complex instruction set computing) devices.

ARM processors are extensively used in consumer electronic devices such as [smartphones](https://searchmobilecomputing.techtarget.com/definition/smartphone), [tablets](https://searchmobilecomputing.techtarget.com/definition/tablet-PC), multimedia players and other mobile devices, such as [wearables](https://internetofthingsagenda.techtarget.com/definition/wearable-computer). Because of their reduced [instruction set](https://whatis.techtarget.com/definition/instruction-set), they require fewer [transistors](https://whatis.techtarget.com/definition/transistor), which enable a smaller die size for the integrated circuitry ([IC](https://whatis.techtarget.com/definition/integrated-circuit-IC)). The ARM processor’s smaller size, reduced complexity and lower power consumption makes them suitable for increasingly miniaturized devices.

ARM processor features include:

* Load/store architecture.
* An [orthogonal](https://searchstorage.techtarget.com/definition/orthogonal) instruction set.
* Mostly single-cycle execution.
* Enhanced power-saving design.
* 64 and 32-bit execution states for scalable high performance.
* [Hardware virtualization](https://searchservervirtualization.techtarget.com/definition/hardware-virtualization) support.



**Figure A1.3: ARM**

* **DSP (Digital Signal Processors):**

Digital Signal Processors (DSP) take real-world signals like voice, audio, video, temperature, pressure, or position that have been digitized and then mathematically manipulate them. A DSP is designed for performing mathematical functions like "add", "subtract", "multiply" and "divide" very quickly.

Signals need to be processed so that the information that they contain can be displayed, analyzed, or converted to another type of signal that may be of use. In the real-world, analog products detect signals such as sound, light, temperature or pressure and manipulate them. Converters such as an Analog-to-Digital converter then take the real-world signal and turn it into the digital format of 1's and 0's. From here, the DSP takes over by capturing the digitized information and processing it. It then feeds the digitized information back for use in the real world. It does this in one of two ways, either digitally or in an analog format by going through a Digital-to-Analog converter. All of this occurs at very high speeds.

A DSP contains these key components:

* **Program Memory**: Stores the programs the DSP will use to process data
* **Data Memory:** Stores the information to be processed
* **Compute Engine:** Performs the math processing, accessing the program from the Program Memory and the data from the Data Memory
* **Input/ Output:** Serves a range of functions to connect to the outside world



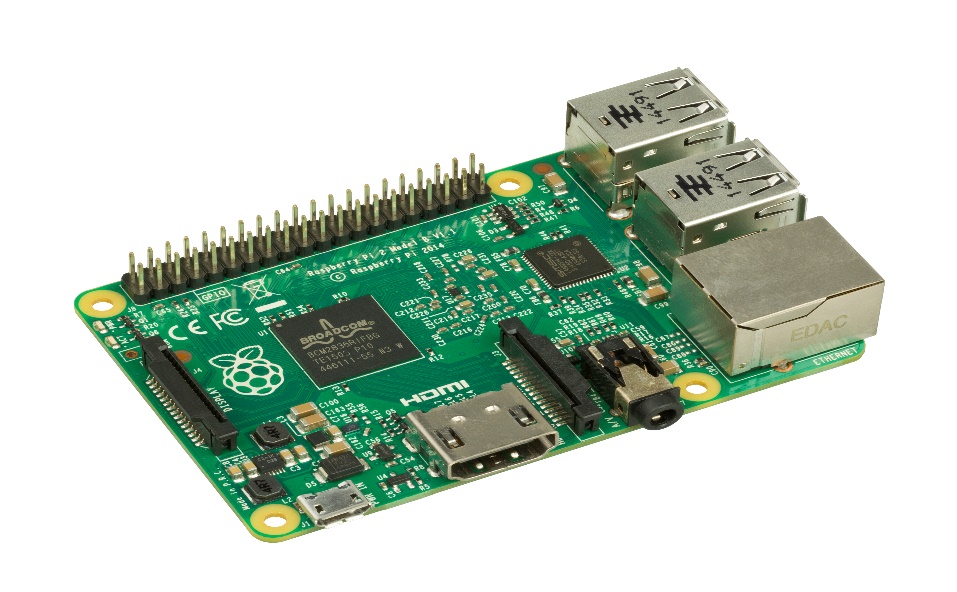
**Figure A1.4: DSP**

* **SOC (System on a Chip):**

A system on a chip is an [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit) (also known as a "chip") that integrates all components of a [computer](https://en.wikipedia.org/wiki/Computer) or other [electronic system](https://en.wikipedia.org/wiki/Electronics). These components typically (but not always) include a [central processing unit](https://en.wikipedia.org/wiki/Central_processing_unit) (CPU), [memory](https://en.wikipedia.org/wiki/Computer_memory), [input/output](https://en.wikipedia.org/wiki/Input/output) ports and [secondary storage](https://en.wikipedia.org/wiki/Computer_data_storage#Secondary_storage) – all on a single [substrate](https://en.wikipedia.org/wiki/Wafer_(electronics)) or microchip, the size of a coin.[[1]](https://en.wikipedia.org/wiki/System_on_a_chip#cite_note-2) It may contain [digital](https://en.wikipedia.org/wiki/Digital_signal_(electronics)), [analog](https://en.wikipedia.org/wiki/Analog_signal), [mixed-signal](https://en.wikipedia.org/wiki/Mixed-signal_integrated_circuit), and often [radio frequency](https://en.wikipedia.org/wiki/Radio_frequency) [signal processing](https://en.wikipedia.org/wiki/Signal_processing) functions, depending on the application. As they are integrated on a single substrate, SOC consume much less power and take up much less area than multi-chip designs with equivalent functionality. Because of this, SoCs are very common in the [mobile computing](https://en.wikipedia.org/wiki/Mobile_computing) (such as in [Smartphones](https://en.wikipedia.org/wiki/Smartphone)) and [edge computing](https://en.wikipedia.org/wiki/Edge_computing) markets. Systems on chip are commonly used in [embedded systems](https://en.wikipedia.org/wiki/Embedded_system) and the [Internet of Things](https://en.wikipedia.org/wiki/Internet_of_things).

Systems on Chip are in contrast to the common traditional [motherboard](https://en.wikipedia.org/wiki/Motherboard)-based [PC](https://en.wikipedia.org/wiki/Personal_computer) [architecture](https://en.wikipedia.org/wiki/Computer_architecture), which separates components based on function and connects them through a central interfacing circuit board. Whereas a motherboard houses and connects detachable or replaceable components, SoCs integrate all of these components into a single integrated circuit, as if all these functions were built into the motherboard. A SoC will typically integrate a CPU, graphics and memory interfaces, hard-disk and USB connectivity, [random-access](https://en.wikipedia.org/wiki/Random-access_memory) and [read-only](https://en.wikipedia.org/wiki/Read-only_memory) [memories](https://en.wikipedia.org/wiki/Computer_memory) and secondary storage on a single circuit die, whereas a motherboard would connect these modules as [discrete components](https://en.wikipedia.org/wiki/Discrete_components) or [expansion cards](https://en.wikipedia.org/wiki/Expansion_card).

A SoC integrates a [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) or [microprocessor](https://en.wikipedia.org/wiki/Microprocessor) with advanced peripherals like [graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit)(GPU), [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) module, or one or more [coprocessors](https://en.wikipedia.org/wiki/Coprocessor).[[4]](https://en.wikipedia.org/wiki/System_on_a_chip#cite_note-8) Similar to how a microcontroller integrates a microprocessor with peripheral circuits and memory, an SoC can be seen as integrating a microcontroller with even more advanced [peripherals](https://en.wikipedia.org/wiki/Peripheral).



**Figure A1.5: SOC**

**XAVIER INSTITUTE OF ENGINEERING**

**Department of Information Technology**

**Class/ Sem/ A.Y: TE IT/ V/ 2019-20**

**Course Name:IoT Mini-Project Lab**

Group No/ Roll No: 17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assignment No. 2: Embedded System Tools** | | | | |
| **LO1:Identify the requirements for the real world problems.**  **LO2: Conduct a survey of several available literatures in the preferred field of study.**  **LO3: Study software/ hardware skills and build the project successfully by hardware requirements, coding, emulating and testing.**  **LO4: To report and present the findings of the study conducted in the preferred domain**  **LO5: Demonstrate an ability to work in teams and manage the conduct of the research study.** | | | | |
| **Rubrics For Assignment Work** | | | | |
| **Roll No.** | **Name of the Student** | **Knowledge & Content**  **(03)** | **Neatness and Timeline**  **(02)** | **Total (05)** |
| XIEIT171805 | SHUBHAM CHAVAN |  |  |  |
| XIEIT171820 | ACHAL KALWAR |  |  |  |
| XIEIT171831 | ROHAN MATHUR |  |  |  |

Ms. Meena Ugale

**ASSIGNMENT NO. 02**

**EMBEDDED SYSTEM TOOLS**

There is a variety of embedded tools available out there for development of hardware and software for embedded systems. These embedded tools include editors, compilers, assemblers, debuggers, and simulators etc. for software part and soldering iron, desoldering gun, Digital Multimeter, oscilloscope, cutter, laptop etc. as hardware tools. All of the embedded tools are necessary and work together. After a short review of software tools, I am going to write about hardware tools which are the main focus of my article.

* **ARDUINO:**

**Introduction:**

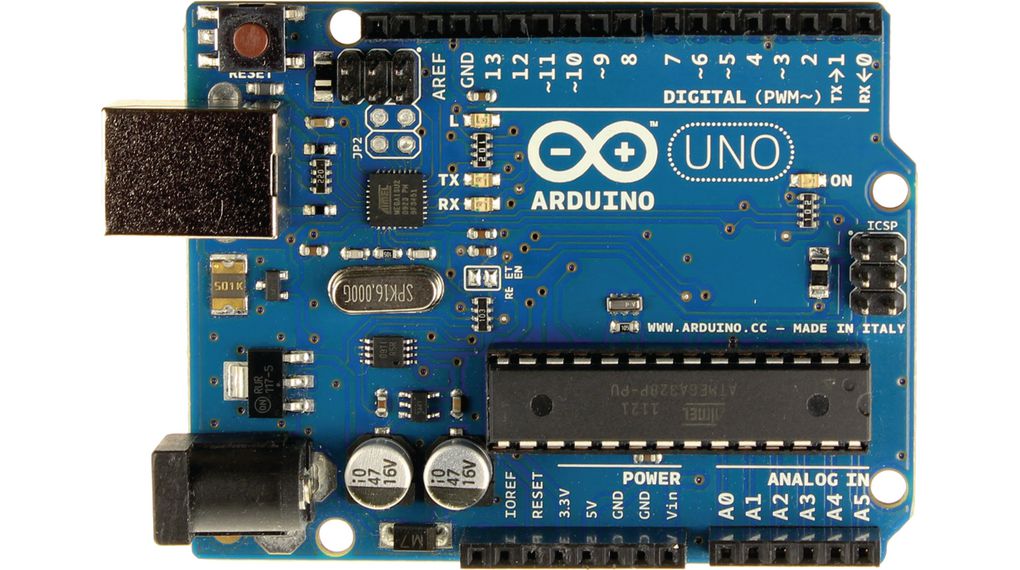
Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

**What does it do?**

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a **huge** variety of Arduino-based projects.

For everything from robots and a heating pad hand warming blanket to honest fortune-telling machines, and even a Dungeons and Dragons dice-throwing gauntlet, the Arduino can be used as the brains behind almost any electronics project.



**Figure A2.1: Arduino**

**Types Of Arduino:**

**Table A2.1 Types of Arduino**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arduino Board | Processor | Memory | Digital I/O | Analogue I/O |
| Arduino Uno | 16Mhz ATmega328 | 2KB SRAM, 32KB flash | 14 | 6 input,  0 output |
| Arduino Due | 84Mhz AT91SAM3X8E | 96KB SRAM, 512KB flash | 54 | 5412 input,  2 output |
| Arduino Mega | 16MHz ATmega2560 | 8KB SRAM, 256KB flash | 54 | 5416 input,  0 output |

* **RASPBERRY PI:**

Generally, the raspberry pi is designed for the educational purpose and it is in small size like a credit card. It will improve the programming skills & hardware programming at the higher education in the schools or colleges. The raspberry pi is slower than the laptop or desktop and it is totally in the Linux operating system. The availability of price in the market is low.



**Figure A2.2: Raspberry Pi**

**Different Types of Raspberry Pi Models**

The different types of raspberry pi models are following:

* **Raspberry Pi 1 model B:**

It is a higher-spec variant of raspberry pi. After this design of this raspberry pi, it has extended to the next model i.e. raspberry pi 2. The specifications of the raspberry pi model B are following, the raspberry pi model B has two USB ports, having a RAM of 512MB and its Ethernet port is 100mb.

* **Raspberry Pi 1 model A:**

This model is usually bought for embedded projects: because it’s missing a few ports and an Ethernet chip, the Model A is lighter and consumes less power than a Model B. Model A is the lower-spec variant of the Raspberry Pi, with 256 MB of RAM, one USB port and no Ethernet port. Model A in combination with a wifi dongle is also ideal for users who just want a Raspberry Pi to act as a media centre running behind their television.

* **Raspberry Pi 1 model B+:**

This model B+ is replaced in the place of raspberry pi model B in the year 2014. Model B+ Rpi is compared with the model B it has more GPIO, more USB,more SD, lower power consumption

* **Raspberry Pi 1model A+:**

The Raspberry Pi 1 Model A+ (A Plus) Is just 56mm long, 12mm thick and uses up to 45% less power than a Raspberry Pi 1 Model B+.

This newly updated version of the Model A does much more than simply bring it in line with the Raspberry Pi Model B+ released in the summer - the Raspberry Pi Model A+ is the same dimensions as a HAT (Hardware Attached on Top) add-on board and can run at nearly half the power consumption as the Model B+.

* **Raspberry Pi Zero:**

It is a half size of the model A+ with twice a utility and for any project, it has the same specification like 1GHz, Single-core CPU, 512MB RAM, Mini-HDMI port, Micro-USB OTG port, Micro-USB power, HAT-compatible 40-pin header, Composite video and reset headers, CSI camera connector (v1.3 only).

* **Raspberry Pi 2:**

The basic image of the raspberry pi 2 is following and the features of the raspberry pi 2 are it has quad-core ARM cortex-A7 processor with a 900MHz; the SDRAM is about the 1GB. It is completely compatible with the raspberry pi 1.

* **Raspberry Pi 3 model B:**

This is the Raspberry Pi Model B 512MB RAM model with two USB ports and a 10/100 Ethernet controller.

As typical of modern computers, generic USB keyboards and mice are compatible with the Raspberry Pi. The Raspberry Pi use Linux-kernel based operating systems. The Raspberry Pi does not come with a real-time clock, so an OS must use a network time server, or ask the user for time information at boot time to get access to time and date info for file time and date stamping. However a real time clock (such as the DS1307) with battery backup can be easily added via the I2C interface.

* **Raspberry Pi Zero W:**

The Raspberry Pi Zero W extends the Pi Zero family and comes with added wireless LAN and Bluetooth connectivity.

Raspberry Pi Zero W is a super-compact, hackable, and ultra-low-cost computer, This is the version with built-in WiFi and Bluetooth, and there is no so call clone version of this super low-cost computer.

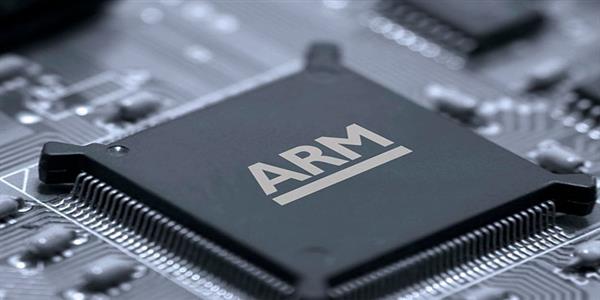
The new Raspberry Pi Zero W offers all the benefits of the Pi Zero, but with one big difference – built-in Wireless (WiFi and Bluetooth)! More specifically, this giant upgrade is the addition of a BCM43143 WiFi chip BUILT-IN to your Raspberry Pi Zero – just like the Pi 3.

**Applications of Raspberry Pi**

The different applications of the raspberry pi model are:

* Tablet computer
* Home automation
* Internet radio
* Controlling robots
* Cosmic Computer
* Arcade machines
* **ARM CORTEX:**

The ARM stands for Advanced RISC machine and it is a 32-bit reduced instructions set computer (RISC) microcontroller. It was first introduced by the Acron computers’ organization in 1987. The ARM is a family of the microcontroller developed by the different manufacturers such as ST microelectronics, Motorola and so on. The ARM microcontroller architecture come with a few different versions such as ARMv1, ARMv2 etc. and each one has its own advantage and disadvantages.



**Figure A2.3: Arm Cortex**

The ARM cortex microcontroller is an advanced microcontroller in the ARM family, which is developed by the ARMv7 architecture. The ARM cortex family divided into three sub-families such as:

* **ARM-Cortex Ax-series:**

The ARM Cortex-A is a group of 32-bit and 64-bit RISC ARM processor cores licensed by Arm Holdings. The cores are intended for application use. The group consists of 32-bit cores: ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15, ARM Cortex-A17 MP Core, and ARM Cortex-A32, and 64-bit cores: ARM Cortex-A35, ARM Cortex-A53, ARM Cortex-A55, ARM Cortex-A57, ARM Cortex-A72, ARM Cortex-A73, ARM Cortex-A75, and ARM Cortex-A76.

* **ARM-Cortex Rx-series:**

The ARM Cortex-R is a family of 32-bit RISC ARM processor cores licensed by Arm Holdings. The cores are optimized for hard real-time and safety-critical applications. Cores in this family implement the ARM Real-time (R) profile, which is one of three architecture profiles, the other two being the Application (A) profile implemented by the Cortex-A family and the Microcontroller (M) profile implemented by the Cortex-M family. The ARM Cortex-R family of microprocessors currently consists of ARM Cortex-R4 (F), ARM Cortex-R5(F), ARM Cortex-R7(F), ARM Cortex-R8(F), and ARM Cortex-R52(F).

* **ARM-Cortex Mx-series:**

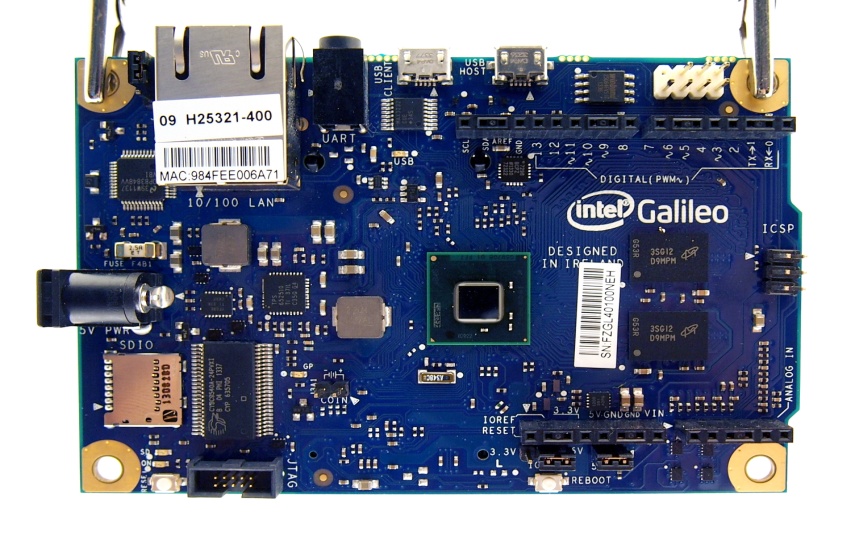
The ARM Cortex-M is a group of 32-bit RISC ARM processor cores licensed by Arm Holdings. They are intended for microcontroller use, and have been shipped in tens of billions of devices. The cores consist of the Cortex-M0, Cortex-M0+, Cortex-M1, Cortex-M3, Cortex-M4, Cortex-M7, Cortex-M23, Cortex-M33, Cortex-M35P. The Cortex-M4 / M7 / M33 / M35P cores have an FPU silicon option, and when included in the silicon these cores are known as "Cortex-Mx with FPU" or "Cortex-MxF", where 'x' is the core number.

* **INTEL GALILEO:**

Galileo is a microcontroller board based on the Intel® Quark SoC X1000 Application Processor, a 32-bit Intel Pentium-class system on a chip (datasheet). It’s the first board based on Intel® architecture designed to be hardware and software pin-compatible with Arduino shields designed for the Uno R3. Digital pins 0 to 13 (and the adjacent AREF and GND pins), Analog inputs 0 to 5, the power header, ICSP header, and the UART port pins (0 and 1), are all in the same locations as on the Arduino Uno R3. This is also known as the Arduino 1.0 pinout.

Galileo is designed to support shields that operate at either 3.3V or 5V. The core operating voltage of Galileo is 3.3V. However, a jumper on the board enables voltage translation to 5V at the I/O pins. This provides support for 5V Uno shields and is the default behavior. By switching the jumper position, the voltage translation can be disabled to provide 3.3V operation at the I/O pins.

Of course, the Galileo board is also software compatible with the Arduino Software Development Environment (IDE), which makes usability and introduction a snap. In addition to Arduino hardware and software compatibility, the Galileo board has several PC industry standard I/O ports and features to expand native usage and capabilities beyond the Arduino shield ecosystem. A full sized mini-PCI Express slot, 100Mb Ethernet port, Micro-SD slot, RS-232 serial port, USB Host port, USB Client port, and 8MByte NOR flash come standard on the board.



**Figure A2.4: Intel Galileo**

* **DIFFERENT TYPES OF SENSORS**

**Sensor:**

Sensor as an input device which provides an output (signal) with respect to a specific physical quantity (input).

We live in a World of Sensors. You can find different types of Sensors in our homes, offices, cars etc. working to make our lives easier by turning on the lights by detecting our presence, adjusting the room temperature, detect smoke or fire, make us delicious coffee, open garage doors as soon as our car is near the door and many other tasks

The following is a list of different types of sensors that are commonly used in various applications. All these sensors are used for measuring one of the physical properties like Temperature, Resistance, Capacitance, Conduction, Heat Transfer etc.

* **Temperature Sensor:**

One of the most common and most popular sensors is the Temperature Sensor. A Temperature Sensor, as the name suggests, senses the temperature i.e. it measures the changes in the temperature.

In a Temperature Sensor, the changes in the Temperature correspond to change in its physical property like resistance or voltage.

There are different types of Temperature Sensors like Temperature Sensor ICs (like LM35), Thermistors, Thermocouples, RTD (Resistive Temperature Devices), etc.



**Figure A2.5: Temperature Sensor**

* **Proximity Sensor**

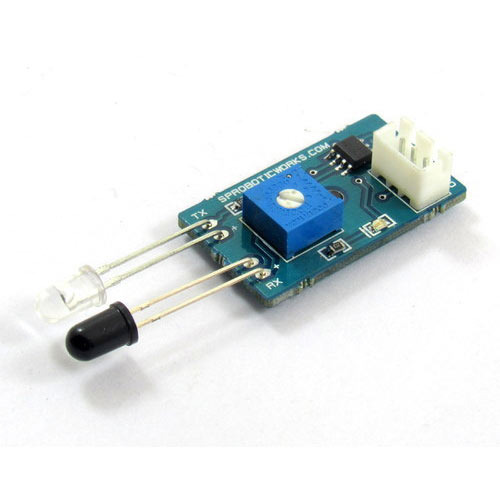
A Proximity Sensor is a non-contact type sensor that detects the presence of an object. Proximity Sensors can be implemented using different techniques like Optical (like Infrared or Laser), Ultrasonic, Hall Effect, Capacitive, etc.



**Figure A2.6: Proximity Sensor**

* **IR Sensor (Infrared Sensor)**

IR Sensors or Infrared Sensor is light based sensor that are used in various applications like Proximity and Object Detection. IR Sensors are used as proximity sensors in almost all mobile phones.



**Figure A2.7: Infrared Sensor**

* **Pressure Sensor:**

A **pressure sensor** is an instrument consisting of a pressure sensitive element to determine the actual pressure applied to the sensor (using different working principles) and some components to convert this information into an output signal.

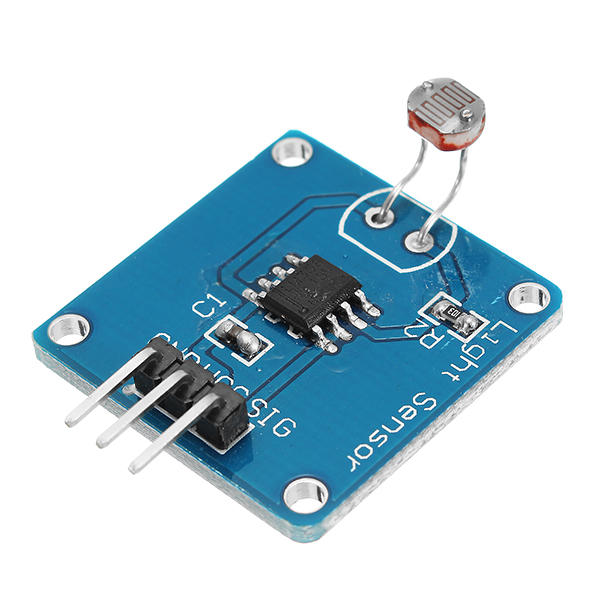


**Figure A2.8: Pressure Sensor**

* **Light Sensor:**

Light Sensors are photoelectric devices that convert light energy (photons) whether visible or infra-red light into an electrical (electrons) signal

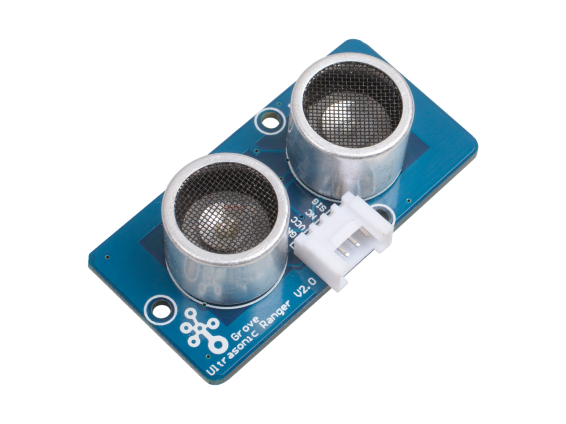
A **Light Sensor** generates an output signal indicating the intensity of light by measuring the radiant energy that exists in a very narrow range of frequencies basically called “light”, and which ranges in frequency from “Infra-red” to “Visible” up to “Ultraviolet” light spectrum.



**Figure A2.9: Light Sensor**

* **Ultrasonic Sensor:**

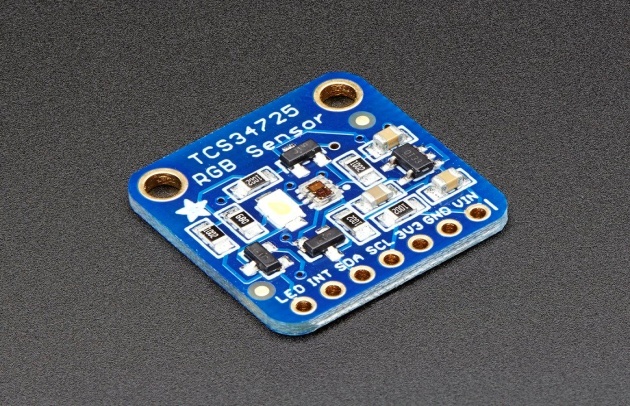
An Ultrasonic Sensor is a non-contact type device that can be used to measure distance as well as velocity of an object. An Ultrasonic Sensor works on the properties of the sound waves with frequency greater than that of the human audible range.



**Figure A2.10: Ultrasonic Sensor**

* **Color Sensor:**

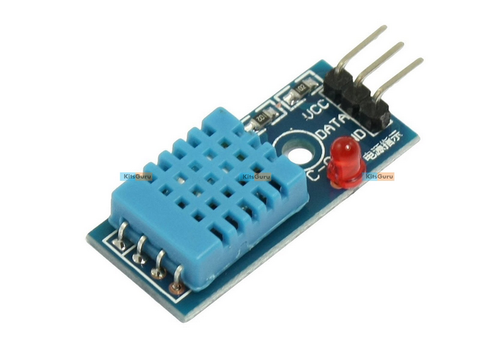
Color sensors detect the color of a surface. The sensors cast light (red, green, and blue LEDs) on the objects to be tested, calculate the chromaticity coordinates from the reflected radiation and compare them with previously stored reference colors. If the color values are within the set tolerance range, a switching output is activated.



**Figure A2.11: Color Sensor**

* **Humidity Sensor:**

A humidity sensor (or hygrometer) senses, measures and reports both moisture and air temperature. The ratio of moisture in the air to the highest amount of moisture at a particular air temperature is called relative humidity. Relative humidity becomes an important factor, when looking for comfort.



**Figure A2.12: Humidity Sensor**