NIGHT PATROL ROBOT

A MINI PROJECT REPORT

Submitted by

S. USHASWI **19RH1A04M0**

S.VAISHNAVI **19RH1A04K4**

V.SUSHMA SWARAJ **19RH1A0445**

Under the Esteemed Guidance of

Dr. N.Sreekanth
Professor & Head

in partial fulfillment of the Academic Requirements for the Degree of

BACHELOR OF TECHNOLOGY

Electronics and Communication Engineering

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN

(Autonomous Institution-UGC, Govt. of India)

Accredited by NBA & NAAC with 'A' Grade

NIRF Indian Ranking, Accepted by MHRD, Govt. of India | Rank Band – Excellent by ARIIA, Accepted by MHRD, Govt. of India

Approved by AICTE, Permanently Affiliated to JNTUH, ISO 9001:2015 Certified Institution

Platinum Rated by AICTE-CII Survey, AAAA+ Rated by Digital Learning Magazine, AAAA Rated by Careers 360,

National Ranking-Top 100 Rank band by Outlook Magazine, Ranked as Top Engineering Colleges of Eminence in India – 2022 by CSR Rankings,

51st National Ranking & 5th Telangana State Ranking by Times of India News Magazine, 86th National Ranking by the Week Magazine of India

Maisammaguda, Dhulapally, Secunderabad, Kompally-500100.

2022-23

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
(Autonomous Institution-UGC, Govt. of India)
Accredited by NBA & NAAC with 'A' Grade
NIRF Indian Ranking, Accepted by MHRD, Govt. of India | Rank Band – Excellent by ARIIA, Accepted by MHRD, Govt. of India
Approved by AICTE, Permanently Affiliated to JNTUH, ISO 9001:2015 Certified Institution
Platinum Rated by AICTE-CII Survey, AAAA+ Rated by Digital Learning Magazine, AAAA Rated by Careers 360,
National Ranking-Top 100 Rank band by Outlook Magazine, Ranked as Top Engineering Colleges of Eminence in India – 2022 by CSR Rankings,
51st National Ranking & 5th Telangana State Ranking by Times of India News Magazine, 86th National Ranking by the Week Magazine of India
Maisammaguda, Dhulapally, Secunderabad, Kompally-500100

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled " **NIGHT PATROL ROBOT** " is carried out by **S. Ushaswi (19RH1A04M0), S. Vaishnavi (19RH1A04K4), V. Sushma Swaraj (19RH1A04P3)** in partial fulfillment for the award of degree of **BACHELOR OF TECHNOLOGY** in Electronics and communication Engineering, Jawaharlal Nehru Technological University, Hyderabad during the academic year 2019-2020.

Supervisor's Signature	**Head of the Department**
Name	**Dr. N. Sreekanth**
Designation	**Professor of ECE**

External Examiner

Date: 22-08-2022

PROJECT COMPLETION CERTIFICATE

This is to certify that the Project work titled "Night Patrol Robot" by the Students of "Malla Reddy Engineering College For Women" has successfully completed Project Work under our guidance during at Dcode soft tech solution pvt ltd , Hyderabad .Their performance in this period is satisfactory.

S. Ushaswi	**19RH1A04M0**
S. Vaishnavi	**19RH1A04K4**
V. Sushma Swaraj	**19RH1A04P3**

Director,

Dcode Soft Tech Solutions Pvt Ltd

No. 1-7-286/1, First Floor, Opp. Genius Grammar School,
Chaitanyapuri, Kothapet, Dilsukhnagar, Hyderabad - 500 060.
Whatsapp & Phone : 7989919045, 9491838806
Email : hr@dcodetechnologies.in, info@dcodetechnologies.in
Website : www.dcodetechnologies.in

ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College for Women** and Department of **Electronics and Communication Engineering** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We would like to deeply thank our Honorable Minister of Telangana State **Sri.Ch. Malla Reddy Garu,** founder chairman MRGI, the largest cluster of institutions in the state of Telangana for providing us with all the resources in the college to make our project success.

We wish to convey gratitude to our **Principal Dr. Y. Madhavee Latha,** for providing us with the environment and mean to enrich our skills and motivating us in our endeavor and helping us to realize our full potential.

We express our sincere gratitude to **Dr. N. Sreekanth, Head of the Department** of Electronics and Communication Engineering for inspiring us to take up a project on this subject and successfully guiding us towards its completion.

We would also like to thank our Project coordinator **Dr. K. Sudhakar,** for his kind encouragement and overall guidance in viewing this program a good asset with profound gratitude.

We would like to thank our internal guide **Mrs. H. Bagya Lakshmi,** and all the Faculty members for their valuable guidance and encouragement towards the completion of our project work.

With Regards and Gratitude

S. Ushaswi(19RH1A04M0)

S. Vaishnavi(19RH1A04K4)

V. Sushma Swaraj(19RH1A04P3)

DECLARATION

We hereby declare that our project entitled **“NIGHT PATROL ROBOT”** submitted to **Malla Reddy Engineering College for Women,** affiliated to **Jawaharlal Nehru Technological University, Hyderabad** for the award of the Degree of Bachelor of Technology in **Electronics and Communication Engineering** is a result of original research work done by us.

It is declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of Degree.

S. Ushaswi(19RH1A04M0)
S. Vaishnavi(19RH1A04K4)
V. Sushma Swaraj(19RH1A04P3)

ABSTRACT

At present the observation during night turned out to be exceptionally testing task. There are a few spots where people can't be engaged with watching. A fundamental prerequisite of this circumstance is a robot which consequently identifies trespassers in the territory like workplaces, home, building and so forth and report close by board security control unit. In the current work, A late evening guarding robot is created with upgraded capacity to recognize and alarm if there is any human movement in the territory to give exact observing framework. The Night Patrolling Robotic vehicle moves in a random path while watching. The framework utilizes IR based way following framework for watching allocated zone. The development of a robot is additionally controlled consequently through deterrent recognizing sensors to stay away from the crash. When IR sensor get on condition time the robot get stopped and buzzer get on condition. Here we are using ESP32 with camera, and here we are using IOT server for live video streaming.

INDEX PAGE

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION ABOUT THE PROJECT

A significant proportion of Indian firms might have faced data theft issues recently. Around 41 per cent of these companies saw such instances, compared to the global average of 29 per cent, showed a survey by a risk management and consultancy firm, released on Monday. India's number is higher than other countries such as the United States (26 per cent), the United Kingdom (32 per cent) and Japan (27 per cent). Autonomous security robots are a new groundbreaking innovation in advanced security and surveillance technology. Guards patrolling an area with flashlights and batons never worked very well; however, today there is an immergence of smart security systems with intelligent sensors, embedded systems, autonomous control mechanism and mobile application. The first security surveillance robot was proposed by Everett, H. & Gage, D.W, 1999 in "Mobile Detection, Assessment and Response System (MDARS)". Since then security robots have become a growing interest with increasing developments in research and application. The idea behind this is to secure the whole area of a building or premises. Any small sound results in the alert and robot automatically goes to the particular area and captures the image of that area and sends it to the user. Raspberry Pi connected with the camera plays an important role in making an automatic robotic system. During the occurrence of an abnormal event like theft, the patrol robot detects the sound and captures the image of the intruder. The image of the intruder is sent to the security department via Email and Short Messaging Service (SMS). An automatic patrolling vehicle acts as a security patroller in the security system, which can monitor those dead zones of the traditional fixed surveillance system. Countless applications today are mobile service security robots, including autonomous navigation, security patrolling, housework, search-and-rescue operations, material handling, manufacturing, and automated transportation systems. Autonomous navigation remains one of the primary challenges in the mobile-robot industry; many control algorithms and techniques have been recently developed that aim to overcome this challenge. We use line follower with IR Sensors to perform night patrol navigation. The security patrol robot will utilize several sensors and motors in order to navigate indoors. It will also be able to communicate and be controlled via Wi-Fi.

1.2LITERATURE SURVEY

After going through various articles and research papers we concluded that some of the papers were beneficial for designing our project and make it a successful one. In Military 2020 Spying Robot by Sarmad Hameed, Muhammad Hamza Khan, Naqi Jafri, the massive tasks is dangerous in war field. In border region it gets difficult for the humans working in the battle field to protect themselves from harm. Both protecting themselves and keeping keen observation on enemy becomes a little bit difficult task so in that situation robot is better option. Consequently robot replaces the trooper. In Spying Robot With Night Vision Camera by Aaruni Jha, Apoorva Singh, Ravinder Turna -The robot sends the flag to the RF collector mounted on the robot through RF transmitter at the base station. Due to this robot records real time footages and videos and can deliver those at our phone screen even in dark also as LED lights are used which even the enemy in the at border region or in suspected area can’t even recognize that something is getting recorded. And the work done by PriyankaYadav, Swati Gawhale-She concluded that during the period of battle against enemy this robot can be used to collect all the necessary data that may weaken the opponent’s plan if in case plotting something dangerous to attack them. In this way the military men would get prepared themselves for anything that the opponent is plotting against them and retort to their action in a better way that the enemy could not even think of at correct time.

Robot navigation problems can be generally classified as global or local, depending upon the environment surrounding the robot. In global navigation, the environment surrounding the robot is known and a path which avoids the obstacles is selected. In one example of the global navigation techniques, graphical maps which contain information about the obstacles are used to determine a desirable path. In local navigation, the environment surrounding the robot is unknown, or only partially known, and sensors have to be used to detect the obstacles and a collision avoidance system must be incorporated into the robot to avoid the obstacles. The artificial potential field approach is one of the well-known techniques which has been developed for this purpose. Krogh, for example, used a generalized potential field approach to obstacle avoidance. Kilm and Khosla used instead harmonic potential functions for obstacle avoidance. On the other hand, Krogh and Fang used the dynamic generation of sub goals using local feedback information. During the past few years, potential field methods (PFM) for obstacle avoidance have gained increased popularity among researchers in the field of robots and mobile robots. The idea of imaginary forces acting on a robot hasbeen suggested by Andrews and Hogan and Khatib. In these approaches' obstacles exert repulsive forces onto the robot, while the target applies an attractive force to the robot. The sum of all forces, the resultant force R, determines the subsequent direction and speed of travel. One of the reasons for the popularity of this method is its simplicity and elegance.This paper introduces histogram in-motion mapping (HIMM), a new method for real-time map building with a mobile robot in motion. HIMM represents data in a two-dimensional array, called a histogram grid, that is updated through rapid in motion sampling of onboard range sensors. Rapid in-motion sampling results in a map representation that is well-suited to modeling inaccurate and noisy range-sensor data, such as that produced by ultrasonic sensors, and requires minimal computational overhead. Fast map-building allows the robot to immediately use the mapped information in real-time obstacle-avoidance algorithms. The benefits of this integrated approach are twofold:quick, accurate mapping;and safe.

Real-time obstacle avoidance is one of the key issues to successful application of mobile robot systems. All mobile robots feature some kind of collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot short of it in order to avoid a collision, through sophisticate algorithms, that enable the robot to detour obstacle. The later algorithms are much more complex, since they involve not only the detection of an obstacle, but also some kind of quantitative measurements concerning the obstacle's dimensions. In our system the ultrasonic sensors are continuously sampled while the robot is moving. If an obstacle produces an echo, the corresponding cell contents are incremented. A solid, motionless obstacle eventually causes a high count in the corresponding cells. Misreading, on the other hand, occur randomly, and do not cause high count in any particular cell. These methods yield a more reliable obstacle representation in spite of the ultrasonic sensor's inaccuracies.

Many definitions of the Internet of Things exist, but at the most fundamental level it can be described as a network of devices interacting with each other via machine to machine (M2M) communications, enabling collection and exchange of data [9], [10], [11]. This technology enables automation within a large range of industries, as well as allowing for the collection of big data. Hailed as the driver of the Fourth Industrial Revolution [12], Internet of Things technology has already found commercial use in areas such as smart parking [14], precision agriculture and water usage management. Extensive research has also been conducted into the use of IoT for developing intelligent systems in areas including traffic congestion minimization structural health monitoring crash-avoiding cars , and smart grids. While the aforementioned fields appear vastly different to healthcare, the research conducted within them verifies the plausibility of an IoT-based healthcare system. Existing systems in other fields have proven that remote monitoring of objects, with data collection and reporting are achievable. This can therefore be expanded and adapted for monitoring the health of people and reporting it to relevant parties such as caretakers, doctors, emergency services, and healthcare centers.

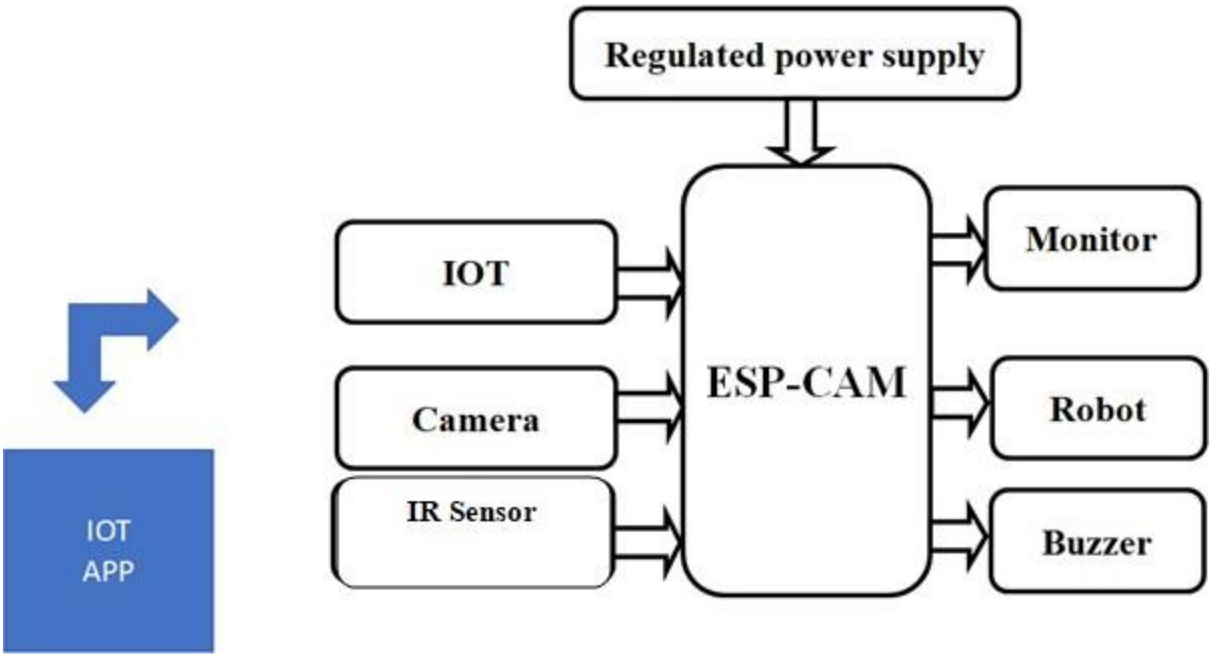
In today's rapid growing generation, the development in the field of new techniques has brought a vast and massive change in the field of mechanics, automation and advancement in all the sectors of our day to day life either related to family or social welfare. In all aspects we are experiencing now-a-day some or the other way kind of changes. All over the world showoffs due to sharply –witted mobiles have brought a drastic revolution in one's living standard and various other aspects of life. One such example is based on android applications which provides us complete open environment to do anything we are pleased with, related to any field we are that we are interested in guiding us in our daily life. The primary objective behind creation of this robot keeping a alert watch especially in war field when something fishy is caught at border side due to some suspected act of enemy or any if any unnatural things is felt to happen. This is done so, in order to avoid loss of human life as the military personnel have great danger of losing their life if they are found to spying any suspected area. So, to avoid it this robot will be useful to use in such cases. This robot vehicle will serve as an suitable material not only in aspect of providing border security but moreover can be utilized for different characteristic adversity and this machine for the defense segment will reduce loss of human life too. It may guide all the military personnel and make them prepared for any misfortune if going to occur within their shelter region. Different Finder can be utilized that can be embedded on mechanical vehicle like metallic finder sensor is utilized to distinguish metallic objects . Fire finder is utilized to distinguish correct heading of fire source. This robot is valuable at places where one cannot reach like mystery spots or little areas. The foremost centre of this sort of model is to supply one extraordinary security degree. The great advancement that we come across in in designing this robot is the use of Wi-Fi. We can use here IOT module also instead of Wi-Fi but IOT have a short range of connection to make the robot work more efficiently as compared to Wi-Fi based system. Wi-Fi technique is useful in case if we are very far from the gadget also but our connection and Wi-Fi network is good then it works more significantly. The Node MCU ESP8266 used here acts as a link between the camera and the motor driver module fixed on the robot. It consists of motor driver module acts as a controller to contol the motion of the robot for working of the wheels of the robot fixed in it. The motor module used is named as L293D and a connector is provided between Node MCU module and motor driver module. That connector will be utilized for supplying external power supply. Four wheels are which operates on DC Motor is used for the motion of the robot. The camera used here can rotate whole 360 degree to record each and everything at every side wherever we wish to figure out the situation at the place where it is used for spying purpose

1.3EXISTING SYSTEM

In the present existing system there is manual things of operations going due to that no faster the applications and cost effective. To make automation we re introducing the robot. This proposed robot will be controlled via manually. We can control the movement of the robot by sending instructions via IOT app from our android phone. a multipurpose Robotic vehicle moves Forward, Left, Right, Backward and Stop directions with night vision spy camera. The existing system has used the 8051 micro controller and Arduino board in order to design the robot. Here we use 8051 series micro controller (AT89C52).

1.4PROPOSED SYSTEM

In this proposed system, Raspberry pi is installed with the night vision camera which help the system to go for the automation and help to find the human or any problem detected using the sound sensor and according to the sound produced it automatically goes to that area. It uses a predefined line to follow its path while patrolling. It stops at particular points and moves to next points if sound are detected. The system uses IR based path following system for patrolling assigned area. It monitors each area to detect any intrusion using 360degree rotating HD camera. It has the ability to monitor sound in the premises. Any sound after company is closed and it starts moving towards the sound on its predefined path. It then scans the area using its camera to detect any human faces detected.



The Attention commands are transferred to the electronic devices. In reverse, the electronic device transfers the stored messages from the wireless module. The micro controller checks the IOT command and after validating the command it performs further certain task on the robot or device. The micro controller used here in this project is ATMEGA 328 incorporated in an Arduino UNO board.

The ongoing revolution of Internet, together With the growing robotics in many activities of everyday life. In this method we use embedded C language for coding and debugging in arduino by using ISP programmer. Arduino-IDE tool. And for controlling we use L293D driver IC. Finding this robot is a reprogrammable, multifunctional manipulator designed to move or pick and place the materials, Parts and tools. These robots are not only used for lifting purpose but also for polishing, sealing, Machine handling and minor surgeries also. The automatic mode robot is programmed within the embedded C Programming and it makes the robot to act as human beings. This version of robot is mainly defined by the factor named Artificial Intelligence. The fig block diagram gives an idea of how the robot works. It shows how the system circuit works and how the current flow goes through it. The wireless communication used is IOT which helps in transferring the data and messages.

CHAPTER 2

EMBEDDED SYSTEMS

2.1 Embedded Systems:

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own.)

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded". A modern example of embedded system is shown in fig: 2.1.

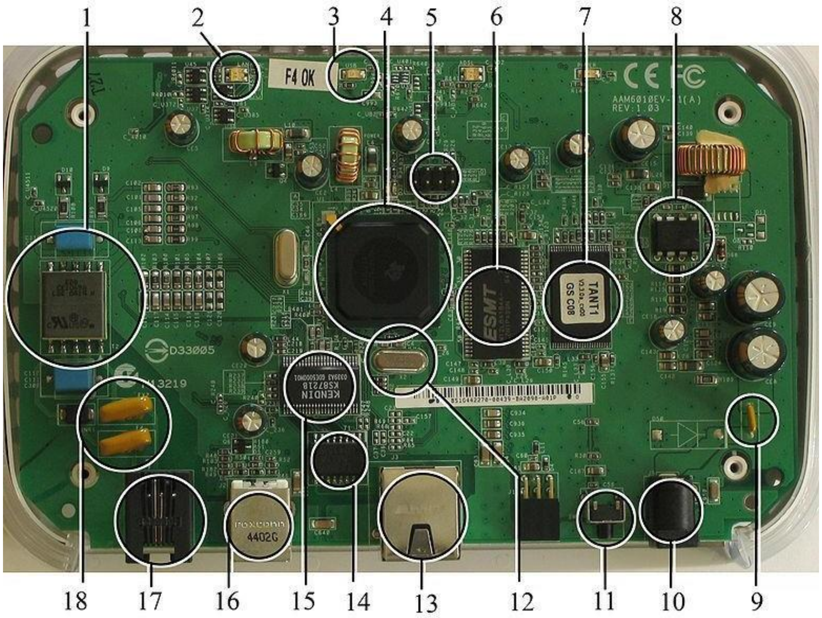


Fig 2.1:A modern example of embedded system

Labeled parts include microprocessor (4), RAM (6), flash memory (7).Embedded systems programming is not like normal PC programming. In many ways, programming for an embedded system is like programming PC 15 years ago. The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

2.1.1 History:

In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology.

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961. It was built from transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits.

2.1.2 Tools:

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort.

Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint when your system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

2.1.3 Resources:

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start, and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well.

Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether.

2.1.4 Real Time Issues:

Embedded systems frequently control hardware, and must be able to respond to them in real time. Failure to do so could cause inaccuracy in measurements, or even damage

hardware such as motors. This is made even more difficult by the lack of resources available. Almost all embedded systems need to be able to prioritize some tasks over others, and to be able to put off/skip low priority tasks such as UI in favor of high priority tasks like hardware control.

2.2 Need For Embedded Systems:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

2.2.1 Debugging:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.
- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.

- Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

  Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software(and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a logic analyzer, for instance.

2.2.2 Reliability:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by themselves if an error occurs. Therefore the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

- The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
- The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.
- The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

  A variety of techniques are used, sometimes in combination, to recover from errors—both software bugs such as memory leaks, and also soft errors in the hardware:

- Watchdog timer that resets the computer unless the software periodically notifies the watchdog
- Subsystems with redundant spares that can be switched over to
- software "limp modes" that provide partial function

- Designing with a Trusted Computing Base (TCB) architecture[6] ensures a

  highly secure & reliable system environment
- An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
- Immunity Aware Programming

2.3 Explanation of Embedded Systems:

2.3.1 Software Architecture:

There are several different types of software architecture in common use.

- Simple Control Loop:

In this design, the software simply has a loop. The loop calls subroutines, each which manages a part of the hardware or software.

- Interrupt Controlled System:

Some embedded systems are predominantly interrupting controlled. This means that tasks performed by the system are triggered by different kinds of events. An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte. These kinds of systems are used if event handlers need low latency and the event handlers are short and simple.

Usually these kinds of systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue
structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

- Cooperative Multitasking:

A non-preemptive multitasking system is very similar to the simple control loop scheme, except that the loop is hidden in an API. The programmer defines a series of tasks, and each task gets its own environment to “run” in. When a task is idle, it calls an idle routine, usually called “pause”, “wait”, “yield”, “nop” (stands for no operation), etc.The advantages and disadvantages are very similar to the control loop, except that adding new software is easier, by simply writing a new task, or adding to the queue-interpreter.

- Primitive Multitasking:

In this type of system, a low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt). This is the level at which the system is generally considered to have an "operating system" kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel.

As any code can potentially damage the data of another task (except in larger systems using an MMU) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronization strategy, such as message queues, semaphores or a non-blocking synchronization scheme.

Because of these complexities, it is common for organizations to buy a real-time operating system, allowing the application programmers to concentrate on device functionality rather than operating system services, at least for large systems; smaller systems often cannot afford the overhead associated with a generic real time system, due to limitations regarding memory size, performance, and/or battery life.

- Microkernels And Exokernels:

A microkernel is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc.

In general, microkernels succeed when the task switching and intertask communication is fast, and fail when they are slow. Exokernels communicate efficiently by normal subroutine calls. The hardware and all the software in the system are available to, and extensible by application programmers. Based on performance, functionality, requirement the embedded systems are divided into three categories:

2.3.2 Stand Alone Embedded System:

These systems takes the input in the form of electrical signals from transducers or commands from human beings such as pressing of a button etc.., process them and produces desired output. This entire process of taking input, processing it and giving output is done in standalone mode. Such embedded systems comes under stand alone embedded systems

Eg: microwave oven, air conditioner etc..

2.3.3 Real-time embedded systems:

Embedded systems which are used to perform a specific task or operation in a specific time period those systems are called as real-time embedded systems. There are two types of real-time embedded systems.

- Hard Real-time embedded systems:

These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment.

Eg: consider a system in which we have to open a valve within 30 milliseconds. If this valve is not opened in 30 ms this may cause damage to the entire equipment. So in such cases we use embedded systems for doing automatic operations.

- Soft Real Time embedded systems:

These embedded systems follow a relative dead line time period i.e.., if the task is not done in a particular time that will not cause damage to the equipment.

Eg: Consider a TV remote control system , if the remote control takes a few milliseconds delay it will not cause damage either to the TV or to the remote control. These systems which will not cause damage when they are not operated at considerable time period those systems comes under soft real-time embedded systems.

2.3.4 Network communication embedded systems:

A wide range network interfacing communication is provided by using embedded systems.

Eg:

- Consider a web camera that is connected to the computer with internet can be used to spread communication like sending pictures, images, videos etc.., to another computer with internet connection throughout anywhere in the world.

- Consider a web camera that is connected at the door lock.

Whenever a person comes near the door, it captures the image of a person and sends to the desktop of your computer which is connected to internet. This gives an alerting message with image on to the desktop of your computer, and then you can open the door lock just by clicking the mouse. Fig: 2.2 show the network communications in embedded systems.

Fig 2.2: Network communication embedded systems

2.3.5 Different types of processing units:

The central processing unit (c.p.u) can be any one of the following microprocessor, microcontroller, digital signal processing

- Among these Microcontroller is of low cost processor and one of the main advantage of microcontrollers is, the components such as memory, serial communication interfaces, analog to digital converters etc.., all these are built on a single chip. The numbers of external components that are connected to it are very less according to the application.

- Microprocessors are more powerful than microcontrollers. They are used in major applications with a number of tasking requirements. But the microprocessor requires many external components like memory, serial communication, hard disk, input output ports etc.., so the power consumption is also very high when compared to microcontrollers.

- Digital signal processing is used mainly for the applications that particularly involved with processing of signals

2.4 APPLICATIONS OF EMBEDDED SYSTEMS:

2.4.1 Consumer applications:

At home we use a number of embedded systems which include microwave oven, remote control, vcd players, dvd players, camera etc….

Fig2.3: Automatic coffee makes equipment

2.4.2 Office automation:

We use systems like fax machine, modem, printer etc…

Fig2.4: Fax machine **Fig2.5: Printing machine**

2.4.3. Industrial automation:

Today a lot of industries are using embedded systems for process control. In industries we design the embedded systems to perform a specific operation like monitoring temperature, pressure, humidity ,voltage, current etc.., and basing on these monitored levels we do control other devices, we can send information to a centralized monitoring station.

Fig2.6: Robot

In critical industries where human presence is avoided there we can use robot**s** which are programmed to do a specific operation.

2.4.5 Computer networking:

Embedded systems are used as bridges routers etc..

Fig2.7: Computer networking

2.4.6 Tele communications:

Cell phones, web cameras etc.

CHAPTER 3

HARDWARE DESCRIPTION

2.1 Arduino:

Figure 2.1 ATMEGA 328 Microcontrollers

2.2 Pin Diagram:

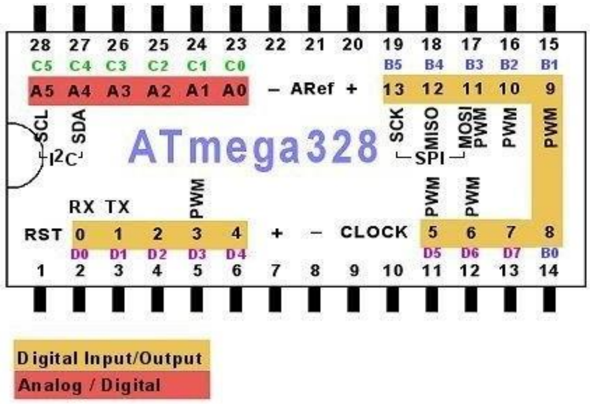


Figure 2.2 ATMEGA 328 PIN diagram

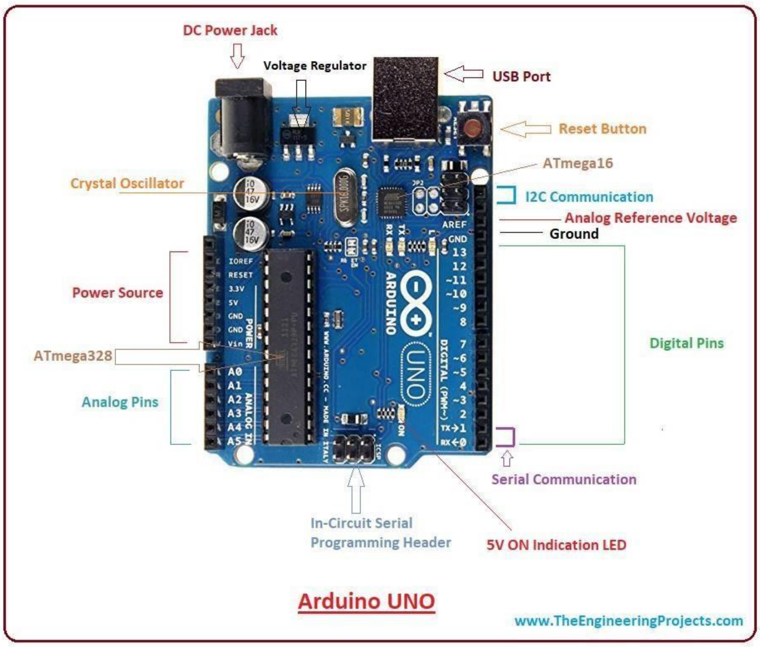


Figure 2.3 ARDUINO Development Board

VCC:
Digital supply voltage magnitude of the voltage range between 4.5 to 5.5 V for the ATmega8 and 2.7 to 5.5 V for ATmega8L

GND:
Ground Zero reference digital voltage supply.

PORTB (PB7.. PB0) :
PORTB is a port I / O two-way (bidirectional) 8-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTB pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

PORTC (PC5.. PC0) :
PORTC is a port I / O two-way (bidirectional) 7-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTC pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

PC6/RESET:
If RSTDISBL Fuse programmed, PC6 then serves as a pin I / O but with different characteristics. PC0 to PC5 If Fuse RSTDISBL not programmed, then serves as input Reset PC6. LOW signal on this pin with a minimum width of 1.5 microseconds will bring the microcontroller into reset condition, although the clock is not running.

PORTD (PD7.. PD0) :
PORTD is a port I / O two-way (bidirectional) 8-bit with internal pull-up resistor can be selected. This port output buffers have symmetrical characteristics when used as a source or sink. When used as an input, the pull-pin low externally will emit a current if the pull-up resistor is activated it. PORTD pins will be in the condition of the tri-state when RESET is active, although the clock is not running.

2.3 Architecture:

Memory: It has 8 Kb of Flash program memory (10,000 Write/Erase cycles durability), 512 Bytes of EEPROM (100,000 Write/Erase Cycles). 1Kbyte Internal SRAM

I/O Ports: 23 I/ line can be obtained from three ports; namely Port B, Port C and Port D.

Interrupts: Two External Interrupt source, located at port D. 19 different interrupt vectors supporting 19 events generated by internal peripherals.

Timer/Counter: Three Internal Timers are available, two 8 bit, one 16 bit, offering various operating modes and supporting internal or external clocking.

SPI (Serial Peripheral interface): ATmega8 holds three communication devices integrated. One of them is Serial Peripheral Interface. Four pins are assigned to Atmega8 to implement this scheme of communication.

USART: One of the most powerful communication solutions is USART and ATmega8 supports both synchronous and asynchronous data transfer schemes. It has three pins assigned for that. In many projects, this module is extensively used for PC-Micro controller communication.

TWI (Two Wire Interface): Another communication device that is present in ATmega8 is Two Wire Interface. It allows designers to set up a commutation between two devices using just two wires along with a common ground connection, As the TWI output is made by means of open collector outputs, thus external pull up resistors are required to make the circuit.

Analog Comparator: A comparator module is integrated in the IC that provides comparison facility between two voltages connected to the two inputs of the Analog comparator via External pins attached to the micro controller.

Analog to Digital Converter: Inbuilt analog to digital converter can convert an analog input signal into digital data of 10bit resolution. For most of the low end application, this much resolution is enough.

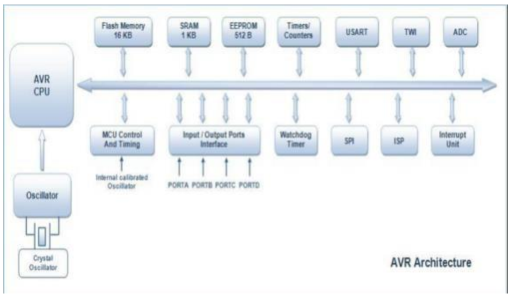


Figure 2.4 AVR Architecture

Features:

• High-performance, Low-power Atmel®AVR® 8-bit Microcontroller

• Advanced RISC Architecture

– 130 Powerful Instructions – Most Single-clock Cycle Execution

– 32 × 8 General Purpose Working Registers

– Fully Static Operation

– Up to 16MIPS Throughput at 16MHz

– On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory segments

– 8Kbytes of In-System Self-programmable Flash program memory

– 1Kbyte Internal SRAM
– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
– Data retention: 20 years at 85°C/100 years at 25°C(1)
– Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
– Programming Lock for Software Security
• Peripheral Features
– Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
– Real Time Counter with Separate Oscillator
– Three PWM Channels
– 8-channel ADC in TQFP and QFN/MLF package
Eight Channels 10-bit Accuracy
– 6-channel ADC in PDIP package
Six Channels 10-bit Accuracy
– Byte-oriented Two-wire Serial Interface
– Programmable Serial USART
– Master/Slave SPI Serial Interface
– Programmable Watchdog Timer with Separate On-chip Oscillator
– On-chip Analog Comparator
• Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated RC Oscillator
– External and Internal Interrupt Sources
– Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

• I/O and Packages
– 23 Programmable I/O Lines
– 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
• Operating Voltages
– 2.7V - 5.5V (ATmega8L)
– 4.5V - 5.5V (ATmega8)
• Speed Grades
– 0 - 8MHz (ATmega8L)
– 0 - 16MHz (ATmega8)
• Power Consumption at 4Mhz, 3V, 25oC
– Active: 3.6mA
– Idle Mode: 1.0mA
– Power-down Mode: 0.5µA

Power Supply:

A regulated power supply converts unregulated AC (Alternating Current) to a constant DC (Direct Current). A regulated power supply is used to ensure that the output remains constant even if the input changes. A regulated DC power supply is also known as a linear power supply; it is an embedded circuit and consists of various blocks. The regulated power supply will accept an AC input and give a constant DC output. The figure below shows the block diagram of a typical regulated DC power supply.

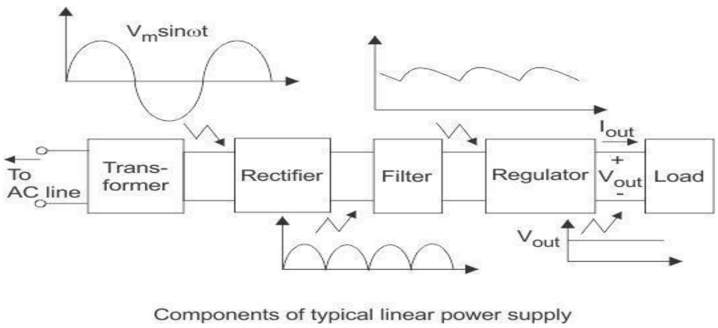


Figure 2.8 Block diagram of power supply

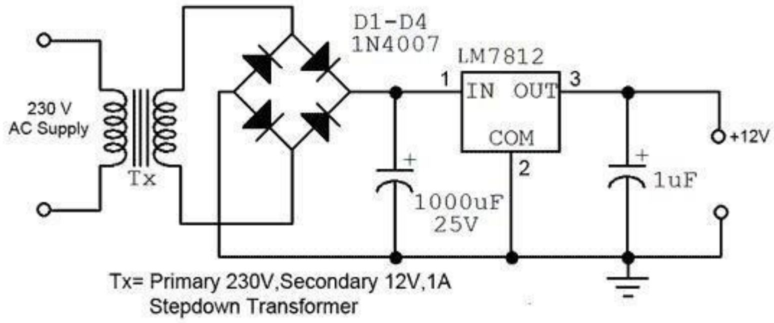


Figure 2.9 Power supply circuit diagram

ROBOT DC MOTOR

A dc motor uses electrical energy to produce mechanical energy, very typically through the interaction of magnetic fields and current-carrying conductors. The reverse process, producing electrical energy from mechanical energy, is accomplished by an alternator, generator or dynamo. Many types of electric motors can be run as generators, and vice versa. The input of a DC motor is current/voltage and its output is torque (speed).

Fig 3.9.1: DC Motor

The DC motor has two basic parts: the rotating part that is called the armature and the stationary part that includes coils of wire called the field coils. The stationary part is also called the stator. Figure shows a picture of a typical DC motor, Figure shows a picture of a DC armature, and Fig shows a picture of a typical stator. From the picture you can see the armature is made of coils of wire wrapped around the core, and the core has an extended shaft that rotates on bearings. You should also notice that the ends of each coil of wire on the armature are terminated at one end of the armature.

The termination points are called the commutator, and this is where the brushes make electrical contact to bring electrical current from the stationary part to the rotating part of the machine

Operation:

The DC motor you will find in modem industrial applications operates very similarly to the simple DC motor described earlier in this chapter. Figure 12-9 shows an electrical diagram of a simple DC motor. Notice that the DC voltage is applied directly to the field winding and the brushes. The armature and the field are both shown as a coil of wire. In later diagrams, a field resistor will be added in series with the field to control the motor speed.

When voltage is applied to the motor, current begins to flow through the field coil from the negative terminal to the positive terminal. This sets up a strong magnetic field in the field winding. Current also begins to flow through the brushes into a commutator segment and then through an armature coil. The current continues to flow through the coil back to the brush that is attached to other end of the coil and returns to the DC power source. The current flowing in the armature coil sets up a strong magnetic field in the armature.

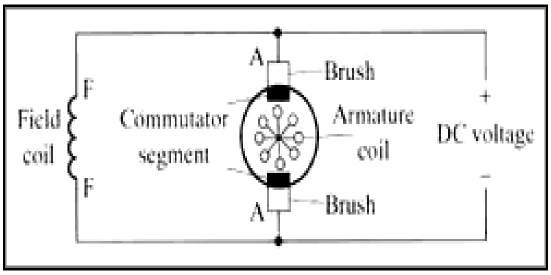


Fig 3.9.2: Simple electrical diagram of DC motor

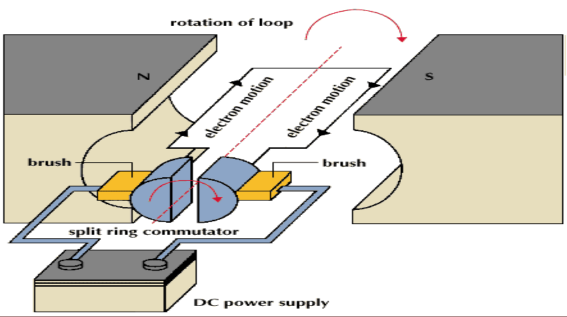


Fig 3.9.3: Operation of a DC Motor

The magnetic field in the armature and field coil causes the armature to begin to rotate. This occurs by the unlike magnetic poles attracting each other and the like magnetic poles repelling each other. As the armature begins to rotate, the commutator segments will also begin to move under the brushes. As an individual commutator segment moves under the brush connected to positive voltage, it will become positive, and when it moves under a brush connected to negative voltage it will become negative. In this way, the commutator segments continually change polarity from positive to negative. Since the commutator segments are connected to the ends of the wires that make up the field winding in the armature, it causes the magnetic field in the armature to change polarity continually from North Pole to South Pole. The commutator segments and brushes are aligned in such a way that the switch in polarity of the armature coincides with the location of the armature's magnetic field and the field winding's magnetic field. The switching action is timed so that the armature will not lock up magnetically with the field. Instead the magnetic fields tend to build on each other and provide additional torque to keep the motor shaft rotating.

When the voltage is de-energized to the motor, the magnetic fields in the armature and the field winding will quickly diminish and the armature shaft's speed will begin to drop to zero. If voltage is applied to the motor again, the magnetic fields will strengthen and the armature will begin to rotate again.

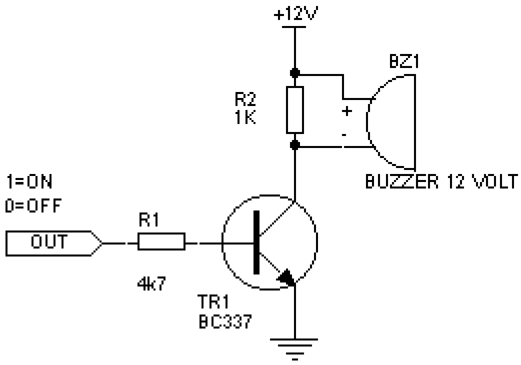
3.8 Buzzer

Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm. A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate (brass or stainless steel, etc.). A piezoelectric ceramic plate is attached to a metal plate with adhesives. Applying D.C. voltage between electrodes of a piezoelectric diaphragm causes mechanical distortion due to the piezoelectric effect. For a misshaped piezoelectric element, the distortion of the piezoelectric element expands in a radial direction. And the piezoelectric diaphragm bends toward the direction. The metal plate bonded to the piezoelectric element does not expand. Conversely, when the piezoelectric element shrinks, the piezoelectric diaphragm bends in the direction Thus, when AC voltage is applied across electrodes, the bending is repeated, producing sound waves in the air.

To interface a buzzer the standard transistor interfacing circuit is used. Note that if a different power supply is used for the buzzer, the 0V rails of each power supply must be connected to provide a common reference.

If a battery is used as the power supply, it is worth remembering that piezo sounders
draw much less current than buzzers. Buzzers also just have one ‘tone’, whereas a piezo sounder is able to create sounds of many different tones.

To switch on buzzer -high 1
To switch off buzzer -low 1
Notice (Handling) In Using Self Drive Method
1) When the piezoelectric buzzer is set to produce intermittent sounds, sound may be heard continuously even when the self drive circuit is turned ON / OFF at the "X" point shown in Fig. 9. This is because of the failure of turning off the feedback voltage.
2) Build a circuit of the piezoelectric sounder exactly as per the recommended circuit shown in the catalog of the transistor and circuit constants are designed to ensure stable oscillation of the piezoelectric sounder.
3) Design switching which ensures direct power switching.
4) The self drive circuit is already contained in the piezoelectric buzzer. So there is no need to prepare another circuit to drive the piezoelectric buzzer.
5) Rated voltage (3.0 to 20Vdc) must be maintained. Products which can operate with voltage higher than 20Vdc are also available.
6) Do not place resistors in series with the power source, as this may cause abnormal oscillation. If a resistor is essential to adjust sound pressure, place a capacitor (about 1μF) in parallel with the piezo buzzer.
7) Do not close the sound emitting hole on the front side of casing.
8) Carefully install the piezo buzzer so that no obstacle is placed within 15mm from the sound release hole on the front side of the casing.



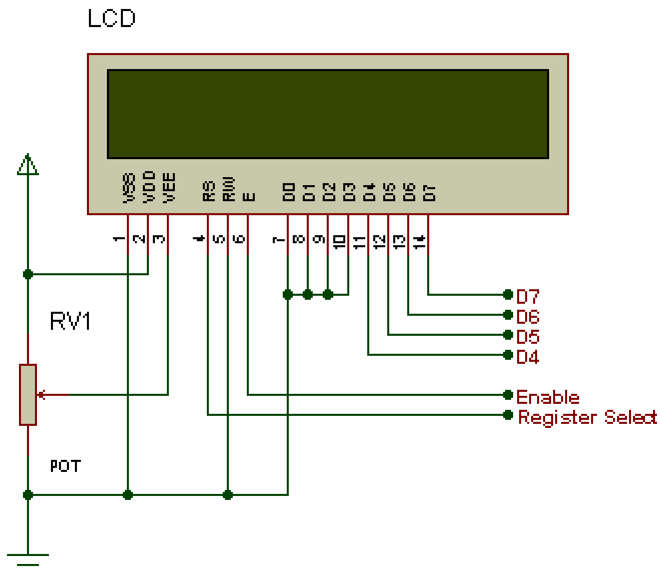
3.7 LCD DISPLAY

LCD Background:

One of the most common devices attached to a micro controller is an LCD display. Some of the most common LCD’s connected to the many microcontrollers are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Basic 16x 2 Characters LCD

Figure 1: LCD Pin diagram



Pin description:

Pin No.	**Name**	**Description**
Pin no. 1	**VSS**	Power supply (GND)
Pin no. 2	**VCC**	Power supply (+5V)
Pin no. 3	**VEE**	Contrast adjust
Pin no. 4	**RS**	0 = Instruction input 1 = Data input
Pin no. 5	**R/W**	0 = Write to LCD module 1 = Read from LCD module
Pin no. 6	**EN**	Enable signal
Pin no. 7	**D0**	Data bus line 0 (LSB)
Pin no. 8	**D1**	Data bus line 1
Pin no. 9	**D2**	Data bus line 2
Pin no. 10	**D3**	Data bus line 3
Pin no. 11	**D4**	Data bus line 4
Pin no. 12	**D5**	Data bus line 5
Pin no. 13	**D6**	Data bus line 6
Pin no. 14	**D7**	Data bus line 7 (MSB)

Table 1: Character LCD pins with Microcontroller

The LCD requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

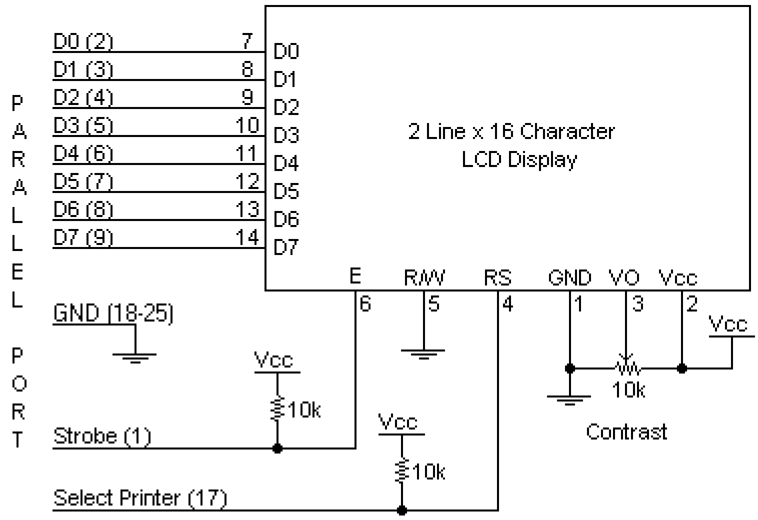
The **EN** line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Schematic:



Circuit Description:

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

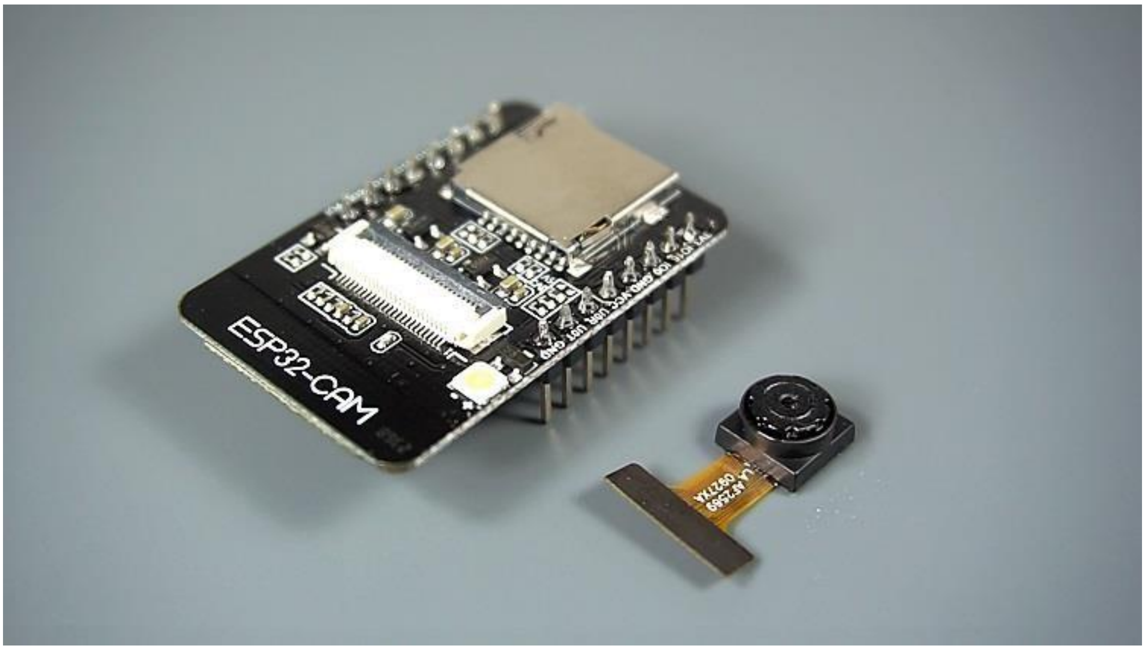
The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. We can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if we have trouble with the circuit working properly.

ESP32-CAMERA

ESP32-CAM is the latest small size camera module released by Essence. The module can work independently as the smallest system, with a size of only 27*40.5*4.5mm, and a deep sleep current as low as 6mA.

ESP32-CAM can be widely used in various IoT applications, suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications .

ESP32-CAM adopts DIP package and can be used directly by plugging in the bottom plate, realizing the rapid production of products, providing customers with high-reliability connection methods, which is convenient for application in various IoT hardware terminal occasions.



The ESP32-CAM is a very small camera module with the ESP32-S chip that costs approximately Besides the OV2640 camera, and several GPIOs to connect peripherals, it also features a microSD card slot that can be useful to store images taken with the camera or to store files to serve to clients.

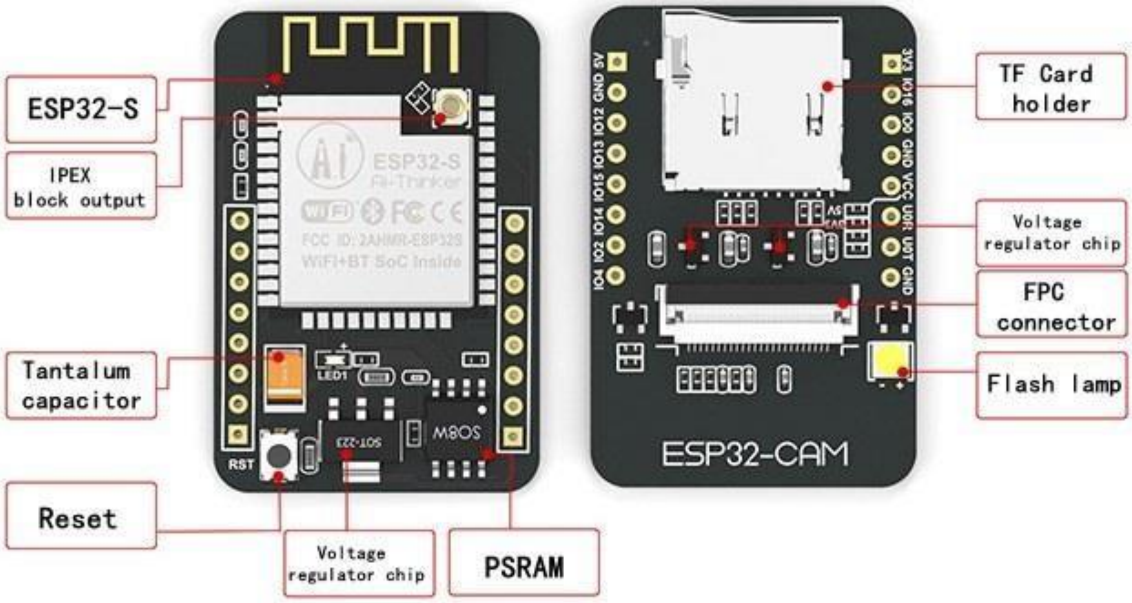
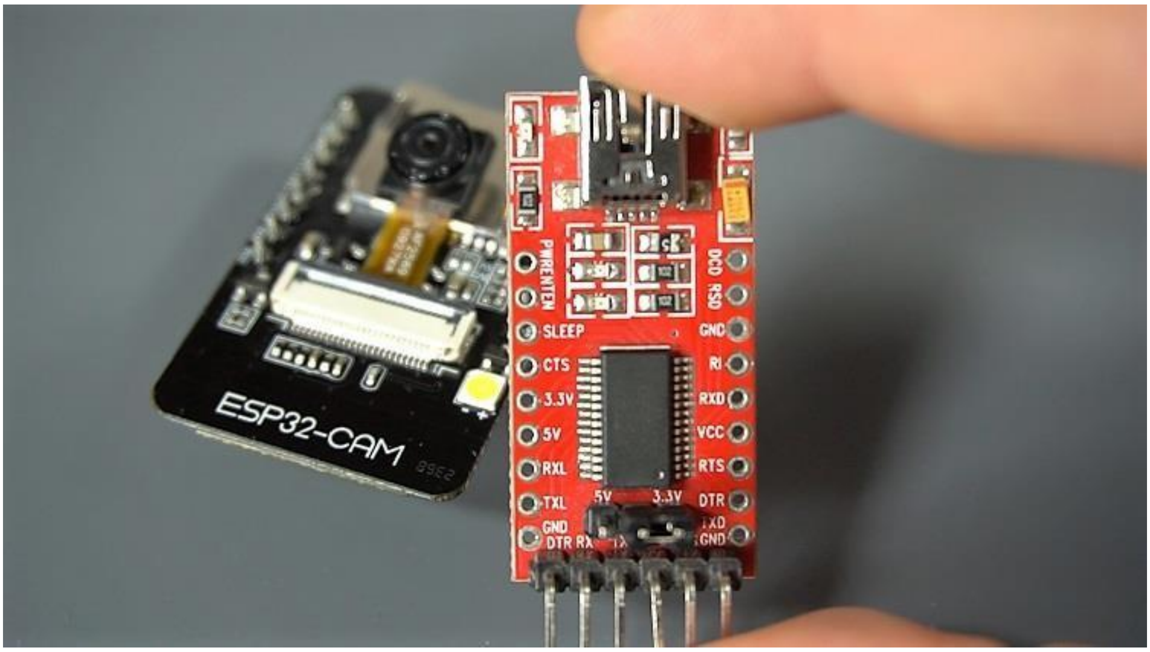


Image source – Seeed Studio

The ESP32-CAM doesn't come with a USB connector, so you need an FTDI programmer to upload code through the U0R and U0T pins (serial pins).



Features

Here is a list with the ESP32-CAM features:

- The smallest 802.11b/g/n Wi-Fi BT SoC module
- Low power 32-bit CPU,can also serve the application processor
- Up to 160MHz clock speed, summary computing power up to 600 DMIPS

- Built-in 520 KB SRAM, external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, built-in flash lamp
- Support image WiFI upload
- Support TF card
- Supports multiple sleep modes
- Embedded Lwip and FreeRTOS
- Supports STA/AP/STA+AP operation mode
- Support Smart Config/AirKiss technology
- Support for serial port local and remote firmware upgrades (FOTA)

ESP32-CAM Pinout

The following figure shows the ESP32-CAM pinout (AI-Thinker module).

Image source – Seeed Studio

There are three GND pins and two pins for power: either 3.3V or 5V.

GPIO 1 and GPIO 3 are the serial pins. You need these pins to upload code to your board. Additionally, GPIO 0 also plays an important role, since it determines whether the ESP32 is in flashing mode or not. When GPIO 0 is connected to GND, the ESP32 is in flashing mode.

The following pins are internally connected to the microSD card reader:

- GPIO 14: CLK
- GPIO 15: CMD
- GPIO 2: Data 0
- GPIO 4: Data 1 (also connected to the on-board LED)
- GPIO 12: Data 2

- GPIO 13: Data 3

Product Features

Using low-power dual-core 32-bit CPU, can be used as an application processor

Main frequency up to 240MHz, computing power up to 600 DMIPS

Built-in 520 KB SRAM, external 8MB PSRAM

Support UART/SPI/I2C/PWM/ADC/DAC and other interfaces

Support OV2640 and OV7670 cameras, built-in flash

Support picture WiFI upload

Support TF card

Support multiple sleep modes.

Embedded Lwip and FreeRTOS

Support STA/AP/STA+AP working mode

Support Smart Config/AirKiss one-click network configuration

Support secondary development

Application scenarios

Home smart device image transmission

Wireless monitoring

Smart agriculture

QR wireless recognition

3.5. LCD MONITOR

A **computer monitor** is an output device that displays information in pictorial form. A monitor usually comprises the visual display, circuitry, casing, and power supply. The display device in modern monitors is typically a thin film transistor liquid crystal display (TFT-LCD) with LED backlighting having replaced cold-cathode fluorescent lamp (CCFL) backlighting. Older monitors used a cathode ray tube (CRT). Monitors are connected to the computer via VGA, Digital Visual Interface (DVI), HDMI, DisplayPort, Thunderbolt, low-voltage differential signaling (LVDS) or other proprietary connectors and signals. Originally, computer monitors were used for data processing while television sets were used for entertainment. From the 1980s onwards, computers (and their monitors) have been used for both data processing and entertainment, while televisions have implemented some computer functionality. The common aspect ratio of televisions, and computer monitors, has changed from 4:3 to 16:10, to 16:9. Modern computer monitors are easily interchangeable with conventional television sets. However, as computer monitors do not necessarily include integrated speakers, it may not be possible to use a computer monitor without external components

The estimation of the monitor size by the distance between opposite corners does not take into account the display aspect ratio, so that for example a 16:9 21-inch (53 cm) widescreen display has less area, than a 21-inch (53 cm) 4:3 screen. The 4:3 screen has dimensions of 16.8 in × 12.6 in (43 cm × 32 cm) and area 211 sq in (1,360 cm^2), while the widescreen is 18.3 in × 10.3 in (46 cm × 26 cm)

The resolution for computer monitors has increased over time. From 320x200 during the early 1980s, to 1024x768 during the late 1990s. Since 2009, the most commonly sold resolution for computer monitors is 1920x1080.[12] Before 2013 top-end consumer LCD monitors were limited to 2560x1600 at 30 in (76 cm), excluding Apple products and CRT monitors. Apple introduced 2880x1800 with Retina MacBook Pro at 15.4 in (39 cm) on June 12, 2012, and introduced a 5120x2880 Retina iMac at 27 in (69 cm) on October 16, 2014. By 2015 most major display manufacturers had released 3840x2160 resolution displays

CHAPTER 4

SOFTWARE

DESCRIPTION

This project is implemented using following software's:

- Express PCB – for designing circuit
- Arduino IDE compiler - for compilation part
- Proteus 7 (Embedded C) – for simulation part

4.1 Express PCB:

Breadboards are great for prototyping equipment as it allows great flexibility to modify a design when needed; however the final product of a project, ideally should have a neat PCB, few cables, and survive a shake test. Not only is a proper PCB neater but it is also more durable as there are no cables which can yank loose.

Express PCB is a software tool to design PCBs specifically for manufacture by the company Express PCB (no other PCB maker accepts Express PCB files). It is very easy to use, but it does have several limitations.

It can be likened to more of a toy then a professional CAD program.

It has a poor part library (which we can work around)

It cannot import or export files in different formats

It cannot be used to make prepare boards for DIY production

Express PCB has been used to design many PCBs (some layered and with surface-mount parts. Print out PCB patterns and use the toner transfer method with an Etch Resistant Pen to make boards. However, Express PCB does not have a nice print layout. Here is the procedure to design in Express PCB and clean up the patterns so they print nicely.

4.1.1 Preparing Express PCB for First Use:

`Express PCB comes with a less then exciting list of parts. So before any project is started head over to Audio logical and grab the additional parts by morsel, ppl, and tangent, and extract them into your Express PCB directory. At this point start the program and get ready to setup the workspace to suit your style.

Click View -> Options. In this menu, setup the units for “mm” or “in” depending on how you think, and click “see through the top copper layer” at the bottom. The standard color scheme of red and green is generally used but it is not as pleasing as red and blue.

4.1.2 The Interface:

When a project is first started you will be greeted with a yellow outline. This yellow outline is the dimension of the PCB. Typically after positioning of parts and traces, move them to their final position and then crop the PCB to the correct size. However, in designing a board with a certain size constraint, crop the PCB to the correct size before starting.

Fig: 4.1 show the toolbar in which the each button has the following functions:

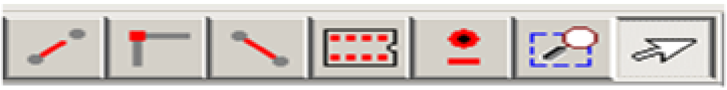


Fig 4.1: Tool bar necessary for the interface

- The select tool: It is fairly obvious what this does. It allows you to move and manipulate parts. When this tool is selected the top toolbar will show buttons to move traces to the top / bottom copper layer, and rotate buttons.
- The zoom to selection tool: does just that.
- The place pad: button allows you to place small soldier pads which are useful for board connections or if a part is not in the part library but the part dimensions are available. When this tool is selected the top toolbar will give you a large selection of round holes, square holes and surface mount pads.
- The place component: tool allows you to select a component from the top toolbar and then by clicking in the workspace places that component in the orientation chosen using the buttons next to the component list. The components can always be rotated afterwards with the select tool if the orientation is wrong.

- The place trace: tool allows you to place a solid trace on the board of varying thicknesses. The top toolbar allows you to select the top or bottom layer to place the trace on.
- The Insert Corner in trace: button does exactly what it says. When this tool is selected, clicking on a trace will insert a corner which can be moved to route around components and other traces.
- The remove a trace button is not very important since the delete key will achieve the same result.

4.1.3 Design Considerations:

Before starting a project there are several ways to design a PCB and one must be chosen to suit the project's needs.

Single sided, or double sided?

When making a PCB you have the option of making a single sided board, or a double sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board without jumpering over traces with a cable. While there's technically nothing wrong with this, it should be avoided if the signal travelling over the traces is sensitive (e.g. audio signals).

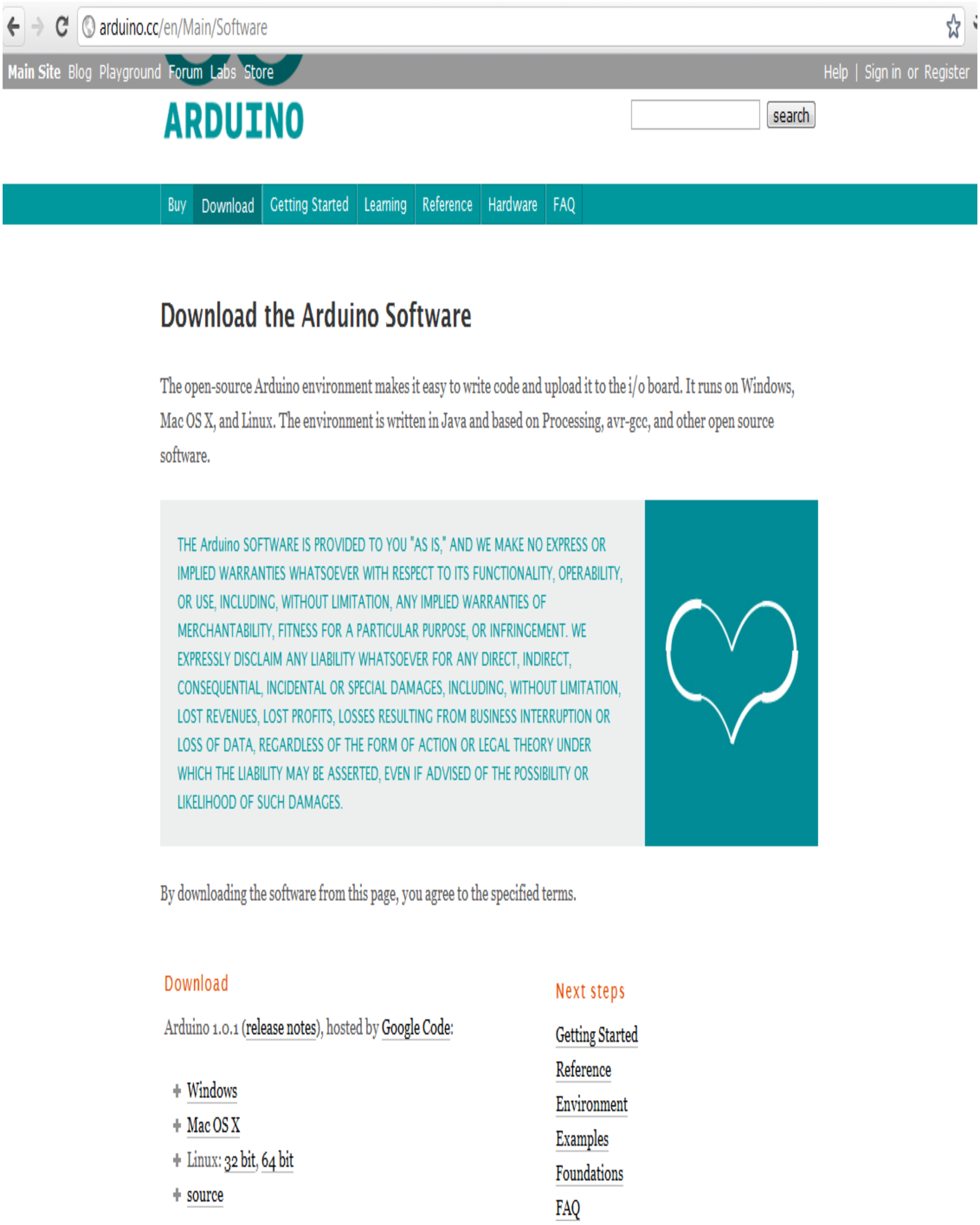
A double sided board is more expensive to produce professionally, more difficult to etch on a DIY board, but makes the layout of components a lot smaller and easier. It should be noted that if a trace is running on the top layer, check with the components to make sure you can get to its pins with a soldering iron. Large capacitors, relays, and similar parts which don't have axial leads can NOT have traces on top unless boards are plated professionally.

Ground-plane or other special purposes for one side

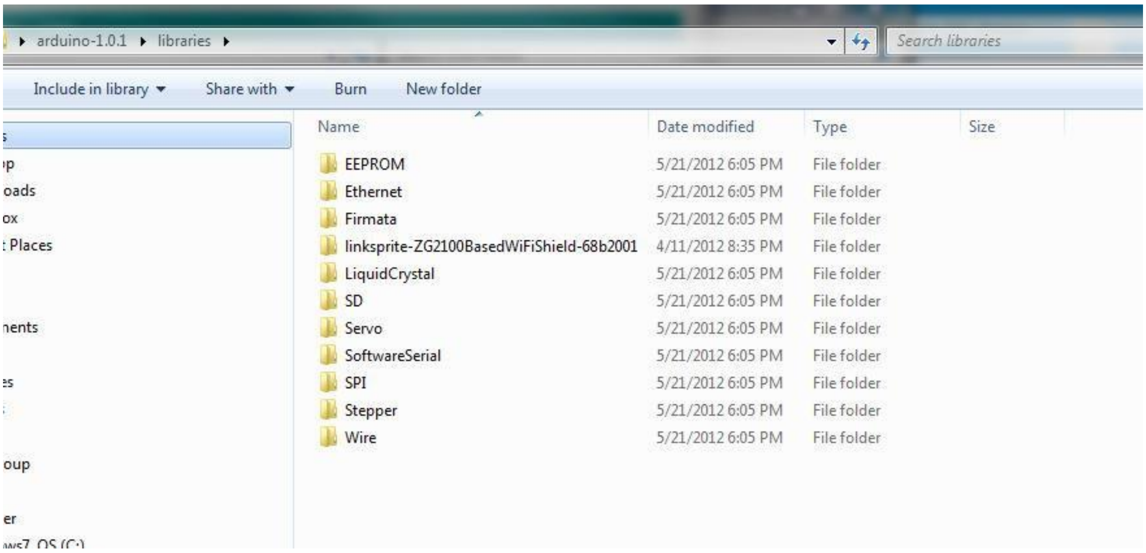
When using a double sided board you must consider which traces should be on what side of the board. Generally, put power traces on the top of the board, jumping only to the bottom if a part cannot be soldiered onto the top plane (like a relay), and vice- versa.

Some projects like power supplies or amps can benefit from having a solid plane to use for ground. In power supplies this can reduce noise, and in amps it minimizes the distance between parts and their ground connections, and keeps the ground signal as simple as possible. However, care must be taken with stubborn chips such as the TPA6120 amplifier from TI. The TPA6120 datasheet specifies not to run a ground plane under the pins or signal traces of this chip as the capacitance generated could effect performance negatively.

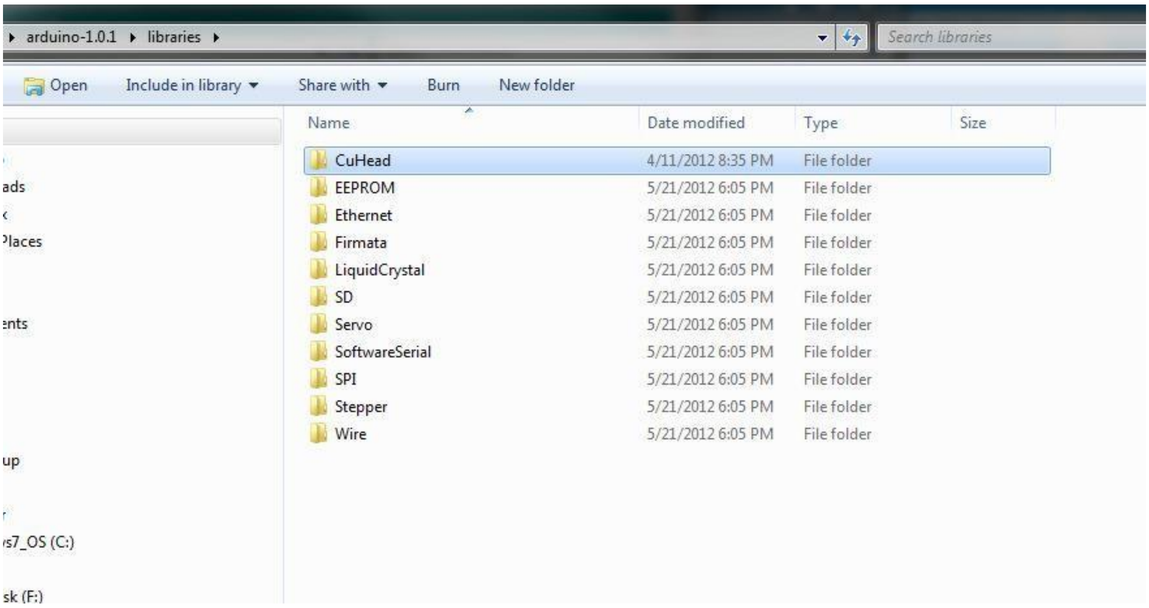
Arduino compiling



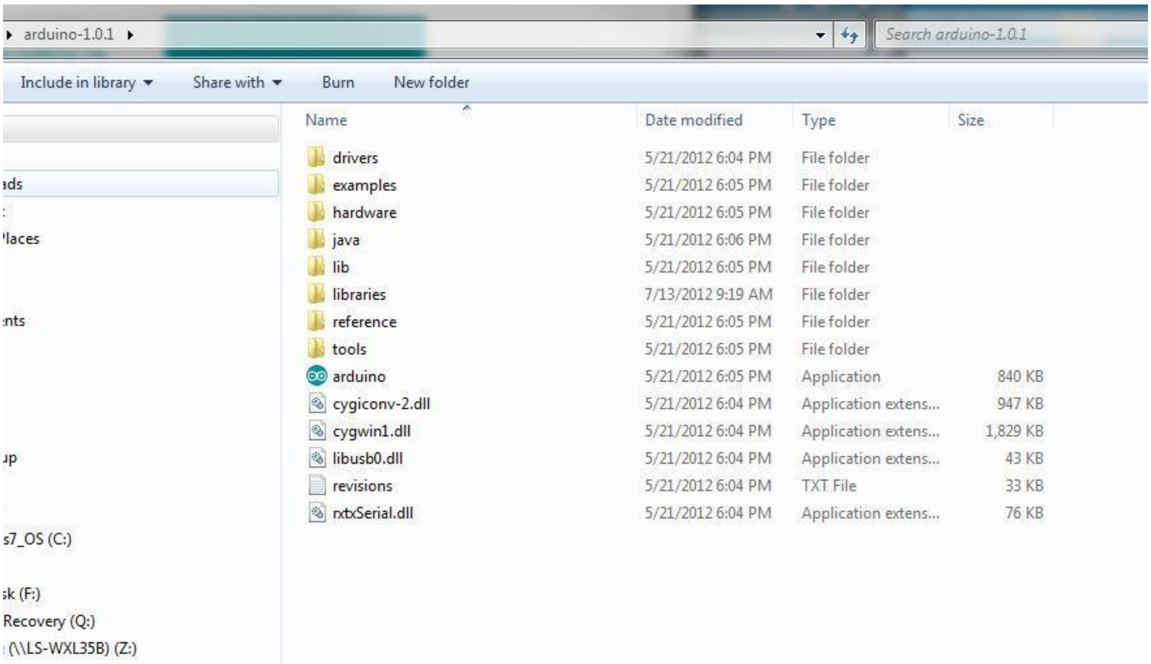
In next step download library



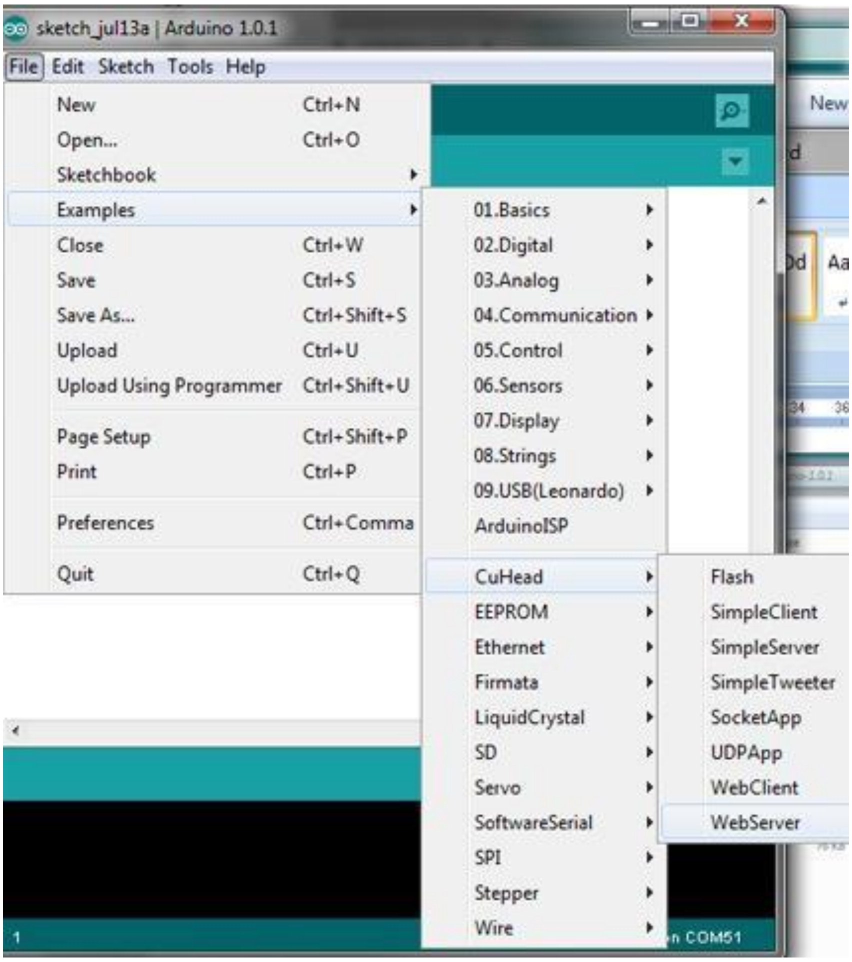
As Arduino doesn’t recognize the directory name, please rename it



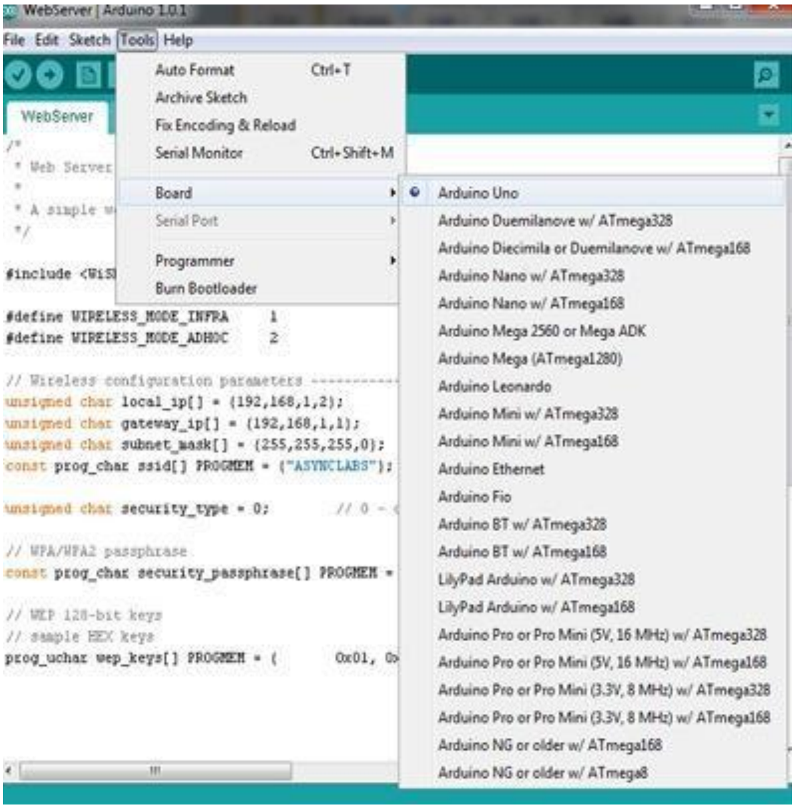
Launch Arduino by double click “arduino” below


arduino-1.0.1
Search arduino-1.0.1
Include in library
Share with
Burn
New folder
Name
Date modified
Type
Size
drivers 5/21/2012 6:04 PM File folder
examples 5/21/2012 6:05 PM File folder
hardware 5/21/2012 6:05 PM File folder
java 5/21/2012 6:06 PM File folder
lib 5/21/2012 6:05 PM File folder
libraries 7/13/2012 9:19 AM File folder
reference 5/21/2012 6:05 PM File folder
tools 5/21/2012 6:05 PM File folder
arduino 5/21/2012 6:05 PM Application 840 KB
cygiconv-2.dll 5/21/2012 6:04 PM Application extens... 947 KB
cygwin1.dll 5/21/2012 6:04 PM Application extens... 1,829 KB
libusb0.dll 5/21/2012 6:04 PM Application extens... 43 KB
revisions 5/21/2012 6:04 PM TXT File 33 KB
rxtxSerial.dll 5/21/2012 6:04 PM Application extens... 76 KB

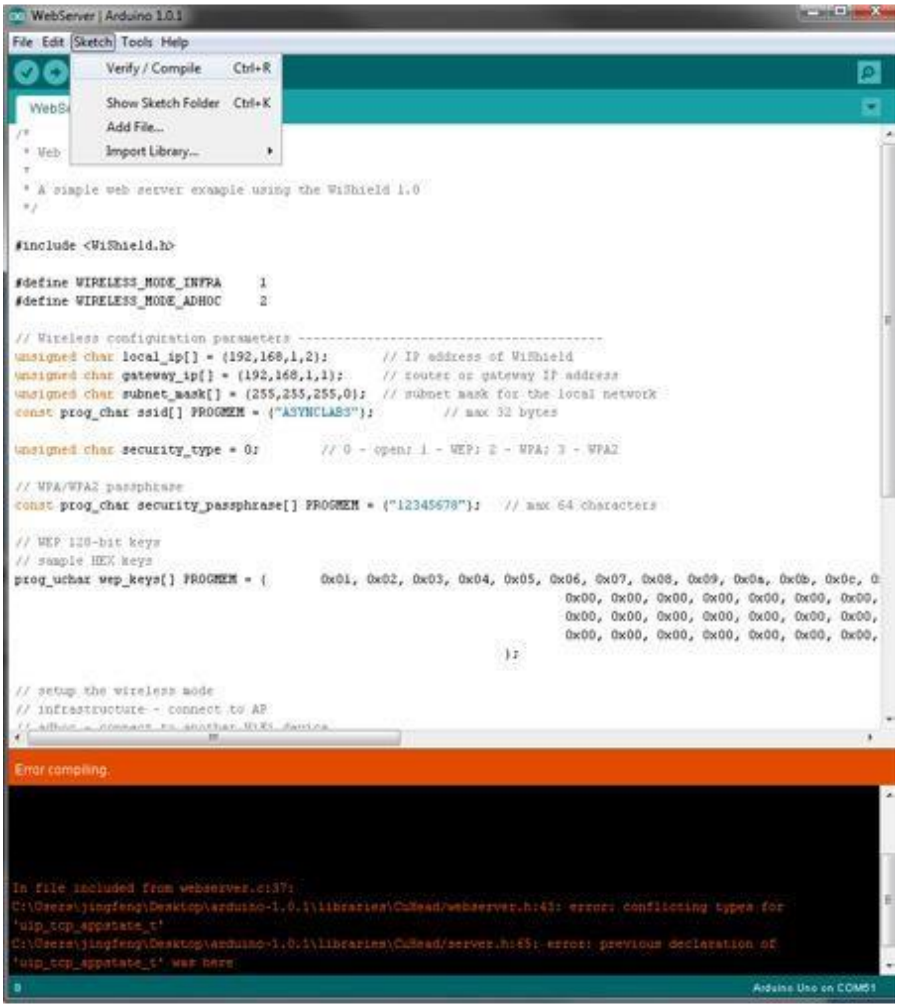
One example


sketch_jul13a | Arduino 1.0.1
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
01.Basics
02.Digital
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB(Leonardo)
ArduinoISP
CuHead
EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire
Flash
SimpleClient
SimpleServer
SimpleTweeter
SocketApp
UDPApp
WebClient
WebServer

Select the target board as “Arduino Uno”:

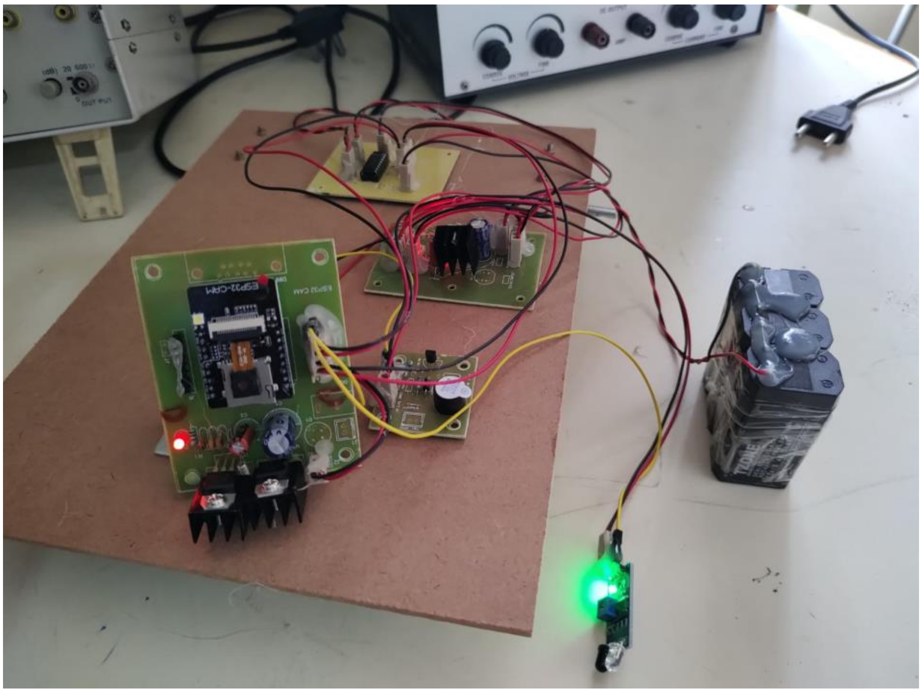


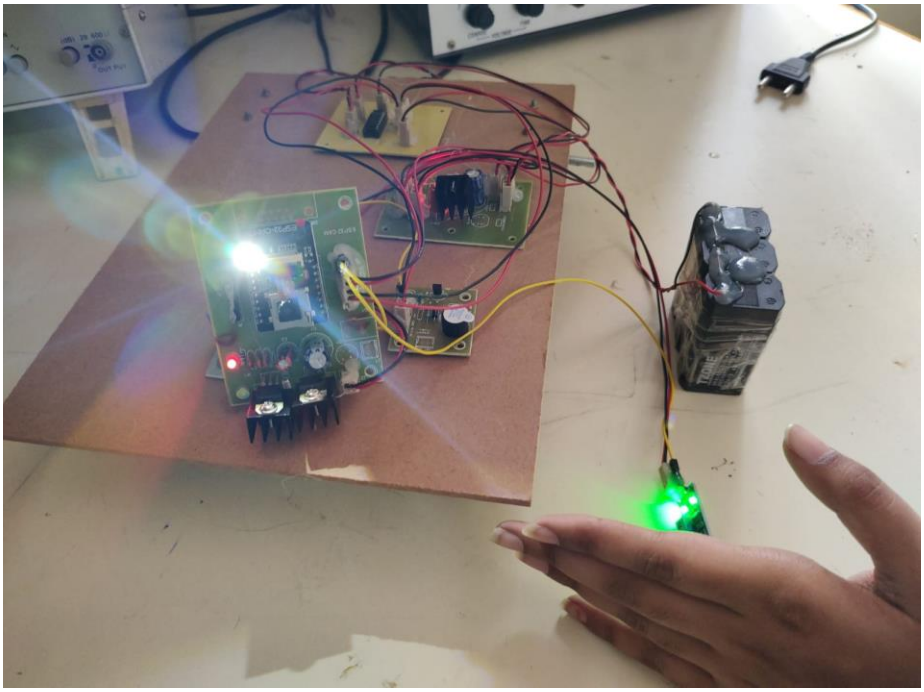
Click Sketch-> Verify/Compile:

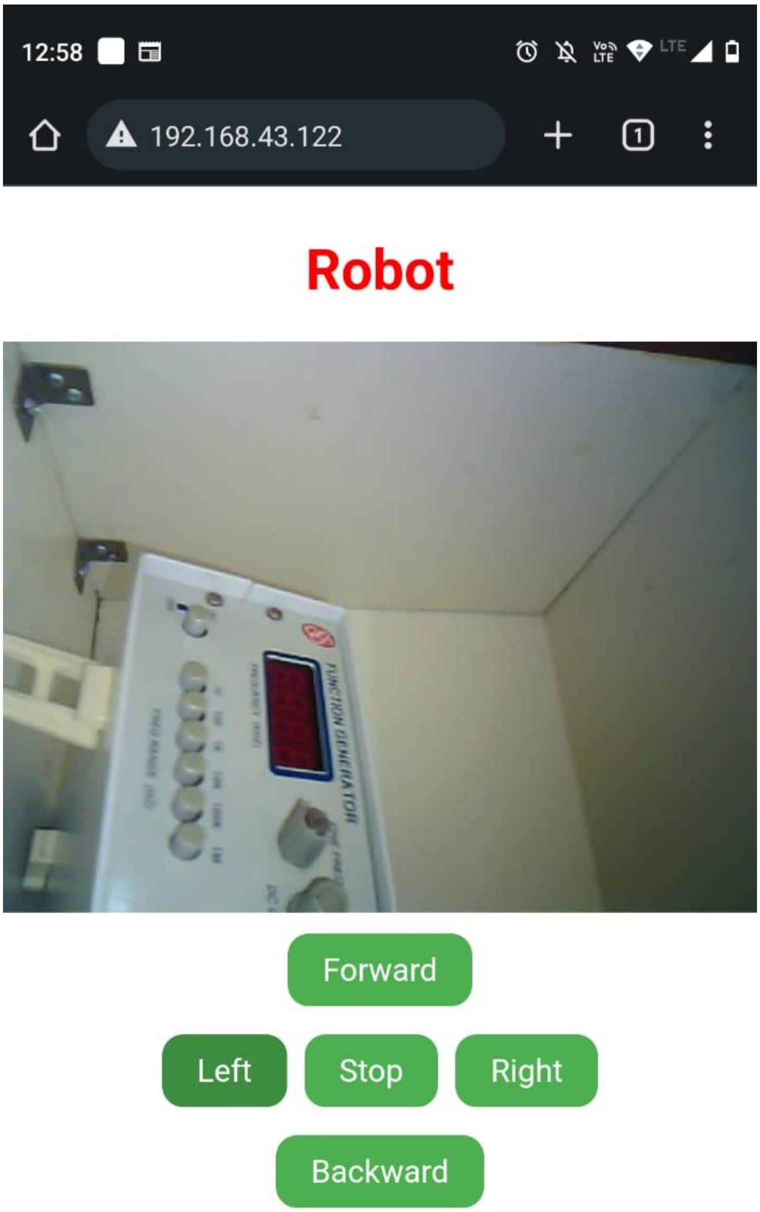

WebServer | Arduino 1.0.1
File Edit Sketch Tools Help
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Add File...
Import Library...
* A simple web server example using the WiShield 1.0
#include <WiShield.h>
#define WIRELESS_MODE_INFRA 1
#define WIRELESS_MODE_ADHOC 2
// Wireless configuration parameters
unsigned char local_ip[] = {192,168,1,2}; // IP address of WiShield
unsigned char gateway_ip[] = {192,168,1,1}; // router or gateway IP address
unsigned char subnet_mask[] = {255,255,255,0}; // subnet mask for the local network
const prog_char ssid[] PROGMEM = {"ASYNCLABS"}; // max 32 bytes
unsigned char security_type = 0; // 0 - open; 1 - WEP; 2 - WPA; 3 - WPA2
// WPA/WPA2 passphrase
const prog_char security_passphrase[] PROGMEM = {"12345678"}; // max 64 characters
// WEP 128-bit keys
// sample HEX keys
prog_uchar wep_keys[] PROGMEM = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c,
// setup the wireless mode
// infrastructure - connect to AP
Error compiling.
In file included from webserver.c:37:
'uip_tcp_appstate_t'
'uip_tcp_appstate_t' was here
Arduino Uno on COM51

CHAPTER 5

RESULTS






12:58
192.168.43.122
Robot
FUNCTION GENERATOR
Forward
Left
Stop
Right
Backward

ADVANTAGES DISADVANTAGES

Advantages:

1. Automation
2. Audio, Video surveillance system
3. wireless data access through Bluetooth
4. robot control-based surveillance
5. Efficient and low-cost design.
6. Low power consumption.

Disadvantages:

1. High cost
2. High Standard Maintenance

Applications.

1. Industrials
2. War field
3. Security
4. home

CONCLUSION

Nowadays, mainly women safety is the biggest issue in all parts of the world due to increased violence against them. The existing system either lags one of these features or require high end microcontroller. Hence, the proposed women safety night patrolling robot makes the best use of its features such as sound sensors, IR sensors, ESP cameras and IoT in order to patrol in its assigned area with least human intervention. The project concludes with a design of security robot for patrolling robot which uses night version camera to securing its premises. The robot move with particular intervals in the same direction. It is also equipped with night vision camera and IR sensors. it is used by a predefined path which is given to the controller for the moment of patrolling. It captures and send the videos directly to the control monitor room, for further actions.

Future scope

In future studies this system integrates with GPS get the exact location of fire and gas detection detected. Module it utilizes an interface GPS sensor to transmit area of the leakage over to the IOT login system, here we use IOT to check, get and show the gas leakage caution and location over IOT.

REFERENCES

[1] Muhammad Hamza, M Atique-ur-Rehman, Hamza Shafqat, Subhan Bin Khalid, “CGI SCRIPT AND MJPG VIDEO STREAMER BASED SURVEILLANCE ROBOT USING ESP-CAM” in 16th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 8th – 12th January, 2019, IEEE 2019.

[2] Juan G. Parada-Salado, Luis E. Ortega-García, Luis F. Ayala-Ramírez, Francisco J. Pérez Pinal, “A Low-Cost Land Wheeled Autonomous MiniRobot For In-Door Surveillance” In Ieee Latin America Transactions, Vol. 16, No. 5, May 2018.

[3] S M Ashish, Madhurya Manjunath, Ravindra L, Mohammed Nadeem, Neelaja K, “Automated Hybrid Surveillance Robot” In International Journal Of Innovations In Engineering Research And Technology [Ijiert], Issn: 2394-3696; Volume 5, Issue 5, May- 2018.

[4] T. Saravanakumar, D. Keerthana, D. Santhiya, J. Sneka, D. Sowmiya, “Surveillance Robot Using ESP-CAM-IoT” in International Journal of Electronics, Electrical and Computational System(IJEECS);ISSN 2348-117X; Volume 7, Issue 3, March 2018.

[5] Harshitha R, Muhammad Hameem Safwat Hussain, “SURVEILLANCE ROBOT USING ESP-CAM AND IOT” in 2018 IEEE - DOI 10.1109/ICDI3C.2018.00018; International Conference on Design Innovations for 3Cs Compute Communicate Control.

[6] Gaurav S Bagul, Vikram C Udawant, Kalpana V Kapade and Jayesh M Zope, “IOT BASED SURVEILLANCE ROBOT” in International Journal of Trend in Research and Development, Mar - Apr 2018; Volume 5(2), ISSN: 2394-9333.

[7] Rutuja. R. Bachhav, Priyanka. S. Bhavsar, Devyani. K. Dambhare, “Spy Robot For Video Surveillance And Metal Detection” In International Journal For Research In Applied Science & Engineering Technology (Ijraset); Issn: 2321-9653; Volume 6 Issue V, May 2018.

[8] Neha Joisher, Sonaljain, Tejas Malviya, Vaishali Gaikwad (Mohite), “Surveillance System Using Robotic Car” In International Conference On Control, Automation And Systems, Ieee 2018.

[9] Fatma Salih, Mysoon S.A. Omer, “ESP-CAM As A Video Server” In The International Conference On Computer, Control, Electrical, And Electronics Engineering (Iccceee) 2018.

[10] Sweeta Deshmukh, Priyadarshini, Mamta, Madhura Deshmukh, Dr.Md. Bakhar, “Iot Based Surveillance Robot” In International Journal Of Innovative Research In Computer And Communication Engineering; Vol.5, Special Issue 4, June 2017

[11] Ghanem Osman Elhaj Abdalla, T. Veeramanikandasamy, “Implementation Of Spy Robot For A Surveillance System Using Internet Protocol Of ESP-CAM” In 2nd Ieee International Conf On Recent Trends In Electronics Information & Communication Technology, May 19-20, 2017