**EMPOWERING WOMEN’S SAFETY WITH SMART IOT TECHNOLOGY: A ROBUST PROTECTION SYSTEM**

**Abstract:**

The project titled "Empowering Women’s Safety with Smart IoT Technology: A Robust Protection System" aims to enhance women's safety by leveraging a combination of Arduino and a Python-installed PC. The system incorporates a webcam connected to the PC to facilitate real-time surveillance. Serial communication is established between the PC and Arduino for seamless coordination. In case of an emergency, a designated switch triggers a series of protective measures. An image of the potential attacker is captured and promptly sent to a pre-configured email address. The system employs a buzzer alert to draw attention to the situation and includes a shock mechanism, utilizing a locally available mosquito bat, to further deter the assailant.

This innovative approach utilizes IoT technology to create a comprehensive safety system that not only documents potential threats through visual evidence but also provides an immediate physical deterrent to enhance women's safety in various environments.

Top of Form

Top of Form

**Keywords: Python installed PC, Emergency switch, Camera, OpenCV, Shock**

**INTRODUCTION**

In recent years, the issue of women's safety has garnered significant attention globally, with concerns arising about the rising incidents of harassment and violence against women. In response to this pressing societal issue, the project "Empowering Women’s Safety with Smart IoT Technology: A Robust Protection System" aims to leverage the advancements in IoT (Internet of Things) technology to create a comprehensive safety solution for women. This project seeks to provide women with a reliable and intuitive system that enhances their sense of security and enables them to navigate public spaces with confidence.

The introduction of smart IoT technology into women's safety initiatives presents a promising opportunity to address the challenges faced by women in various environments. By integrating IoT devices such as wearable sensors, GPS trackers, and communication modules, this project envisions creating a networked system that can detect and respond to potential threats in real-time. Through the seamless connectivity offered by IoT technology, the protection system can relay vital information to designated contacts or authorities, enabling prompt assistance in emergency situations.

Moreover, the project recognizes the importance of user-centric design and accessibility in ensuring the effectiveness of the safety solution. With a focus on user experience and interface design, the system aims to be intuitive and easy to use, catering to the diverse needs of women from different backgrounds and age groups. By empowering women with a smart IoT-based protection system, the project strives to contribute to a safer and more inclusive society where women can live and thrive without fear of violence or harassment

**Embedded system implementation**

**Introduction:**

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, and store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic controlsystems, etc.

User interface

Embedded system

Hardware

Software

Inputs

Output

Link to other systems

Fig: Overview of embedded system

**Embedded system:**

Embedded system includes mainly two sections, they are

1. Hardware

2. Software

Serial communication ports

Application specific circuits

Power supply and oscillator circuits

Timers

Processor

Interrupt controller

Parallel ports

Memory

Output devices interfacing

Input devices interfacing

and driver circuits

Block diagram of embedded system

**Embedded System Hardware:**

As with any electronic system, an embedded system requires a hardware platform on which it performs the operation. Embedded system hardware is built with a microprocessor or microcontroller. The embedded system hardware has elements like input output (I/O) interfaces, user interface, memory and the display. Usually, an embedded system consists of:

* Power Supply
* Processor
* Memory
* Timers
* Serial communication ports
* Output/Output circuits
* System application specific circuits

Embedded systems use different processors for its desired operation. Some of the processors used are

1. Microprocessor

2. Microcontroller

3. Digital signal processor

**Microprocessor vs. Microcontroller**

**Microprocessor**

* **CPU** on a chip.
* We can attach required amount of ROM, RAM and I/O ports.
* Expensive due to external peripherals.
* Large in size
* general-purpose

**Microcontroller**

* **Computer** on a chip
* fixed amount of on-chip ROM, RAM, I/O ports
* Low cost.
* Compact in size.
* Specific –purpose

**Embedded System Software:**

The embedded system software is written to perform a specific function. It is typically written in a high level format and then compiled down to provide code that can be lodged within a non-volatile memory within the hardware. An embedded system software is designed to keep in view of the three limits:

* Availability of system memory
* Availability of processor’s speed
* When the system runs continuously, there is a need to limit power dissipation for events like stop, run and wake up.

**Bringing software and hardware together for embedded system:**

To make software to work with embedded systems we need to bring software and hardware together .for this purpose we need to burn our source code into microprocessor or microcontroller which is a hardware component and which takes care of all operations to be done by embedded system according to our code.

Generally we write source codes for embedded systems in assembly language, but the processors run only executable files.The process of converting the source code representation of your embedded software into an executable binary image involves three distinct steps:

1. Each of the source files must be compiled or assembled into an object file.
2. All of the object files that result from the first step must be linked together to produce a single object file, called the re-locatable program.
3. Physical memory addresses must be assigned to the relative offsets within the re-locatable program in a process called relocation.

The result of the final step is a file containing an executable binary image that is ready to run on the embedded system.

Source code

Linker

Locator

Assembler

Executable file

Processor

Flow of burning source code to processor

**Applications:**

Embedded systems have different applications. A few select [applications of embedded systems](https://www.elprocus.com/embedded-systems-real-time-applications/) are smart cards, telecommunications, satellites, missiles, digital consumer electronics, computer networking, etc.

[Embedded Systems in Automobiles](http://www.edgefx.in/importance-of-embedded-systems-in-automobiles-with-applications/)

* Motor Control System
* Engine or Body Safety
* [Robotics](http://www.edgefx.in/top-list-robotics-projects-for-engineering-beginners/) in Assembly Line
* Mobile and E-Com Access

Embedded systems in Telecommunications

* Mobile computing
* Networking
* [Wireless Communications](http://www.edgefx.in/multiple-input-and-multiple-output-mimo-wireless-communications/)

Embedded Systems in Smart Cards

* Banking
* Telephone
* [Security Systems](http://www.edgefx.in/microcontroller-based-projects-on-car-security-systems-using-gsm/)

**Implementation flow:**

**Stage 1:**

Considering the problems of existing methods and giving solution to that problem by considering the basic requirements for our proposed system

**Stage 2:**

Considering the hardware requirement for the proposed system

For this we need to select the below components:

1. Microcontroller

2. Inputs for the proposed system (ex: sensors, drivers etc..,)

3. Outputs (ex: relays, loads)

**Stage 3:**

After considering hardware requirements, now we need to check out the software requirements. Based on the microcontroller we select there exists different software for coding, compiling, debugging. we need to write source code for that proposed system based on our requirements and compile, debug the code in that software .

After completing all the requirements of software and hardware we need to bring both together to work our system. For this we need to burn our source code into microcontroller, after burning our source code to microcontroller then connect all input and output modules as per our requirement.

**EXISTING SYSTEM**

In the current scenario, there is no standardized system for ensuring women's safety with integrated IoT technology. Traditional methods rely on personal alert devices, mobile applications, or emergency hotlines, which may not offer real-time surveillance or a robust defense mechanism. The absence of a unified and technologically advanced solution leaves women vulnerable to potential threats, limiting the effectiveness of existing safety measures. This project aims to address these shortcomings by proposing a comprehensive system that combines Arduino and a Python-installed PC, leveraging the capabilities of IoT to provide a more proactive and responsive approach to women's safety.

**Drawbacks:**

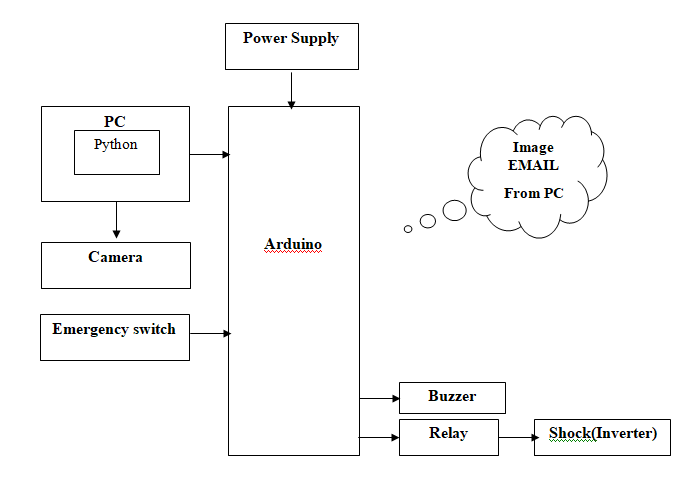
1. Limited Real-time photo:
2. Dependency on External Services:
3. Inadequate Defense Mechanism:
4. Lack of Automated Alert Systems:
5. Absence of Integrated Shock Mechanism:

**PROPOSED SYSTEM**

The proposed method enhances women's safety through the integration of advanced technologies. An Arduino microcontroller, coupled with a Python-installed PC, forms the core of the safety system. A webcam connected to the PC enables real-time monitoring of the surroundings. In case of an emergency, a designated switch triggers a sequence of actions. The system captures an image of the potential threat, and sends the image to a predefined email address. This immediate response allows for swift documentation of the situation, aiding in subsequent investigations.

To further enhance security, a buzzer alert is activated simultaneously to attract attention and deter potential threats. Additionally, a shock mechanism is incorporated into the system using a locally available mosquito bat. This serves as a non-lethal yet effective deterrent, providing an extra layer of defense for the user. The integration of these features creates a comprehensive safety system that combines surveillance, evidence collection, and an immediate response mechanism to empower women in ensuring their security.

**Block Diagram:**

****

**ADVANTAGES**

**Advantages:**

1. Real-time Photo capture
2. Immediate Documentation
3. Email Alerts
4. Buzzer Alert
5. Non-lethal Deterrent

**APPLICATIONS**

Personal Safety Monitoring: The primary application of this project is to provide women with a personal safety monitoring system that they can carry with them wherever they go. The system includes wearable IoT devices equipped with sensors to detect abnormal movements or distress signals. In case of an emergency, the device can trigger alerts, such as sending GPS coordinates and live audio/video feeds to designated contacts or authorities, enabling timely intervention.

Safe Transportation: Another key application is in the realm of transportation safety, particularly for women using public transportation or ride-sharing services. By integrating IoT technology into vehicles or transportation apps, the system can monitor the route in real-time, detect deviations from the planned route, and alert both the passenger and emergency contacts if any suspicious activity or detours occur.

Campus Safety: Universities and educational institutions can implement this project to enhance campus safety for female students and staff. IoT-enabled security cameras, panic buttons, and geofencing systems can be deployed across campus premises to detect and respond to potential threats. Additionally, students can utilize wearable safety devices connected to the campus security network for immediate assistance in distress situations.

Workplace Security: Employers can deploy IoT-based safety solutions in workplaces to ensure the safety and well-being of female employees. Wearable devices equipped with panic buttons and biometric sensors can alert security personnel or HR departments in case of emergencies, such as intruders or harassment incidents. Moreover, IoT-enabled access control systems can enhance security measures by restricting unauthorized access to sensitive areas.

Public Events and Gatherings: Organizers of public events, festivals, and gatherings can utilize this project to implement comprehensive safety measures for attendees, especially women. IoT devices embedded in event venues can monitor crowd movements, identify potential security threats, and facilitate quick response actions. Additionally, attendees can opt to wear wearable safety devices for personalized protection and emergency assistance.

Community Safety Initiatives: Local authorities and community organizations can collaborate to deploy IoT-based safety solutions in neighborhoods and residential areas. Smart surveillance cameras, motion sensors, and smart streetlights can help deter criminal activities and provide early detection of suspicious behavior. Community members can also participate by using mobile applications connected to the safety network to report incidents and request assistance.

Travel Safety: For women travelers, especially those exploring unfamiliar destinations or traveling solo, IoT technology can offer invaluable safety benefits. Wearable devices with GPS tracking capabilities can provide real-time location updates to trusted contacts or family members. Additionally, integration with travel apps and emergency response services can ensure swift assistance in case of emergencies, such as medical emergencies or harassment incidents.

Online Safety: In the digital realm, where cyberbullying and online harassment are prevalent concerns, this project can extend its reach to ensure women's online safety. IoT-enabled parental control devices, social media monitoring tools, and cybersecurity solutions can help protect women and girls from online threats, including identity theft, cyberstalking, and harmful content exposure.

Emergency Response Systems: IoT technology can be integrated into existing emergency response systems to streamline communication and coordination during crises. By equipping emergency responders with IoT devices and mobile applications, they can receive real-time alerts, access critical information about the incident, and coordinate rescue efforts more effectively, thereby reducing response times and saving lives.

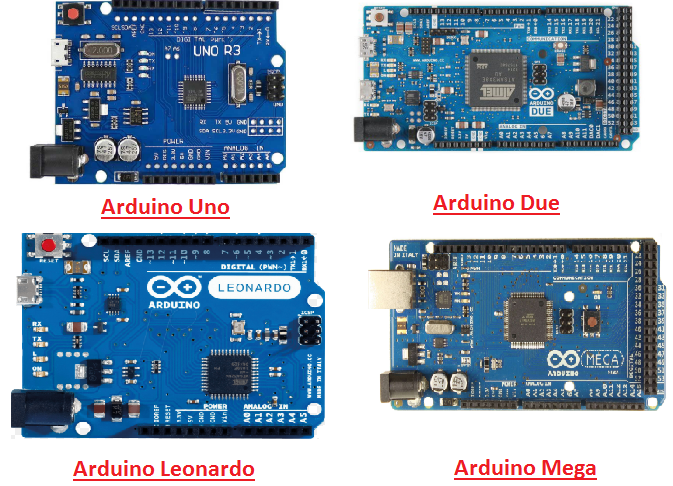
Domestic Violence Prevention: Finally, this project can contribute to domestic violence prevention efforts by providing victims with discreet and accessible safety tools. Wearable panic buttons, hidden cameras, and smart home security systems can help victims alert authorities or support networks discreetly and gather evidence of abuse for legal proceedings. Additionally, IoT-enabled monitoring systems can provide remote monitoring and support for victims who may be isolated or unable to seek help openly.

**HARDWARE REQUIREMENTS**

**Arduino:**

Arduino Uno is a very valuable addition in the electronics that consists of USB interface, 14 digital I/O pins, 6 analog pins, and Atmega328 microcontroller. It also supports serial communication using Tx and Rx pins.

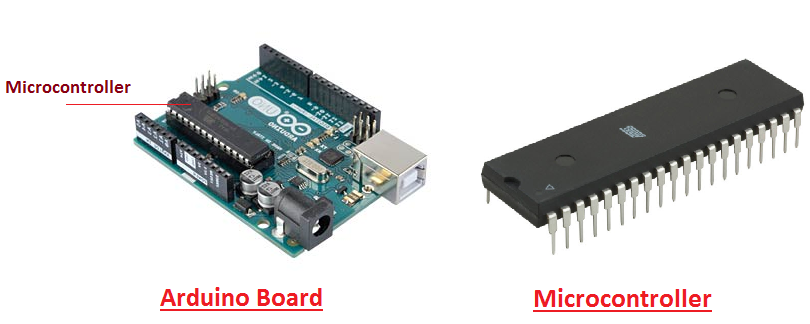
There are many versions of Arduino boards introduced in the market like Arduino Uno, Arduino Due, Arduino Leonardo, Arduino Mega, however, most common versions are Arduino Uno and Arduino Mega. If you are planning to create a project relating to digital electronics, embedded system, robotics, or IoT, then using Arduino Uno would be the best, easy and most economical option.



It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality.

The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and required some basic skills to learn it. It can be programmed using C and C++ language.

Some people get confused between **Microcontroller and Arduino**. While former is just an on system 40 pin chip that comes with a built-in microprocessor and later is a board that comes with the microcontroller in the base of the board, bootloader and allows easy access to input-output pins and makes uploading or burning of the program very easy.

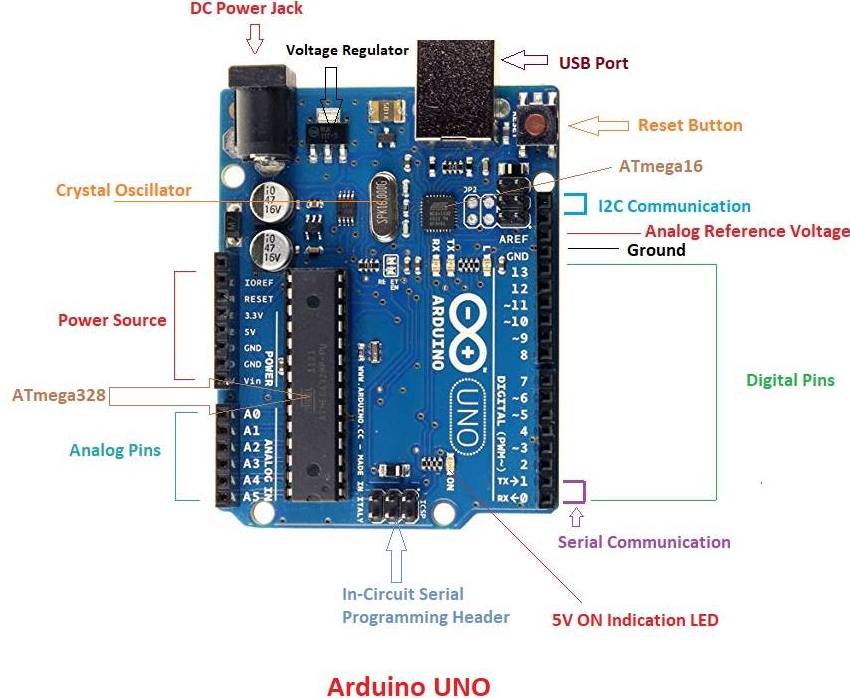


While learning microcontroller requires some expertise and skills.

Nevertheless, we can say every Arduino is basically a [microcontroller](https://www.theengineeringprojects.com/2018/03/introduction-to-microcontrollers.html) but not every microcontroller is an Arduino.

**Introduction to Arduino**

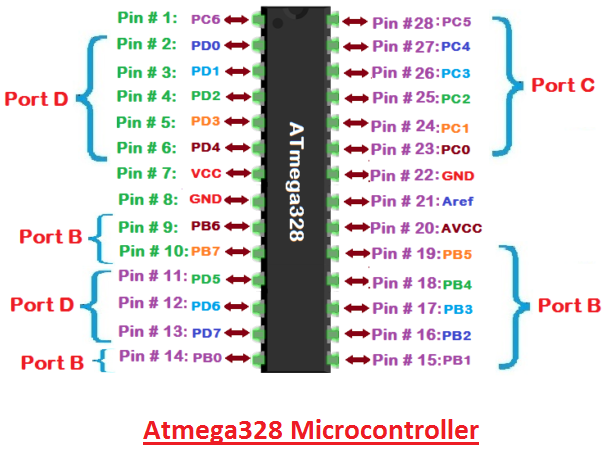
* **Arduino Uno** is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.
* First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.
* The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.
* It allows the designers to control and sense the external electronic devices in the real world

****

* This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. IDE is equally compatible with Windows, MAC or Linux Systems, however, Windows is preferable to use. Programming languages like C and C++ are used in IDE.
* Apart from USB, battery or AC to DC adopter can also be used to power the board.
* Arduino Uno boards are quite similar to other boards in Arduino family in terms of use and functionality, however, Uno boards don’t come with FTDI USB to Serial driver chip.
* There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno are the most official versions that come with Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.
* When nature and functionality of the task go complex, Mirco SD card can be added in the boards to make them store more information.

**Features of Arduino**

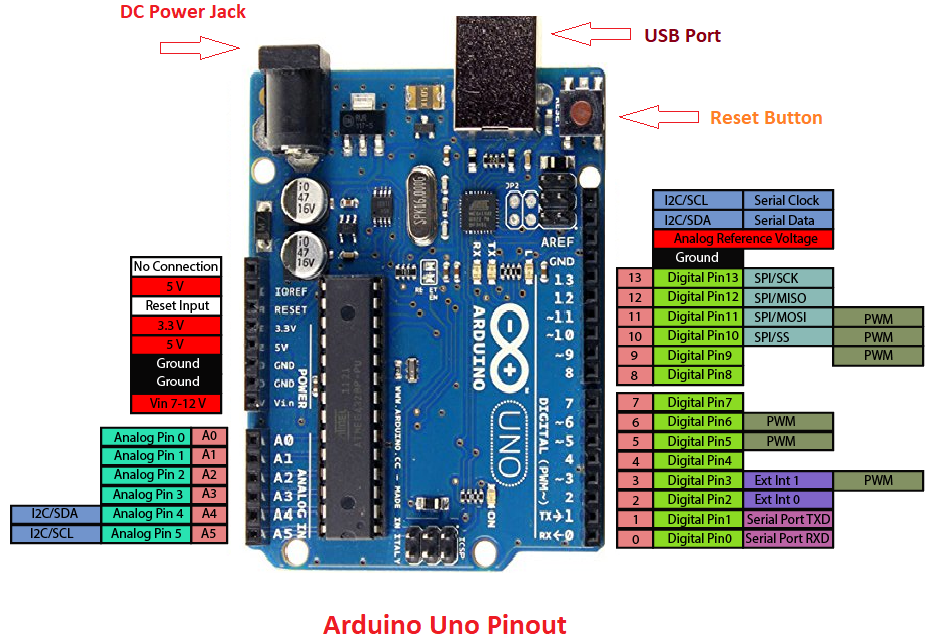
* Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer.
* [Atmega328](https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html) microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

****

* It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
* This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
* Reset pin is added in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
* There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.
* The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analogReference() function and AREF pin.
* 13KB of flash memory is used to store the number of instructions in the form of code.
* Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

**Arduino Pinout**

* Arduino Uno is based on AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. Following figure shows the pinout of the Arduino Uno Board

****

**Pin Description:**

There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resisters useless and damages the device.

**LED.** Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

**Vin.** It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

**5V.** This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack.

USB supports voltage around 5V while Vin and Power Jack support a voltage ranges between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins, they result in bypassing the voltage regulation that can damage the board if voltage surpasses from its limit.

**GND.** These are ground pins. More than one ground pins are provided on the board which can be used as per requirement.

**Reset.** This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

**IOREF.** This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.

**PWM.** PWM is provided by 3, 5, 6,9,10, 11pins. These pins are configured to provide 8-bit output PWM.

**SPI.** It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.

**AREF.** It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

**TWI.** It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

**Serial Communication.** Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx pin is used to receive data while Tx pin is used to transmit data.

**External Interrupts.** Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

### **Arduino Uno Technical Specifications**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 8 bit AVR family microcontroller |
| Operating Voltage | 5V |
| Recommended InputVoltage | 7-12V |
| Input Voltage Limits | 6-20V |
| Analog Input Pins | 6 (A0 – A5) |
| Digital I/O Pins | 14 (Out of which 6 provide PWM output) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB is used for Bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

**Communication and Programming:**

Arduino Uno comes with an ability of interfacing with other other Arduino boards, microcontrollers and computer. The Atmega328 placed on the board provides serial communication using pins like Rx and Tx.

The Atmega16U2 incorporated on the board provides a pathway for serial communication using USB com drivers. Serial monitor is provided on the IDE software which is used to send or receive text data from the board. If LEDs placed on the Rx and Tx pins will flash, they indicate the transmission of data.

Arduino Uno is programmed using Arduino Software which a cross-platform application called IDE is written in Java. The AVR microcontroller Atmega328 laid out on the base comes with built-in boot loader that sets you free from using a separate burner to upload the program on the board.

****

**Applications:**

Arduino Uno comes with a wide range of applications. A larger number of people are using Arduino boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board.

* [Embedded System](https://www.theengineeringprojects.com/2016/10/what-is-embedded-systems.html)
* Security and Defense System
* Digital Electronics and Robotics
* Parking Lot Counter
* Weighing Machines
* Traffic Light Count Down Timer
* Medical Instrument
* Emergency Light for Railways
* Home Automation
* Industrial Automation

There are a lot of other microcontrollers available in the market that are more powerful and cheap as compared to Arduino board. So, why you prefer Arduino Uno?

Actually, Arduino comes with a big community that is developing and sharing the knowledge with a wide range of audience. Quick support is available pertaining to technical aspects of any electronic project. When you decide Arduino board over other controllers, you don’t need to arrange extra peripherals and devices as most of the functions are readily available on the board that makes your project economical in nature and free from a lot of technical expertise.

**Buzzer:**

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play.



**Buzzer Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Positive | Identified by (+) symbol or longer terminal lead. Can be powered by 5V DC |
| 2 | Negative | Identified by short terminal lead. Typically connected to the ground of the circuit |

**Buzzer Features and Specifications**

* Rated Voltage: 6V DC
* Operating Voltage: 4-8V DC
* Rated current: <30mA
* Sound Type: Continuous Beep
* Resonant Frequency: ~2300 Hz
* Small and neat sealed package
* Breadboard and Perf board friendly

**How to use a Buzzer**

A **buzzer**is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on [breadboard](https://components101.com/misc/breadboard-connections-uses-guide), Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customized with help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

**Applications of Buzzer**

* Alarming Circuits, where the user has to be alarmed about something
* Communication equipment’s
* Automobile electronics
* Portable equipment’s, due to its compact size

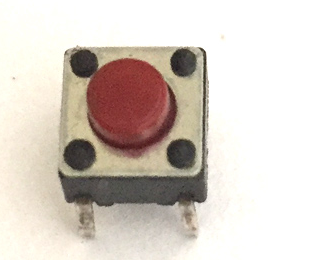
**PUSH BUTTON SWITCH:**

Push buttons are power-controlling switches of a machine or appliance. These are usually made of metal or thermoplastic and offer very simple access to the user. In electric circuits, the power flows continuously through the devices, and to regulate this power supply, we use push buttons. These are, simply push buttons that can be seen or felt and are easily operable with our hands and fingers.

You will find these switches in calculators, push-buttons, telephones, kitchen appliances, locks and so many appliances which are used across homes and industries. In cases of large-scale machinery, there are provisions for separate colors for varied push buttons. These colors indicate the function of the push button, making it easy to use. In our everyday use, we often have one or a maximum of 2 push buttons on our appliances and devices, thus making them simpler to use.

**How Push Buttons Work**

Push buttons work with a simple yet ingenious mechanism. These switches are usually equipped with an internal spring mechanism. It is this spring that returns the button to its 'in' and 'out' positions. This spring comes into contact with two wires, thus allowing the power supply to the appliance, and the push button is then said to be in the 'on' position. When the spring’s contact with the two wires is interrupted, the push button is said to be in the 'off' position.



**BATTERY:**

A rechargeable battery is an energy storage device that can be charged again after being discharged by applying [DC](https://whatis.techtarget.com/definition/DC-direct-current) current to its terminals.

Rechargeable [batteries](https://searchmobilecomputing.techtarget.com/definition/battery) allow for multiple usages from a cell, reducing waste and generally providing a better long-term investment in terms of dollars spent for usable device time. This is true even factoring in the higher purchase price of rechargeable and the requirement for a charger.  
A rechargeable battery is generally a more sensible and sustainable replacement to one-time use batteries, which generate current through a chemical reaction in which a reactive anode is consumed. The anode in a rechargeable battery gets consumed as well but at a slower rate, allowing for many charges and discharges.

In use, rechargeable batteries are the same as conventional ones. However, after discharge the batteries are placed in a charger or, in the case of built-in batteries, an [AC](https://whatis.techtarget.com/definition/alternating-current-AC)/DC adapter is connected.

While rechargeable batteries offer better long term cost and reduce waste, they do have a few cons. Many types of rechargeable cells created for consumer devices, including AA and AAA, C and D batteries, produce a lower voltage of 1.2v in contrast to the 1.5v of alkaline batteries. Though this lower voltage doesn't prevent correct operation in properly-designed electronics, it can mean a single charge does not last as long or offer the same power in a session. This is not the case, however, with lithium polymer and [lithium ion batteries](https://searchmobilecomputing.techtarget.com/definition/Lithium-Ion-battery).

Some types of batteries such as [nickel cadmium](https://searchmobilecomputing.techtarget.com/definition/Nickel-Cadmium-battery) and [nickel-metal hydride](https://searchmobilecomputing.techtarget.com/definition/Nickel-Metal-Hydride-battery) can develop a [battery memory effect](https://whatis.techtarget.com/definition/battery-memory-effect) when only partially discharged, reducing performance of subsequent charges and thus [battery life](https://whatis.techtarget.com/definition/battery-life) in a given device.

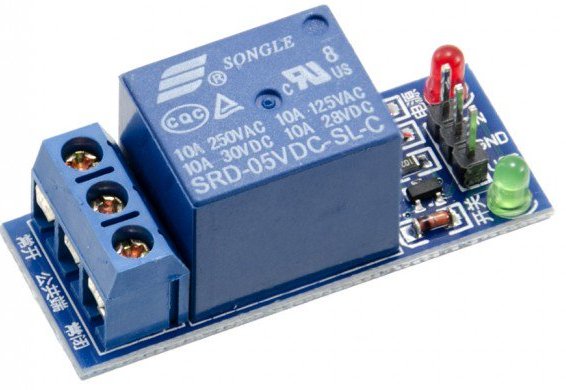
Rechargeable batteries are used in many applications such as cars, all manner of consumer electronics and even off-grid and supplemental facility power storage.

**Relay:**

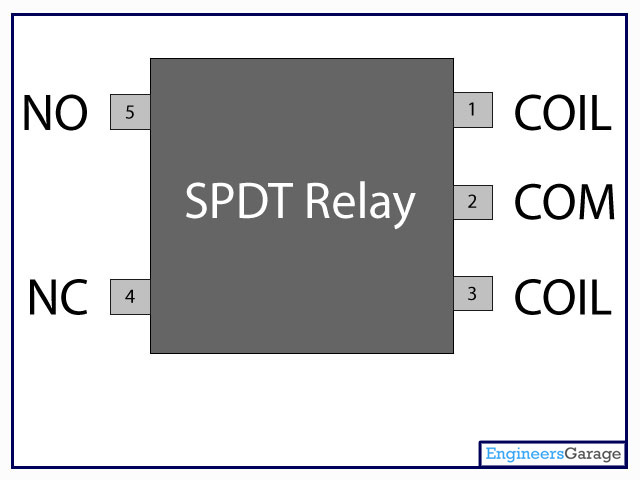
**What is a relay?**

A relay is an electromagnetic switch that is used to turn on and turn off a circuit by a low power signal, or where several circuits must be controlled by one signal.

Most of the high end industrial application devices have relays for their effective working. Relays are simple switches which are operated both electrically and mechanically. Relays consist of an electromagnet and also a set of contacts. The switching mechanism is carried out with the help of the electromagnet. There are also other operating principles for its working. But they differ according to their applications. Most of the devices have the application of relays.



### **Pin Diagram:**



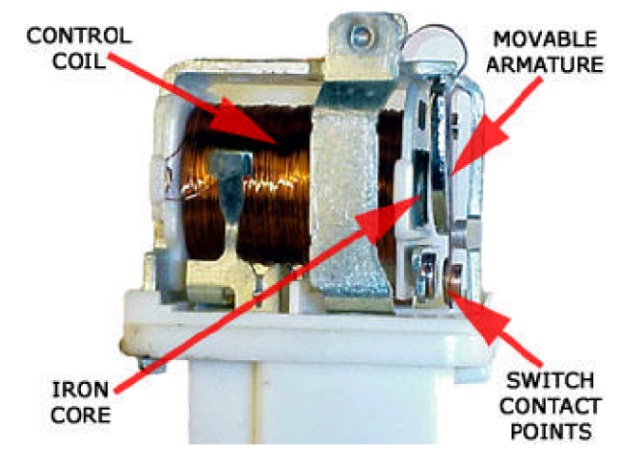
**Why is a relay used?**

The main operation of a relay comes in places where only a low-power signal can be used to control a circuit. It is also used in places where only one signal can be used to control a lot of circuits. The application of relays started during the invention of telephones. They played an important role in switching calls in telephone exchanges. They were also used in long distance telegraphy. They were used to switch the signal coming from one source to another destination. After the invention of computers they were also used to perform Boolean and other logical operations. The high end applications of relays require high power to be driven by electric motors and so on. Such relays are called contactors.

### **Relay Design**

* There are only four main parts in a relay. They are
* Electromagnet
* Movable Armature
* Switch point contacts
* Spring

The figures given below show the actual design of a simple relay.



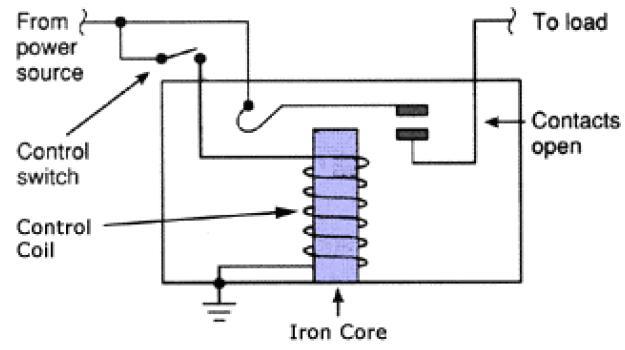
**Relay Construction**

It is an electro-magnetic relay with a wire coil, surrounded by an iron core. A path of very low reluctance for the magnetic flux is provided for the movable armature and also the switch point contacts.

The movable armature is connected to the yoke which is mechanically connected to the switch point contacts. These parts are safely held with the help of a spring. The spring is used so as to produce an air gap in the circuit when the relay becomes de-energized.

### **How relay works?**

The relay function can be better understood by explaining the following diagram given below.



**Relay Design**

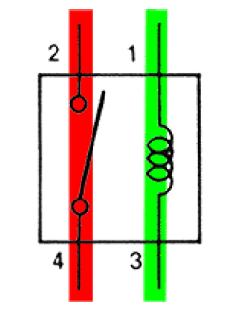
The diagram shows an inner section diagram of a relay. An iron core is surrounded by a control coil. As shown, the power source is given to the electromagnet through a control switch and through contacts to the load. When current starts flowing through the control coil, the electromagnet starts energizing and thus intensifies the magnetic field. Thus the upper contact arm starts to be attracted to the lower fixed arm and thus closes the contacts causing a short circuit for the power to the load. On the other hand, if the relay was already de-energized when the contacts were closed, then the contact move oppositely and make an open circuit.

As soon as the coil current is off, the movable armature will be returned by a force back to its initial position. This force will be almost equal to half the strength of the magnetic force. This force is mainly provided by two factors. They are the spring and also gravity.

Relays are mainly made for two basic operations. One is low voltage application and the other is high voltage. For low voltage applications, more preference will be given to reduce the noise of the whole circuit. For high voltage applications, they are mainly designed to reduce a phenomenon called arcing.

### **Relay Basics**

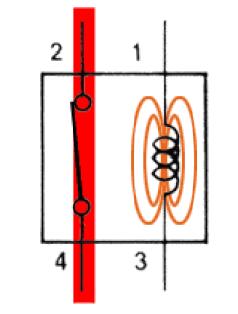
The basics for all the relays are the same. Take a look at a 4 pin relay shown below. There are two colors shown. The green color represents the control circuit and the red color represents the load circuit. A small control coil is connected onto the control circuit. A switch is connected to the load. This switch is controlled by the coil in the control circuit. Now let us take the different steps that occur in a relay.



**Relay operation**

* **Energized Relay (ON)**

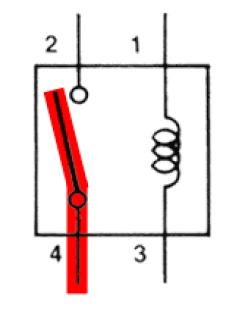
As shown in the circuit, the current flowing through the coils represented by pins 1 and 3 causes a magnetic field to be aroused. This magnetic field causes the closing of the pins 2 and 4. Thus the switch plays an important role in the relay working. As it is a part of the load circuit, it is used to control an electrical circuit that is connected to it. Thus, when the electrical relay in energized the current flow will be through the pins 2 and 4.



**Energized Relay (ON)**

* **De – Energized Relay (OFF)**

As soon as the current flow stops through pins 1 and 3, the relay switch opens and thus the open circuit prevents the current flow through pins 2 and 4. Thus the relay becomes de-energized and thus in off position.



**De-Energized Relay (OFF)**

**In simple, when a voltage is applied to pin 1, the electromagnet activates, causing a magnetic field to be developed, which goes on to close the pins 2 and 4 causing a closed circuit. When there is no voltage on pin 1, there will be no electromagnetic force and thus no magnetic field. Thus the switches remain open.**

### **Pole and Throw**

Relays have the exact working of a switch. So, the same concept is also applied. A relay is said to switch one or more poles. Each pole has contacts that can be thrown in mainly three ways. They are

* **Normally Open Contact (NO):**  NO contact is also called a make contact. It closes the circuit when the relay is activated. It disconnects the circuit when the relay is inactive.
* **Normally Closed Contact (NC):**  NC contact is also known as break contact. This is opposite to the NO contact. When the relay is activated, the circuit disconnects. When the relay is deactivated, the circuit connects.
* **Change-over (CO) / Double-throw (DT) Contacts:**  This type of contacts are used to control two types of circuits. They are used to control a NO contact and also a NC contact with a common terminal. According to their type they are called by the names **break before make** and **make before break** contacts.

Relays can be used to control several circuits by just one signal. A relay switches one or more poles, each of whose contacts can be thrown by energizing the coil.

Relays are also named with designations like

* **Single Pole Single Throw (SPST)**: The SPST relay has a total of four terminals. Out of these two terminals can be connected or disconnected. The other two terminals are needed for the coil to be connected.
* **Single Pole Double Throw (SPDT):** The SPDT relay has a total of five terminals. Out of these two are the coil terminals. A common terminal is also included which connects to either of two others.
* **Double Pole Single Throw (DPST):**  The DPST relay has a total of six terminals. These terminals are further divided into two pairs. Thus they can act as two SPST which are actuated by a single coil. Out of the six terminals two of them are coil terminals.
* **Double Pole Double Throw (DPDT)**: The DPDT relay is the biggest of all. It has mainly eight relay terminals. Out of these two rows are designed to be change over terminals. They are designed to act as two SPDT relays which are actuated by a single coil.

### **Relay Applications**

* A relay circuit is used to realize logic functions. They play a very important role in providing safety critical logic.
* Relays are used to provide time delay functions. They are used to time the delay open and delay close of contacts.
* Relays are used to control high voltage circuits with the help of low voltage signals. Similarly they are used to control high current circuits with the help of low current signals.
* They are also used as protective relays. By this function all the faults during transmission and reception can be detected and isolated.

#### **Application of Overload Relay**

Overload relay is an electro-mechanical device that is used to safeguard motors from overloads and power failures. Overload relays are installed in motors to safeguard against sudden current spikes that may damage the motor. An overload relay switch works in characteristics with current over time and is different from circuit breakers and fuses, where a sudden trip is made to turn off the motor. The most widely used overload relay is the thermal overload relay where a bimetallic strip is used to turn off the motor. This strip is set to make contact with a contactor by bending itself with rising temperatures due to excess current flow. The contact between the strip and the contactor causes the contactor to de-energize and restricts the power to the motor, and thus turns it off.

Another type of overload motor is the electronic type which continuously watches the motor current, whereas the thermal overload relay shuts off the motor depending on the rise of temperature/heat of the strip.

All overload relays available to buy comes in different specifications, the most important of them being the current ranges and response time. Most of them are designed to automatically reset to work after the motor is turned back on.

### **Relay Selection**

You must note some factors while selecting a particular relay. They are

* Protection Different protections like contact protection and coil protection must be noted. Contact protection helps in reducing arcing in circuits using inductors. Â Coil protection helps in reducing surge voltage produced during switching.
* Look for a standard relay with all regulatory approvals.
* Switching time Ask for high speed switching relays if you want one.
* Ratings There are current as well as voltage ratings. The current ratings vary from a few amperes to about 3000 amperes. Â In case of voltage ratings, they vary from 300 Volt AC to 600 Volt AC. There are also high voltage relays of about 15,000 Volts.
* Type of contact used whether it is a NC or NO or closed contact.
* Select Make before Break or Break before Make contacts wisely.
* Isolation between coil circuit and contacts

**WEB CAMERA**

A **webcam** is a [video camera](https://en.wikipedia.org/wiki/Video_camera) that feeds or [streams](https://en.wikipedia.org/wiki/Streaming_media) an image or video in real time to or through a [computer network](https://en.wikipedia.org/wiki/Computer_network), such as the [Internet](https://en.wikipedia.org/wiki/Internet). Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video.

Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)), such streams usually use [compressed formats](https://en.wikipedia.org/wiki/Video_compression). The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions



## **SOFTWARE REQUIREMENTS**

**Arduino IDE:**

**Arduino IDE**where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

**Introduction to Arduino IDE:**

* Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
* It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
* It is easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
* A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, [Arduino Micro](https://www.theengineeringprojects.com/2018/09/introduction-to-arduino-micro.html) and many more.
* Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
* The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
* The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
* This environment supports both C and C++ languages.

**How to install Arduino IDE:**

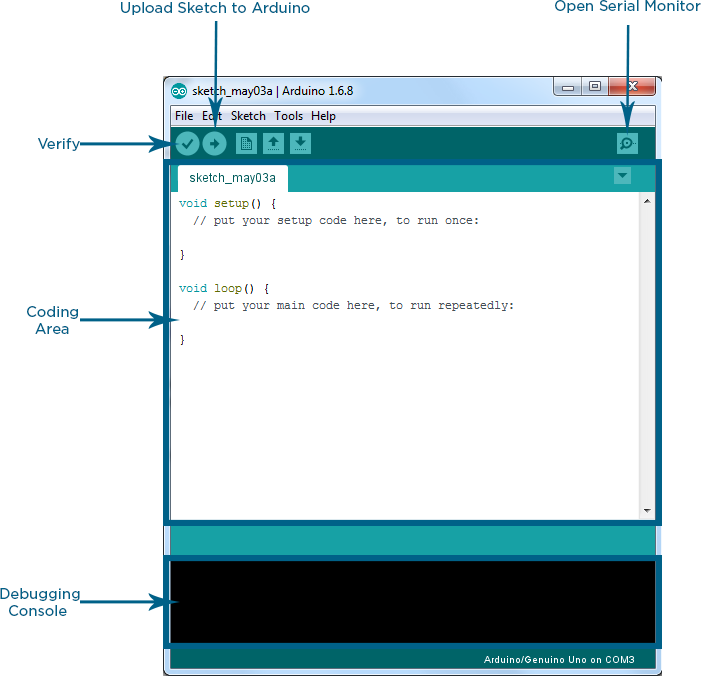
You can download the Software from [Arduino](https://www.arduino.cc/en/Main/Software) main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

* If you aim to download Windows app version, make sure you have Windows 8.1 or Windows 10, as app version is not compatible with Windows 7 or older version of this operating system.

The IDE environment is mainly distributed into three sections

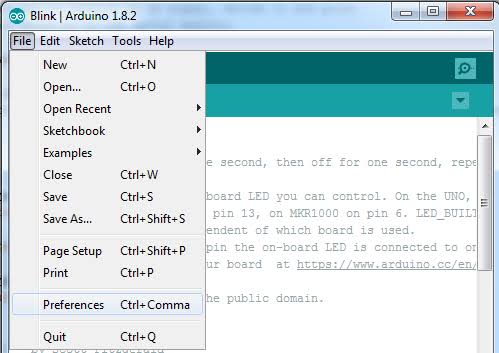
* **1. Menu Bar**
* **2. Text Editor**
* **3. Output Pane**

As you download and open the IDE software, it will appear like an image below.

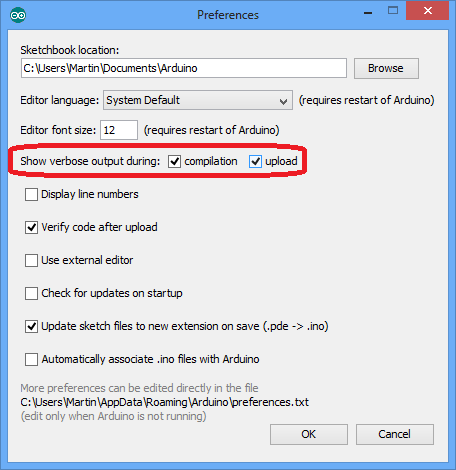


The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

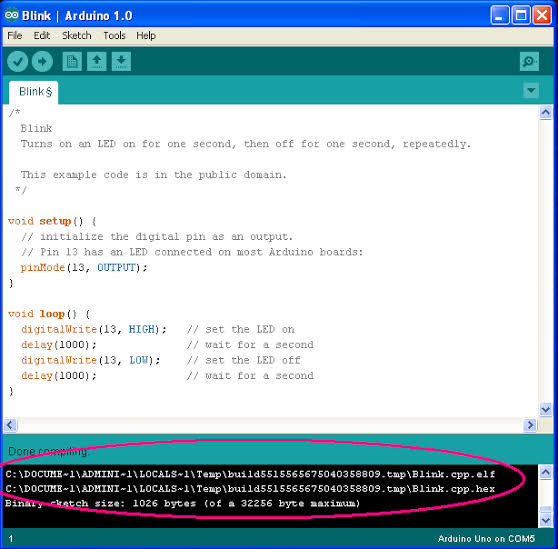
* **File** – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.

****

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

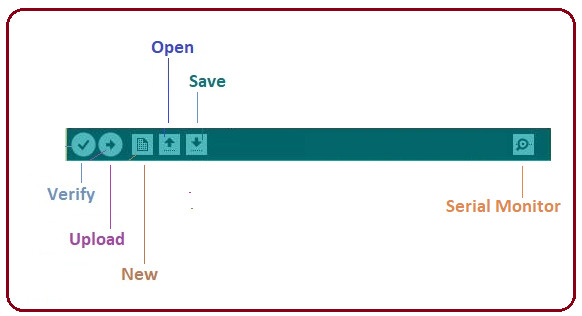


And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

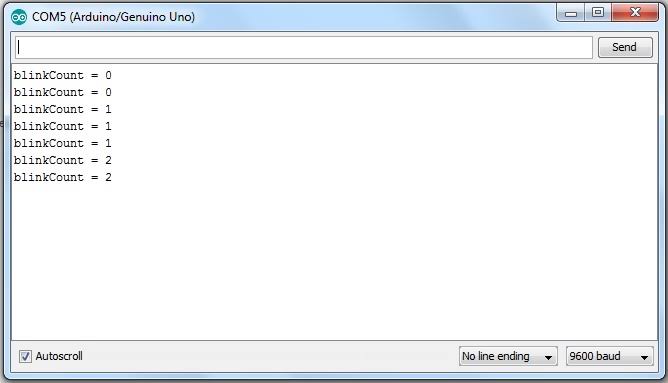
****

* **Edit** – Used for copying and pasting the code with further modification for font
* **Sketch** – For compiling and programming
* **Tools** – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
* **Help** – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

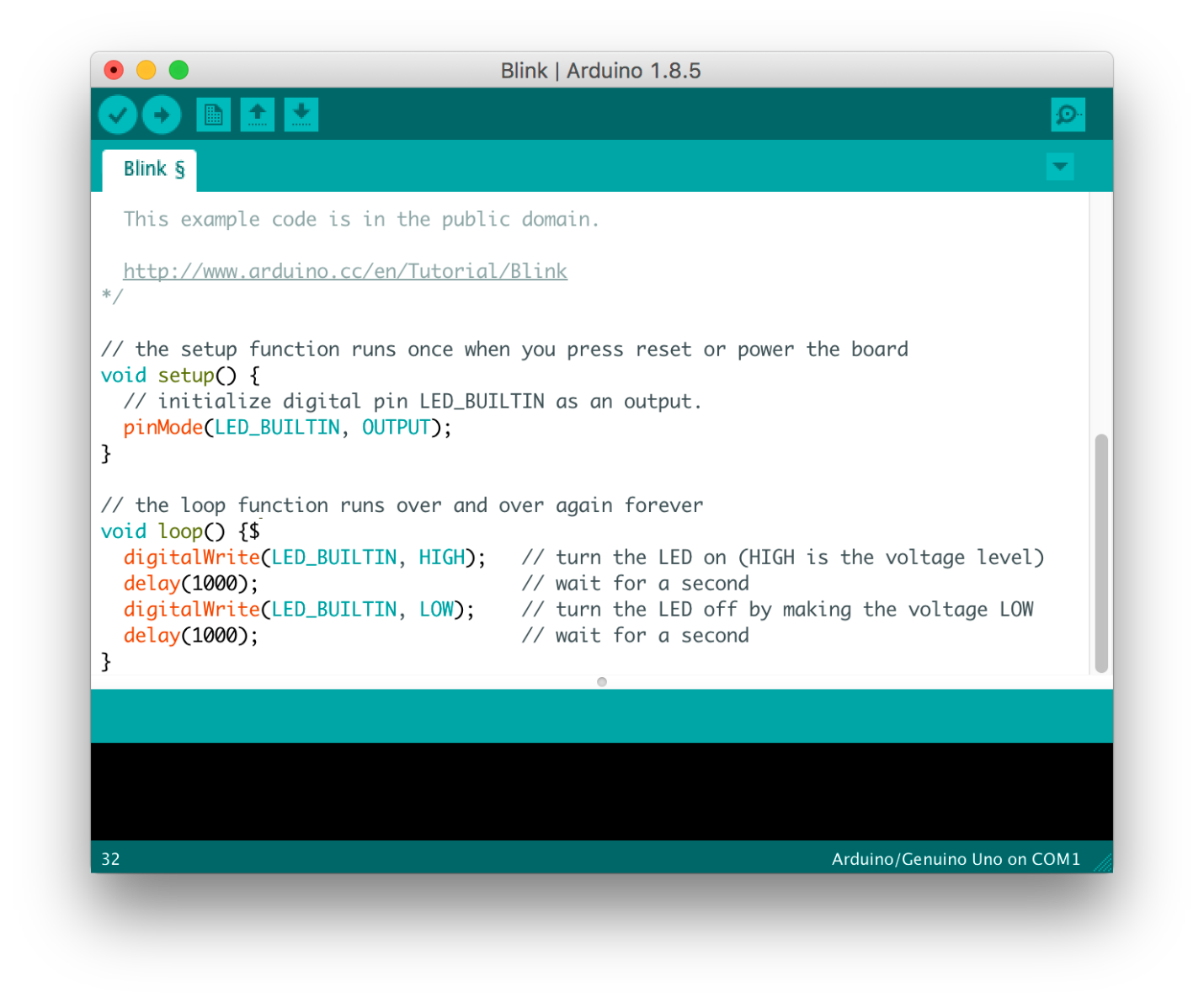
The **Six Buttons** appearing under the Menu tab are connected with the running program as follow.

****

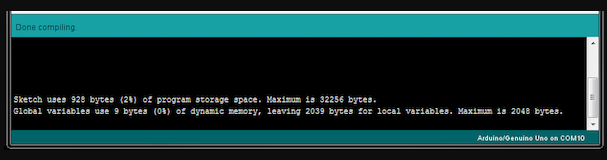
* The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.
* The arrow key will upload and transfer the required code to the Arduino board.
* The dotted paper is used for creating a new file.
* The upward arrow is reserved for opening an existing Arduino project.
* The downward arrow is used to save the current running code.
* The button appearing on the top right corner is a **Serial Monitor** – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
* You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.



The main screen below the Menu bard is known as a simple text editor used for writing the required code.



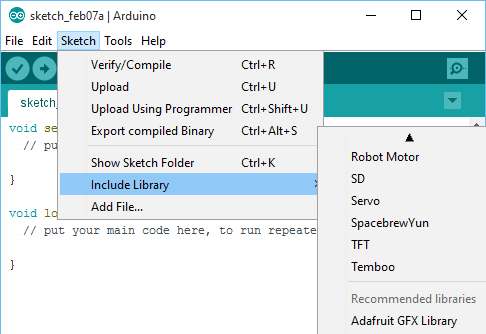
The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.



More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

**Libraries:**

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.



As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.h>.

Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from the external sources.

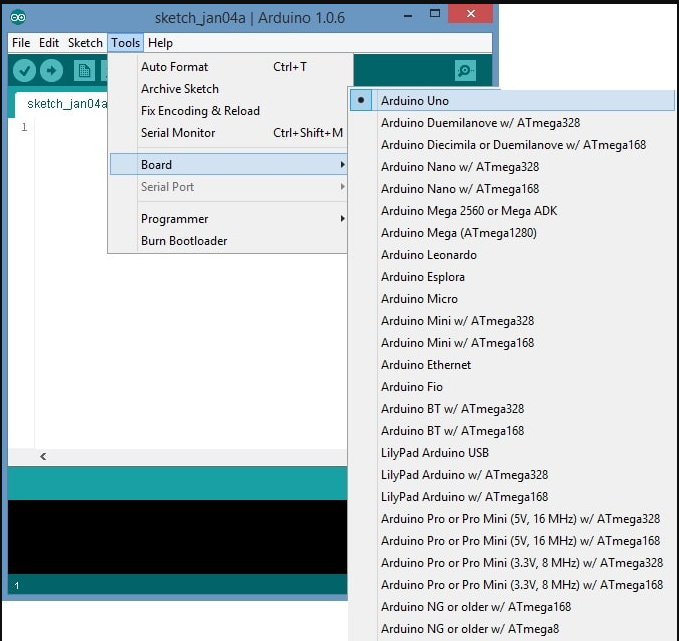
**Making pins Input and output:**

The digitalRead and [digitalWrite](https://www.theengineeringprojects.com/2018/09/how-to-use-digitalwrite-arduino-command.html) commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W”. Writing it down with Digitalwrite or digitalwrite won’t be calling or addressing any function.

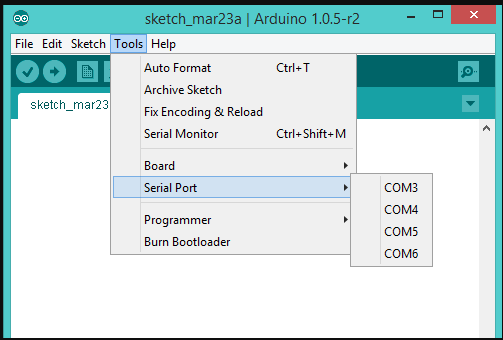
**How to select the board:**

In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the Tools on the Menu, it will open like the figure below.



* Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the ports section of the Windows Device Manager.

Following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with COM4 port at the right bottom corner of the screen.



* After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six button section or you can go to the Sketch section and press verify/compile and then upload.
* The sketch is written in the text editor and is then saved with the file extension .ino.

It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

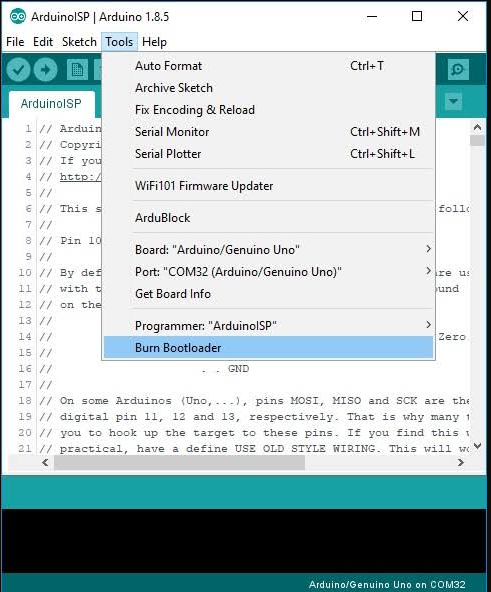
* Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

**Note**: The port selection criteria mentioned above is dedicated for Windows operating system only, you can check this [Guide](https://www.arduino.cc/en/Guide/Environment) if you are using MAC or Linux.

* The amazing thing about this software is that no prior arrangement or bulk of mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment.

**BootLoader:**

As you go to the Tools section, you will find a bootloader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.



When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

**PYTHON:**

Python is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. It is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development. It's syntax and dynamic typing with its interpreted nature, makes it an ideal language for scripting and rapid application development. It supports multiple programming patterns, including object oriented, imperative and functional or procedural programming styles. It is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.It makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

**Python Features**

Python provides lots of features that are listed below.

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

5) Free and Open Source

Python language is freely available at address. The source-code is also available. Therefore it is open source.

6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support

Graphical user interfaces can be developed using Python.

10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

**Python History**

* Python laid its foundation in the late 1980s.
* The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.
* In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
* In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
* Python 2.0 added new features like: list comprehensions, garbage collection system.
* On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language.
* *ABC programming language* is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
* Python is influenced by following programming languages:
  + ABC language.
  + Modula-3

**Python Version**

Python programming language is being updated regularly with new features and supports. There are lots of updations in python versions, started from 1994 to current release.

A list of python versions with its released date is given below.

Python Version Released Date

Python 1.0 January 1994

Python 1.5 December 31, 1997

Python 1.6 September 5, 2000

Python 2.0 October 16, 2000

Python 2.1 April 17, 2001

Python 2.2 December 21, 2001

Python 2.3 July 29, 2003

Python 2.4 November 30, 2004

Python 2.5 September 19, 2006

Python 2.6 October 1, 2008

Python 2.7 July 3, 2010

Python 3.0 December 3, 2008

Python 3.1 June 27, 2009

Python 3.2 February 20, 2011

Python 3.3 September 29, 2012

Python 3.4 March 16, 2014

Python 3.5 September 13, 2015

Python 3.6 December 23, 2016

Python 3.6.4 December 19, 2017

**Python Applications Area**

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Here, we are specifying applications areas where python can be applied.

1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and delelop web based applications. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

2) Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

3) Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

4) Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

5) Business Applications

Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.

6) Console Based Application

We can use Python to develop console based applications. For example: **IPython**.

7) Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

8) 3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

9) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

10) Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc. There are several such applications which can be developed using Python.

**CONCLUSION**

In conclusion, the implementation of "Empowering Women’s Safety with Smart IoT Technology: A Robust Protection System" marks a significant step towards addressing the pressing issue of women's safety and security in various spheres of life. Through the integration of innovative IoT solutions, this project aims to provide women with reliable tools and mechanisms to enhance their personal safety, whether at home, in transit, at work, or in public spaces. By leveraging the power of IoT devices, real-time data analysis, and swift communication channels, the project endeavors to create a safer environment where women feel empowered and supported.

Furthermore, the collaborative nature of this project encourages partnerships between stakeholders, including government agencies, law enforcement authorities, community organizations, and technology developers. By working together, these entities can synergize their efforts to deploy comprehensive safety solutions that address the multifaceted challenges faced by women in different contexts. Through joint initiatives, such as community safety programs, workplace safety protocols, and public awareness campaigns, the project fosters a collective commitment to creating safer and more inclusive environments for women.

Ultimately, the success of this project hinges on its ability to make a tangible difference in the lives of women by reducing the risks they face and providing timely assistance in emergencies. By harnessing the potential of IoT technology to deliver proactive safety measures, rapid response capabilities, and personalized support services, the project seeks to instill greater confidence and peace of mind among women. As we continue to innovate and evolve in the realm of women's safety, it is imperative to prioritize inclusivity, accessibility, and empowerment, ensuring that every woman can lead a life free from fear and insecurity

**REFERENCES**

[1] Shaista Khanam, Trupti Shah, (2019) Self Defence Device with GSM Alert and GPS Tracking with Fingerprint Verification for Women Safety, International Conference on Electronics Communication and Aerospace Technology [ICECA], IEEE.

[2] N. Islam, Md. Anisuzzaman, (2019) Sikder Sunbeam Islam, Mohammed Rabiul Hossain, Abu Jafar Mohammad Obaidullah, Design and Implementation of Women Auspice System by Utilizing GPS and GSM, International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE.

[3] Sharifa Rania Mahmud, Jannatul Maowa, Ferry Wahyu Wibowo, (2017) Women Empowerment: One Stop Solution for Women, 2 nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), IEEE.

[4] Anand Jatti, Madhvi Kannan, Alisha RM, Vijayalakshmi P, Shrestha Sinha, (2016) Design and Development of an IOT Based Wearable Device for The Safety and Security of Women and Girl Children, International Conference on Recent Trends in Electronics Information Communication Technology, IEEE.

[5] Sunil K Punjabi, Suvarna Chaur, Ujwala Ravale, Deepti Reddy, (2018) Smart Intelligent System for Women and Child Security, 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE.

[6] M. Kavitha, V. Sivachidam baranathan, (2018) Women Self-Protecting System Using Internet of Things, International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE.

[7] R. Pavithra, S. Karthikeyan, (2017) Survey on Women’s Safety Mobile App Development, International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE.

[8] Madhura Mahajan, KTV Reddy, Manita Rajput, (2016) Design and Implementation of Rescue System for Safety of Women International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE.

[9] Nandita Viswanath, Naga Vaishnavi Pakyala, G. Muneeswari, (2016) Smart Foot Device for Women Safety, IEEE Region Ten Symposium (TENSYMP), IEEE