CHAPTER-1
INTRODUCTION

In 1954, humans were introduced to the world's first fully functioning industrial robot "The Unimate" and after that, scientists and engineers have come together to create dynamic and diverse changes in the field of automation and robotics to make the daily humane tasks easier and faster. The use of robots in development and automation fields is increasing day by day and there is no doubt about the future being largely controlled by robots and artificial intelligence (AI).

The Surveillance System closely observes and analyzes the surrounding and get instant information about the conditions. It is mainly required in areas of high risk, borders, public places, and prison or in industries which is mainly used for monitoring behavior and activities of a group or any individual. The need of surveillance robots arises when the life risk is too high and the user wants the information to be highly accurate. Robots are nothing but fully automated electronic and internet-controlled devices that are capable of performing various tasks that a normal human might not be able to do. Thus, use of robots for surveillance is one of the greatest advancements in the field of automation. These multifunctional robots are able to perform tasks in dangerous situations like collapsing buildings or radioactive zones. One of its best uses is in the protection and rescue works after unexpected tragedy or unwanted invasions like Ukraine-Russia Cold war or tragedies like Chernobyl/Bhopal Gas Plant. There are many obstacles faced by the rescue forces during inspection of such sudden and unexpected events like narrow spacing, collapsing of damaged structures. It becomes difficult for an ordinary human to deal with such risky tasks to enter areas without knowing the present information. These robots being autonomous in nature are designed to perform efficiently without human interference and have high mobility.

Back in 1999, Kevin Ashton introduced the term 'Internet of Things' to the world in one of his presentations. IoT connected people with everything on the internet from anywhere around the world and since then the definition of IoT has evolved and growth has rapidly increased.

This IoT Based Autonomous multi-purpose surveillance and rescue robot is built on mainly two systems. First, the motorized working of the robot with all the connections and second, the communication of the device with the user and smooth data transfer from the sensors to the cloud platform. These systems help in carrying out task properly. The main aim of this project is to combine the two different systems into one machine that would make them work simultaneously and perform the required tasks. To achieve this aim, an IoT based monitoring system is also included with the robot which can be used to monitor by the user through their device.

1.1 PROPOSED SYSTEM

The robot consists of Arduino UNO micro controller which acts as heart piece of the robot. This robot also consists of DC motors, wheel chassis, battery, Wi-Fi module and various types of sensors such as camera for monitoring. Surveillance is essential for various sectors such as security, monitoring, and automation. Traditional surveillance systems have limitations in terms of mobility and flexibility. The proposed system aims to address these limitations by introducing an IoT-based surveillance robot.

Objectives:

- Aim is to design a robot to replace humans in dangerous areas
- Long range and night vision camera for clear night monitoring
- Using WIFI module with Arduino make better connection
- Develop a surveillance robot capable of remote monitoring and control.
- Integrate IoT technologies for real-time data transmission and analysis.
- Enable autonomous navigation and obstacle avoidance.
- Implement advanced features such as object detection and recognition.

1.2 METHODOLOGY

Methodology for an IoT based surveillance robot:

- **Hardware setup:** Begin by gathering the necessary components for surveillance robot, including an Arduino uno board, a camera module, wheels, and another peripheral. connect and assemble them according to the provided instructions.

- **Software installation:** Install the Arduino IDE on computer and connect the Arduino uno board to the computer. write and upload the necessary code to control the motors, camera and other functionalities of the robot.

- **Camera integration:** connect the camera module to the Arduino uno board and configure it to capture video footage. Adjust the camera settings to achieve the desired field of view and image quality.
- **IoT Connectivity:** Configure the communication protocols (MQTT, HTTP, WebSocket) for transmitting sensor data and receiving commands from remote devices. Implement security measures (encryption, authentication) to protect data transmission and access to the robots control interface. Set up a cloud-based IoT platform or server for data storage, management, and remote monitoring.
- **Remote control:** By integrating with cloud services, users can remotely access control the surveillance robot from anywhere viewing live videos.

- **Data transmission:** Implement bidirectional data transmission between the robot and the remote-control device. This includes sending control commands (e.g., move forward, turn left, stop) from the user interface to the robot, as well as transmitting sensor data (e.g., camera feed, environmental data) back to the user interface for monitoring.
- **Cloud Integration:** The robot collects various types of data from its sensors, including video streams from cameras, environmental data, and positional information. Data is collected continuously as the robot moves and performs its surveillance tasks. the collected data is transmitted to the cloud using internet connectivity. cloud-based services are used to process and analyse the collected data. the results

of data processing and analysis can be used to generate insights, trigger alerts or notifications, and automate certain actions based on predefined rules or thresholds.

- **Power management:** Develop power management systems to optimize battery life. Implement features such as sleep modes, low-power sensors, and efficient motor control algorithms.
- **Testing and Optimization:** Test the remote-control functionality in various scenarios to ensure reliability and responsiveness. This may involve simulated testing as well as real-world trails in different environments. Optimise the remote-control system for latency, bandwidth efficiency, and overall user experience, considering factors such as network condition and device capabilities.

CHAPTER-2
LITERATURE REVIEW

Table 1. Literature Review of IoT Based Surveillance and Rescue Robot using Various existing solutions.

Year	Title	Proposed technique	Limitations
2016 [6]	Design and Implementation of IoT-based Intelligent Surveillance Robot	-SMART-I a mobile robot and moved on fixed line tracing - Real time Video Transmission -Smartphone App	Limited Battery Backup and No Night Vision Camera
2017 [7]	Military Robot for Reconnaissance and Surveillance using Image Processing	-Robot for detecting land mines and can move on any terrain - Face recognition as per the database -Updates a new person after taking 20 pictures - Gas leak, Radiation, Heat Detection	No Battery Backup, Connected to Vulnerable Cloud
2017 [8]	Autonomous Surveillance Robots	-Robots for human assistance - gesture sensing like waving the camera for assistance	Decision making and the robot can't cover large areas Irregular sensor data
2018 [9]	IOT-Based Wi-Fi Surveillance Robot with Real-Time Audio and Video Streaming	-Smartphone app with easy UI - PIR sensors along with gas sensors, night vision camera instead of IP cam	Using third party app may create hindrance with the security concerns like IP cam breach
2018 [10]	Smart Surveillance Robot for Real-Time Monitoring and Control System Monitoring and Control System	-Smartphone app -PIR/IR and Night Vision for patrolling	Only related to data collection about environmental aspects
2020 [11]	IoT Based Smart Multi Application Surveillance Robot	Instead of cayenne use cloud platform self-made UI to customize another feature also	Limited data storage up to 64gb Laser gun to be replaced with much more powerful weapon

CHAPTER-3
INTERNET OF THINGS

3.1 INTRODUCTION TO INTERNET OF THINGS

IoT, or the Internet of Things, refers to a network of interconnected physical devices, vehicles, appliances, and other objects embedded with sensors, actuators, software, and connectivity capabilities that enable them to collect, exchange, and act on data. The fundamental idea behind IoT is to create a vast ecosystem where everyday objects can communicate with each other, share information, and make intelligent decisions without human intervention.

The Internet of Things (IoT) refers to the interconnection of everyday objects, devices, machines to the internet, allowing them to collect, exchange, and share data. This interconnected network enables devices to communicate and collaborate seamlessly, leading to increased efficiency, automation, and convenience in various industries and sectors. IoT devices can include everything from household appliances like smart thermostats and refrigerators to industrial machinery, vehicles, and wearable devices. These devices are equipped with sensors, actuators, and software that enable them to gather and transmit data, often in real-time. This data can then be analyzed and used to make informed decisions, improve processes, and enhance user experiences.

The IoT has numerous applications, such as smart cities, where sensors monitor traffic, energy usage, and waste management to improve urban planning and sustainability. In healthcare, IoT devices can track patients' vital signs remotely and alert healthcare providers of any abnormalities. In agriculture, sensors can monitor soil conditions and weather patterns to optimize crop yield and reduce water usage.

Despite its many benefits, the IoT also raises concerns about privacy, security, and data breaches, as the sheer number of connected devices increases the potential attack surface for hackers. As the IoT continues to evolve, it has the potential to revolutionize how we interact with technology and the world around .

3.2CHARACTERISTICS OF IoT(INTERNET OF THINGS)

1. **Interconnectivity:** Devices and systems are connected to the internet and to each other enabling communication and data exchange.
2. **Sensors and Actuators:** IoT devices often include sensors to collect data from their environment and actuators to perform actions based on that data.
3. **Data Collection:** IoT devices continuously gather and transmit data, which can be used for monitoring, analysis, and decision-making.
4. **Communication Protocols:** IoT relies on various communication protocols to facilitate data exchange between devices.
5. **Scalability:** IoT systems are designed to scale up, accommodating an increasing number of devices and data volume.
6. **Real-Time Analytics:** IoT systems can analyze data in real-time, allowing for immediate insights and responses.
7. **Security and Privacy:** Ensuring the society and privacy of data isa critical aspect of IoT, given the potential for vulnerabilities and data breaches.
8. **Energy Efficient:** Many IoT devices are designed to be energy efficient to prolong battery life and reduce energy consumption.
9. **Self-Configuration:** Some IoT devices have the capability to self-configure and adapt to new environments or changes in their network.
10. **Autonomy:** IoT devices can operate independently, making decisions based on data and programming.

3.3 APPLICATIONS OF IoT(INTERNET OF THINGS)

The Internet of Things (IoT) refers to a network of interconnected physical devices that can collect, exchange, and act on data. These devices, often embedded with sensors, software, and other technologies, range from everyday household items to sophisticated industrial tools. IoT enables these devices to communicate with each other and with other internet-enabled systems, creating a vast network of smart objects. The Internet of Things (IoT) has many applications in both the public and private sectors, including:

- **Smart Home:** IoT is used in smart homes to control devices such as lights, shower and kitchen.
- **Healthcare:** IoT is used in healthcare for point-of-care diagnostics, remote monitoring, and managing chronic diseases.
- **Agriculture:** IoT is used in agriculture to monitor soil moisture and nutrients, and to streamline crop management processes.
- **Smart Cities:** IoT can be used to improve urban planning and infrastructure maintenance, such as measuring air quality, detecting maintenance needs, and managing parking.
- **Manufacturing:** IoT is used in manufacturing to monitor machines and products in real time to improve efficiency and quality.
- **Supply Chain:** IoT is used in supply chain management to monitor inventory levels and track customer’s spending habits.
- **Energy:** IoT is used in smart grids to switch between different power sources, such as traditional power plants and solar and wind power plants.
- **Smart door access:** IoT is used in smart door access systems to control doors using a web interface or smartphone application.
- **Vehicle Tracking:** IoT is used in tracking of vehicle location, speed, and performance.
- **Environmental Monitoring:** IoT is used to track air, water, and soil quality, and monitor weather patterns.

CHAPTER-4
CONSTRUCTION

4.1 BLOCK DIAGRAM

The main components that the proposed system consists of are Arduino uno, Wi-Fi camera module, DC motor, LCD, Wheels, Power Supply, L293D motor driver.

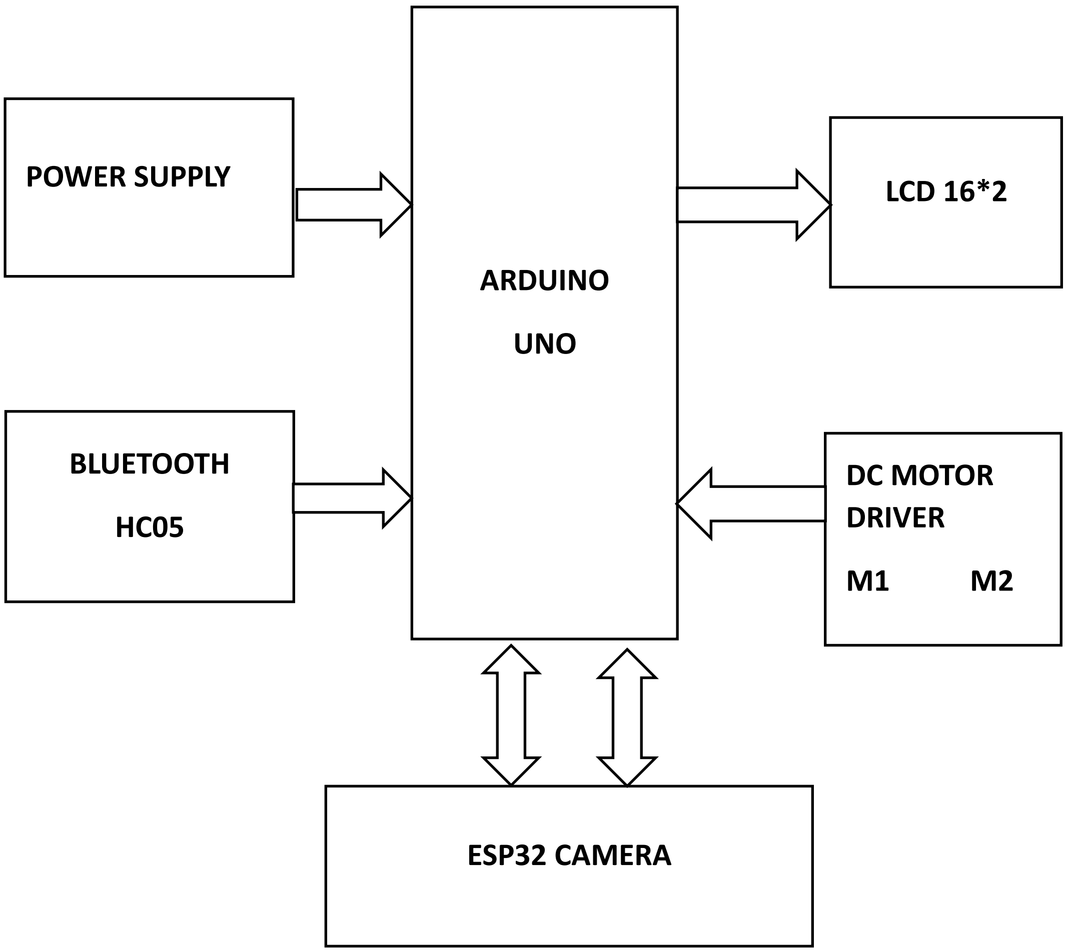


Figure 4.1: Block Diagram of IoT based surveillance robot

The system consists of two major sections – one is the user control section and the other one is the mobility section. The robot has an Arduino uno microcontroller which acts as the brain of system as it is used for controlling the motors and movement of the robot. In the user section, with the help of a mobile or a laptop user control is much better as compared to the old versions where a big control instructed panel was required to make the robot move and perform tasks. The communication from the user end to the robotic mobile section can be performed in various ways, using a Zigbee device or any Bluetooth controlled technology, but in these the range of communication is limited. So, in order to make the robot remotely controlled from a far distance, user Section can be connected with the internet using the IoT technology. For connecting the user with the robot from far distance an IoT based technology is used .

The Wi-Fi Camera Module is connected to the Arduino Board that further provides the Wi-Fi facility to the Robot. After the full connection, the robot can send the data collected from the sensors using the Wi-Fi modules and the IoT platform to the users.

In mobility part, the robot consists of wheels, motors, battery, Wi-Fi Module and various different sensors. The Arduino board is placed on the body of robot which gives input supplies to the DC Motors and the motors are further connected to the wheels which move according to the user instructions. The Arduino microcontroller is coded through its IDE software where the code is defined for appropriate movement.

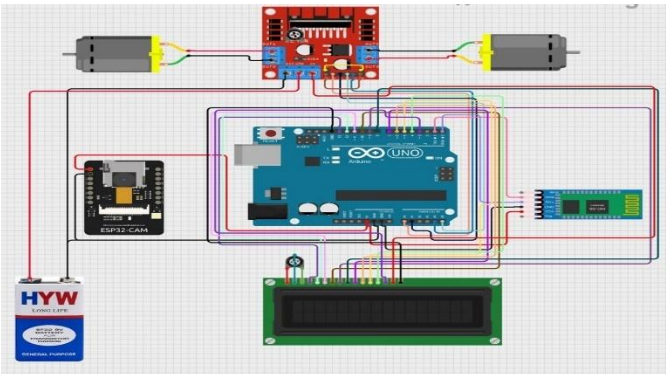


Figure 4.2 Circuit diagram of IoT Based Surveillance Robot

CHAPTER-5
HARDWARE COMPONENTS

5.1 LIQUID CRYSTAL DISPLAY

A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to a controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display.

They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS, RW, D7, D6, D5, D4, D3, D2, D1, D0).

For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS, RW, D7, D6, D5, D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Figure 5.1 16x2 LCD Display

5.1.1 USES

The LCD s used exclusively in watches, calculators and measuring instruments is the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCD s being extensively used in telecommunications and entertainment electronics. The LCD s has even started replacing the cathode ray tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

5.1.2 LCD PIN DIAGRAM

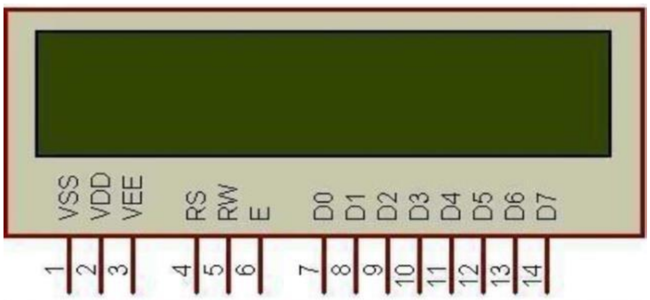


Figure 5.2 Pin Diagram of LCD

5.1.3 CONTROL LINES

EN:

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your 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.

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.

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.

- **CONTRAST CONTROL**

To have a clear view of the characters on the LCD, contrast should be adjusted. To adjust the contrast, the voltage should be varied. For this, a preset is used which can behave like a variable voltage device. As the voltage of this preset is varied, the contrast of the LCD can be adjusted.

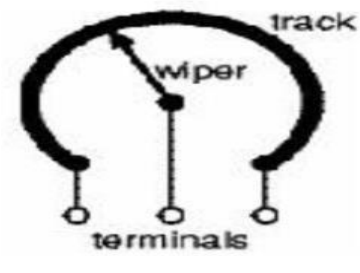


Figure.5.3 variable resistor

- **POTENTIOMETER**

Variable resistors used as potentiometers have all three terminals connected. This arrangement is normally used to vary voltage, for example to set the switching point of a circuit with a sensor, or control the volume (loudness) in an amplifier circuit. If the terminals at the ends of the track are connected across the power supply, then the wiper terminal will provide a voltage which can be varied from zero up to the maximum of the supply.

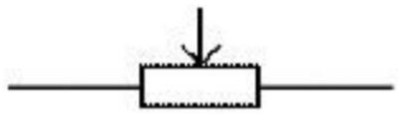


Figure 5.4 Potentiometer Symbol

5.2 ARDUINO UNO

5.2.1 INTRODUCTION:

Microcontroller as the name suggest, a small controller. They are like single chip computers that are often embedded into other systems to function as processing/controlling unit. For example, the control you are using probably has microcontrollers inside that do decoding and other controlling functions. They are also used in automobiles, washing machines, microwaves ovens, toy set, where automation is needed.

5.2.2 ARDUINO UNO MICROCONTROLLER:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5Vpin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts The power pins are as follows:

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

Input and Output:

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor of 20-50 k Ohms. In addition, some pins have specialized functions:

- **Serial**: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a

change in value. See the attach Interrupt () function for details.

- **PWM: 3, 5, 6, 9, 10, and11.** Provide 8-bit PWM output with analog write () function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCLK).** These pins support SPI communication, which although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED**: **13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- The Uno has 6 analog inputs, each of which provide 10 bits of resolution.
- **I2C: 4(SDA) and 5(SCL).** Support I2C(TWI) communication using the Wire library. There are a couple of other pins on the board.
- **AREF:** Reference voltage for the analog inputs. Used with analogy Reference ().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

5.2.3 ARDUINO UNO BOARD:

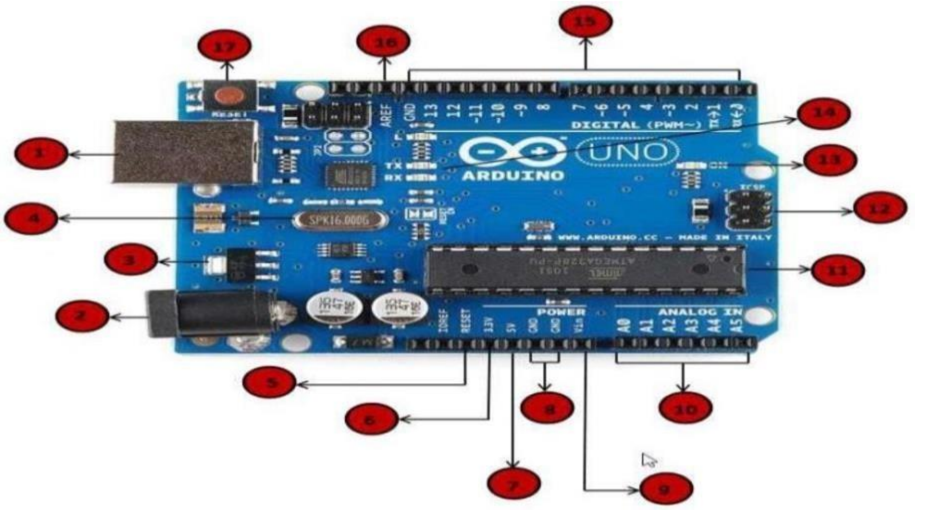


Figure 5.5 Arduino UNO Board

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Technical specifications:

FEATURE	SPECIFICATION
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 5.1: Arduino Uno specifications

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

PIN DIAGRAM OF ARDUINO UNO:

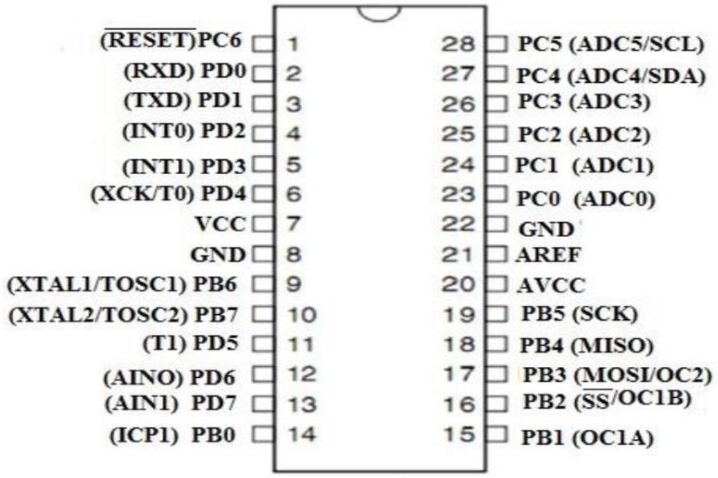


Figure 5.6. Pin diagram

5.3 ESP32 CAMERA

The integration of a Wi-Fi camera in a surveillance robot allows for real-time video streaming, remote monitoring, and control. Below is an overview of how a Wi-fi camera works with in a surveillance robot:

1. **Hardware Setup:**

- The Wi-Fi camera is mounted on the surveillance robot's chassis or body in a position that provides a clear view of the surroundings.
- The camera is connected to the robot’s onboard microcontroller (e.g., Arduino, or computer system)

2. **Power on initialization:**

- Upon powering on the surveillance robot, the Wi-Fi camera will initialize itself and establishes a connection to local Wi-Fi network.
- The camera requires authentication credentials (username & password) to connect to the network.

3. **Capture and Processing:**

- The camera continuously captures videos of the surroundings**.** It is used to analyzed to perform tasks like encoding and streaming.

4. **Wi-Fi Transmission:**

- ➢ Processed video frames are transmitted over the Wi-Fi network in real time.
- ➢ The camera utilizes the Wi-fi protocol (e.g., IEEE 802.11) to establish a wireless connection with the network router/access point.
- ➢ Video data is transmitted using TCP/IP protocols over the Wi-Fi connection.

5. **Remote Access and control:**

- ➢ Users can access the live video stream from the surveillance robot remotely using a compatible device (e.g., smartphone, tablet, computer).

6. **Security and Encryption:**

- ➢ To ensure the security of the video stream, encryption techniques such as WPA/WPA2 (Wi-Fi Protected Access) are employed for securing the Wi-Fi network.
- ➢ Additionally, HTTPS (Hypertext Transfer Protocol Secure) or other encryption protocols may be used for transmitting video data over the network.

7. **Latency and quality considerations:**

- ➢ The latency of the video stream may vary depending on factors such as network congestion, signal strength, and processing capabilities of the camera and microcontroller/computer.
- ➢ To maintain a smooth and high-quality video stream, optimizations such as frame rate adjustment, video compression, and adaptive bitrate streaming may be implemented.

- ➢ By incorporating a Wi-Fi camera into a surveillance robot, users can remotely monitor and control their surroundings in real-time, enhancing security, surveillance, and situational awareness.

ADVANTAGES

- **Wireless connectivity:** The most obvious advantage is the absence of wires. This allows for flexible installation options without the need to run cables, making it easier to set up and move the camera as needed.

- **Remote access:** Wi-Fi cameras can be accessed remotely via the internet, allowing users to view live feeds and recordings from anywhere with an internet connection. This is particularly useful for surveillance purposes or checking in on pets or loved ones while away from home.

- **Easy installations:** Since there are no cables to run, installation is typically simpler and quicker compared to wired cameras. This makes Wi-Fi cameras ideal for DIY installations.

- **Scalability:** Wi-Fi cameras can easily be added to an existing network without the need for additional infrastructure, making them scalable to accommodate changing needs or expanding surveillance coverage.
- **Flexibility:** Wi-Fi cameras can be placed virtually anywhere within the range of the Wi-Fi network, providing flexibility in positioning to capture the desired area or angle.

- **Cost Effectiveness:** In many cases, Wi-Fi cameras can be more cost-effective than wired alternatives, especially when considering installation costs associated with running cables.

- **Cloud-storage options:** Many Wi-Fi cameras offer cloud storage options for recorded footage, eliminating the need for on-site storage devices and providing convenient access to recordings from anywhere.

Figure 5.7 Camera module

5.4 BLUETOOTH

The Bluetooth HC-05 is a popular Bluetooth module used for wireless communication between devices. The module can be used in master or slave configuration.

HC-05 has red LED which indicates the connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds. This module works on 3.3V. It can connect 5V supply voltage as well since the module has on board 5 to 3.3V regulator.

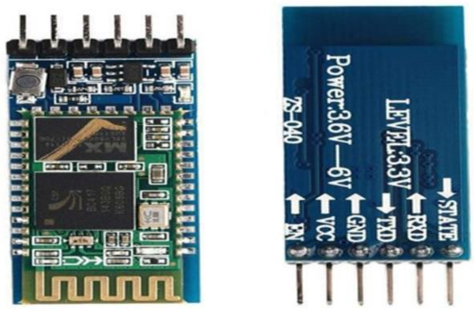


Figure 5.8: Bluetooth Module

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

It has 6 pins,

1. Key/EN: It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

1. Data mode: Exchange of data between devices.
2. Command mode: It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

2. VCC: Connect 5 V or 3.3 V to this Pin.

3. GND: Ground Pin of module.

4. TXD: Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)

5. RXD: Receive data serially (received data will be transmitted wirelessly by Bluetooth module).

6. **State:** It tells whether module is connected or not.

5.4.1 SPECIFICATIONS

- Bluetooth Version: 2.0 + EDR (Enhanced Data Rate)
- Frequency: 2.4 GHz
- Transmission Power: 4dBm
- Receiving Sensitivity: -80dBm
- Data Transfer Rate: Up to 2Mbps
- Transmission Range: Up to 30 meters
- Power Supply: 3.3V to 5V
- Current Consumption: Around 30mA
- Interface: UART (Universal Asynchronous Receiver Transmitter)

5.5 DC MOTOR

A DC Motor is an electrical machine that converts direct current (DC) electrical energy into mechanical energy.

Basic components:

Stator: Stationary part of the motor.

Rotor: Rotating part of the mirror.

Armature: Coil of wire wrapped around the rotor.

Commutator: Switch that reserves current flow.

Figure 5.9 DC Motor

5.6 MOTOR DRIVER

Motor driver L293D isa dual H-bridge motor driver integrated circuit. Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H-bridge driver circuits. Motor drivers are crucial for precise control over motor operation, allowing for smooth and efficient movement in various applications.

They act as a bridge between the control system and the motor, providing the necessary power and control signals to regulate the motor's speed, direction, and torque. The speed of the motor controlled by Enable 2 pin using analog write () function. While the direction of the motor is controlled by In 3 and In 4 pins.

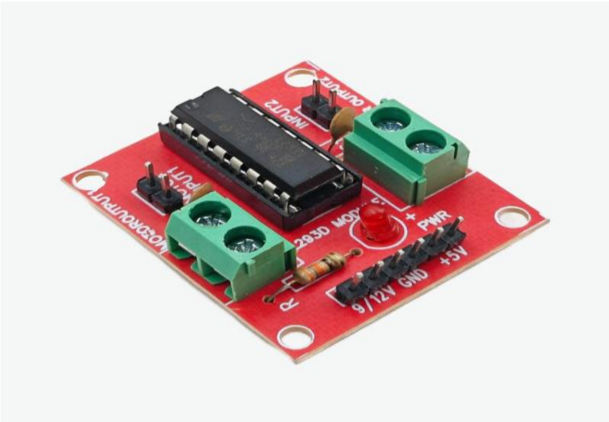


Figure 5.10 Motor Driver

SPECIFICATIONS:

- Operating Voltage: 4.5v to 36v
- Power Dissipation: 4W
- Output Current: 1A per channel
- Number of Channels: 4
- Switching Frequency: Up to 100KHz
- Package Type: 16-pin package
- Control Interface: Logic-level inputs

5.7 POWER SUPPLY UNIT

The power supply for this system is shown below.

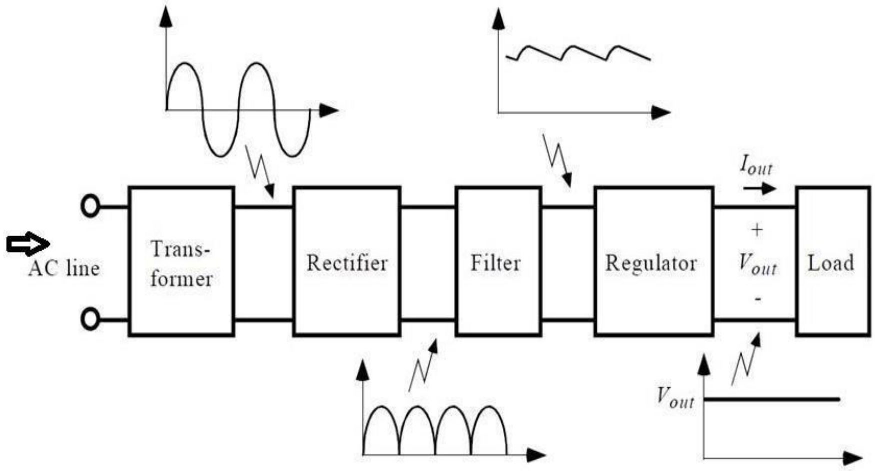


Figure 5.11. Power supply

In this project we have power supplies with +5V & -5V option normally +5V is enough for total circuit. Another (-5V) supply is used in case of OP amp circuit.

Transformer primary side has 230/50HZ AC voltage whereas at the secondary winding the voltage is step downed to 12/50 Hz and this voltage is rectified using two full wave rectifiers the rectified output is given to a filter circuit to fitter the unwanted ac in the signal. After that the output is again applied to a regulator LM7805 (+5v) regulator. WhereasLM7905 is for providing –5Vregulation. Z (+12V circuit is used for stepper motors, Fanand Relay by using LM7812 regulator same process like above supplies).

CHAPTER-6

SOFTWARE IMPLEMENTATION

6.1 INTRODUCTION TO ARDUINO IDE

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

Key Features:

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

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

Arduino Data Types:

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

Example: void, Boolean, char, Unsigned char, Byte, Unsigned int, Word, Long, Float, Double.

Example:

```
Void Loop ()
{
// rest of the code
}
```

Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Figure 6.1: USB Cable

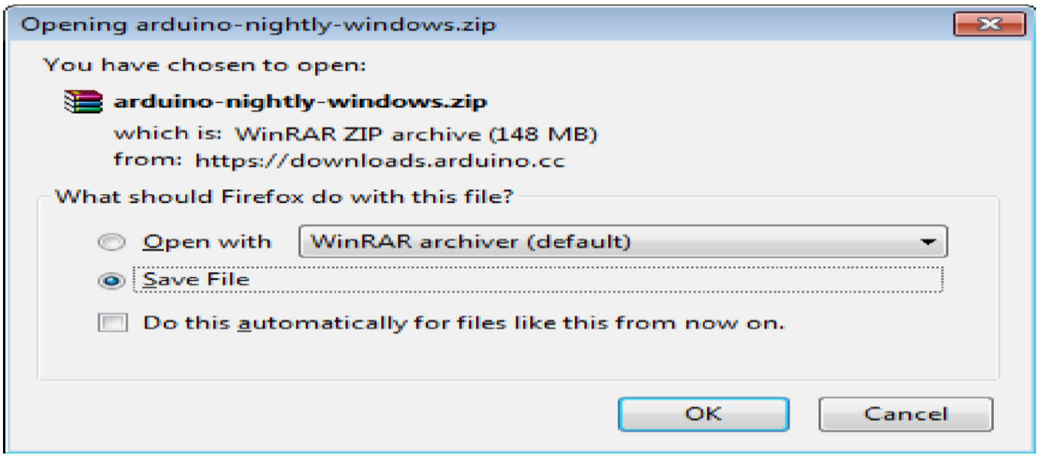
Step 2: Download Arduino IDE Software.

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

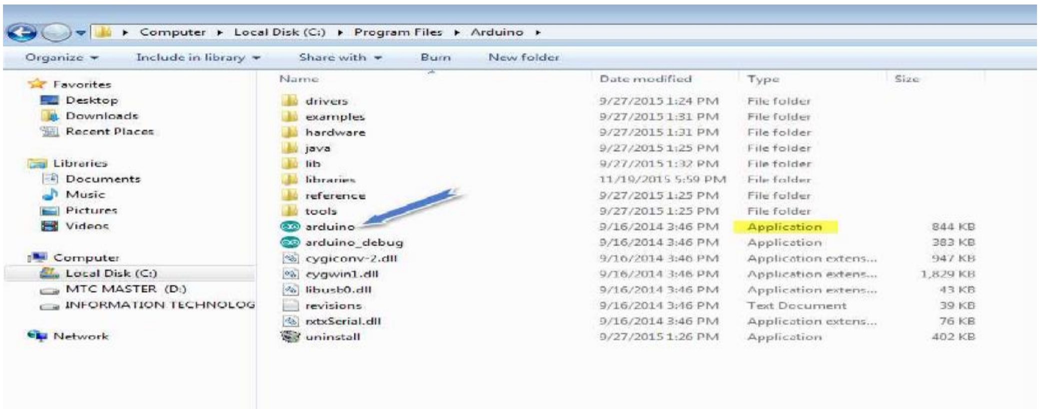
Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4: Launch Arduino IDE.



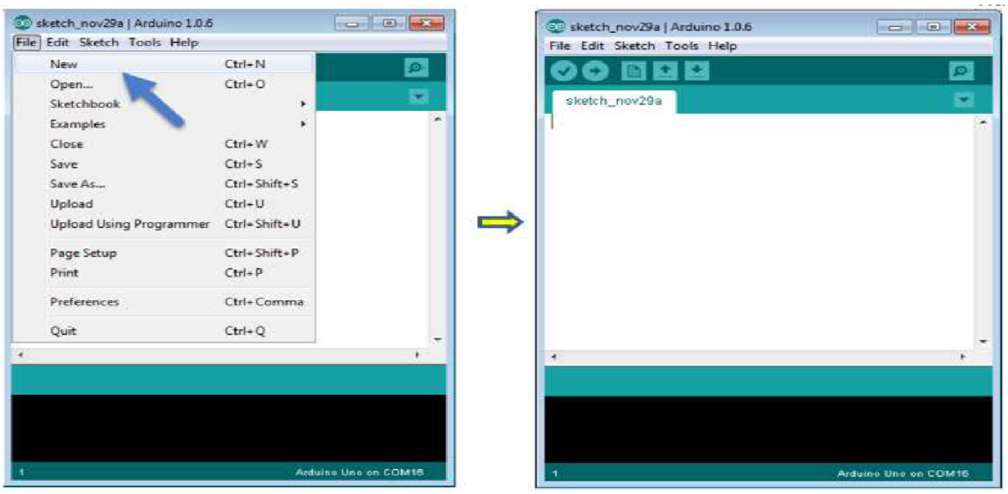
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double click the icon to start the IDE.

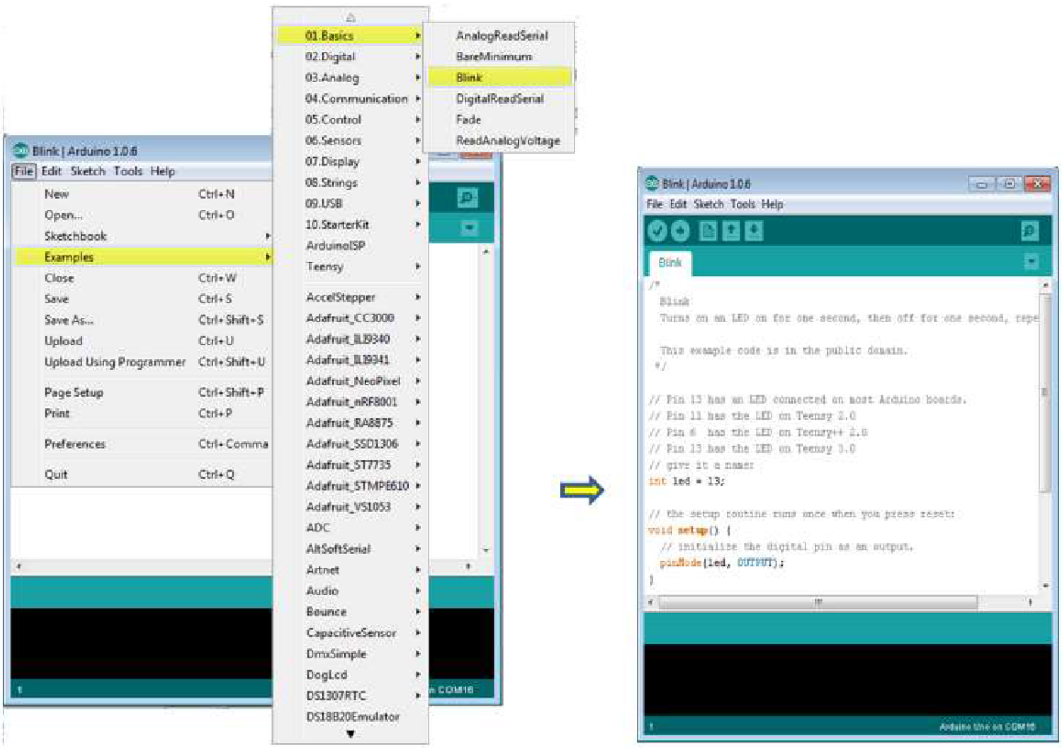


Step 5: Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example
- To create a new project, select File --> New. To open existing project example, select File -> Example -> Basics -> Blink.

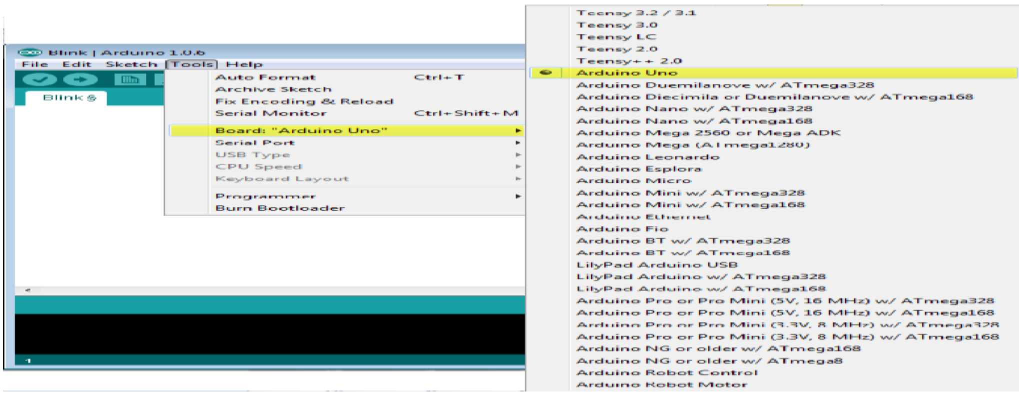




Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with sometime delay. You can select any other example from the list.

Step 6: Select your Arduino board.

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

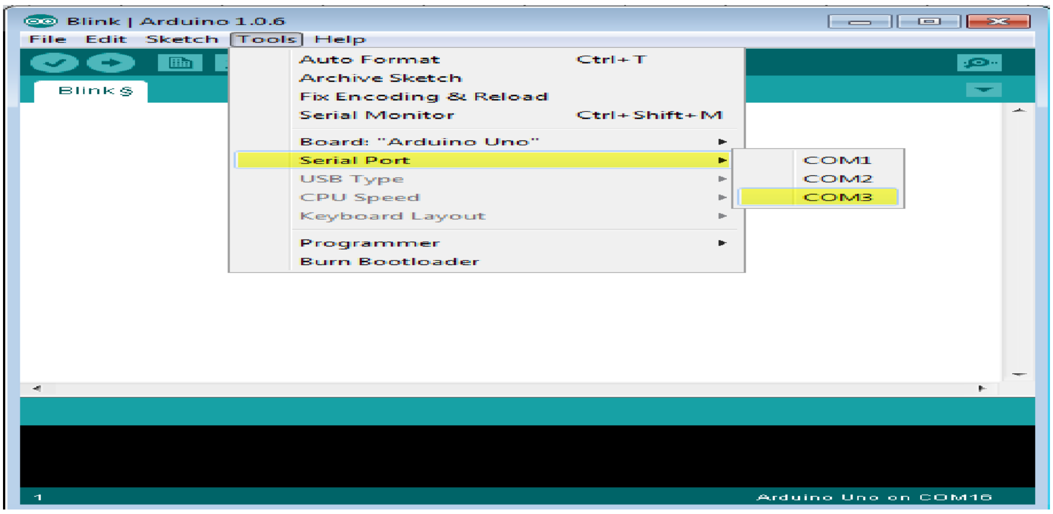


Go to Tools -> Board and select your board

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

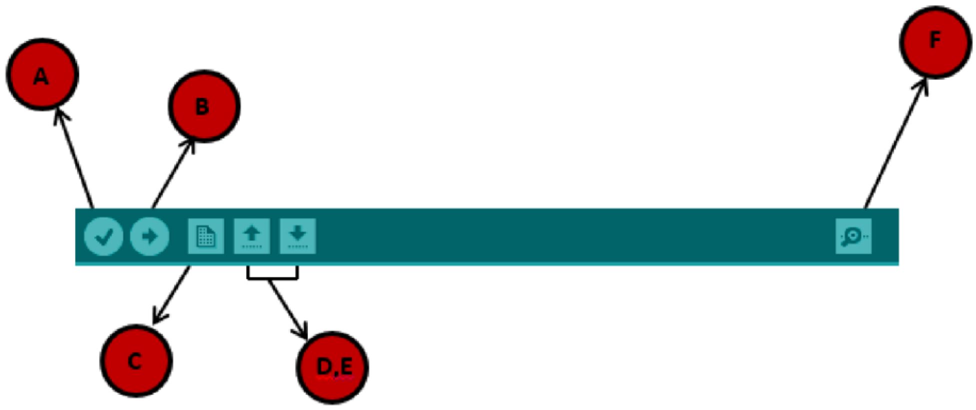
Step 7: Select your serial port.

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



Step 8: Upload the program to your board.

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



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

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

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketches.

E- Used to save your sketch.

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

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

Note: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Arduino programming structure

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

Sketch: The first new terminology is the Arduino program called "**sketch**".

Structure: Arduino programs can be divided in three main parts: **Structure, Values** (variables and constants), and **Functions**.

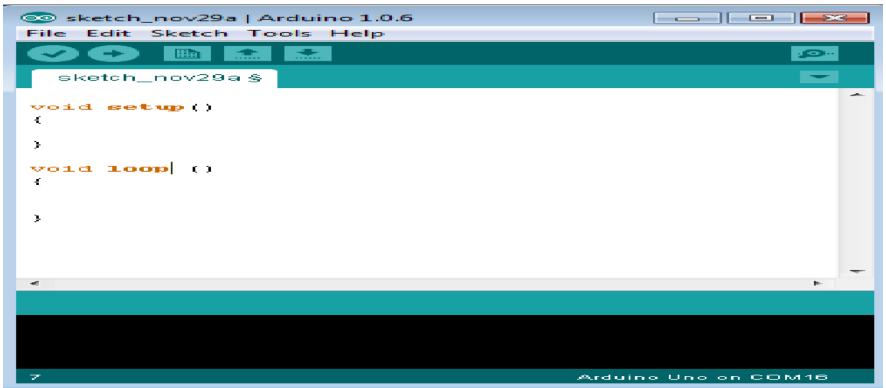
Let us start with the **Structure**. Software structure consists of two main functions:

- Setup () function
- Loop () function

Void setup ()

{

}



PURPOSE:

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

INPUT

OUTPUT

RETURN

Void Loop ()

{

}

CHAPTER-7

WORKING PRINCIPLE

An IoT-based surveillance robot is a robotic system that incorporates Internet of Things (IoT) technology to enhance surveillance capabilities. The primary purpose of such a robot is to monitor and gather information about its surroundings, providing real- time data to users who can remotely control and monitor the robot.

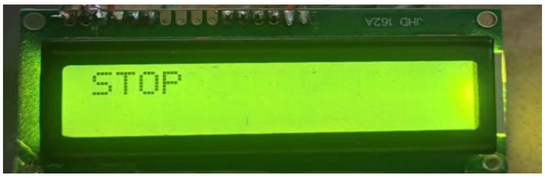
The working principle of an IoT-based surveillance robot involves several key components and functionalities.

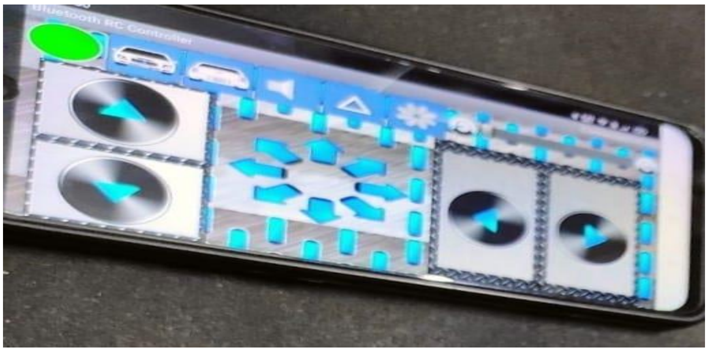
1. **Connectivity:** The robot is equipped with communication modules, such as Wi-Fi or cellular connectivity, allowing it to establish a connection to the internet. This connectivity enables seamless communication between the robot and the control system.
2. **Sensors:** The robot is equipped with a sensor to capture and collect data about its environment. The sensor may include camera. The data collected by the sensor is crucial for surveillance purposes.
3. **Data Acquisition:** The sensor on the robot continuously gather data from the environment. For example, the camera captures live video feeds. All this data is processed and transmitted to the control system.
4. **Data Transmission:** The collected data is transmitted from the robot to the control system through the internet connection. This transmission can occur in real-time, ensuring that the control system receives up-to-date information about the robot's surroundings.
5. **Control System:** The control system serves as the central hub for monitoring and controlling the surveillance robot. It can be accessed through a smartphone, computer, or any other device with internet connectivity. The control system provides a user interface where users can view the live video feed from the robot's cameras and control its movements.

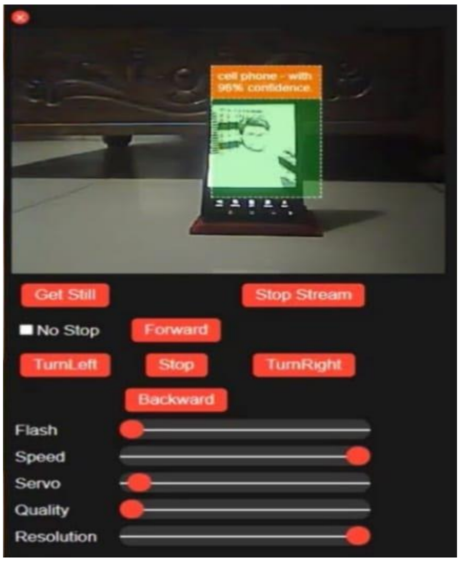
6. **Remote Control:** Users can remotely control the surveillance robot through the control system. They can command the robot to move in different directions. This remote-control capability allows users to navigate the robot to specific areas of interest and gather more detailed surveillance information.
7. **Data Analysis and Storage:** The control system can analyse and process the collected data from the robot's sensor. The processed data can be stored for future reference or used to trigger alerts or notifications based on predefined criteria.
8. **User Interaction:** Users can interact with the control system to receive real-time updates, set surveillance schedules, configure alert.

CHAPTER-8

RESULTS







CHAPTER-9
ADVANTAGES DISADVANTAGES & APPLICATIONS

9.1 ADVANTAGES

- **Enhanced Monitoring:** The IoT-based surveillance robot allows for real-time monitoring of an area, providing a constant stream of video footage and sensor data. This enables remote surveillance and monitoring, enhancing security and situational awareness.
- **Flexibility and Mobility:** The robot's compact size and mobility enable it to navigate various terrains and access hard-to-reach areas, providing surveillance in locations where fixed cameras may not be feasible. It can be deployed in indoor and outdoor environments, making it versatile for different surveillance needs.
- **Remote Control and Accessibility:** With IoT connectivity, the surveillance robot can be controlled and accessed remotely from anywhere using a mobile app or web interface. This allows users to monitor and control the robot's movements, camera angles, and other functionalities conveniently.
- **Energy Efficiency:** With power management techniques, the IoT-based robot can optimize energy usage, prolonging battery life and reducing power consumption. This ensures longer operation times and reduces the need for frequent battery replacements or recharging.

9.2 DISADVANTAGES

- **Security Risks:** Vulnerable to hacking and data breaches, compromising sensitive information.
- **Dependence on Internet:** Requires a stable internet connection, which can be unreliable in certain areas.

- **Power Consumption:** IoT devices can consume significant power, leading to increased energy costs and environmental impact.
- **Limited Battery Life:** Battery-powered robots may have limited operating time, requiring frequent charging.
- **High Cost:** IoT-based surveillance robots can be expensive, making them inaccessible to some individuals or organizations.

9.3 APPLICATIONS

- **Home Security:** The surveillance robot can be used to monitor and secure homes, providing real-time video footage and alerts in case of any suspicious activities or breaches. It allows homeowners to remotely monitor their property.
- **Industrial Surveillance:** In industrial settings, the robot can be deployed to monitor manufacturing facilities, warehouses, or construction sites. It can provide real-time video feed and sensor data to ensure safety, detect anomalies, and prevent unauthorized access.
- **Agricultural Monitoring:** The surveillance robot can be used in agriculture for monitoring crop fields, greenhouses, or livestock. It can help farmers detect pests, monitor irrigation systems, and ensure the well-being of their crops or animals.
- **Public Safety:** The robot can be utilized by law enforcement agencies or security personnel for surveillance in public spaces, such as parks, stadiums, or crowded events. It can help monitor for potential threats, crowd management, and ensure public safety.
- **Disaster Management:** During natural disasters or emergency situations, the robot can be deployed for surveillance and assessment purposes. It can help in search and rescue operations, providing real-time visuals and data to aid in decision-making.
- **Healthcare Facilities:** The robot can be used in healthcare settings, such as hospitals or nursing homes, for monitoring patient rooms, ensuring the safety of staff and patients, and detecting emergencies or unusual activities.

- **Educational Institutions:** Deploying the surveillance robot in schools or universities can enhance security measures and provide real-time monitoring of campus premises. It can help in preventing unauthorized access, ensuring student safety, and detecting potential threats.
- **Construction Sites:** The robot can be used in construction sites for monitoring worker safety, detecting potential hazards, and ensuring compliance with safety.
- **Traffic Monitoring:** Deploying the robot in traffic management systems can enable real-time monitoring of traffic flow, congestion, or accidents. It can assist in optimizing traffic management and improving overall transportation efficiency.
- **Environmental Monitoring:** The surveillance robot can be used for environmental monitoring, such as monitoring air quality, water pollution, or wildlife habitats. It can collect data and provide valuable insights for environmental conservation efforts.

CHAPTER-10
CONCLUSION & FUTURE SCOPE

10.1 CONCLUSION

An IoT-based surveillance robot is proposed which can solved the problems regarding inspection of difficult areas and unexpected situations. The robot is fully capable of replacing humans and providing extremely accurate data to the user. It overcomes the problem of short ranged communication with the help of IoT platform and broadcasts the live videos to the user. The robot is small in size and is capable of maneuvering hard terrains, also it rotates in all directions. There are many applications to this robot such as surveillance while being steady or in motion, analyzing the surrounding areas, displaying land mines, spying and other militarized operations.

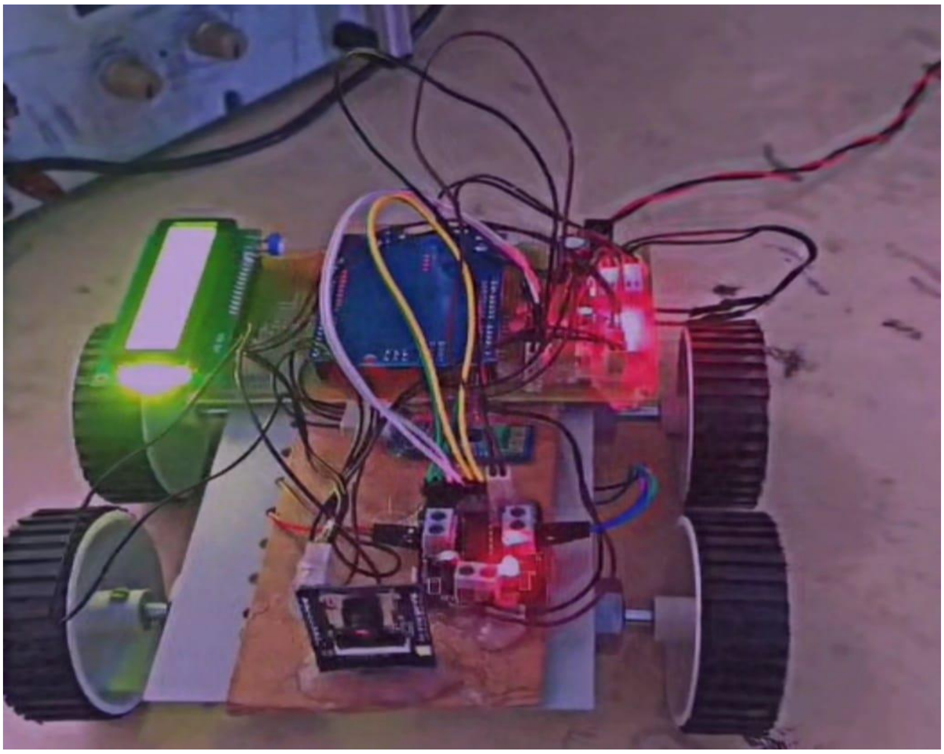


Figure 10.1 IoT Based Surveillance Robot

10.2 FUTURE SCOPE

New technologies can be implemented to make the robot more efficient. a proper HD camera can be used and a separate network can be made for fast and accurate transfer of data. Instead of using the IoT platform an API can be built for fetching data. The design of the robot can be upgrades as per the needs and an arm can be attached to the robot, which will have new functionalities and a few more sensors. Instead of using battery, a proper solar powered battery might be planted the working of robot. Further, thermal imaging or face/identity detection systems can be installed on the robot that will help in identifying people and thermal imaging can be used to get information regarding any hostage or people in restrict.

REFERENCES

[1]. Gasparetto A, Scalera L. From the Unimate to the Delta Robot: The Early Decades of Industrial Robotics. In: Zhang, B., Ceccarelli, M. (eds) Explorations in the History and Heritage of Machines and Mechanisms. History of Mechanism and Machine Science: Springer, Cham; 2019. 37.

[2]. Basan A, Basan E, Makarevich O. Analysis of ways to secure group control for autonomous mobile robots. In: Proceedings of the 10th International Conference on Security of Information and Networks (SIN '17) 13 October 2017: IEEE; 2017., pp.
134139

[3]. Mahmud H, Ahamed JU. An Autonomous Surveillance Robot with IoT based Rescue System Enhancement. International Journal on Emerging Technologies.2020; 11(5): 489 – 494.

[4]. Thiele LP. Rise of the Centaurs: The Internet of Things Intelligence Augmentation.

In: Keskin, T., Kiggins, R.D. (eds) Towards an International Political Economy of Artificial Intelligence. International Political Economy Series. Palgrave Macmillan: Cham; 2021.39 – 61.

[5]. Akilan T, Kumari P, Chaudhary S, Utkarsh P. A Survey on Surveillance Robot in Hazardous Place Using IoT Technology, International Journal of Research in Engineering, Science and Management. 2020; 3(6): 310 – 315.

[6]. Hin MS, Kim B C, Hwang SM, Ko MC. Design and Implementation of IoT-based Intelligent Surveillance Robot. Studies in Informatics and Control. 2016; 25(4); 421- 432.

[7]. Ghouse Z, Hiwrale N, Ranjan N. Military robot for reconnaissance and surveillance using image processing. International Research Journal of Engineering and Technology. 2017; 4(5): 2395-0072

[8]. Witwicki S, Castillo JC, Messias J, Capitan J, Melo FS, Lima PU,

Veloso M. Autonomous surveillance robots: A decision-making framework for networked multi agent systems. IEEE Robotics & Automation Magazine.2017; 24(3):52-64.

[9]. Singh, D., Nand Gaonkar, A. IOT-Based Wi-Fi Surveillance Robot with Real-Time Audio and Video Streaming. In: Iyer, B., Nalbalwar, S., Pathak, N. (eds) Computing, Communication and Signal Processing. Advances in Intelligent Systems and Computing: Springer; 2019.

[10]. Nayyar, A., Puri, V., Nguyen, N.G., Le, D.N. Smart Surveillance Robot for Real Time Monitoring and Control System in Environment and Industrial Applications. In: Bhateja, V., Nguyen, B., Nguyen, N., Satapathy, S., Le, DN. (eds) Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing: Springer; 2018. vol 672.

[11]. Telkar AK, Gadgay B. IoT Based Smart Multi Application Surveillance Robot. Second International Conference on Inventive Research in Computing Applications (ICIRCA): IEEE; 2020. pp. 931-935.

[12]. Pravallika GL, Rakesh P, Srikanth N, Hariguru YP. Multi-Application Surveillance Robot using IoT. International Research Journal of Engineering and Technology. 2021; 8(4): 4554 – 4558.

[13]. Anandravisekar G, Clinton A, Mukesh T, Naveen L. IOT Based Surveillance Robot. International Journal of Engineering Research & Technology. 2018; 7(4): 315 – 320.

[14]. IoT Analytics – Thing Speak Internet of Things, www.thingspeak.com [Accessed: 2022-05-05]

APPENDIX

Source code

Code for L293D(Bluetooth)

```
#include <LiquidCrystal.h>
// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define enA 11  //Enable1 L298 Pin enA
#define in1 A0  //Motor1  L298 Pin in1
#define in2 A1  //Motor1  L298 Pin in1
#define in3 A4  //Motor2  L298 Pin in1
#define in4 A5  //Motor2  L298 Pin in1
#define enB 5   //Enable2 L298 Pin enB
#define Speed 225
#define extra 12
#define leds 4
char value;
bool bl = 0;
void updateLcd(String text) {
  lcd.clear();
  lcd.print(text);
  delay(200);
}
void setup() {
 delay(3000);
  Serial.begin(9600);
```

```
  pinMode(extra, OUTPUT);     //extra pins
  digitalWrite(extra, HIGH);  //extra pin
  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("Surveillance ROBO");
  lcd.setCursor(0, 1);
 lcd.print("CAR using FPV");
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(leds, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  analogWrite(enA, 110);
  analogWrite(enB, 110);
  delay(2000);
  updateLcd("Starting");
  bl = false;
}
void loop() {
  Btc();
}
void Btc() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
```

```
  }
  if (value == 'F') {
    forward();
    updateLcd("FORWARD");
  } else if (value == 'B') {
    backward();
    updateLcd("BACKWARD");
  } else if (value == 'L') {
    left();
    updateLcd("LEFT");
 } else if (value == 'R') {
    right();
    updateLcd("RIGHT");
  } else if (value == 'S') {
    Stop();
    updateLcd("STOP");
    bl = true;
  } else if (value == 'X') {
    digitalWrite(leds, LOW);  //extra pin
  } else if (value == 'x') {
 digitalWrite(leds, HIGH);  //extra pin
  } else if (value == 'D') {
    bl = false;
    digitalWrite(leds, HIGH);  //extra pin
  }
}
void forward() {
  digitalWrite(in1, LOW);   //Left Motor backword Pin
  digitalWrite(in2, HIGH);  //Left Motor forword Pin
```

```
  digitalWrite(in3, LOW);   //Right Motor forword Pin
  digitalWrite(in4, HIGH);  //Right Motor backword Pin
}
void backward() {
  digitalWrite(in1, HIGH);  //Left Motor backword Pin
  digitalWrite(in2, LOW);   //Left Motor forword Pin
  digitalWrite(in3, HIGH);  //Right Motor forword Pin
  digitalWrite(in4, LOW);   //Right Motor backword Pin
}
void right() {
  digitalWrite(in1, LOW);   //Left Motor backword Pin
  digitalWrite(in2, HIGH);  //Left Motor forword Pin
  digitalWrite(in3, HIGH);  //Right Motor forword Pin
  digitalWrite(in4, LOW);   //Right Motor backword Pin


}
void left() {
  digitalWrite(in1, HIGH);  //Left Motor backword Pin
  digitalWrite(in2, LOW);   //Left Motor forword Pin
  digitalWrite(in3, LOW);   //Right Motor forword Pin
  digitalWrite(in4, HIGH);  //Right Motor backword Pin
}
void Stop() {
 digitalWrite(in1, LOW);  //Left Motor backword Pin
  digitalWrite(in2, LOW);  //Left Motor forword Pin
  digitalWrite(in3, LOW);  //Right Motor forword Pin
  digitalWrite(in4, LOW);  //Right Motor backword Pin
}
```

Code for ESP32 CAM

```
#include "esp_camera.h"
#include <WiFi.h>
#include "camera_pins.h"
// const char *ssid = "cam_car";
// const char *password = "123456789";
const char *ssid = "jkc";
const char *password = "jkc@12345";
void startCameraServer();
void setupLedFlash(int pin);
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;

  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
```

```
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.pin_sccb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.frame_size = FRAMESIZE_UXGA;
  // config.pixel_format = PIXFORMAT_JPEG;  // for streaming
  config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition
  config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
  config.fb_location = CAMERA_FB_IN_PSRAM;
  config.jpeg_quality = 12;
  config.fb_count = 1;
  if (config.pixel_format == PIXFORMAT_JPEG) {
    if (psramFound()) {
      config.jpeg_quality = 10;
      config.fb_count = 2;
      config.grab_mode = CAMERA_GRAB_LATEST;
    } else {
      config.frame_size = FRAMESIZE_SVGA;
      config.fb_location = CAMERA_FB_IN_DRAM;
    }
  } else {
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3


 config.fb_count = 2;
#endif
  }
```

```
#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  Sensor_t *s = esp_camera_sensor_get();
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);        // flip it back
    s->set_brightness(s, 1);   // up the brightness just a bit
    s->set_saturation(s, -2);  // lower the saturation
  }
  if (config.pixel_format == PIXFORMAT_JPEG) {
    s->set_framesize(s, FRAMESIZE_QVGA);
  }
#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif
#if defined(CAMERA_MODEL_ESP32S3_EYE)
  s->set_vflip(s, 1);
#endif
// Setup LED FLash if LED pin is defined in camera_pins.h
#if defined(LED_GPIO_NUM)
  setupLedFlash(LED_GPIO_NUM);
```

```
#endif
  WiFi.begin(ssid, password);
  WiFi.setSleep(false);
  Serial.print("WiFi connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  startCameraServer();
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}
void loop() {
  // Do nothing. Everything is done in another task by the web server
  delay(10000);
}
```