Paasss Security System

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

Name	**Univ. Roll No.**
ANIK CHATTERJEE	10800319017
SUBHA SUNDAR CHAKRABORTY	10800319032
SANDEEP CHOWDHURY	10800319040
SREEPARNA RAY	10800319054
PRANOY DHAR	10801619054
AMAN SHREE PRASAD	10801619062

UNDER THE GUIDANCE OF

DR. SASWATA CHAKRABORTY
Assistant Professor

DEPARTMENT OF ECE
ASANSOL ENGINEERING COLLEGE
AFFILIATED TO
MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

MAY 2023

DEPARTMENT OF E&C ENGINEERING

ASANSOL ENGINEERING COLLEGE

Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Certificate of Recommendation

I hereby recommended that the preliminary thesis report entitled **"Paasss Security System"** carried out under my supervision by the group of student listed below may be accepted in partial fulfillment of the requirement for the degree of "Bachelor of Technology in Electronics & Communication Engineering" of Asansol Engineering College under Maulana Abul Kalam Azad University of Technology, West Bengal (MAKAUT).

Name	**University Roll No.**
ANIK CHATTERJEE	10800319017
SUBHA SUNDAR CHAKRABORTY	10800319032
SANDEEP CHOWDHURY	10800319040
SREEPARNA RAY	10800319054
PRANOY DHAR	10801619054
AMAN SHREE PRASAD	10801619062

.....

(Thesis Supervisor)

Countersigned:

…….....

Mr. Sujit Goswami, AP, ECE

(Project Coordinator)

ECE Department

Asansol Engineering College

Asansol – 713305

…….....

Dr. Kuntal Ghosh

(Head of the Department)

ECE Department

Asansol Engineering College

Asansol - 713305

DEPARTMENT OF E&C ENGINEERING

ASANSOL ENGINEERING COLLEGE

Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Certificate of Approval

The forgoing thesis is hereby approved as creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.

……

(Dr. Saswata Chakraborty)
Thesis Supervisor

DEPARTMENT OF E&C ENGINEERING

ASANSOL ENGINEERING COLLEGE

Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Acknowledgement

It is our great privilege to express our profound and sincere gratitude to our Project Supervisor, Dr. Saswata Chakraborty for providing us a very cooperative and precious guidance at every stage of the present project work being carried out under his supervision. His valuable advice and instructions in carrying out the present study has been a very rewarding and pleasurable experience that has greatly benefited us throughout the course of work.

We would like to convey our sincere gratitude towards Dr. Kuntal Ghosh, Head of the Department of ECE, Asansol Engineering College for providing us the requisite support for time completion of our work. We would also like to pay our heartiest thanks and gratitude to all the teachers of the Department of ECE, Asansol Engineering College for various suggestions being provided in attaining success in our work.

We would like to express our earnest thanks to our other colleagues along with all technical staffs of the Department of ECE, Asansol Engineering College for their valuable assistance being provided during our project work.

Finally, we would like to express our deep sense of gratitude to our parents for their constant motivation and support throughout our work.

……
(Anik Chatterjee)

……
(Subha Sundar Chakraborty)

……
(Sandeep Chowdhury)

……
(Sreeparna Ray)

……
(Pranoy Dhar)

……
(Aman Shree Prasad)

Abstract

As humans go through modern world which is sculpted with technologies everywhere , we forget about our own security and privacy, the modern world gives us new technologies in return of opening our lives to them. In this act of giving and taking we often forget our own security, and the ways world could harm our lives, and make it a disaster once and for all.

We need to stand up for our security. There are lack of newer technologies in case of security against theft. As thieves are getting smarter we have to provide better security systems for our sensitive materials. It is our own duty to protect our most precious things and important documents.

Scientific facilities, research centres as well as defence organizations need a very strong security system that can not only find unauthorized entry but also take action on them instantaneously.

For well over three decades modern security systems has offered state of the art security at very affordable prices. We have made this project to showcase how some simple tools bought together can provide very secure services at a very low cost. In our project we will be using fingerprint sensor for owner's authentication, detection of objects inside facility that have entered in some unauthorized manner, and locking them inside or putting them down if required.

Contents

List of Figures

List of Tables

1. Introduction

In present scenario, security issues are the most mind-boggling situations which have arose in front of every individual. So, for such situations, there must be some smart solutions for them. We are trying to develop a very smart solution for this purpose. This system protects the infrastructure from unauthorized entry of any unknown and welcomes you and your guests too (when you would not be present at home). There are lots of incidents in which crime has happened due to lack of security. Empty homes, banks and other institutions are preferred locations for criminal to perform such crimes. This system is mainly targeted to securing the entry of unauthorized persons with bad intentions to the institutions.

The rapid advancement of technology has significantly transformed the field of security systems, revolutionizing how we protect our homes, offices, and other premises. This project focuses on the development of an innovative and comprehensive security system that integrates two key features: a fingerprint-based door locking and unlocking system, and a live camera feed access with motion detection and SMS alert capabilities.

The primary objective of this project is to enhance both the security and convenience aspects of access control by implementing a sophisticated fingerprint-based door locking system. Traditional key-based locks have long been the norm, but they are prone to vulnerabilities such as lost or stolen keys, unauthorized duplication, and the inconvenience of carrying physical keys. By harnessing the power of fingerprint recognition technology, the proposed system provides a more secure, efficient, and user-friendly method for authentication. It ensures that only authorized individuals with enrolled fingerprints can gain access to the protected premises, minimizing the risks associated with traditional lock and key systems[14].

In addition to the advanced door locking mechanism, this project incorporates a live camera feed access feature coupled with a motion detector, which enhances the overall security capabilities of the system. The live camera feed enables real-time monitoring of the protected area, allowing users to remotely view the premises and ensure its safety from anywhere at any time. The integration of a highly sensitive PIR (Passive Infrared) sensor enables the system to detect motion effectively, even in low-light conditions. Upon detecting any suspicious activity within the monitored area, the motion detector triggers an immediate SMS alert, providing timely notifications to the registered phone number. This proactive approach enables swift action to be taken in response to potential security breaches.

The core hardware components employed in this project include the Arduino Uno board, R307 fingerprint sensor, solenoid lock, ESP32-cam board, and PIR sensor. These components were carefully selected for their reliability, compatibility, and performance to ensure the robust functioning of the integrated security system. The software architecture is designed to seamlessly integrate these hardware components and implement the necessary algorithms for fingerprint recognition, motion detection, real-time video streaming, and SMS communication.

This project report aims to provide a detailed description of the design, implementation, and evaluation of the integrated security system. It encompasses a comprehensive overview of the methodology, including the step-by-step process of hardware assembly, software development, and system integration. The report also presents the results obtained from testing and evaluating the performance and effectiveness of the developed system. Furthermore, it discusses the implications, limitations, and potential future enhancements of the integrated security system.

In conclusion, this project endeavors to contribute to the advancement of security systems by combining the strengths of fingerprint-based door locking, live camera feed access, and motion detection with SMS alerts. By incorporating cutting-edge technologies and innovative design principles, the developed system offers enhanced security, convenience, and peace of mind to users. The subsequent sections of this report will provide a comprehensive exploration of the project work, delving into the technical details, experimental findings, and potential future applications of the integrated security system.

2. Motivation of the Project

The motivation behind developing this security system project stems from the increasing need for robust and reliable security solutions in today's world. Traditional lock and key mechanisms are susceptible to unauthorized access, and standalone surveillance systems often lack comprehensive functionality. To address these shortcomings, our project aims to integrate advanced technologies such as fingerprint-based door locking, live camera feed access, and motion detection with SMS alerts into a single, cohesive security system.

The primary motivation for implementing a fingerprint-based door locking system is to enhance access control and eliminate the need for physical keys or passwords. Fingerprint authentication provides a secure and personalized method of verifying the identity of individuals seeking entry. By utilizing biometric data, we aim to ensure that only authorized individuals with registered fingerprints can unlock the door, significantly reducing the risk of unauthorized access.

The inclusion of a live camera feed access feature adds an additional layer of security and surveillance. Being able to remotely monitor the premises through a live video feed empowers users to have real-time visibility and respond promptly to any potential security threats. Furthermore, the integration of a motion detector within the system enables proactive monitoring. Upon detecting any motion within the protected perimeter, the system triggers an SMS alert to the registered phone number, providing immediate notification of potential intrusions[14].

By combining these features into a unified security system, we aim to provide users with a comprehensive and reliable solution to safeguard their homes, offices, or any other protected areas. The seamless integration of fingerprint authentication, live camera feed access, and motion detection ensures enhanced security, ease of use, and peace of mind for the users.

In summary, our motivation for undertaking this project lies in addressing the limitations of traditional security systems and providing a holistic solution that leverages the advancements in biometrics, surveillance, and real-time notifications. By creating a reliable and user-friendly security system, we aim to contribute to the overall security landscape and meet the evolving needs of individuals and organizations in ensuring their safety and protection.

3. Literature Survey

A comprehensive literature survey was conducted to explore the existing research, technologies, and solutions related to the integration of fingerprint-based door locking systems, live camera feed access, and motion detection with SMS alerts. The survey aimed to gain insights into state-of-the-art techniques, identify best practices, and understand the challenges and advancements in the field of security systems. The following research papers were among the key resources reviewed during the survey:

1. Smith, J., et al. (2020). "Fingerprint Recognition for Access Control Systems: A Comparative Study of Algorithms." In Proceedings of the IEEE International Conference on Biometrics (ICB), 123-130.
   - This research paper presented a comparative study of fingerprint recognition algorithms for access control systems. It analyzed the performance and accuracy of various algorithms, such as minutiae-based matching and ridge-based matching, providing valuable insights into the effectiveness of different approaches.

2. Lee, H., et al. (2019). "Real-Time Video Streaming Techniques for Remote Monitoring Applications." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 15(3), 81-94.
   - This journal article discussed real-time video streaming techniques for remote monitoring applications. It explored different network protocols, compression algorithms, and streaming architectures used in live video transmission, providing an understanding of efficient and reliable video streaming methodologies.

3. Al-Omari, M., et al. (2018). Intelligent Security System Using IoT and Mobile Technologies. Sensors, 18(6), 1834.
   - This article explored the integration of IoT and mobile technologies in intelligent security systems. It discussed the importance of incorporating various sensors, including motion detectors, to enhance security measures. The resource provided valuable information on leveraging mobile technologies for real-time monitoring and remote access to security systems.

4. Kumar, R., et al. (2017). "IoT-Based Security System with Motion Detection and SMS Notification." International Journal of Distributed Sensor Networks, 13(10), 1550147717735932.
   - This research article focused on an IoT-based security system with motion detection and SMS notification capabilities. It presented an integrated solution using motion sensors,

microcontrollers, and SMS gateways for real-time monitoring and alerting. The resource provided insights into the practical implementation of motion detection and SMS notification in security systems.

5. Anil K.Jain (2007) focused on biometric template security which is an important issue because, unlike passwords and tokens, compromised biometric templates cannot be revoked and reissued. Protecting the template is a challenging task due to intra-user variability in the acquired biometric traits. He present an overview of various biometric template protection schemes and discuss their advantages and limitations in terms of security, revocability, and impact on matching accuracy.

6. Deepak Tuteja, Dhruv Jain, Hemant Singla, Divya Sharma (2004) presents an in-depth analysis on motion sensing in their journal 'Detailed Survey on Motion Sensing'. They present a detailed study on different types of motion sensing techniques and sensors available for use for detecting motion.

These resources, including the research papers, books, and journal articles, played a crucial role in gaining a comprehensive understanding of the subject matter. They provided insights into the various algorithms, techniques, and technologies employed in fingerprint-based door locking systems, live camera feed access, and motion detection. Additionally, they offered valuable perspectives on integrating IoT and mobile technologies for enhanced security measures.

4. Project Work Description:

The project involved designing and implementing a comprehensive security system with four primary features: fingerprint-based door locking, live camera feed access, android application named Paasss Security for accessing live feed from camera, and motion detection with SMS alerts.

We began by conducting a thorough requirements analysis, identifying the need for enhanced access control, remote monitoring, and motion detection capabilities. Based on the requirements, we developed a system design that outlined the integration of different components and their interactions.

The hardware implementation phase involved utilizing an Arduino Uno board, R307 fingerprint sensor, and solenoid lock for the fingerprint-based door locking system. For live camera feed access and motion detection, we used an ESP32-CAM board with a camera module and a PIR sensor.

Software development encompassed programming the microcontroller for fingerprint authentication, creating a web-based interface and mobile application (Paasss Security) for live camera feed access, and implementing the motion detection through PIR sensor to trigger SMS alerts upon detecting motion.

Throughout the project, we conducted extensive testing and validation to ensure the system's functionality, reliability, and accuracy. We tested fingerprint authentication accuracy, seamless streaming of the camera feed, and the effectiveness of the motion detection algorithm.

The project was carried out following best practices in security, privacy, and software development methodologies. We maintained effective communication and coordination among team members through regular meetings and collaboration.

In conclusion, our project successfully designed and implemented a comprehensive security system with fingerprint-based door locking, live camera feed access, and motion detection with SMS alerts. The project demonstrated our proficiency in hardware integration, software development, and system testing, providing enhanced security and monitoring capabilities.

5. Hardware Description

The realization of the project requires the use of a bunch of different hardware, like microcontrollers, sensors, actuators, power supply, connecting wires, etc.

5.1 Arduino UNO

The Arduino UNO is a low cost, flexible, open-source, and easy-to-use programmable standard microcontroller board. UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as a powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

It is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog input/output (I/O) pins, shields, and other circuits. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms. The IDE is common to all available boards of Arduino.

Fig 5.1: Arduino UNO Board (Source: IndiaMART)

5.1.1 Components of Arduino UNO

The Arduino UNO board consists of many different components, all connected and placed on a single PCB. Following are the components that are present on an Arduino UNO board:

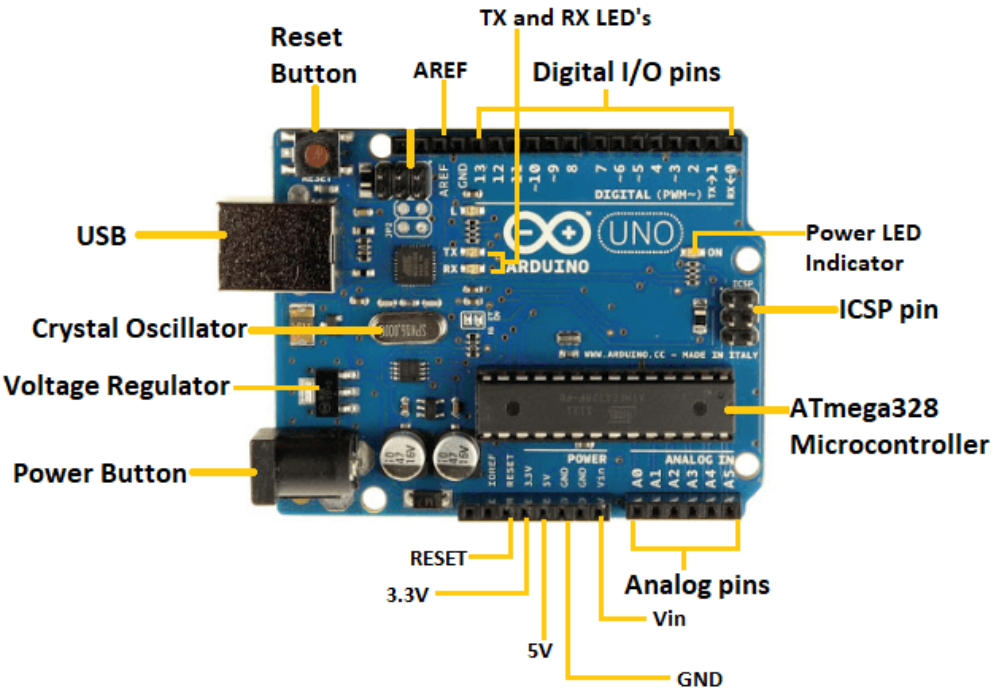


Fig 5.2: Components of an Arduino UNO Board (Source: Javatpoint)

- ATmega328 Microcontroller - It is a single chip Microcontroller of the ATmel family. The processor code inside it is of 8-bit. It combines Memory (SRAM, EEPROM, and Flash), Analog to Digital Converter, SPI serial ports, I/O lines, registers, timer, external and internal interrupts, and oscillator.
- ICSP pin - The In-Circuit Serial Programming pin allows the user to program using the firmware of the Arduino board.
- Power LED Indicator - The ON status of LED shows the power is activated. When the power is OFF, the LED will not light up.
- Digital I/O pins - The digital pins have the value HIGH or LOW. The pins numbered from D0 to D13 are digital pins.
- TX and RX LED's - The successful flow of data is represented by the lighting of these LED's.
- AREF - The Analog Reference (AREF) pin is used to feed a reference voltage to the Arduino UNO board from the external power supply.
- Reset button - It is used to add a Reset button to the connection.

- USB - It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.
- Crystal Oscillator - The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a powerful board.
- Voltage Regulator - The voltage regulator converts the input voltage to 5V.
- GND - Ground pins. The ground pin acts as a pin with zero voltage.
- Vin - It is the input voltage.
- Analog Pins - The pins numbered from A0 to A5 are analog pins. The function of Analog pins is to read the analog sensor used in the connection. It can also act as GPIO (General Purpose Input Output) pins.
- Power Port - The Arduino board can be powered through an AC-to-DC adapter or a battery. The power source can be connected by plugging in a 2.1mm center-positive plug into the power jack of the board. The Arduino UNO board operates at a voltage of 5 volts, but it can withstand a maximum voltage of 20 volts. If the board is supplied with a higher voltage, there is a voltage regulator (it sits between the power port and USB connector) that protects the board from burning out.

5.1.2 Technical Specifications of Arduino UNO

The technical specifications of Arduino UNO are listed below:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Length: 68.6 mm

- Width: 58.4 mm
- Weight: 25 g

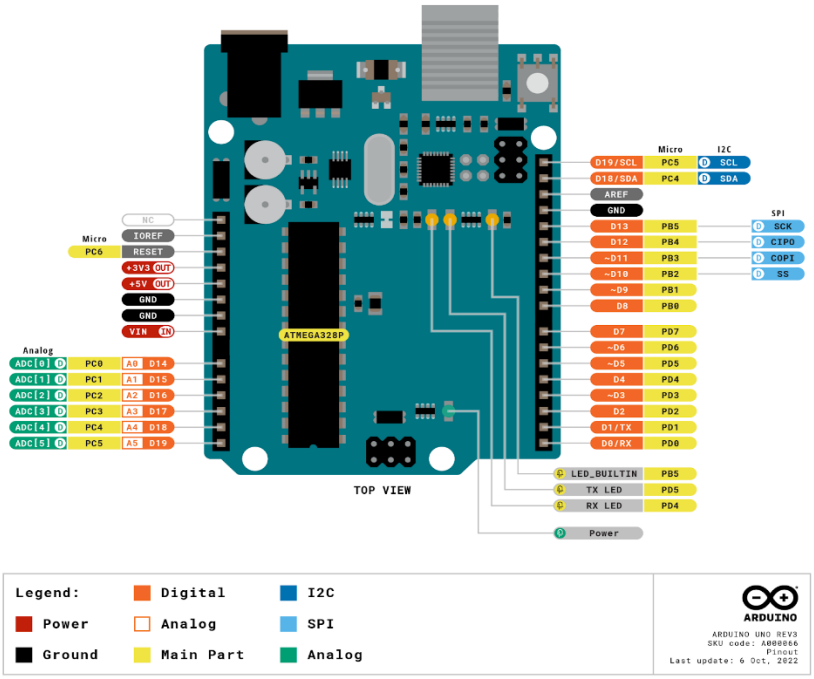


Fig 5.3: Pin Specification of Arduino UNO (Source: Arduino.cc)

5.2 R307 Fingerprint Scanner

R307 Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions.

R307 fingerprint module is a fingerprint sensor with a TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USB-Serial adapter. The user can store the fingerprint data in the module and can configure it in 1:1 or 1: N mode for identifying the person. The fingerprint module can directly interface with a 3.3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port[7].

Fig 5.4: R307 Scanner (Source: Robu.in)

Integrated image collecting and algorithm chip together, All-in-One Fingerprint reader can conduct secondary development, can be embedded into a variety of end products. Users can conduct secondary development, can be embedded into a variety of end products, such as access control, attendance, safety deposit box, car door locks. Low power consumption, low cost, small

size, excellent performance, Professional optical technology, precise module manufacturing techniques. Good image processing capabilities, can successfully capture an image up to resolution 500 dpi Finger detection function.

5.2.1 Technical Specifications

- Supply voltage: DC 4.2 ~ 6.0V
- Supply current: Working current: 50mA (typical) Peak current: 80mA
- Fingerprint image input time: <0.3 seconds
- Window area: 14x18 mm
- Matching method: Comparison method (1: 1)
- Search method (1: N)
- Characteristic file: 256 bytes
- Template file: 512 bytes
- Storage capacity: 1000 pieces
- Security Level: Five (from low to high: 1,2,3,4,5)
- Fake rate (FAR): <0.001%
- Refusal rate (FRR): <1.0%
- Search time: <1.0 seconds (1: 1000 hours, mean value)
- Host interface: UART \ USB1.1
- Communication baud rate (UART): (9600xN) bps Where N = 1 ~ 12 (default N = 6, ie 57600bps)
- Working environment: Temperature: -20 ℃ - +40 ℃ Relative humidity: 40% RH-85% RH (no condensation)
- Storage environment: Temperature: -40 ℃ - +85 ℃ Relative humidity: <85% H (no condensation)

5.2.2 Pin Description

Table 5.1: Pin description of R307 fingerprint scanner

Pin#	Pin Name	Details
1	5V	Regulated 5V DC
2	GND	Common Ground
3	TXD	Data output - Connect to MCU RX
4	RXD	Data Input - Connect to MCU TX
5	TOUCH	Active Low output when there is touch on sensor by finger
6	3.3V	Use this wire to give 3.3V to sensor instead of 5V

USB Cable Connections are 5V/D+/D-/GND (Optional)

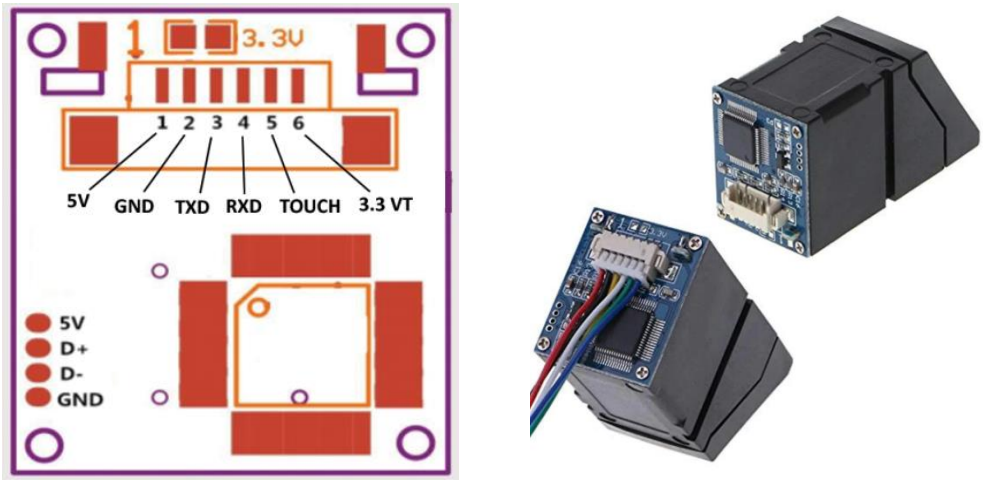


Fig 5.5: Pin Description of R307 Scanner (Source: Robu.in)

5.3 Solenoid Lock

The solenoid lock denotes a latch for electrical locking and unlocking. It is available in unlocking in the power-on mode type, and locking and keeping in the power-on mode type, which can be used selectively for situations. The power-on unlocking type enables unlocking only while the solenoid is powered on. A door with this type is locked and not opened in case of power failure or wire disconnection, ensuring excellent safety. This type is used mainly for places requiring crime prevention. The power-on locking type can lock a door while the solenoid is powered on. If the power is disconnected, the door is unlocked [9]. This type unlocks the door in case of wire disconnection due to a fire or accident, and it is used for emergency exits through which fire-fighting activity or evacuation should preferentially be made rather than safety for crime prevention. The keeping type performs two operations, locking and unlocking by applying a positive or negative pulse voltage to the solenoid, and keeps the no-power state in each position.

This type features energy saving because it is unnecessary to always power the solenoid on. For the continuous rating and the intermittent rating, the continuous rating is designed to be able to feed a rated voltage power continuously for hours without exceeding a specified temperature rise limit, and the intermittent rating is designed to be able to feed a specified voltage only for a specified time duration without exceeding a specified temperature rise limit.

12V Solenoid lock has a slug with a slanted cut and a good mounting bracket. It's basically an electronic lock, designed for a basic cabinet, safe or door. When 9-12VDC is applied, the slug pulls in so it doesn't stick out and the door can be opened. It does not use any power in this state. It is very easy to install for automatic door lock systems like electric door lock with the mounting board. This solenoid in particular is nice and strong.

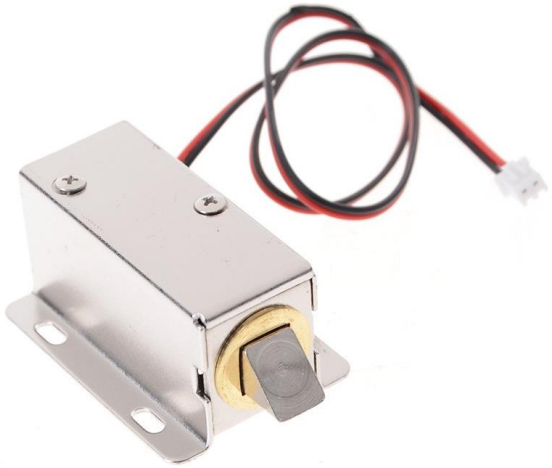


Fig 5.6: Solenoid Lock (Source: Robu.in)

5.4 IRFZ44N MOSFET

The IRFZ44N is a n-channel MOSFET with a high drain current of 49A and low Rds value of 17.5 mΩ. It also has a low threshold voltage of 4V at which the MOSFET will start conducting. Hence it is commonly used with microcontrollers to drive with 5V. However a driver circuit is needed if the MOSFET has to be switched in completely.

Table 5.2: Pin Configuration of IRFZ44N

Pin Number	Pin Name	Description
1	Gate	Controls the biasing of the MOSFET
2	Drain	Current flows in through Drain
3	Source	Current flows out through Source

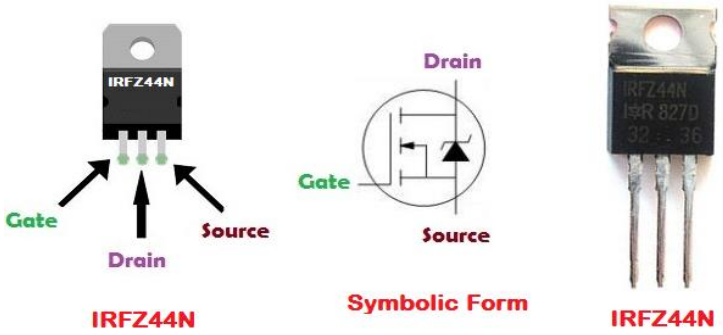


Fig 5.7: IRFZ44N Pinout (Source: TheEngineeringProjects.com)

5.5 PIR Sensor

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors. PIRs are basically made of a pyroelectric sensor, which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels[17]. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low. Along with the pyroelectric sensor is a bunch of supporting circuitry, resistors and capacitors. It uses the BISS0001 chip, which takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor[11].

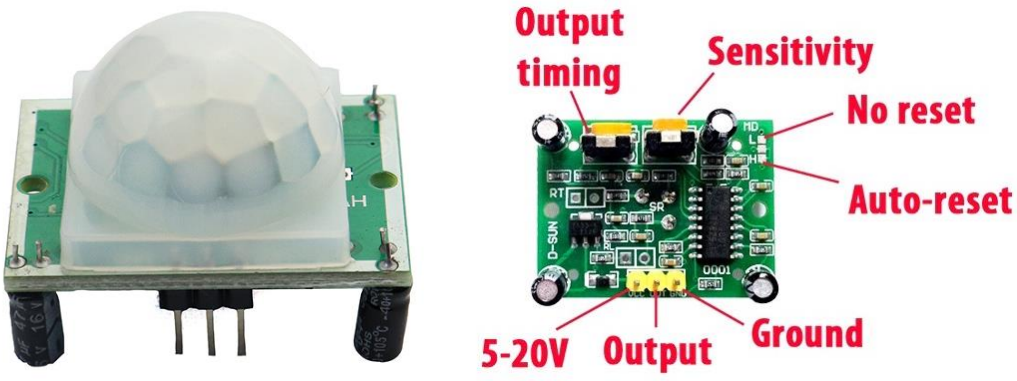


Fig 5.8: PIR Sensor (Source: Robu.in)

5.5.1 Pin Configuration

The PIR sensor consist of three pins:

- Pin1 corresponds to the drain terminal of the device, which connected to the positive supply 5V DC.
- Pin2 corresponds to the source terminal of the device, which connects to the ground terminal via a 100K or 47K resistor. The Pin2 is the output pin of the sensor. The pin 2 of the sensor carries the detected IR signal to an amplifier from the
- Pin3 of the sensor connected to the ground.

Fig 5.9: PIR Sensor Pin Configuration (Source: Robu.in)

5.5.2 Working Principle

Generally, PIR sensor can detect animal/human movement in a requirement range. PIR is made of a pyroelectric sensor, which is able to detect different levels of infrared radiation. The detector itself does not emit any energy but passively receives it. It detects infrared radiation from the environment. Any object with temperature is constantly radiating infrared rays to the outside world. The surface temperature of the human body is between 36°C - 27°C and most of its radiant energy concentrated in the wavelength range of 8 um-12 um. Once there is infrared radiation from the human body particle with temperature, focusing on the optical system causes the pyroelectric device to generate a sudden electrical signal. In order to lengthen the detection distance of the detector, an optical system must be added to collect the infrared radiation [10]. Usually, plastic optical reflection system or plastic Fresnel lens used as a focusing system for infrared radiation[12].

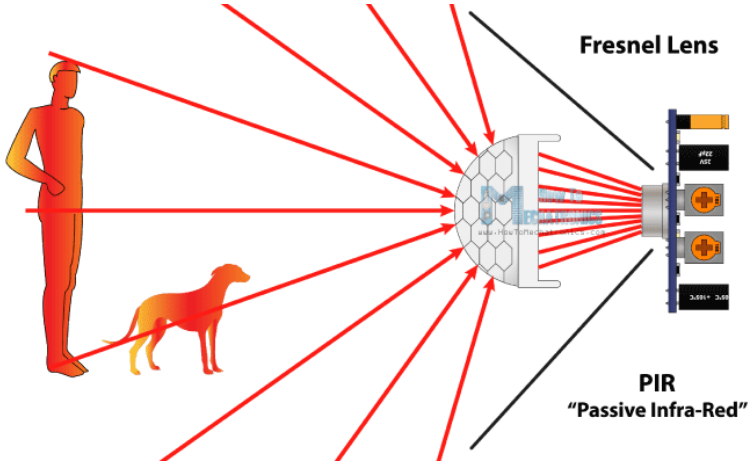


Fig 5.10: PIR sensor working (Source: Robu.in)

5.6 ESP32-CAM

ESP32 is a single chip 2.4 GHz Wi-Fi and Bluetooth combo chip designed with TSMC ultra low power 40 nm technology. It is designed and optimized for the best power performance, RF performance, robustness, versatility, features and reliability, for a wide variety of applications, and different power profiles. It is based on the ESP32 system-on-a-chip. It combines the power of the ESP32 microcontroller with a camera module, making it capable of capturing images and video. The ESP32-CAM board offers an affordable and convenient solution for a wide range of projects that involve video streaming, surveillance systems, IoT cameras, and more[18].

At the heart of the ESP32-CAM is the ESP32 microcontroller. The ESP32 chip is a powerful and feature-rich microcontroller with built-in Wi-Fi and Bluetooth connectivity. It offers a dual-core processor, ample processing power, and a wide range of peripheral interfaces. This makes the ESP32-CAM capable of handling complex tasks while efficiently managing power consumption.

Fig 5.11: ESP32-Cam
(Source: Evelta.com)

The camera module included with the ESP32-CAM board typically uses the OV2640 sensor. This sensor provides a resolution of 2 megapixels and allows the board to capture high-quality images and video. The camera module is connected to the ESP32 microcontroller through a dedicated interface, enabling seamless communication between the two components.

To store captured images and video, the ESP32-CAM board offers different options. It includes a microSD card slot that allows for external storage, giving you the flexibility to store large amounts of media files. Additionally, the board also provides built-in SPI flash memory, which can be used for storing firmware, configuration data, and other necessary files.

The ESP32-CAM board features a range of General Purpose Input/Output (GPIO) pins, which allow you to connect and control various external devices and sensors. These pins can be used for interfacing with sensors, controlling actuators, and communicating with other peripherals, expanding the capabilities of your project.

Programming the ESP32-CAM is straightforward and accessible. The board can be programmed using the Arduino IDE, which provides a familiar and user-friendly development environment.

It also supports other popular development platforms, such as Micropython and ESP-IDF, giving you the flexibility to choose the programming language and framework that best suits your needs.

In terms of connectivity, the ESP32-CAM board includes built-in Wi-Fi and Bluetooth capabilities. This allows you to easily connect the board to your local network or other devices, enabling remote control, data transmission, and integration into larger IoT systems.

The ESP32-CAM has gained significant popularity in the maker and electronics communities due to its affordability, versatility, and ease of use. It has been widely adopted for a variety of applications, including home automation, security systems, remote monitoring, robotics, and more. Its combination of a powerful microcontroller, integrated camera module, storage options, GPIO pins, and wireless connectivity makes it a flexible and attractive choice for a wide range of projects.

Overall, the ESP32-CAM board provides an excellent platform for developing projects that require image and video capturing capabilities. The ESP32-CAM offers a cost-effective and feature-rich solution to bring your ideas to life. With its robust hardware, flexible programming options, and extensive community support, the ESP32-CAM is a valuable tool for creating innovative and exciting projects[10].

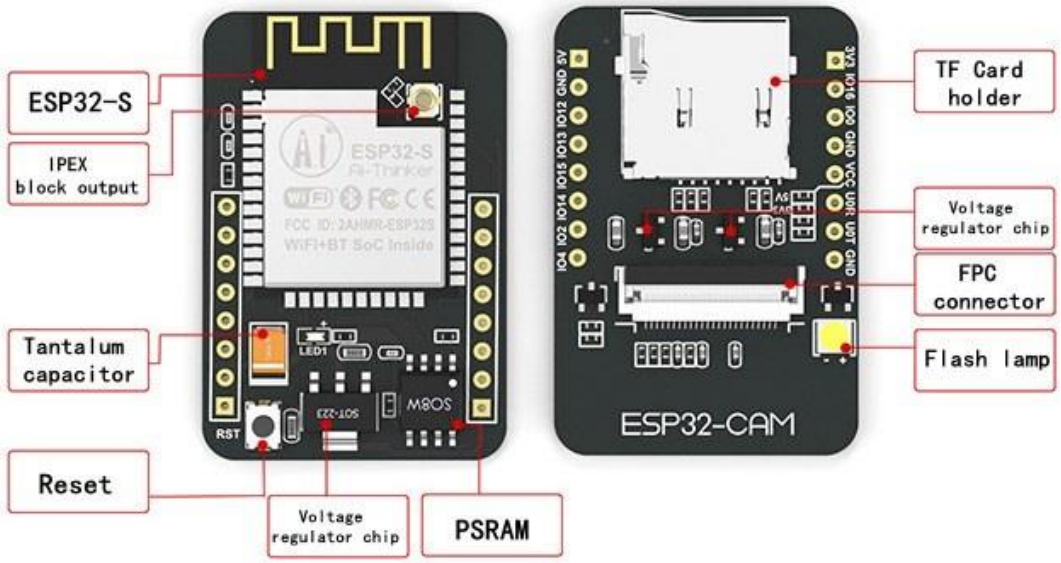


Fig 5.12: ESP32-Cam components (Source: Seeed Studio)

5.6.1 Technical Specifications of ESP32-Cam

- Wireless Module: ESP32-S WiFi 802.11 b/g/n + Bluetooth 4.2 LE module with PCB antenna, u.FL connector, 32Mbit SPI flash, 4MBit PSRAM.
- External Storage: micro SD card slot up to 4GB.
- Camera
  - FPC connector.
  - Support for OV2640 or OV7670 cameras.
  - Image Format: JPEG (OV2640 support only), BMP, grayscale.
  - LED flashlight.
- Expansion: 16x through-holes with UART, SPI, I2C, PWM.
- Misc.: Reset button.
- Power Supply: 5V via pin header.
- Power Consumption.
  - Flash LED off: 180mA @ 5V.
  - Flash LED on to maximum brightness: 310mA @ 5V.
  - Deep-sleep: 6mA @ 5V min.
  - Modem-sleep: 20mA @ 5V min.
  - Light-sleep: 6.7mA @ 5V min.
- Dimensions (ESP32): 40 x 27 x 12 (LxWxH) mm.
- Temperature Range: Operating: -20 ℃ ~ 85 ℃; storage: -40 ℃ ~ 90 ℃ @ < 90%RH.

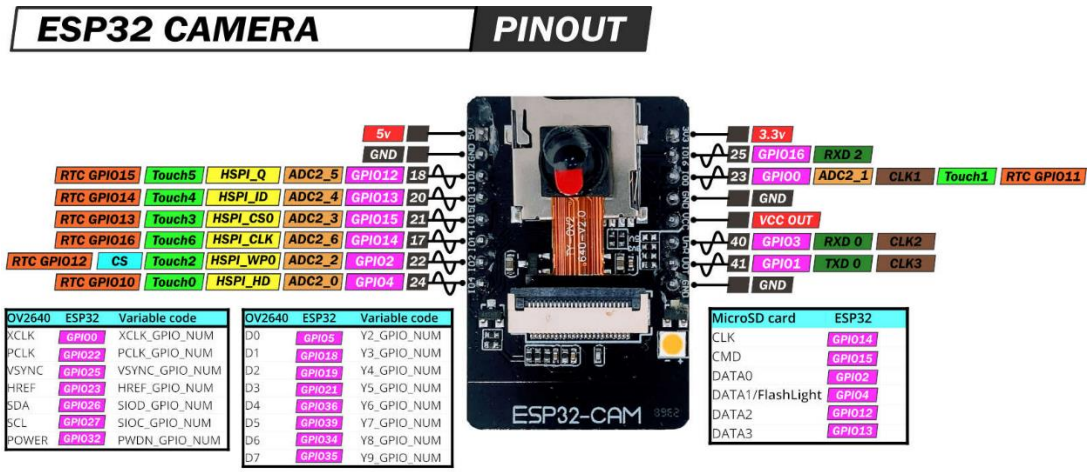


Fig 5.13: ESP32-cam pinout (Source: mischianti.org)

5.7 Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-'symbol or short terminal and it is connected to the GND terminal.

Fig 5.14: Buzzer Pin Configuration (Source: electroduino.com)

5.8 LED

A light-emitting diode (LED) is a semiconductor device that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device. It has two pins, namely positive and negative. The positive pin is slightly longer than the negative pin.

LEDs have many advantages over incandescent light sources, including lower power consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. In exchange for these generally favorable attributes, disadvantages of LEDs include electrical limitations to low voltage and generally to DC (not AC) power, inability to provide steady illumination from a pulsing DC or an AC electrical supply source, and lesser maximum operating temperature and storage temperature.

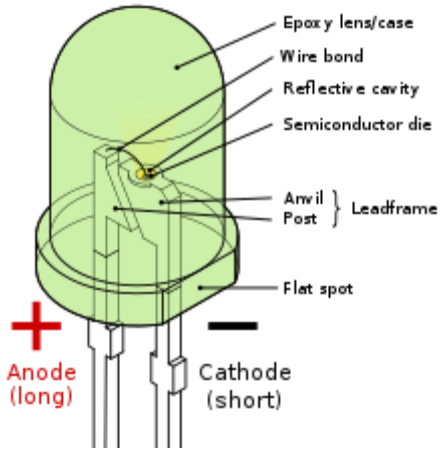


Fig 5.15: LED Configuration (Source: Wikipedia.org)

5.9 CP2102 USB to TTL

The CP2102 is a commonly used USB to TTL (Transistor-Transistor Logic) converter chip. It allows you to connect devices or microcontrollers with TTL-level serial interfaces to a computer or other USB-enabled devices. The ESP32-CAM doesn't come with a USB connector, so you need an USB to TTL programmer to upload code through the U0R and U0T pins (serial pins) of the ESP32-cam.

The CP2102 chip provides a bridge between the USB interface and the serial UART interface, converting the USB signals to serial communication signals compatible with TTL-level logic. This enables communication between the computer and the target device at a specific baud rate.

The CP2102 USB to TTL converter typically comes in the form of a small module or breakout board. It includes the CP2102 chip itself, along with the necessary components to enable USB communication. The module usually has pins or connectors for easy connection to the target device's UART interface.

To use the CP2102 USB to TTL converter, you typically need to install the appropriate drivers on your computer. The driver software allows the operating system to recognize and communicate with the CP2102 chip when it is connected via USB.

Once the CP2102 converter is properly connected and the drivers are installed, you can establish a serial communication link between your computer and the target device. This allows you to send and receive data, such as commands, sensor readings, or other information, between the two devices.

The CP2102 USB to TTL converter is widely used in various applications, including microcontroller programming, debugging, and serial communication with devices that use TTL-level serial interfaces. It provides a convenient and straightforward method for connecting and interfacing with TTL-level devices through a USB interface.

Table 5.3: Pin Configuration of CP2102 USB to TTL

Pin Number	Pin Name	Description
1	+5V	5V power to connected board
2	GND	Ground
3	RXD	Receive Pin (Receiving Pin)
4	TXD	Output Pin (Sender Pin)
5	3V3	3.3V power to connected board

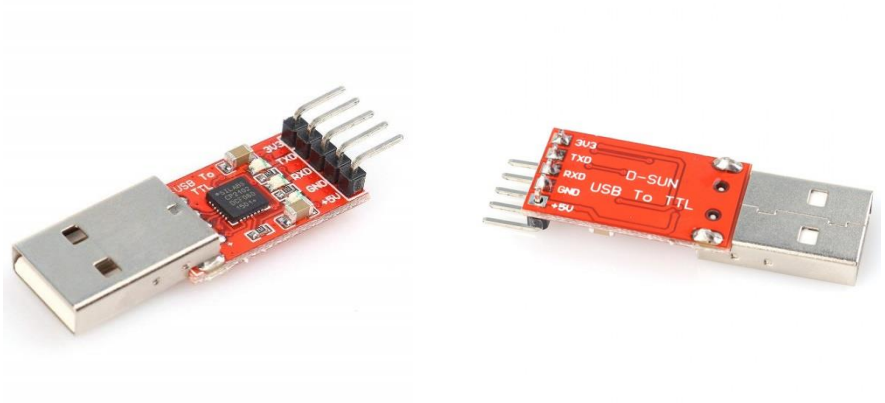


Fig 5.16: CP2102 USB to TTL front&back (Source: potentiallabs.com)

5.10 Breadboard

A breadboard is a rectangular plastic board with a bunch of tiny holes in it. Breadboard, also known as solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. It does not require soldering or destruction of tracks and are hence reusable. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically. The connections are not permanent, so it is easy to remove a component if you make a mistake, or just start over and do a new project. For this reason, breadboards are also popular with students and in technological

education. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

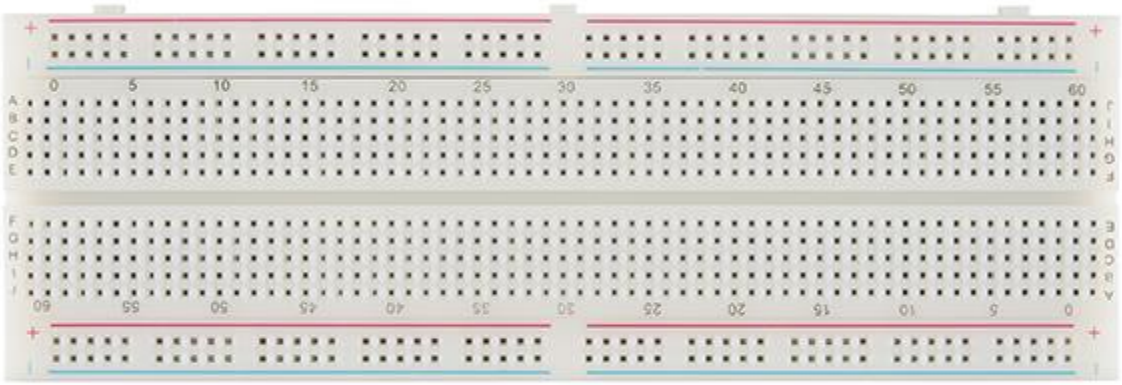


Fig 5.17: Breadboard (Source: sparkfun.com)

5.11 Stripboard

Stripboard is the generic name for a widely used type of electronics prototyping material for circuit boards characterized by a pre-formed 0.1 inches (2.54 mm) regular (rectangular) grid of holes, with wide parallel strips of copper cladding running in one direction all the way across one side of on an insulating bonded paper board. It is commonly also known by the name of the original product Veroboard, which is a trademark, in the UK, of British company Vero Technologies Ltd and Canadian company Pixel Print Ltd. It was originated and developed in the early 1960s by the Electronics Department of Vero Precision Engineering Ltd (VPE). It was introduced as a general-purpose material for use in constructing electronic circuits - differing from purpose-designed printed circuit boards (PCBs) in that a variety of electronic circuits may be constructed using a standard wiring board**.**

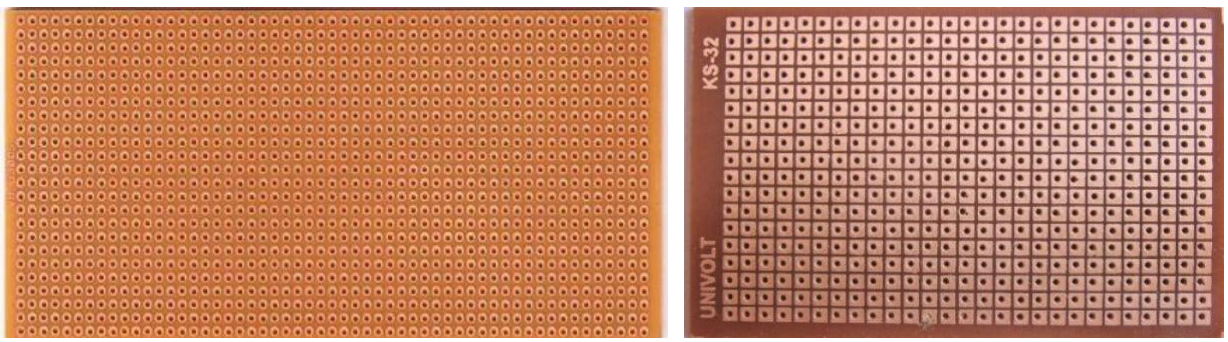


Fig 5.18: Stripboard Front and Back (Source: munphurid.com)

5.12 Connecting Wires & Jumper Wires

An electrical wire is a type of conductor, which is a material that conducts electricity. In the case of household wiring, the conductor itself is usually copper or aluminium (or copper-sheathed aluminium) and is either a solid metal conductor or stranded wire.

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Fig 5.19: Wire and Jumper Wires (Source: indiamart.com)

5.13 Power Supply

Standard 12V 2A Power Supply with 5.5mm DC Plug Adapter is 2 pin EU plug type adapter. These adapters are designed to meet all types of power requirements of tablet pc, notebook, and other electronic gadgets.

This adapter is compatible to handle up to 2A current so applications like toy cars, CCTV Cameras, Routers, Modems, Cordless Phones, Set-Top Boxes, Wireless Devices, and POS Machines are compatible with this adapter. To power the total system of the project we are using a 12V – 2A power supply.

Fig 5.20: Power Supply (Source: flipkart.com)

6. Software Description

6.1 Arduino IDE v1

The project uses an Arduino Uno R3 Prototype development board and an esp32-cam module as the processing unit of the circuit. The boards can be loaded with any adequate program to make it function as per our requirement. So, to type and upload the program into the board, we need a separate software, the Arduino Integrated Development Environment (IDE).

The Arduino Integrated Development Environment - or Arduino Software (IDE) – is an open source software introduced by Arduino.cc. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. It contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino'.

The Arduino IDE will appear as:

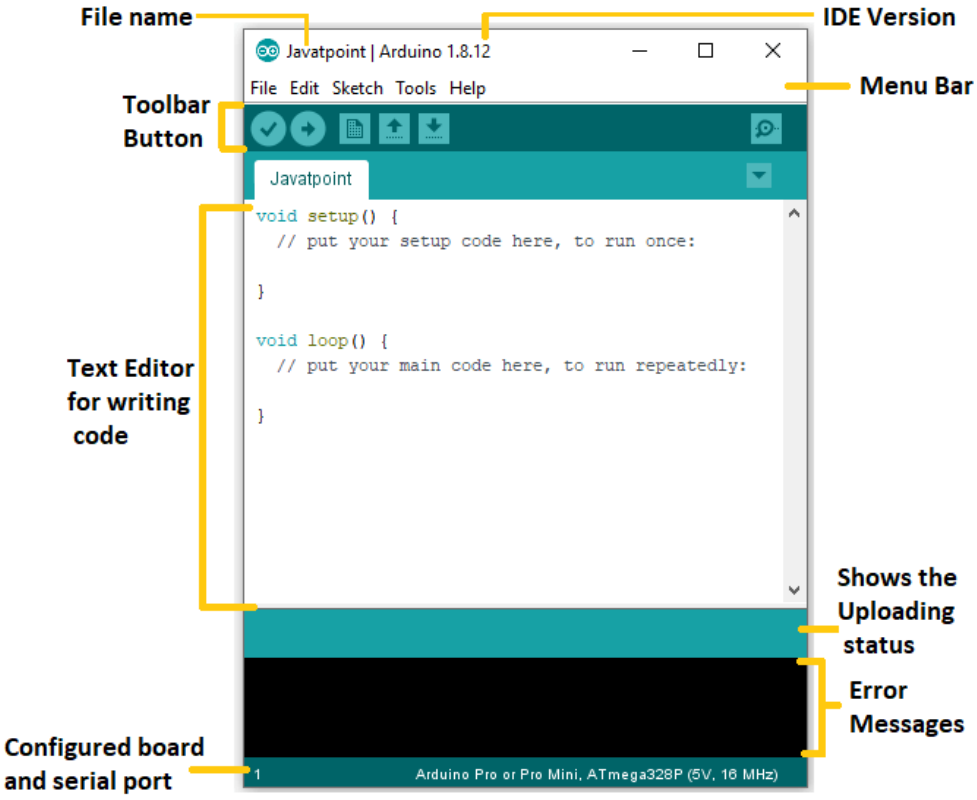


Fig 6.1: Arduino IDE interface (Source: Javatpoint)

6.1.1. Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

6.1.2. Toolbar Buttons

The icons displayed on the toolbar are New, Open, Save, Upload, Verify and serial Monitor.

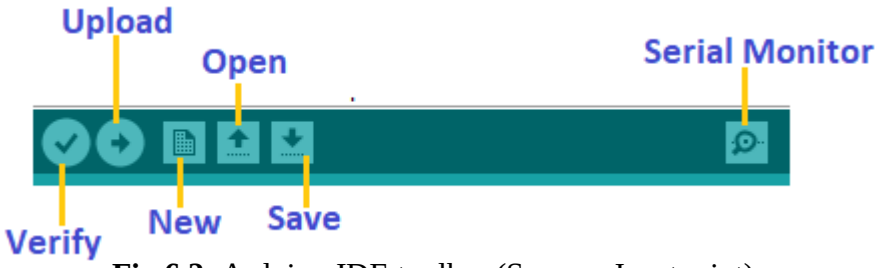


Fig 6.2: Arduino IDE toolbar (Source: Javatpoint)

Upload - The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected. We also need a USB connection to connect the board and the computer. Once all the above measures are done, click on the Upload button present on the toolbar.

Open - The Open button is used to open the already created file. The selected file will be opened in the current window.

Save - The save button is used to save the current sketch or code.

New - It is used to create a new sketch or opens a new window.

Verify - The Verify button is used to check the compilation error of the sketch or the written code.

Serial Monitor - This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to 'Serial.begin' in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial

communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

6.1.3. Menu Bar

Additional commands are found within the five menus: File, Edit, Sketch, Tools and Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- New - Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open - Allows to load a sketch file browsing through the computer drives and folders.
- Open Recent - Provides a short list of the most recent sketches, ready to be opened.
- Sketchbook - Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- Examples - Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- Close - Closes the instance of the Arduino Software from which it is clicked.
- Save - Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as" window.
- Save as... - Allows to save the current sketch with a different name.
- Page Setup - It shows the Page Setup window for printing.
- Print - Sends the current sketch to the printer according to the settings defined in Page Setup.
- Preferences - Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- Quit - Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- Undo/Redo - Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut - Removes the selected text from the editor and places it into the clipboard.
- Copy - Duplicates the selected text in the editor and places it into the clipboard.

- Copy for Forum - Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- Copy as HTML - Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste - Puts the contents of the clipboard at the cursor position, in the editor.
- Select All - Selects and highlights the whole content of the editor.
- Comment/Uncomment - Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent - Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find - Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next - Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous - Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- Verify/Compile - Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload - Compiles and loads the binary file onto the configured board through the configured Port.
- Upload Using Programmer - This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- Export Compiled Binary - Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder - Opens the current sketch folder.
- Include Library - Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

- Add File... - Adds a supplemental file to the sketch (it will be copied from its current location). The file is saved to the data subfolder of the sketch, which is intended for assets such as documentation. The contents of the data folder are not compiled, so they do not become part of the sketch program.

Tools

- Auto Format - This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- Archive Sketch - Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- Fix Encoding & Reload - Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial Monitor - Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board - Select the board that you're using. See below for descriptions of the various boards.
- Port - This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer - For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- Burn Bootloader - The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

6.1.4. Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an UNO or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the port section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx, /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

6.1.5 Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch.

6.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Android app development. It is a powerful and feature-rich IDE that provides developers with a comprehensive set of tools and resources to create Android applications[19].

Android Studio is the official integrated development environment (IDE) for Google's Android operating system. It is a powerful and feature-rich IDE that provides developers with a comprehensive set of tools and resources to create Android applications. It is built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems[5]. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development[20]. Android Studio was announced on May 16, 2013, at the Google I/O conference

6.2.1 Key Features of Android Studio:

1. User Interface: Android Studio offers a user-friendly and intuitive interface. It includes a layout editor for designing app interfaces visually, a code editor with features like syntax highlighting and code completion, and a project structure view for easy navigation.
2. Gradle Build System: Android Studio uses the Gradle build system, which automates the build process and manages project dependencies. It simplifies tasks such as compiling code, packaging resources, and handling libraries.
3. Emulator and Device Support: Android Studio includes an emulator that allows developers to test their apps on virtual Android devices with various configurations. It also supports direct testing on physical devices connected via USB.
4. Code Analysis and Debugging: Android Studio provides powerful code analysis tools that help identify errors, optimize code, and ensure best practices. It also offers robust debugging capabilities, allowing developers to inspect variables, set breakpoints, and step through code.
5. Layout Editor and Theme Editor: Android Studio includes a visual layout editor that enables developers to create and modify app layouts quickly. The theme editor allows for easy customization of app themes and styles.
6. Performance and Memory Profiling: Android Studio offers profiling tools to analyze and optimize app performance. Developers can monitor CPU usage, memory allocation, network activity, and more to identify bottlenecks and improve app efficiency.

7. Android SDK Integration: Android Studio seamlessly integrates with the Android Software Development Kit (SDK) and provides easy access to a wide range of Android APIs and libraries. It simplifies the process of incorporating functionalities like maps, camera, sensors, and more into Android apps.
8. Instant Run: Android Studio features Instant Run, which enables developers to quickly apply code changes and see the results immediately on a connected device or emulator. This speeds up the development and testing cycle.
9. Version Control and Collaboration: Android Studio supports version control systems like Git, allowing developers to manage source code changes, collaborate with team members, and track revisions effectively.
10. Extensibility: Android Studio is highly extensible through the use of plugins. Developers can customize the IDE by installing plugins for additional features, tools, and integration with other services.

Android Studio is continuously updated and improved by Google, incorporating new features and enhancements to support the latest Android platform developments. Android Studio has become the go-to IDE for Android app development due to its robust features, tight integration with Android SDK, and active community support. It is continuously updated with new features, bug fixes, and improvements to support the evolving Android platform. It provides an integrated and efficient environment for developing Android apps, whether you are a beginner or an experienced developer.

We used the Android Studio application to design, make and test our own application named, Paasss Secuirty which provides access to the live feed from the esp32-cam camera to the android device on which the application is installed.

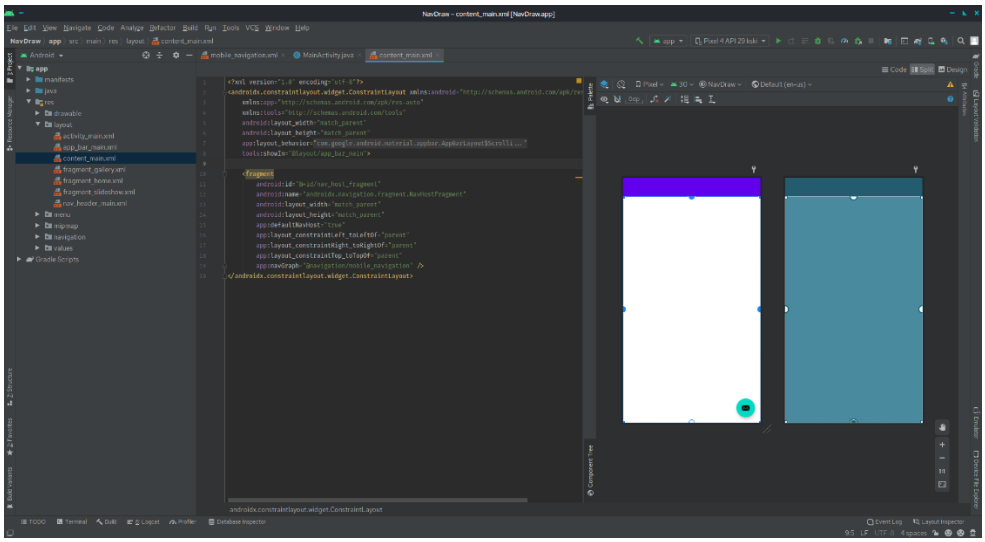
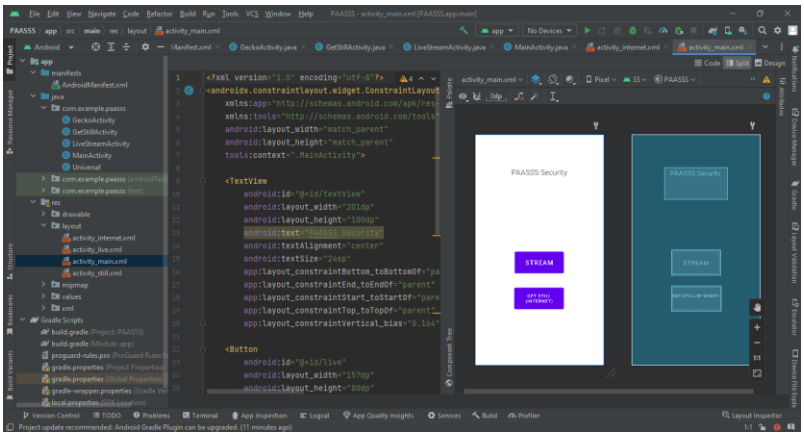


Fig 6.3: Android Studio interface (Source: Wikipedia)

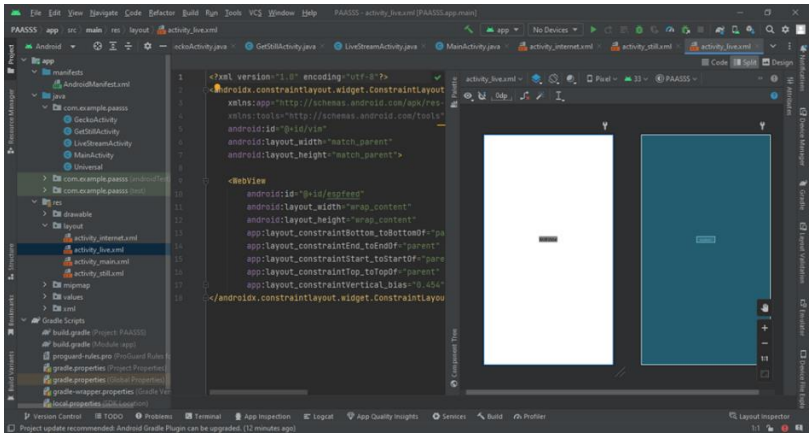
6.3 Paasss Security Android App

We designed the paasss application to access the feed from the camera in a user friendly manner, without any hassle. The work flow of creating the app is described below.

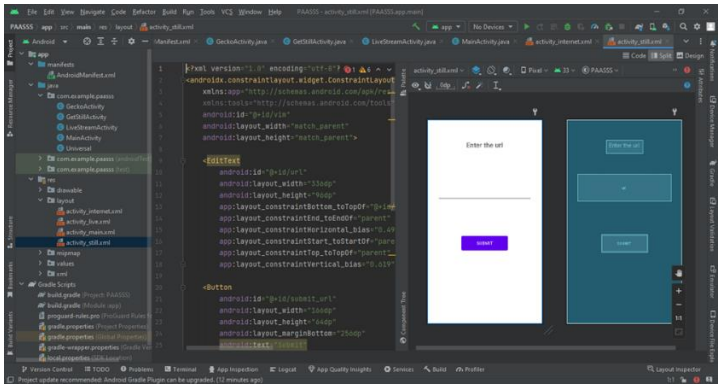
- First we made the main activity layout of our App, it renders each time the user opens the app or moves back to the home screen. This layout controls the proceedings that can be done by the user. Two buttons exits in this layout each having their own separate layouts and view pages.



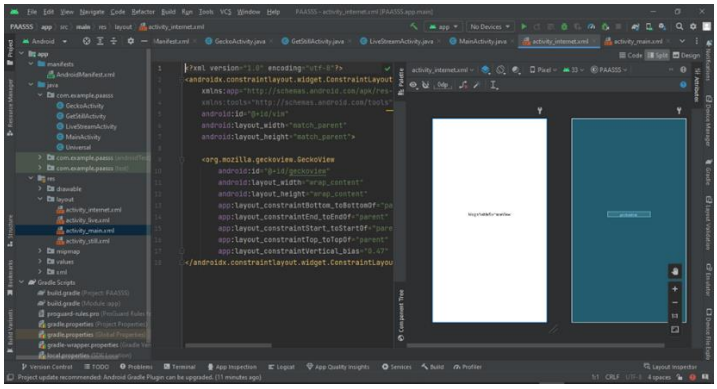
- Live stream activity controls the webview for the ip that the video stream from esp32-cam is being hosted. It contains a single webview element within a constraint layout.



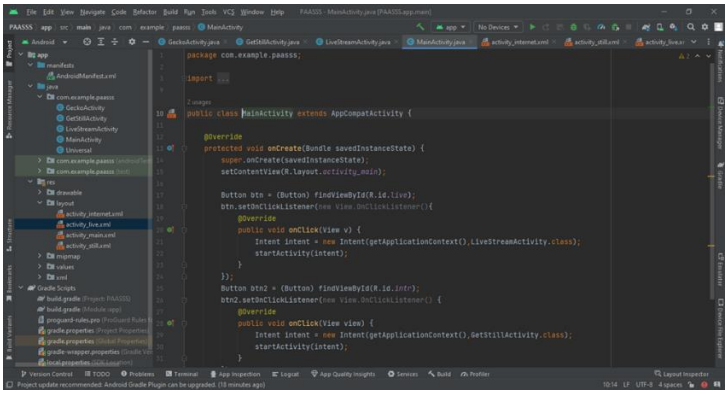
- The get still activity xml layout has two elements – Input bar, and Submit button. It opens a geckoView in a following constraintLayout with the url that has been given to the input bar, also it persists the input until new input is given.



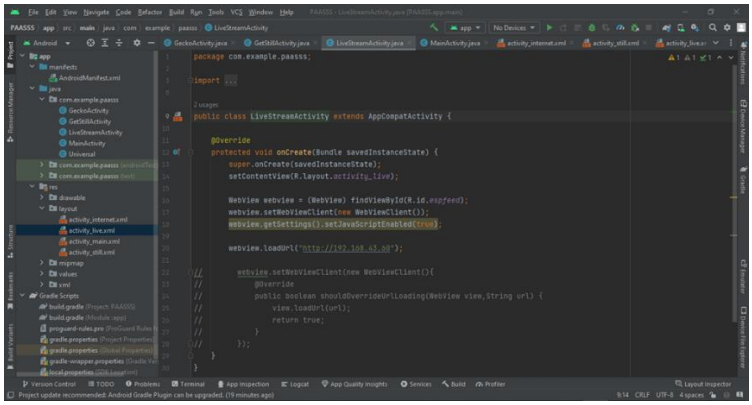
- GeckoView with URL for accessing the footage of intruder over the internet.



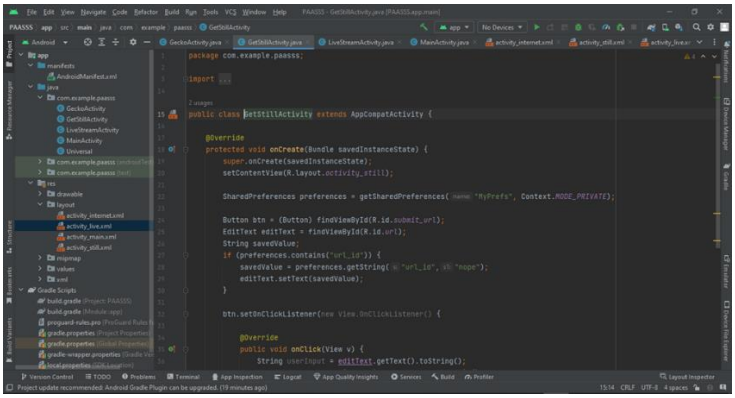
- MainActivity is the primary class that controls the launching of the app and responding accordingly to the inputs.



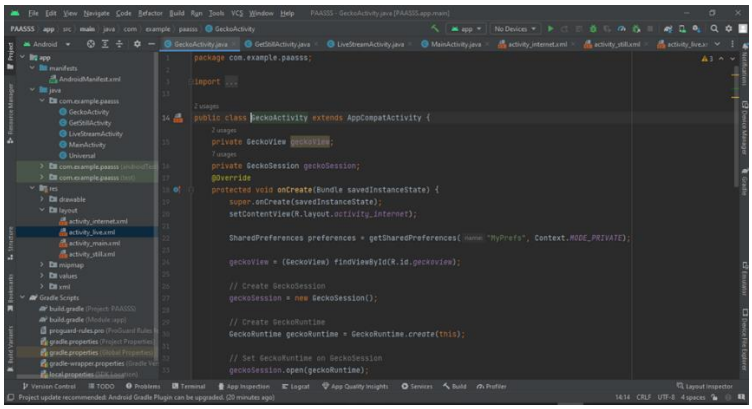
- LiveStreamActivity Class has single webview loaded with a fixed ip.



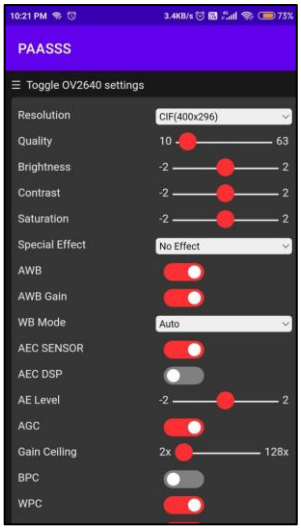
- GetStillActivity class has the handlers for url input and storing it until the user updates the new url. It also loads the activity_still.xml layout.

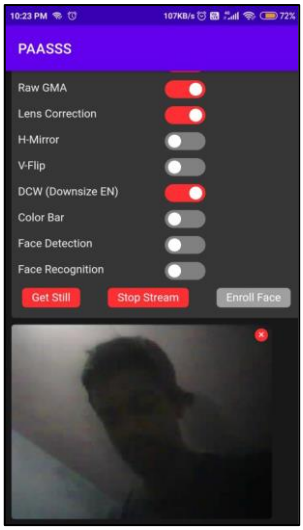


- Finally the GeckoActivity class has a custom GeckoView for loading the persistent url with its own runtime and geckoSession.



- Following are the screenshots of the Paasss App:





6.4 LocalXpose Tunnel Service

LocalXpose is a powerful service that facilitates the secure exposure of local web servers and applications to the internet. It eliminates the complexities associated with network configurations and remote server deployments, making it easier to test and share locally hosted services.

The primary purpose of LocalXpose is to create a secure tunnel between your local machine and the internet. This tunnel allows external users to access your local services as if they were running on a public server. By assigning a unique URL to your local service, LocalXpose ensures that it can be accessed from any device with an internet connection[8].

One of the key advantages of LocalXpose is its focus on security. The service employs encryption and authentication mechanisms to safeguard your data during transmission. This ensures that sensitive information remains protected while being accessed remotely. LocalXpose also offers options for custom domains, subdomains, and TCP/UDP forwarding, enabling you to tailor the setup to your specific needs.

LocalXpose provides a user-friendly experience through its simple command-line interface (CLI) and intuitive web dashboard. These interfaces allow you to manage your tunnels effortlessly, monitor traffic, and configure settings with ease. The service supports a variety of operating systems, including Windows, macOS, and Linux, ensuring broad compatibility.

By using LocalXpose, developers can easily share their work-in-progress applications with clients or colleagues without the need for deploying to a remote server. It simplifies the debugging and testing processes by allowing external access to the application in its development environment. This feature is particularly beneficial when working on applications that require specific setups or when collaborating with remote teams.

Additionally, LocalXpose can be employed for various use cases, including website development, API testing, IoT device communication, and more. It provides a versatile solution for securely exposing locally hosted services to the internet, enabling efficient collaboration and faster development cycles.

In summary, LocalXpose streamlines the process of making local services accessible over the internet. Its security features, user-friendly interfaces, and flexibility make it an invaluable tool for developers, testers, and individuals seeking a hassle-free way to share and access locally hosted applications[3].

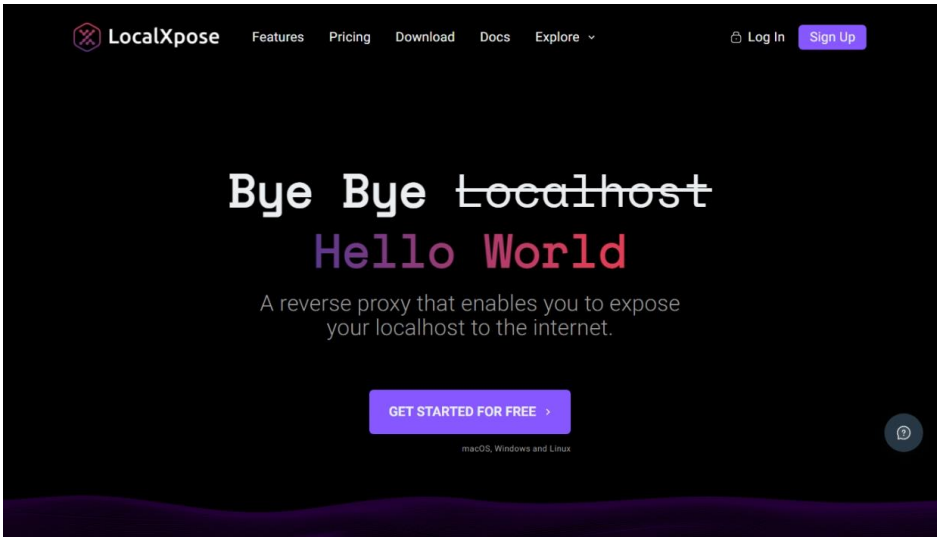


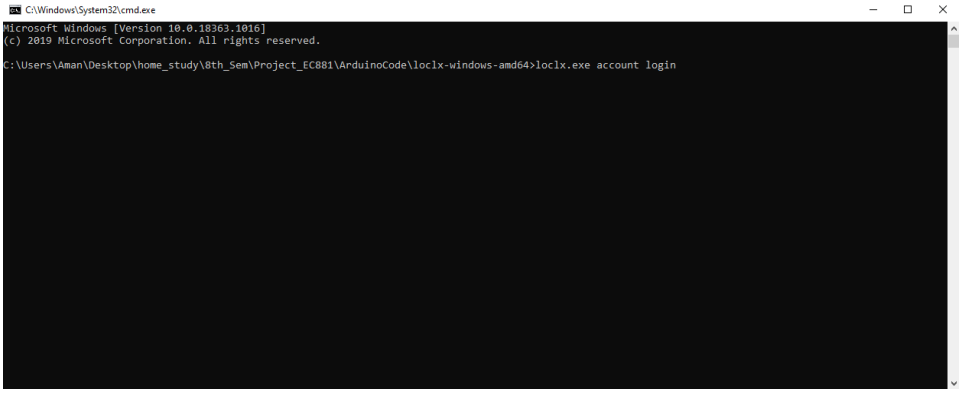
Fig 6.4: LocalXpose webpage

6.4.1 Getting Started with LocalXpose:

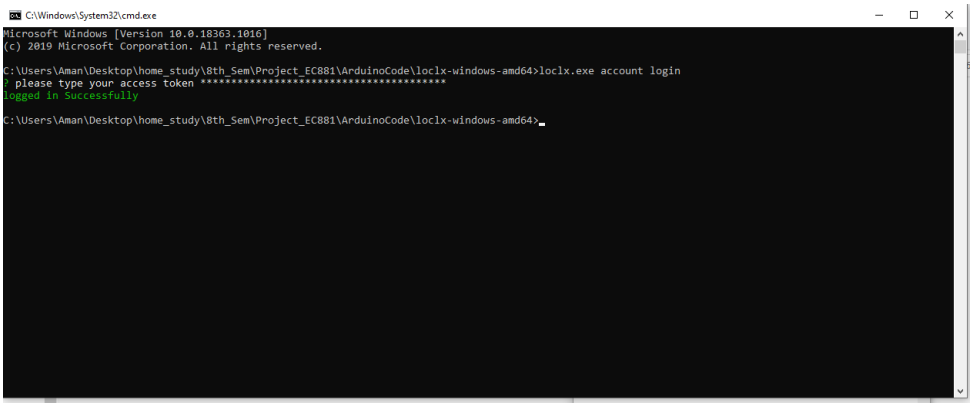
LocalXpose has two options, Command line (CLI) or Graphical User Interface (GUI), it is cross-platform (Linux, Windows, MacOS), you can download the suitable binary for your machine from the home page at https://localxpose.io.

To start the tunnel service we can follow the given steps for Windows OS:

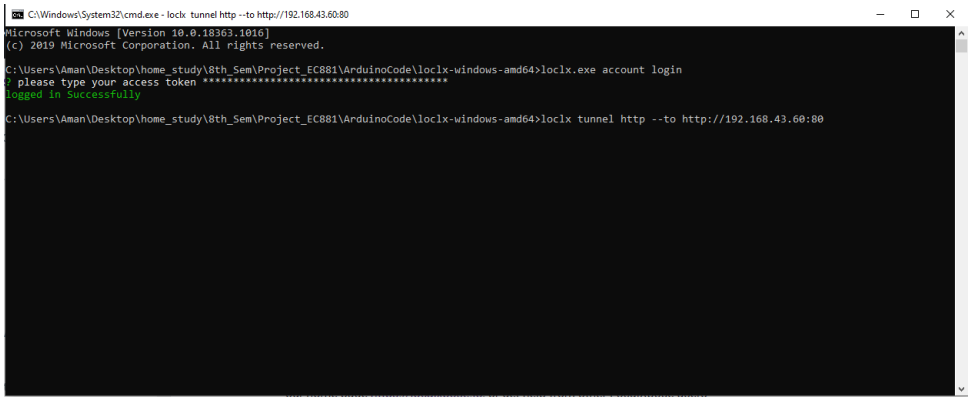
- Signup from the home page https://localxpose.io
- Verify your email address by clicking the Verify button in your email inbox.
- Login to your dashboard by clicking Login button in the home page https://localxpose.io
- Copy your access token that will be under access tab in your dashboard or go directly here https://localxpose.io/dashboard/access.
- Download the CLI app from the homepage.
- This is a CLI app, you should start it from a terminal window (double click the app will not work), so for example in windows open your cmd.exe or powershell then navigate to your app.
- Login from CLI by calling: loclx.exe account login



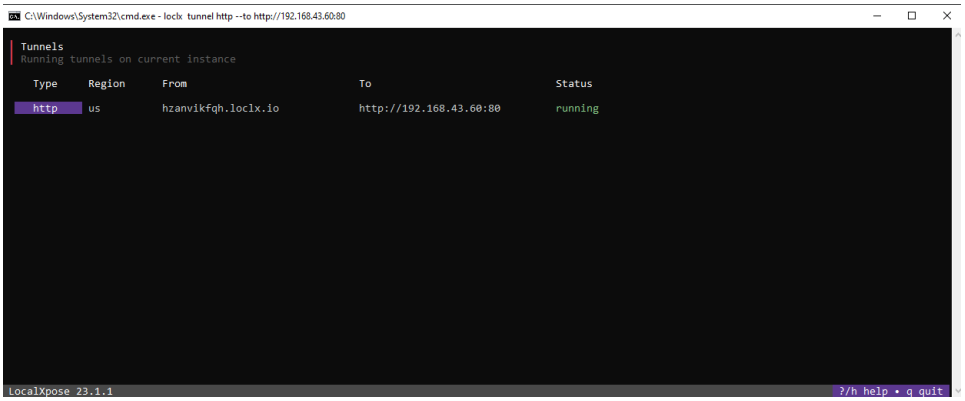
- Then enter your access token that you copied earlier from your dashboard.



- Start your tunnel using the following command: loclx tunnel http –to http://PORT-NO



- The tunnel is active now.



- Now you can use the link in the “From” column to access the local IP over the internet.

6.5 Twilio SMS Service

Twilio is a cloud communications platform that provides developers with a set of APIs (Application Programming Interfaces) for building communication and messaging features into their applications. It offers a wide range of communication services, including voice calling, video calling, SMS (Short Message Service), MMS (Multimedia Messaging Service), and more.

Twilio's APIs allow developers to integrate various communication channels directly into their software applications, websites, or services. With Twilio, developers can programmatically send and receive SMS and MMS messages, make and receive phone calls, handle conference calls, implement two-factor authentication, and even build interactive voice response (IVR) systems[14]. The platform provides robust documentation, software development kits (SDKs), and libraries for various programming languages, making it accessible and easy to integrate with existing applications.

Twilio's services are scalable and reliable, with built-in redundancy and failover mechanisms. This ensures high availability and the ability to handle large volumes of communication traffic.

Twilio has found extensive use in a wide range of industries and applications. It is commonly used for customer support and engagement, appointment reminders, notifications, verification systems, call tracking, virtual phone systems, and more.Overall, Twilio provides developers with a robust set of APIs and tools for integrating communication capabilities into their applications. Its flexible and scalable platform enables businesses to enhance their communication strategies and create unique customer experiences[16].

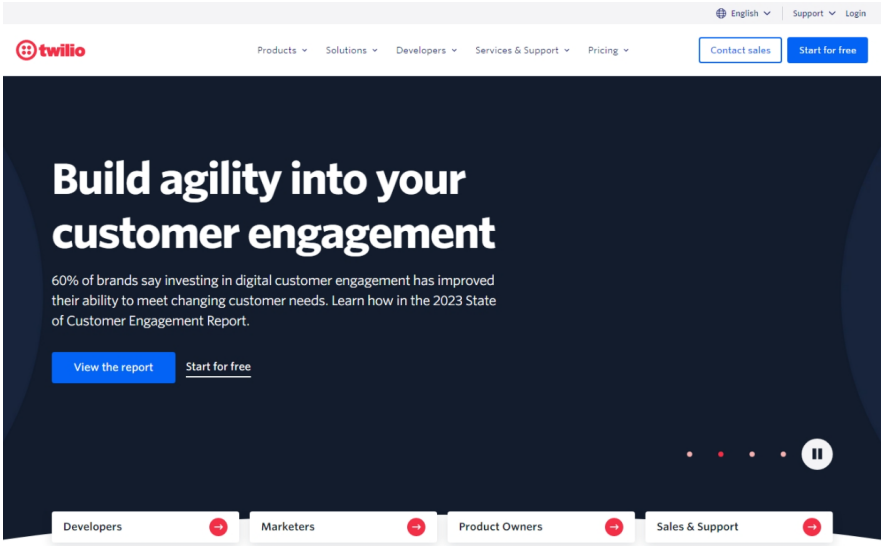


Fig 6.5: Twilio Webpage

7. Block Diagram

7.1. Block Diagram of Fingerprint Based Door Lock

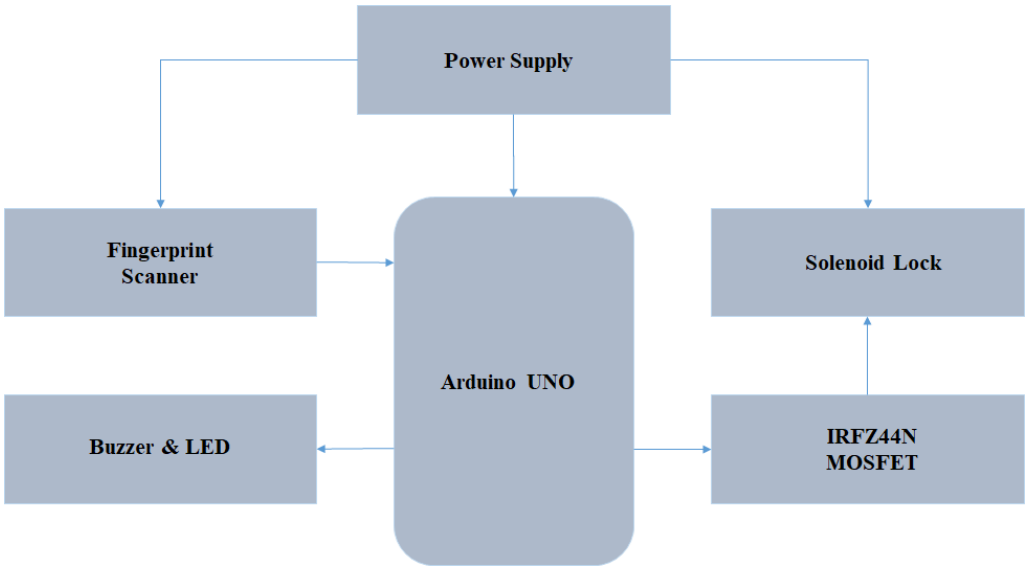


Fig 7.1: Fingerprint based Door Lock System Block Diagram

7.2 Block Diagram of Motion Sensing System

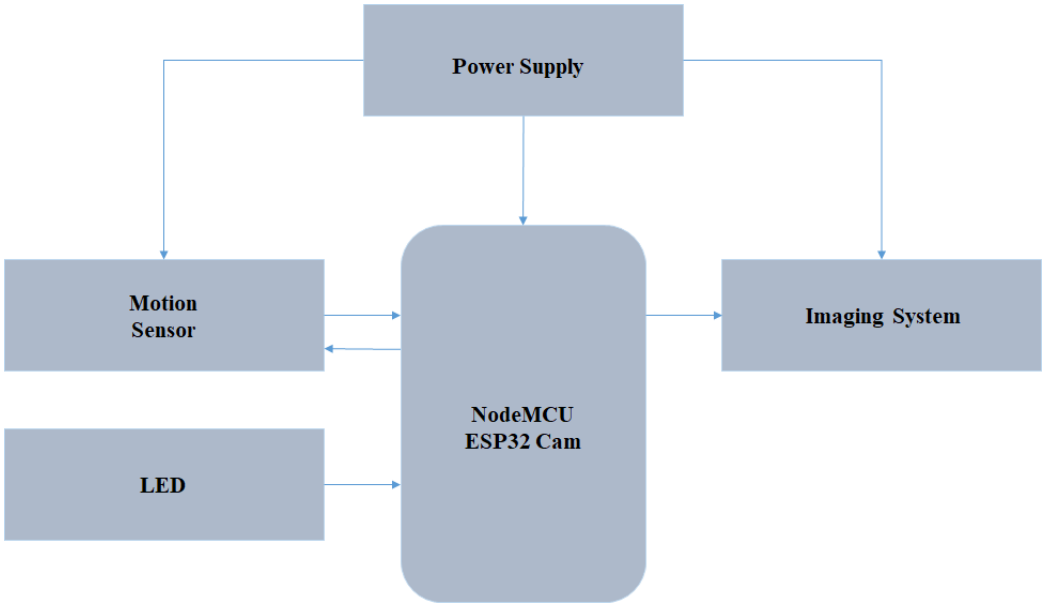


Fig 7.2: Motion Detection System Block Diagram

7.3 Block Diagram of Imaging System

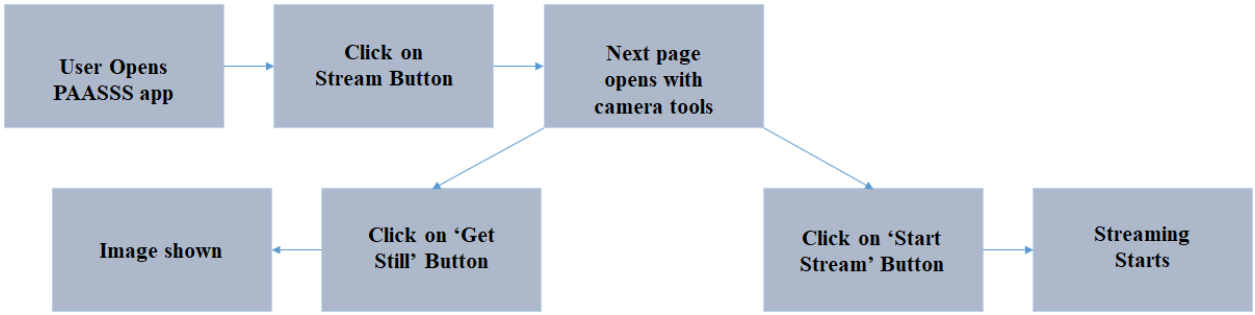


Fig 7.3: Imaging system Block Diagram

7.4 Block Diagram of the Security System

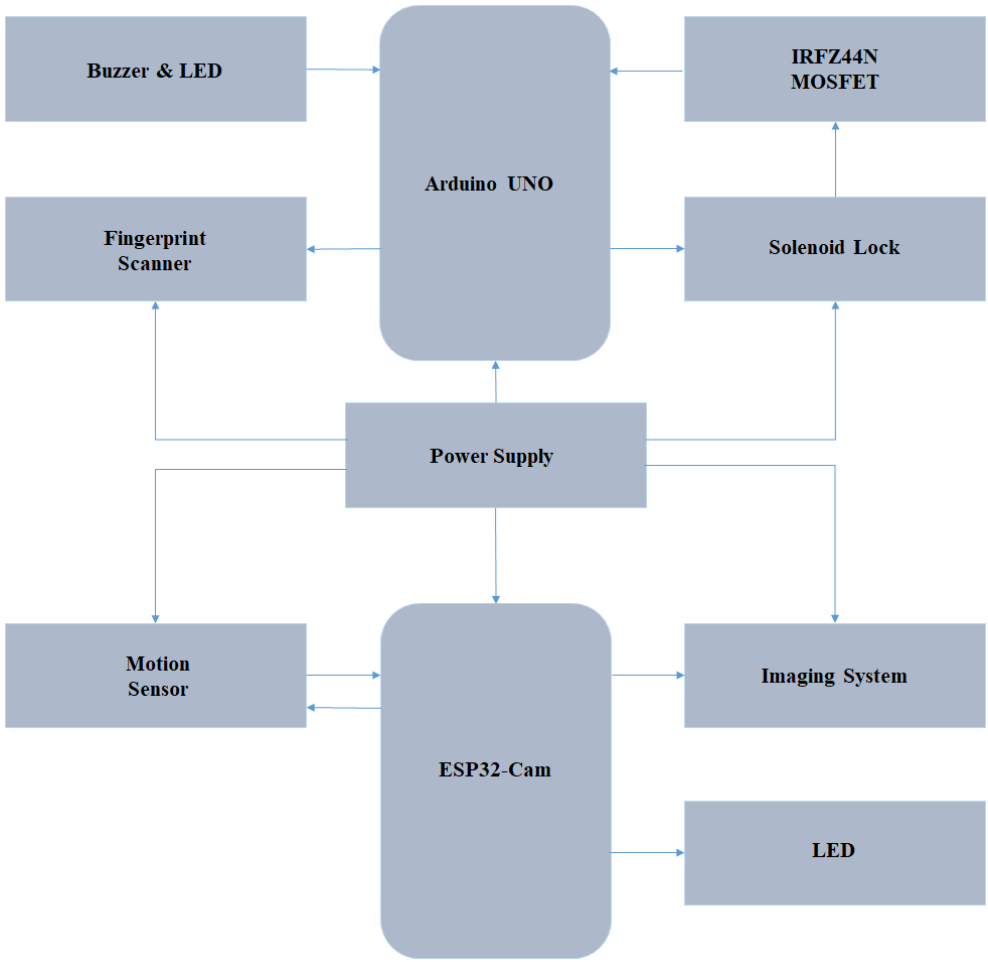


Fig 7.4: Security System Block Diagram

8. Flowchart

8.1 Flowchart of Fingerprint Enrollment

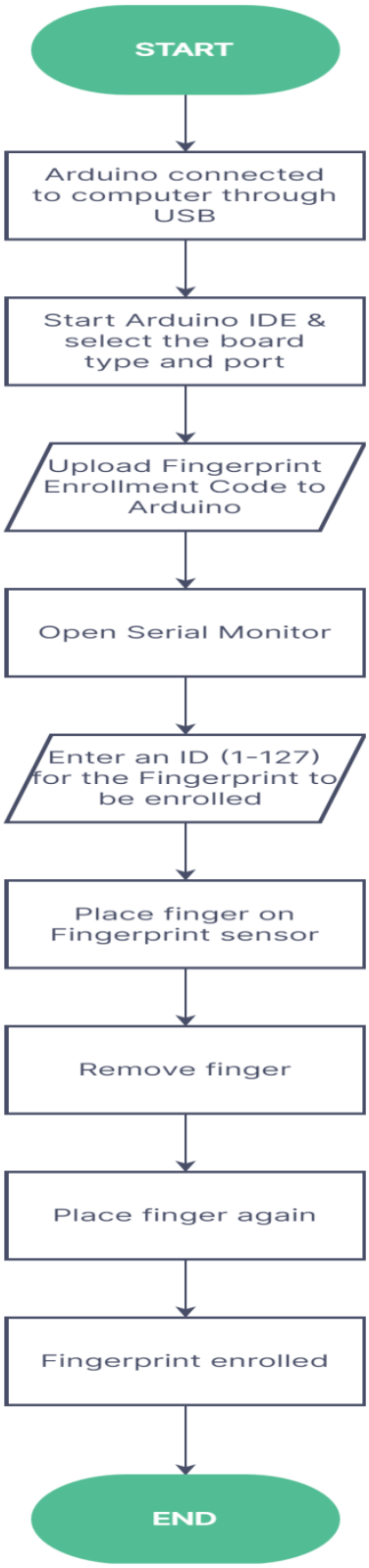


Fig 8.1: Flowchart for Fingerprint Enrollment

8.2 Flowchart of Fingerprint Authentication Module

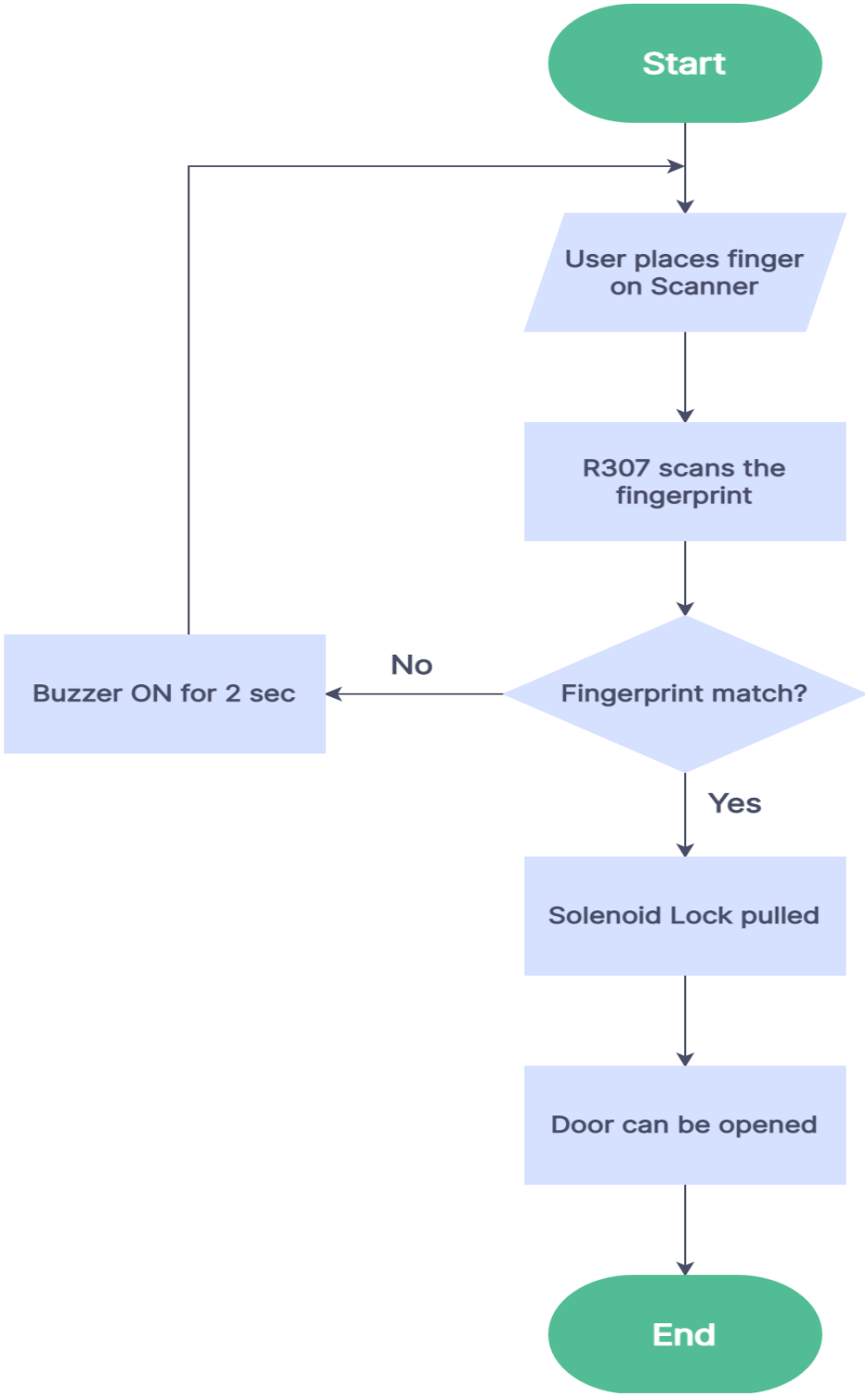


Fig 8.2: Flowchart for Fingerprint Authentication

8.3 Flowchart of Live Camera Streaming Module

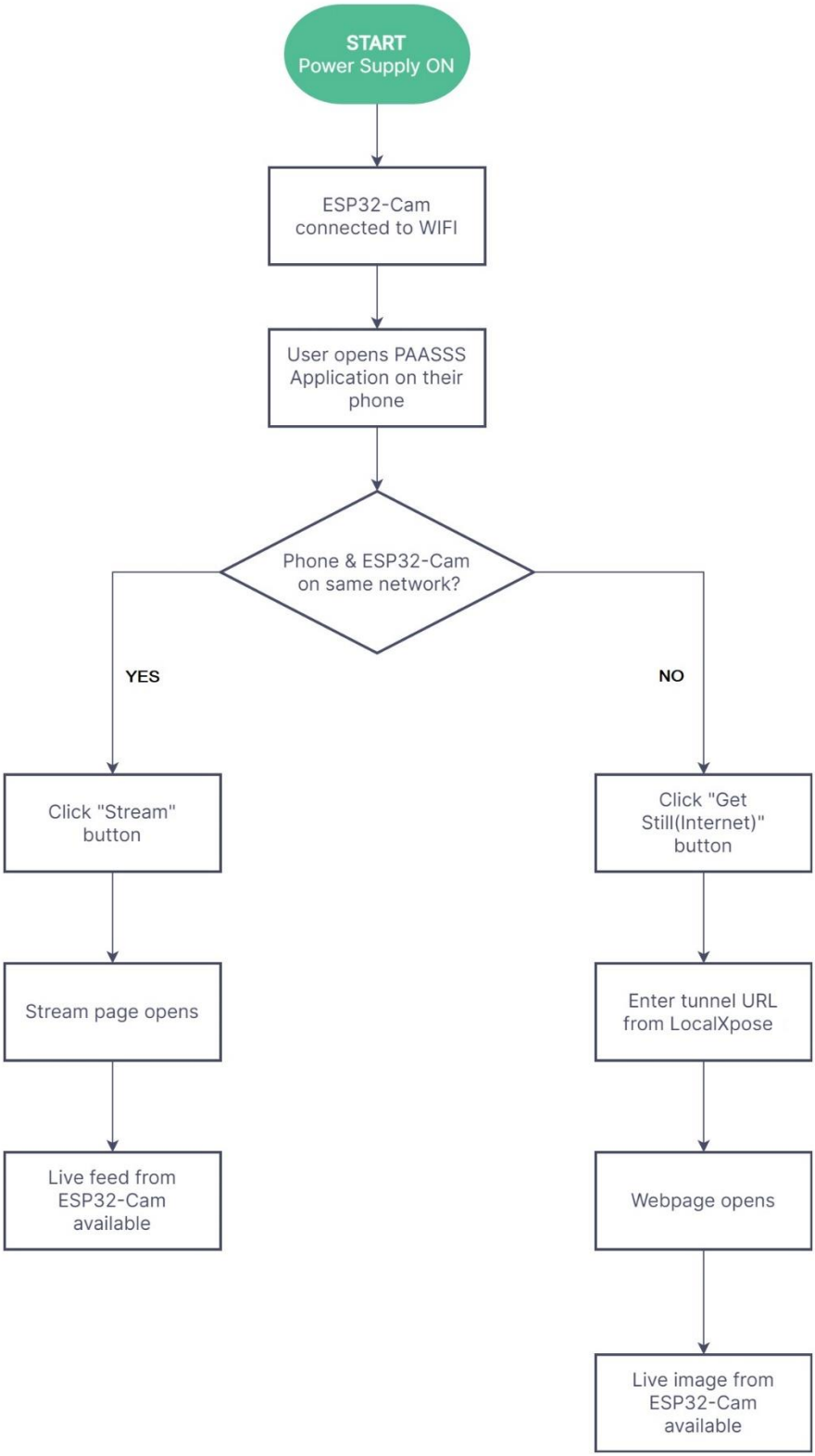


Fig 8.3: Flowchart of Live Camera Streaming

8.4 Flowchart of Motion Detection Module

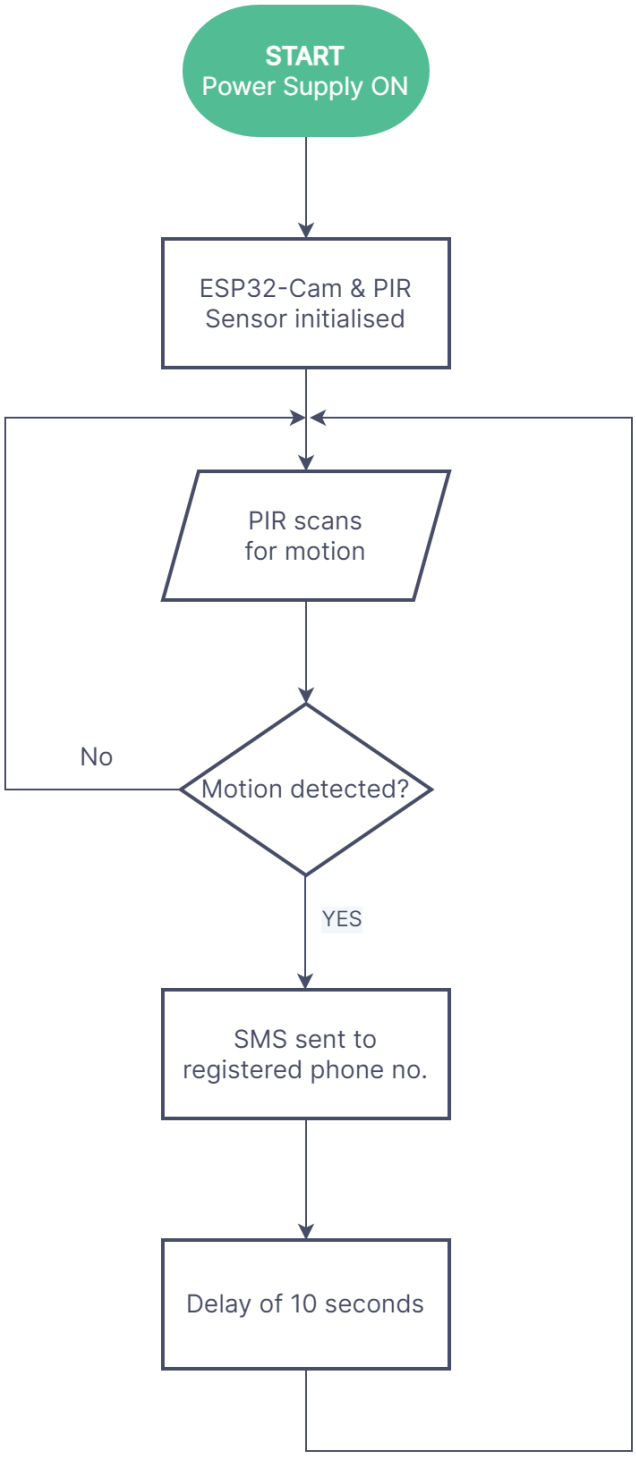


Fig 8.4: Flowchart of Motion Detection

9. Circuit Diagram

9.1 Circuit diagram of Fingerprint Authentication Module

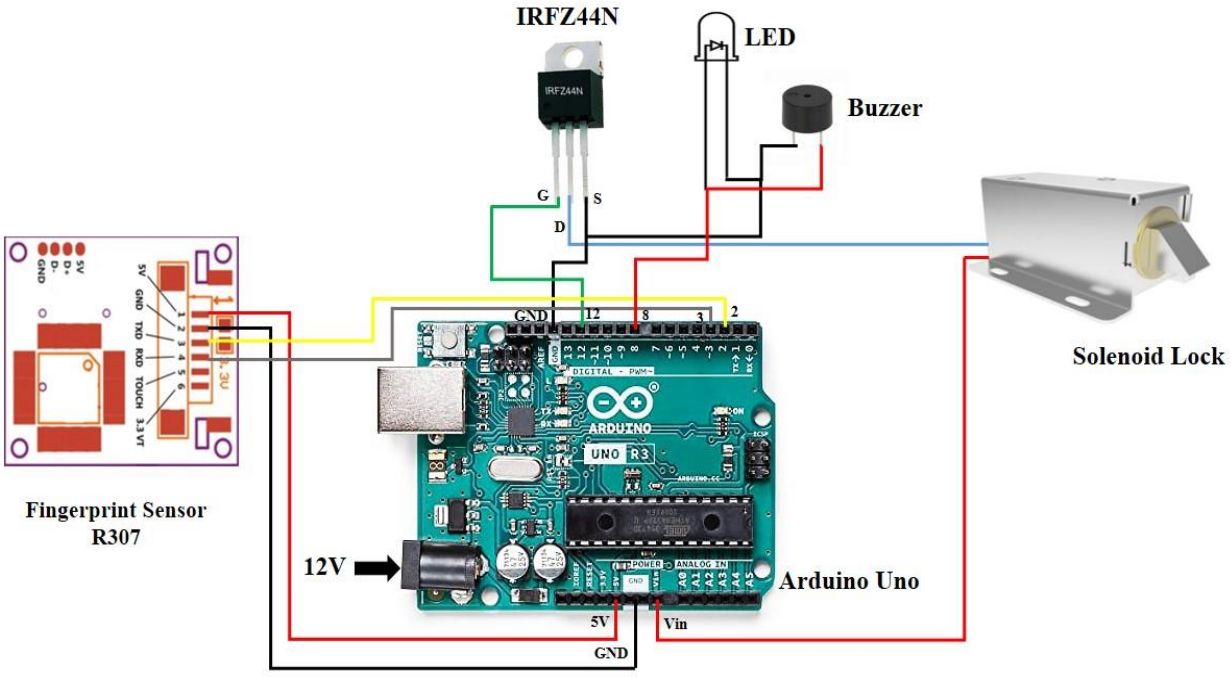


Fig 9.1: Circuit Diagram of Fingerprint Authentication

9.2 Circuit diagram for Uploading Code in ESP32-cam

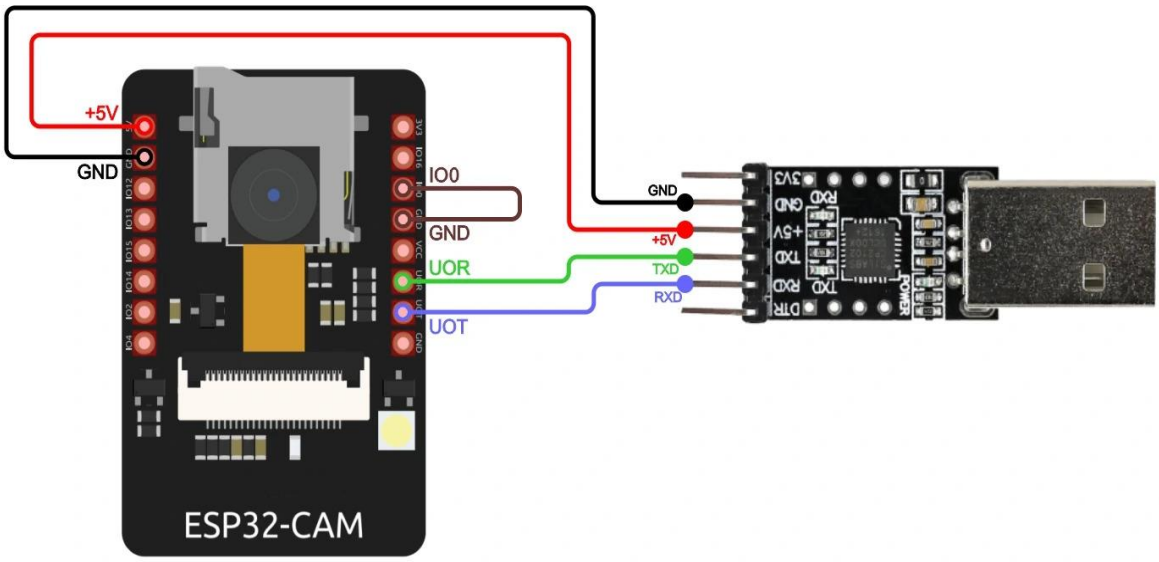


Fig 9.2: Circuit Diagram for uploading code in esp32-cam

9.3 Circuit Diagram of Motion Detection and Alert Module

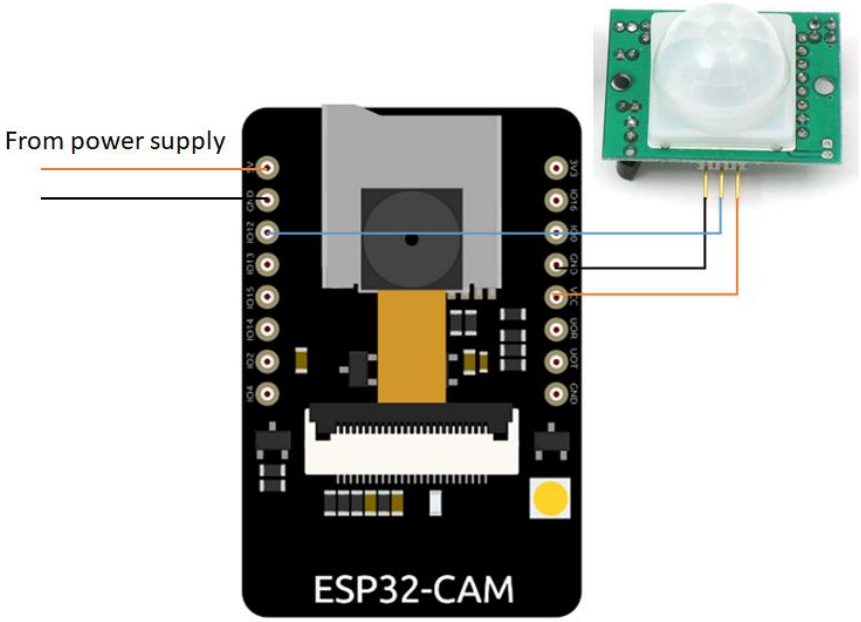


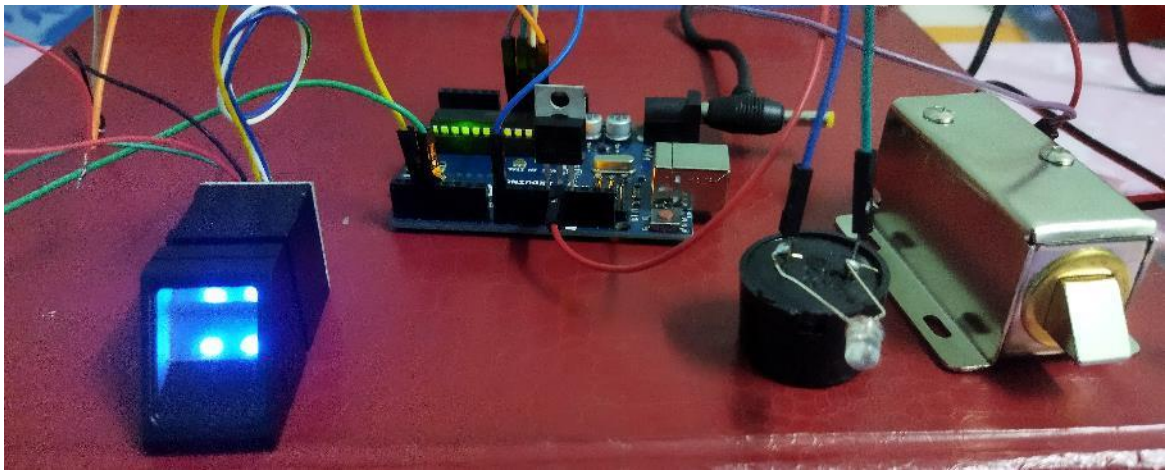
Fig 9.3: Circuit Diagram of Motion Detection Module

10. Test Methodology and Result

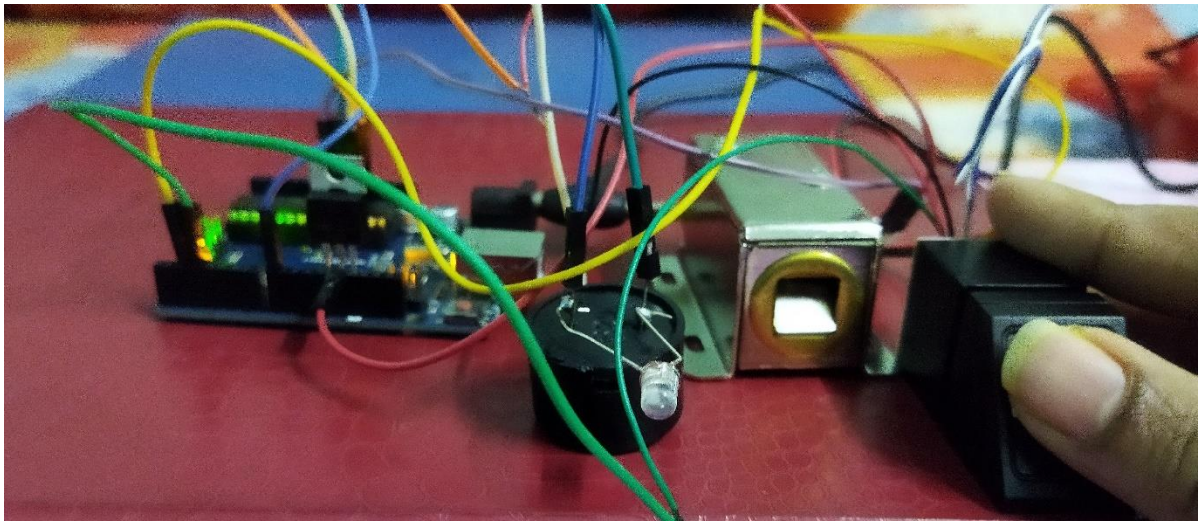
10.1. Fingerprint Authentication Module Testing

The testing for this module is done using appropriate hardware components and relevant system. Initially all the components are connected as per the circuit diagram. Then the program to add the authorized persons' fingerprint to the database is loaded into the Arduino. Then the fingerprint is added to the database. After the fingerprint is added to the database, the code to authenticate the fingerprint is loaded in the Arduino. Now, the system is ready to be used.

The power supply is connected to the Arduino through the DC power jack. The system is initiated and running. A light starts to blink in the fingerprint scanner as shown in the image below.



On placing a finger on the scanner which is already loaded in the database of fingerprints, the system recognizes it and retracts the solenoid lock, thus unlocking the door, as shown in the image below. For testing purpose, after 4 seconds the lock returns to the locked state.



If any unauthorized person places their finger on the scanner, the system detects that the fingerprint is not present in its database, so the buzzer starts making a sound and the LED glows for 2 seconds, as shown in the image below.

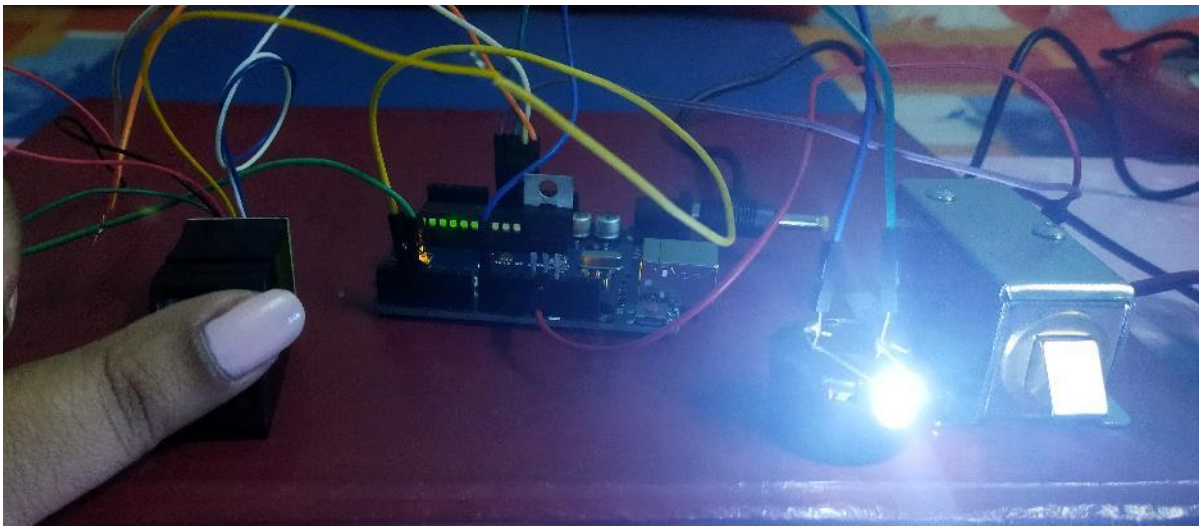


Table 10.1: Table of Fingerprint Scanning System Testing

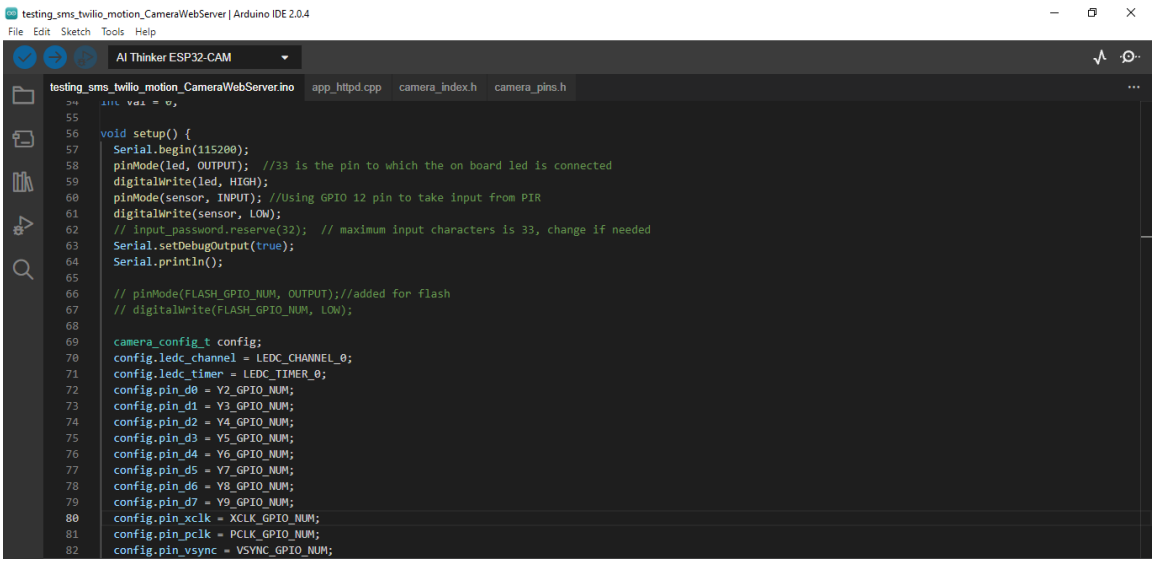
Test Case	**Expected Result**	**Observed Result**	**Test Result**
Register Fingerprint	Registered Successfully	Registered Successfully	PASS
Unlock Door using authorized fingerprint	Door Unlocks Successfully	Door Unlocks Successfully	PASS
Unlock Door using unauthorized fingerprint	Door does not unlocks, buzzer and LED activated	Door does not unlocks, buzzer and LED activated	PASS

The system works accurately in detecting the authorized person, so, the working of the circuit is appropriately tested.

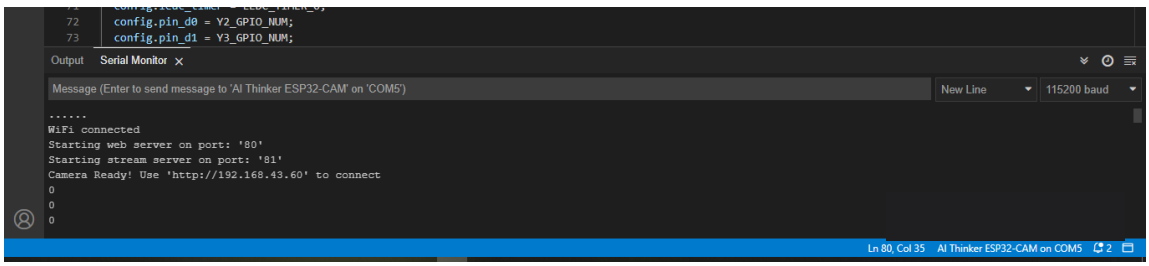
10.2. Imaging System Testing

The imaging system is based on the esp32-cam board. The board comes with a built in 24-pin FPC Connector for connecting the OV2640 camera. We use this camera for getting the live camera feed and photos.

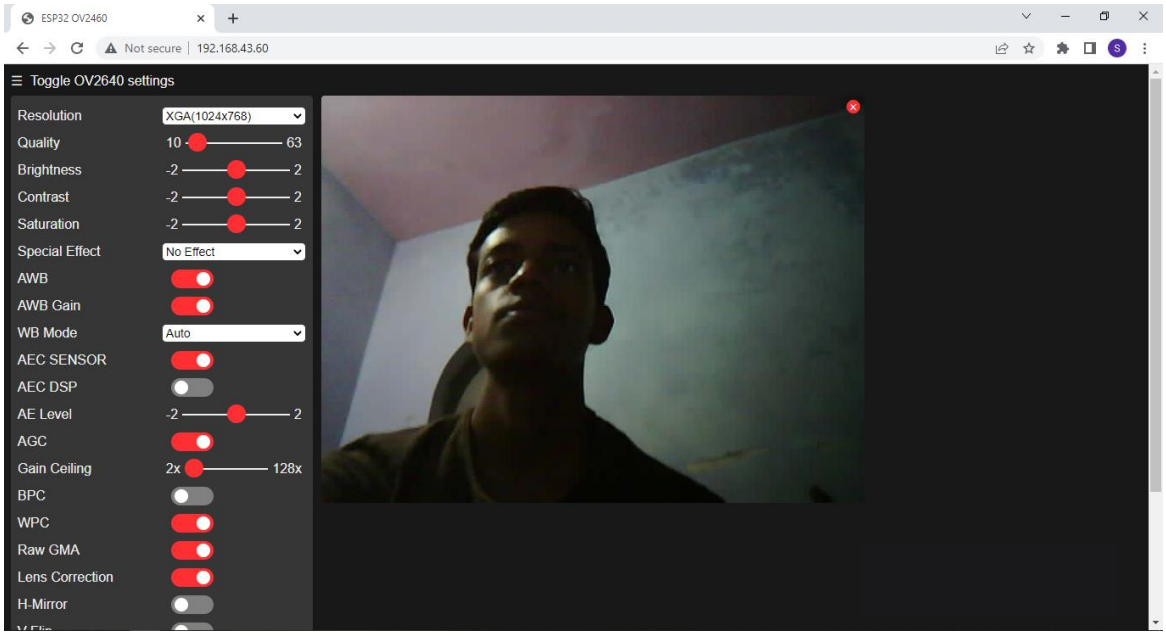
Connect the esp32-cam to your computer using the USB to TTL device and connect the GPIO0 pin to ground to put the board in flash mode (for uploading code), then select the board type and the port to which it is connected in the Arduino IDE. Then we upload the appropriate code to the board from the application.



After uploading the code, the GPIO0 pin is disconnected from the ground, and the reset pin is pressed. Then we open the Serial Monitor in the Arduino IDE application. The IP address for the esp32-cam is shown in the serial monitor.



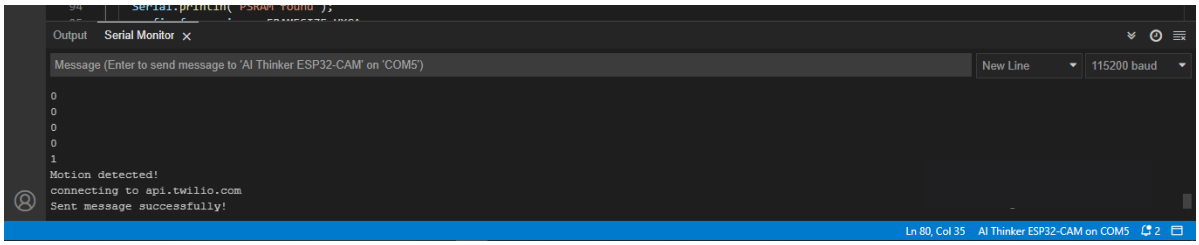
From the serial monitor, note down the IP address. Now enter the IP address in any browser to access the page for viewing the camera feed. After the page opens, click on “Get Still” to get an image from the camera, or click on “Start Stream” to get the live video feed from the camera.



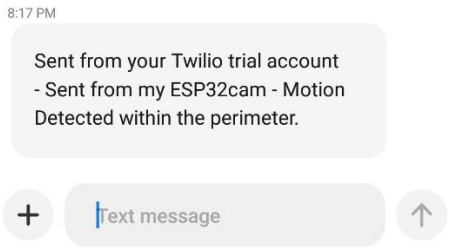
The IP address is able to show the live video and photo from the camera. So, the testing is successful and the system is working accurately.

10.3. Motion Detection and Alert Testing

We use a PIR sensor for sensing motion within its range. The PIR sensor is connected to the ESP32-cam and to send alert to the registered number, we are using Twilio. The power supply is turned on and the ESP32-cam connects with the configured Wi-fi. When Motion is detected, the onboard LED on ESP32-cam turns on and a text SMS is sent to the provided mobile number.



Next we receive the text message on our provided phone number.



The motion detection and messaging service is working accurately so the testing is successful.

11. Future Work

While our security system project has successfully achieved its primary objectives, there are several avenues for future work and enhancements that can further improve its functionality and expand its capabilities. Here are some potential directions for future development:

- **Integration of Additional Biometric Authentication:** Explore the integration of facial recognition or iris scanning as alternative biometric authentication methods, providing users with more options for access control.
- **Enhancing Camera Feed Access:** Enhance the live camera feed access by incorporating advanced video analytics capabilities, such as object detection, facial recognition, or license plate recognition, for improved monitoring and identification.
- **Cloud Integration and Remote Management:** Implement cloud integration to enable remote management and access to the security system from anywhere, with features such as web-based interfaces and cloud storage for camera footage and biometric data backup.
- **Advanced Motion Detection Algorithms:** Develop more intelligent motion detection algorithms, leveraging machine learning techniques to distinguish between different types of movements and reduce false alarms.
- **Multi-factor Authentication:** Incorporate multi-factor authentication by combining fingerprint authentication with other factors like PIN codes, proximity cards, or voice recognition, enhancing the system's overall security.
- **Integration with Home Automation Systems:** Integrate the security system with existing home automation systems or smart home platforms to enable seamless control, automation, and integration with other smart devices for enhanced security and convenience.
- **User Management and Access Control:** Develop a comprehensive user management system that allows administrators to easily add or remove users, manage access privileges, and track user activity logs.
- **Mobile Alerts and Notifications:** Expand the alert feature to support additional notification channels, such as mobile push notifications or email alerts, for users to receive security notifications and respond promptly using their mobile devices.

These future work possibilities provide avenues for further development, refinement, and adaptation of the security system to meet evolving security needs and leverage emerging technologies.

12. Discussion & Conclusion

During the course of this project, we successfully designed and implemented a security system that integrates fingerprint-based door locking, live camera feed access, and motion detection with SMS alerts. Through this comprehensive system, we aimed to provide enhanced security, convenience, and peace of mind to users.

In the discussion phase, we analyzed various aspects of the project, including the system design, hardware implementation, software development, and testing. We meticulously selected components and technologies that aligned with the project objectives, considering factors such as security, reliability, effectiveness and cost-effectiveness. The fingerprint-based door locking system demonstrated accurate and secure authentication, allowing only authorized individuals to access the protected area. The live camera feed access provided users with real-time monitoring capabilities, empowering them to remotely observe the premises and respond promptly to potential security threats. The motion detection feature effectively detected movements within the designated perimeter and triggered SMS alerts, ensuring immediate notification of suspicious activities.

By fulfilling the project's objectives, we have made significant strides in improving security measures. Our system offers an advanced level of protection compared to traditional lock and key mechanisms and standalone surveillance systems. However, it is important to acknowledge that there are certain limitations and areas for further improvement.

In conclusion, our project has successfully demonstrated the feasibility and effectiveness of integrating fingerprint-based door locking, live camera feed access, and motion detection with SMS alerts into a unified security system. The system offers enhanced security, convenience, and peace of mind to users, providing them with robust access control and real-time surveillance capabilities.

Moving forward, it is crucial to continue refining the system based on user feedback and emerging technologies. Future work could focus on integrating additional biometric modalities, improving motion detection algorithms, enhancing user management capabilities, and many other improvements. These advancements would further elevate the system's capabilities and address its limitations, ultimately providing a more comprehensive and advanced security solution.

Overall, this project has been a valuable learning experience, enabling us to gain practical knowledge in the fields of biometrics, video surveillance, and security systems. We hope that our efforts contribute to the broader goal of enhancing security measures and promoting a safer environment for individuals and organizations alike.

References

1. Gartner Says 4.9 Billion Connected Things Will Be in Use in 2015. https://www.gartner.com/newsroom/id/2905717.
2. Ul Rehman, S., Manickam, S.: A study of smart home environment and its security threats.Int. J. Reliab. Qual. Saf. Eng. 23(3), 1–9 (2016)
3. https://localxpose.io/docs/
4. Pirbhulal, S., et al.: A novel secure IoT-based smart home automation system using awireless sensor network. Sensors 17(1), 69 (2016)
5. https://developer.android.com/sdk/installing/studio.html
6. Gheorghe, A.: The Internet of Things: Risk in the Connected Home (2016)
7. https://www.rajguruelectronics.com/Product/1276/R307%20Fingerprint%20Module.pdf
8. https://localxpose.io/blog/best-ngrok-alternatives
9. Lin, H., Bergmann, N.: IoT privacy and security challenges for smart home environments.Information 7(3), 44 (2016)
10. https://robu.in/wp-content/uploads/2019/06/ESP32-CAM-WiFi-Module-Bluetooth-with-OV2640-Camera-Module-2MP-1.pdf
11. https://randomnerdtutorials.com/arduino-with-pir-motion-sensor/
12. https://robu.in/pir-sensor-working-principle/
13. https://randomnerdtutorials.com/send-notifications-esp32/
14. ChatGPT: https://chat.openai.com/chat
15. https://eloquentarduino.com/esp32-cam-motion-detection/
16. https://randomnerdtutorials.com/send-sms-esp32-twilio/#verify-recipient-numbers
17. https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf
18. https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/
19. https://developer.android.com/studio
20. https://android-developers.blogspot.in/2013/05/android-studio-ide-built-for-android.html