IOT Based Smart Door Lock System Using Arduino

A Project report submitted in partial fulfilment

of the requirements for the degree of B. Tech in Electrical Engineering
By

Arnab Debnath (11701619018)
Saswata Samanta (11701619021)
Poulami Das (11701619022)

Under the supervision of

Dr. Alok Kole
Professor
Department of Electrical Engineering

श्रमम् बिना न किमपि साध्यम्

Department of Electrical Engineering

RCC INSTITUTE OF INFORMATION TECHNOLOGY

CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL

Maulana Abul Kalam Azad University of Technology (MAKAUT)

श्रमम् बिना न किमपि साध्यम्

Department of Electrical Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL
PHONE: 033-2323-2463-154, FAX: 033-2323-4668
Email: hodeercciit@gmail.com, Website: http://www.rcciit.org/academic/ee.aspx

CERTIFICATE

To whom it may concern This is to certify that the project work entitled **IOT Based Smart Door Lock System Using Arduino** is the bonafide work carried out by **SASWATA SAMANTA (11701619021), ARNAB DEBNATH(11701619018), POULAMI DAS(11701619022)** the student of B.Tech in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, India, during the academic year 2022-23, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electrical Engineering and that this project has not submitted previously for the award of any other degree, diploma and fellowship.

Dr. Alok Kole
Professor
Department of Electrical Engineering
RCC Institute of Information Technology

Countersigned by
Dr. Shilpi Bhattacharya
HOD, Electrical Engineering
Dept RCC Institute of Information Technology

Acknowledgement

It is my great fortune that I have got opportunity to carry out this project work under the supervision of Dr. Alok Kole, Professor in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, India. I express my sincere thanks and deepest sense of gratitude to my guide for his constant support, unparalleled guidance and limitless encouragement.

I would also like to convey my gratitude to all the faculty members and staffs of the Department of Electrical Engineering, RCCIIT for their whole hearted cooperation to make this work turn into reality.

I am very thankful to our department and to the authority of RCCIIT for providing all kinds of infrastructural facility towards the research work.

ARNAB DEBNATH (11701619018)

SASWATA SAMANTA(11701619021)

POULAMI DAS (11701619022)

To,

The Head of the Department
Department of Electrical Engineering
RCC Institute of Information Technology
Canal South Rd. Beliaghata, Kolkata-700015.

Respected Ma'am,

In accordance with the requirements of the degree of Bachelor of Technology in the Department of Electrical Engineering, RCC Institute of Information Technology, we present the following thesis entitled "**IOT Based Smart Door Lock System Using Arduino**". This work was performed under the valuable guidance of Dr. Alok Kole, Professor, in the Dept. of Electrical Engineering.

We declare that the thesis submitted is our own, expected as acknowledge in the test and reference and has not been previously submitted for a degree in any other Institution.

Yours Sincerely,

ARNAB DEBNATH (11701619018)

SASWATA SAMANTA (11701619021)

POULAMI DAS (11701619022)

CONTENT

LIST OF PICTURES

Topic **Page no.**

LIST OF TABLES

Topic **Page no.**

Abstract

A smart lock is a new line in home security, and along with the likes of Amazon's Alexa and Google Home, it's the next step towards creating the smart homes of the future. Put simply, it's an electronic lock that can be locked or unlocked remotely using your smartphone or by using your fingerprint. Removing the need for physical keys, which can be easily lost or forgotten, smart locks secure your home with a biometric system. The expected outcome of this project is to make a smart door lock using Arduino and ESP32 module and a camera integrated with ESP32 to wirelessly operate the door lock and also to integrate fingerprint sensor to unlock the door lock. The door lock will get power from 12 Volt DC supply. The door lock is wirelessly controlled by Blynk application. This is smart and cost-effective approach to make a smart door lock system. This progress report contains the block diagram and working principle of the smart door lock.

CHAPTER 1

Introduction

1.1 Background of Study and Motivation

These days office/corporate environment security is a major threat faced by every individual when away from home or at the home. When it comes to security systems, it is one of the primary concerns in this busy competitive world, where human cannot find ways to provide security to his/her confidential belongings manually. Instead, he/she finds an alternative solution which provides better, reliable and atomized security. This is an era where everything is connected through network, where anyone can get hold of information from anywhere around the world. Thus chances of one's info being hacked are a serious issue. Due to these risks it's very important to have some kind of personal identification system to access one's own information. Now a day, personal identification is becoming an important issue all around. Among mainstream personal identification methods, we mostly see password and identification cards techniques. But it is easy to hack password now and identification cards may get lost, thus making these methods quite unreliable.

There are certain situations which are very annoying like when a person locks himself out of his house or office or he leaves his key inside or sometimes when a thief just breaks the lock and steals everything. These kinds of situations always trouble people who use manual door lock with keys. Although in some places people use smart cards, there might arise a situation when someone loses the card or keeps the card inside. Then in other scenarios there are caretakers for locking houses or offices and keeping the keys safe. But then again there are times when a person in charge of the keys might not be available or has gone to some emergency routine, which can cause unwanted delay for people who need the key straightaway. These are some of the hassles that people might face when using keys or smart cards. That is when our system, fingerprint door lock system comes into play. Our design is implemented to provide better securities as users don't need to remember passwords and don't need any sort of keys or cards that often get lost. If someone's fingerprint is authorized in the systems he/she would not face any sort of delays to enter a room. Fingerprint recognition is one of the most secure systems because a fingerprint of one person never matches with others. Therefore, unauthorized access can be restricted by designing a lock that stores the fingerprints of one or more authorized users and unlock the system when a match is found. Bio-metrics authorization proves to be one of the best traits because the skin on our palms and soles exhibits a flow like pattern of ridges on each fingertip which is unique and immutable. This makes fingerprint a unique identification for everyone. The popularity and reliability on fingerprint scanner can be easily guessed from its use in recent hand-held devices like mobile phones and laptops.

This paper is about solving the problem regarding security of unauthorized people trespassing in our home, shops or offices. Security issues can be fixed using traditional locks but there is always possibility of someone opening the lock even without breaking it with the use of duplicate key. Using these kinds of locks also create problem if we lose keys and also we have to carry keys along with us always. Again, using patterns in the locks can increase security but again it can be opened if somehow the passwords or patterns are known. So, leaving every system in this project we will implement a system using biometrics. Incase-of biometrics, the pattern which will be used as key will be unique. Here, to implement the project we will use fingerprint as the key This Arduino project will make use of different devices for the implementation of the security lock where there will be different features to increase the security level. In simple words, we can say that we are implementing a door access system using Arduino which make use of fingerprints to identify whom to allow and who not to allow inside our homes, offices, shops, etc. We are trying to implement it using a normal and simple door lock which is fitted in every home so as to minimize the cost of the device as a product.

1.2 Project Objectives

The goal of this project is to research and analyze a suitable collection of components for developing a smart door lock using Arduino that provides excellent security and quick access.

The following are the specific project goals:

- Familiarity with a smart door locking system based on a microcontroller.
- Using Arduino to create a simple and smart door locking system.

1.3 A brief outline of the report

This project is divided into 6 chapters.

Chapter 1 present the Introduction of this project. Chapter one also presents objectives and a brief outline.

Chapter 2 provides the literature review of this project.

Chapter 3 includes the Theory and introduces the project methodology and modeling like working principle, process of work, component, implementation, testing and cost analysis.

Chapter 4 presents the Hardware Modelling

Chapter5 presents the Logic operation behind the system.

Chapter 6 Conclude the project

CHAPTER 2

LITERATURE REVIEW

LITERATURE REVIEW

1) Various smart locks are previously available. The majority of them are expensive. In this paper “Arduino based electronic lock using RFID and password” which was proposed by “Ni Ni San Hlaing, San SanLwin”. This digital door lock runs on the technology of audio-frequency identification and passcode-based with the help of an Arduino Uno MCU.

2) In another paper named “Secured password-based lock system” was put forward by “Arpita Mishra, Siddharth Sharma, Sachin Dubey, S.K.Dubey”. This methodology is targeted to prevent unlocking of the door by unknown individuals. The formation of the home safety Service consists of the numeric keypad, the hook which is used for lifting, and a GSM module to establish dependable connection for communication conferred with the MCU. The control panel conferred with the device is employed because the passcode access combination opens/closes the door.

3) In another paper named “Smart Lock System Using RFID” was proposed by “ShrinidhiGindi, NaiyerShaikh, KashifBeig, AbdealiSabuwala”. Here may be a Room security solution supported IoT using RFID, the system is often monitored from anywhere within the world thanks to the continual updating of the status of the door.

4) Moving forward to another paper named “An OTP-based wireless smart door locking system” was proposed by “Mr. L. David William Raj, M.Deepika, V. Bhubaneshwar, R. Harshitha, K. Haripriya”. In this innovation, the key phrase for security is initially put away within the Electrically Erasable Programmable ROM . At the purpose when the client enters the proper secret phrase then the two-way confirmation of a haphazardly produced OTP is shipped off the client gadget. On the off chance that the OTP is coordinated, the framework is going to be opened, and therefore the required capacity is often started.

5) Coming to the next paper named “SMART DOOR UNLOCK SYSTEM USING FINGERPRINT” was proposed by K.Rajesh, Asst.Prof. B.VenkataRao, P.AV.S.K.Chaitanya, A.Ruchitha Reddy. In our paper, we apply the fingermark detector to scan one's character to instinctually function the gate of the car, under such situation we prefer to use a MCU for enabling for both opening and closing of the door if both the match for scanned and existing facts are true.

6) In the upcoming document termed“DOORWAY ROBOTIZATION network supported by CORDLESS for android Smartphone” was proposed by “Lia Kamelia, Alfin Noorhassan S.R, Mada Sanjaya, and W.S., Edi Mulyana”. In this a tool called a automated door lock with the support of Bluetooth and Android smartphone door locks automation system using Bluetooth-based Android Smartphone's is recommended and prototyped. The equipment structure forthe door lock setup is that a combination with an android.

CHAPTER3

Theory

3.1 Smart Door Lock

Smart Door lock is an electromechanical lock that is designed to perform locking and unlocking operations on a door when it receives when it receives a prompt via an electronic keypad, biometric sensor, access card, Bluetooth, or Wi-Fi from a registered mobile device. These locks are called smart locks because they use advanced technology and Internet communication to enable easier access for users and enhanced security from intruders. The main components of a smart lock include the physical lock, the key (which can be electronic, digitally encrypted, or a virtual key to provide keyless entry), a secure Bluetooth or Wi-Fi connection, and a management mobile app. Smart lock may also monitor access and send alerts in response to the different events it monitors as well as other critical events related to the status of the device. Smart locks can be considered part of a smart home. Most smart locks are installed on mechanical locks (simple types of locks, including deadbolts) and they physically upgrade the ordinary lock. Recently, smart locking controllers have also appeared at the market. Smart locks, like the traditional locks, need two main parts to work: the lock and the key. In the case of these electronic locks, the key is not a physical key but a smartphone or a special key card configured explicitly for this purpose which wirelessly performs the authentication needed to automatically unlock the door. Smart lock controlled by a phone app Smart lock allow users to grant access to a third party by means of a virtual key. This key can be sent to the recipient smartphone over standard messaging protocols such as e-mail or SMS, or via a dedicated application. Once this key is received the recipient will be able to unlock the smart lock using their mobile device during the timeframe previously specified by the sender. Certain smart locks include a built-in Wi-Fi connection that allows for monitoring features such as access notifications or cameras to show the person requesting access. Some smart locks work with a smart doorbell to allow the user to see who and when someone is at a door. Many smart locks now also feature biometric features, such as fingerprint sensors. Biometrics are becoming increasingly popular because they offer more security than passwords alone. This is because they use unique physical characteristics rather than stored information. Smart locks may use Bluetooth Low Energy and SSL to communicate, encrypting communications using 128/256-bit AES.

3.1.1 Smart Locks are Internet of Things

Smart locks are IoT-enabled keyless entry devices that allow users remote access to door locks through their smartphone. Smart locks leverage IoT-enabled sensors to operate keyless entry devices that allow users to access doors remotely, through a smartphone or other internetconnected device. Smart locks provide users the ability to unlock their door without a key, from anywhere. The global smart lock market reached a value of US$ 1.6 Billion in 2021. Looking forward, IMARC Group expects the market to reach US$ 4.9 Billion by 2027, exhibiting at a CAGR of 21.8% during 2022-2027. There are two types of locks available. One lock provides additional functionality to existing locking mechanisms and must be retrofitted to the door lock already in place and the other completely replaces the locking mechanism on the door. While smart locks must be powered, many also allow a physical key to serve as a backup in case of a service or internet outage. Smart locks offer additional functionality through compatibility with other IoT devices, smart assistants, or smart home management systems. These functions can include automating processes, like turning on your lights and adjusting your thermostat when the door is unlocked, or triggering the security system to record and send video if the door is unlocked outside of expected hours.

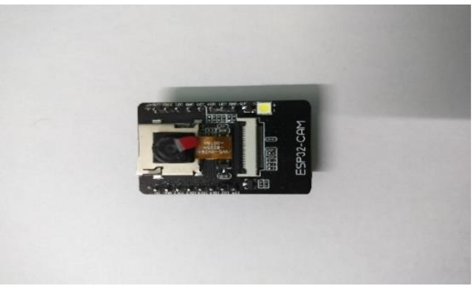
3.1.2 Advantage of Smart Door Lock

• Check on the status of a door remotely, ensuring that it is locked no matter how far from home they are.
• Give and revoke remote access to visitors, enabling service providers to access the door only at specific times, or to give unlimited access to trusted friends or family.
• Receive notifications whenever the door is opened – allowing users to be immediately alerted in case of unexpected access.

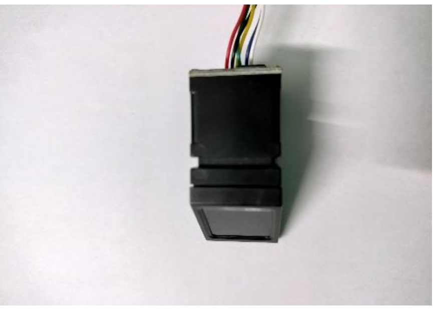
3.1.3 TECHNOLOGIES USED

A. Arduino Uno the Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.[4] It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

B. ESP32 with cam ESP32 is a series of low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISCV microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 microcontroller.

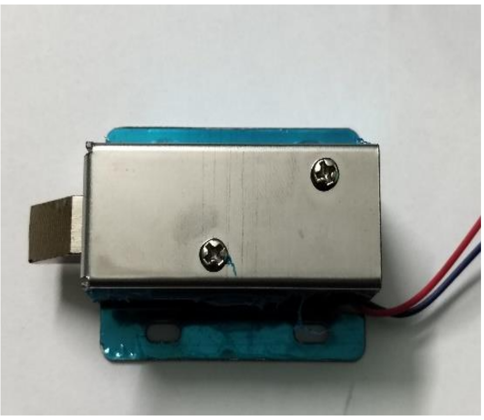


C. Fingerprint sensor (R307) R307 Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry,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



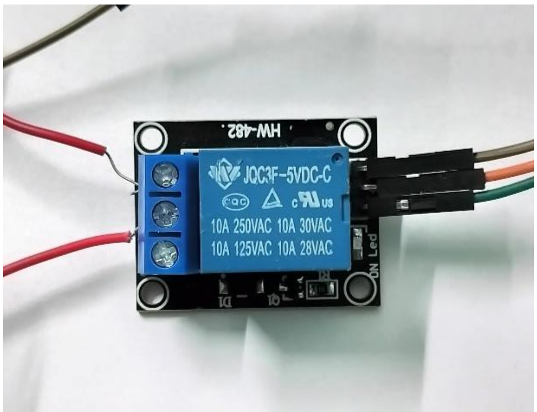
D. Solenoid Lock

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.

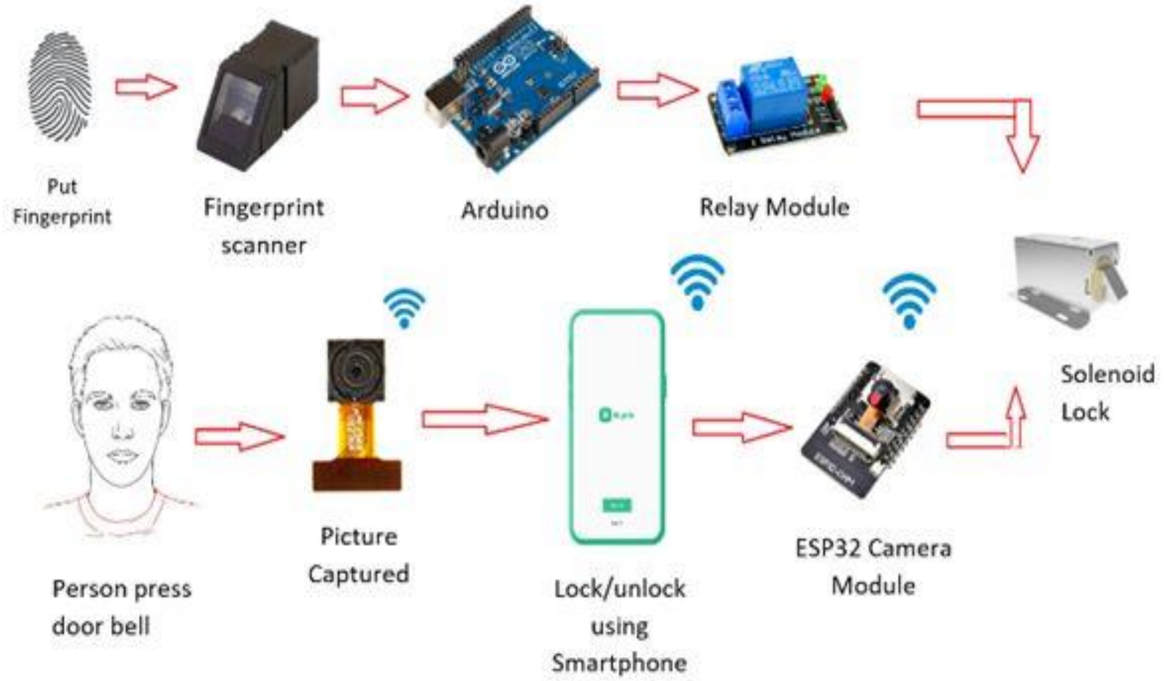


E. Blynk Application Blynk is a full suite of software required to prototype, deploy, and remotely manage connected electronic devices at any scale: from personal IoT projects to millions of commercial connected products. With Blynk anyone can connect their hardware to the cloud and build a no-code iOS, Android, and web applications to analyse real-time and historical data coming from devices, control them remotely from anywhere in the world, receive important notifications, and much more

F. Relay is an electro-mechanical device which acts as a switch. DC electrical current is used to energize the relay coil which opens or closes the contact switches. Internal circuit of a single channel 5V relay consists of normally open contacts, normally closed contacts and a coil.



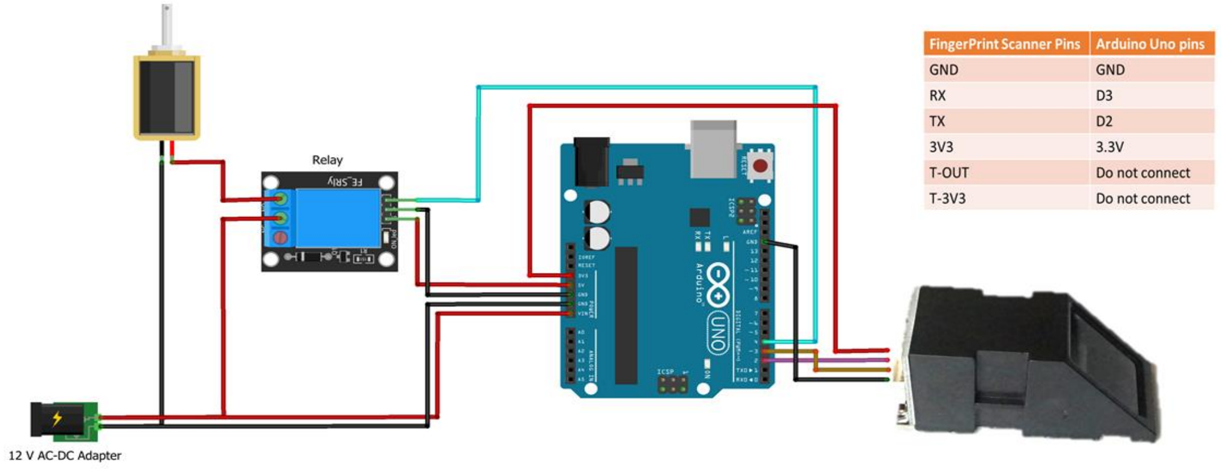
3.2 Hardware Architecture



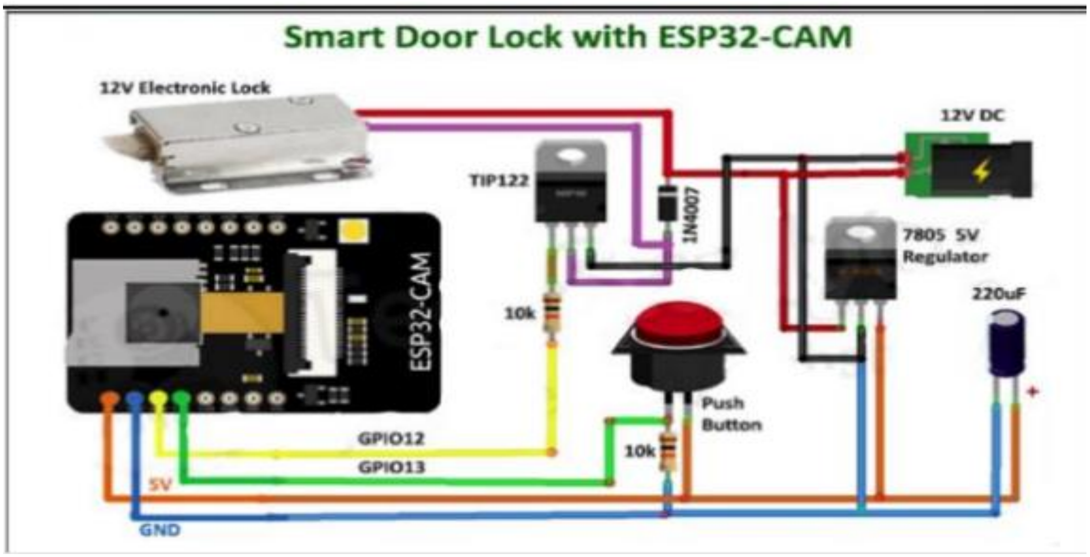
Overview of the complete project

According to our project there is two ways to unlock the solenoid lock system that is by biometric system and wirelessly by Blynk app. To use the wireless system, we need to ensure that the ESP32 is connected to a wi-fi network. If the user clicks the unlock button in the Blynk app then the ESP32 send signal to the solenoid lock and the lock opens. The lock can also be opened by the fingerprint sensor. The user needs to put the finger in the sensor and if the database matches with the user's fingerprint, then the relay is triggered and the solenoid lock opens.

3.3 Schematic Diagram



FingerPrint Scanner Pins	Arduino Uno pins
GND	GND
RX	D3
TX	D2
3V3	3.3V
T-OUT	Do not connect
T-3V3	Do not connect



Complete circuit Diagram of the Project

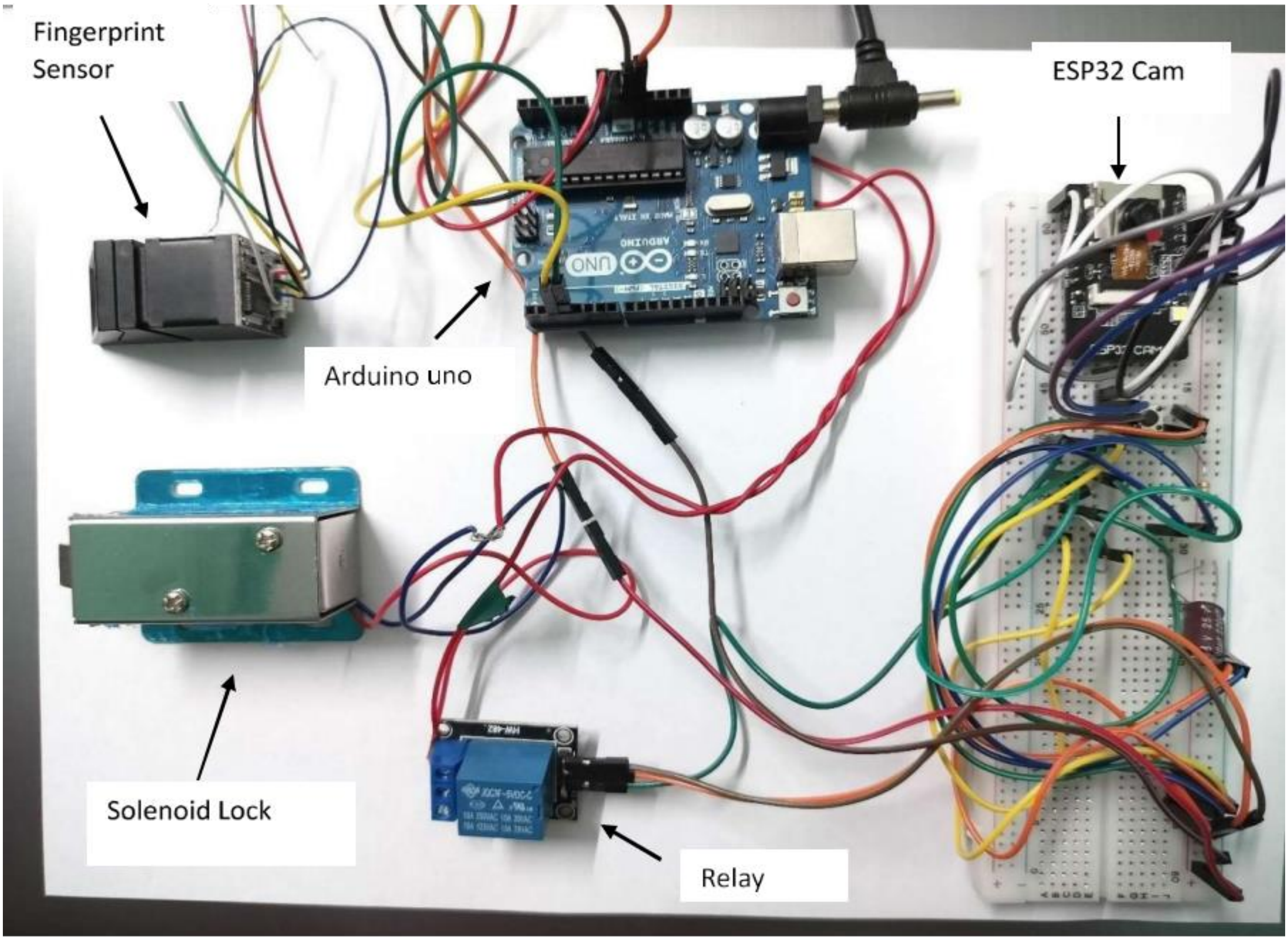
CHAPTER 4

Hardware Implementation of the Project

4.1 Main features of the prototype

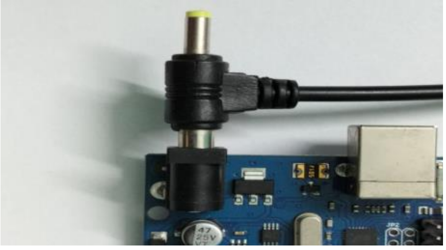
- Fingerprint sensor (very secure)
- Wireless connectivity through Blynk
- Only accessible through Blynk app (very secure)
- 12-volt operation • Cost Effective solution

4.2 Photographs of the prototype



Main prototype

4.3 Step by step operation of the prototype



1. Connect the 12 V supply and via adapter. Switch on the Arduino by connecting the pin of adapter to board.

2. Now we will load the fingerprints on the fingerprint sensor and load it to the fingerprint enrolment code on the laptop.

3. We are now connecting the circuit and here the black wire of fingerprint sensor (5V VCC Pin) is connected to VIN pin on the Arduino. Now connect the brown wire of fingerprint sensor to the ground pin of Arduino, white wire to pin 2 and yellow wire to digital pin 3 of Arduino, respectively. Now we will connect the relay whose VCC is connected to the 5V. The ground pin of relay will be connected to the ground pin of Arduino, And the signal pin will be connected to the 12 no pin of Arduino. Now the fingerprint is loaded on the fingerprint sensor via the program. Refer Fig 2

4.The solenoid red wire will be connected to the common pin on the Arduino. After the whole circuit is completed, we place the smart lock on the door and now we will give our fingerprint on the sensor which will get checked with the previously enrolled fingerprints and if it matches any of it, the door is unlocked.

refer Fig 1.

5. We also have a camera module (ESP32) with our circuit to detect anyone through the camera. The camera module is operated by the Blynk application where we can capture the picture or video of the person currently standing outside the door. Refer Fig 3.

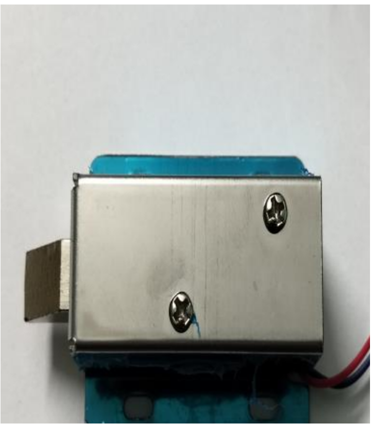


Fig 1. Solenoid Lock

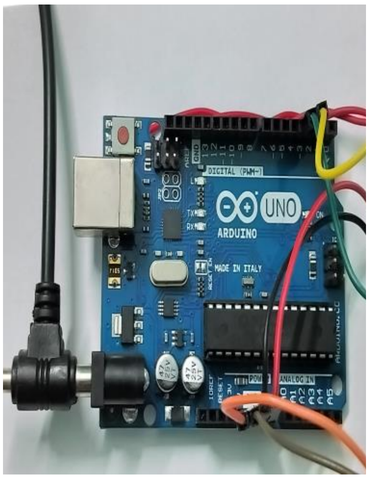


Fig 2. Arduino connection

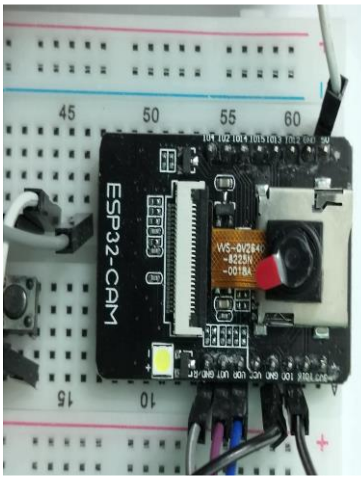
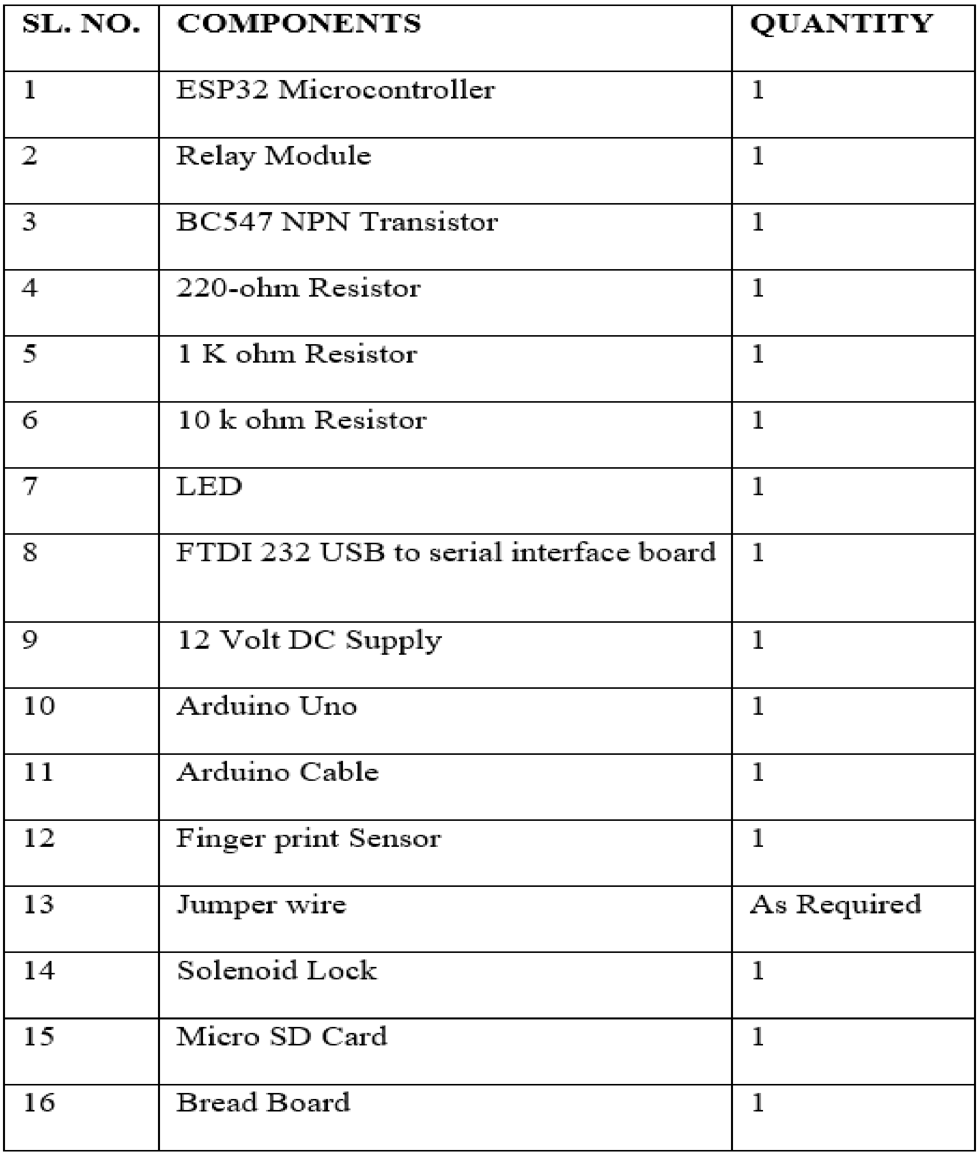


Fig 3. ESP32 connection

4.4Components required

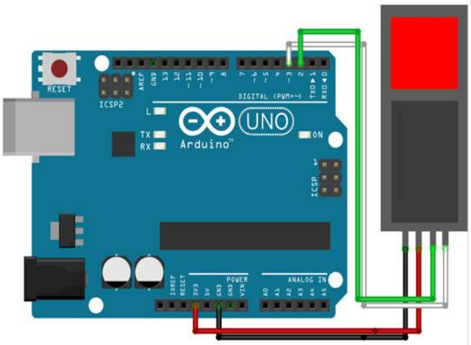


SL. NO.	COMPONENTS	QUANTITY
1	ESP32 Microcontroller	1
2	Relay Module	1
3	BC547 NPN Transistor	1
4	220-ohm Resistor	1
5	1 K ohm Resistor	1
6	10 k ohm Resistor	1
7	LED	1
8	FTDI 232 USB to serial interface board	1
9	12 Volt DC Supply	1
10	Arduino Uno	1
11	Arduino Cable	1
12	Finger print Sensor	1
13	Jumper wire	As Required
14	Solenoid Lock	1
15	Micro SD Card	1
16	Bread Board	1

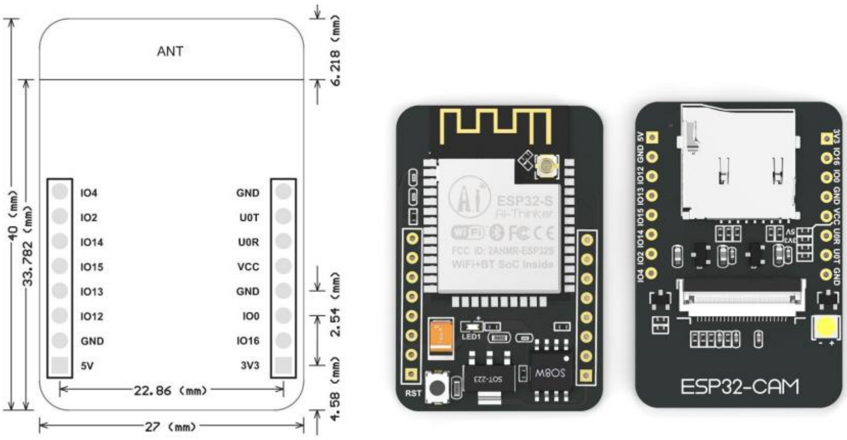
4.5Hardware interfacing

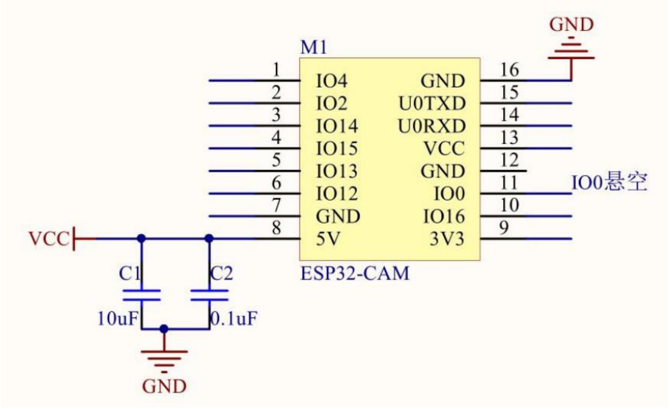
4.5.1 Fingerprint Interface

The R307 fingerprint module has two interface TTL UART and USB2.0, USB2.0 interface can be connected to the computer; RS232 interface is a TTL level, the default baud rate is 57600, can be changed, refer to a communication protocol; can and microcontroller, such as ARM, DSP and other serial devices with a connection, 3.3V 5V microcontroller can be connected directly. Needs to connect the computer level conversion, level conversion note, embodiments such as a MAX232 circuit.



4.5.2 ESP32 INTERFACE





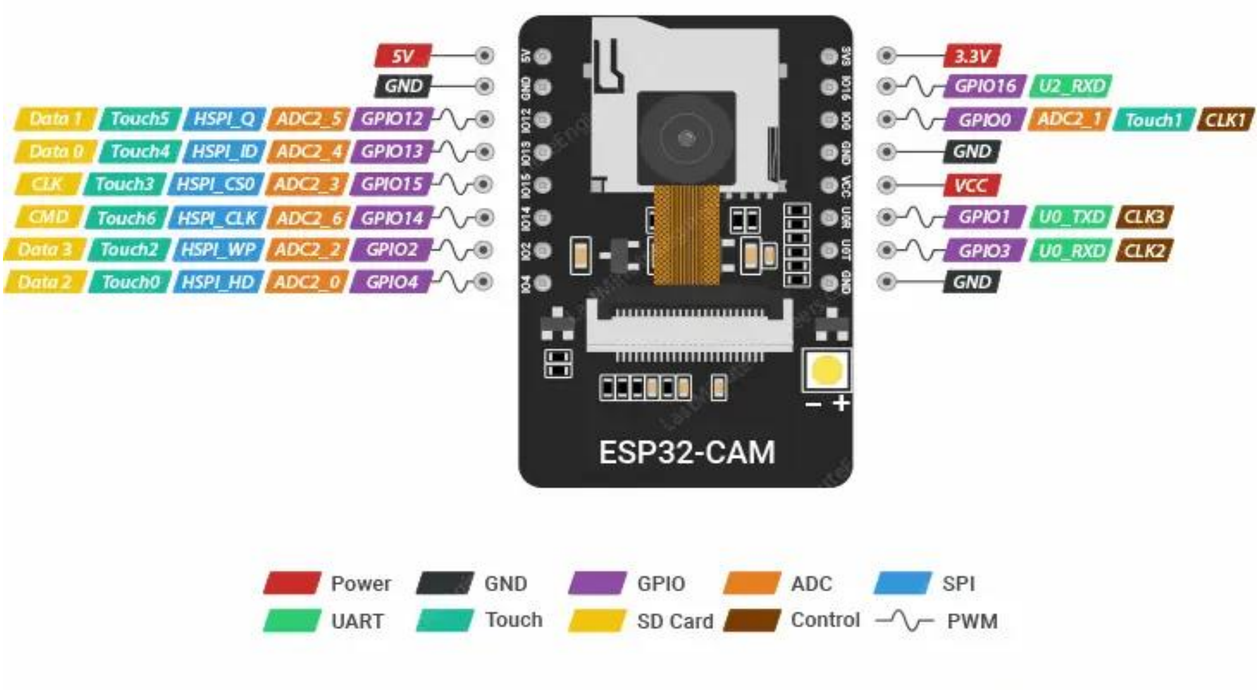
The power consumption of the ESP32-CAM varies depending on what you're using it for.It ranges from 80 mAh when not streaming video to around 100~160 mAh when streaming video; with the flash on, it can reach 270 mAh.

Operation mode:	Power Consumption
Stand by:	80 mAh
In streaming:	100~160 mAh
In streaming with flash:	270 mAh

CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

ESP32-CAM Pinout

The ESP32-CAM has 16 pins in total. For convenience, pins with similar functionality are grouped together. The pinout is as follows:



Power Pins There are two power pins: 5V and 3V3. The ESP32-CAM can be powered via the 3.3V or 5V pins. Since many users have reported problems when powering the device with 3.3V, it is advised that the ESP32-CAM always be powered via the 5V pin. The VCC pin normally outputs 3.3V from the on-board voltage regulator. It can, however, be configured to output 5V by using the Zero-ohm link near the VCC pin.

GND is the ground pin.

GPIO Pins The ESP32-S chip has 32 GPIO pins in total, but because many of them are used internally for the camera and the PSRAM, the ESP32-CAM only has 10 GPIO pins available. These pins can be assigned a variety of peripheral duties, such as UART, SPI, ADC, and Touch.

UART Pins The ESP32-S chip actually has two UART interfaces, UART0 and UART2. However, only the RX pin (GPIO 16) of UART2 is broken out, making UART0 the only usable UART on the ESP32-CAM (GPIO 1 and GPIO 3). Also, because the ESP32-CAM lacks a USB port, these pins must be used for flashing as well as connecting to UART-devices such as GPS, fingerprint sensors, distance sensors, and so on.

MicroSD Card Pins are used for interfacing the microSD card. If you aren't using a microSD card, you can use these pins as regular inputs and outputs.

ADC Pins On the ESP32-CAM, only ADC2 pins are broken out. However, because ADC2 pins are used internally by the WiFi driver, they cannot be used when Wi-Fi is enabled.

Touch Pins The ESP32-CAM has 7 capacitive touch-sensing GPIOs. When a capacitive load (such as a human finger) is in close proximity to the GPIO, the ESP32 detects the change in capacitance.

SPI Pins The ESP32-CAM features only one SPI (VSPI) in slave and master modes.

PWM Pins The ESP32-CAM has 10 channels (all GPIO pins) of PWM pins controlled by a PWM controller. The PWM output can be used for driving digital motors and LEDs.

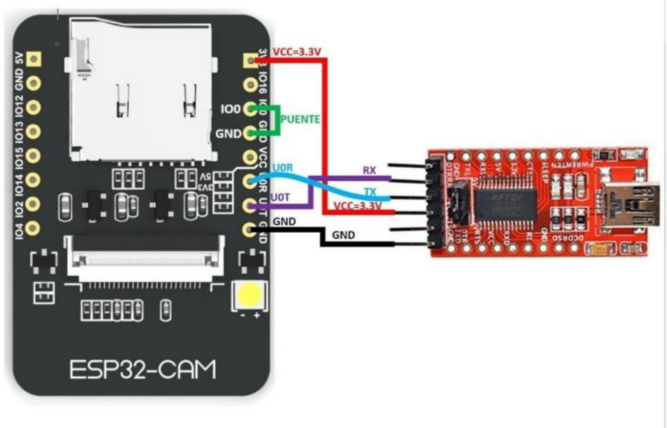
Programming the ESP32-CAM

Programming the ESP32-CAM can be a bit of a pain as it lacks a built-in USB port. Because of that design decision, users require additional hardware in order to upload programs from the Arduino IDE. None of that is terribly complex, but it is inconvenient.

To program this device, you'll need either a USB-to-serial adapter (an FTDI adapter) or an ESP32-CAM-MB programmer adapter.

Using the FTDI Adapter

If you've decided to use the FTDI adapter, here's how you connect it to the ESP32-CAM module.


5V GND IO12 IO13 IO15 IO14 IO2 IO4
3V3 IO16 IO0 GND VCC U0R U0T GND
IO0
GND
VCC=3.3V
PUENTE
U0R
U0T
GND
RX
TX
VCC=3.3V
GND
ESP32-CAM

Many FTDI programmers have a jumper that lets you choose between 3.3V and 5V. As we are powering the ESP32-CAM with 5V, make sure the jumper is set to 5V.

GPIO 0 pin is connected to Ground. This connection is only necessary while programming the ESP32-CAM. Once you have finished programming the module, you must disconnect this connection.

4.6 RESULTS

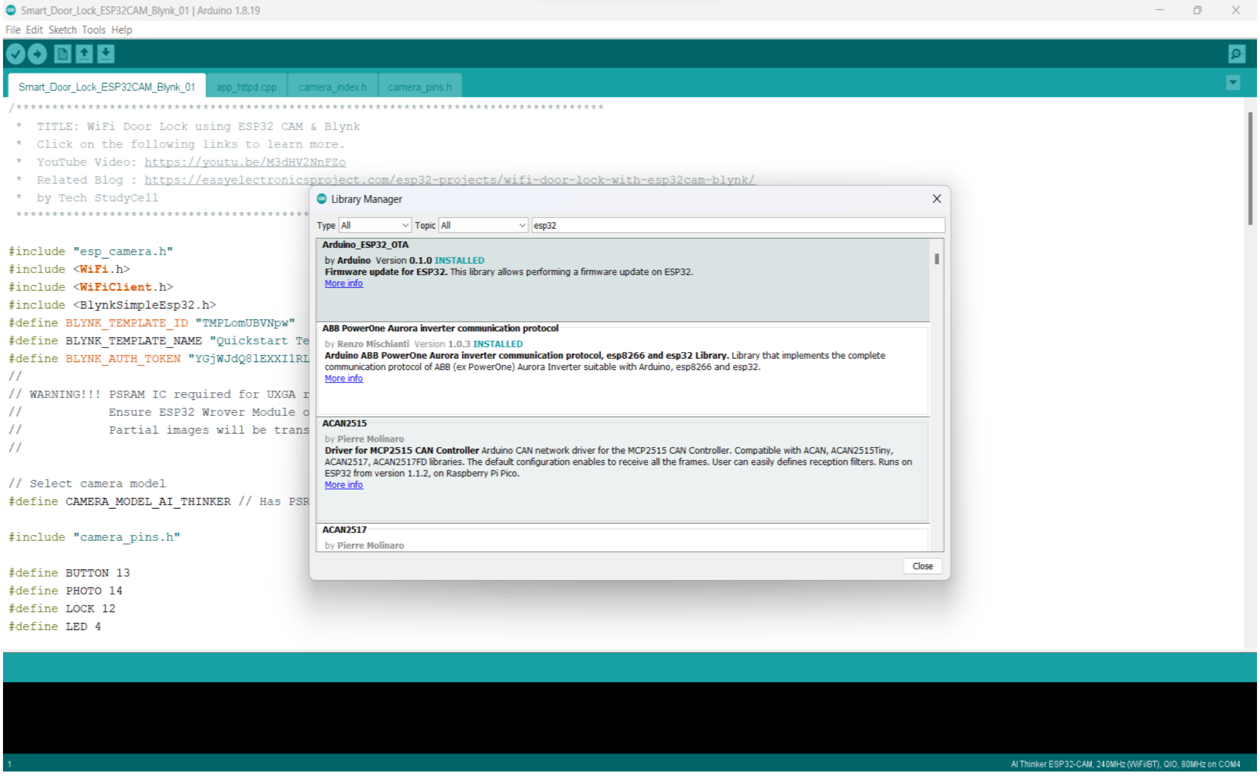
The prototype was made according to the circuit diagram and the results were as expected. The solenoid worked that is it unlocked when the user used the registered finger in the fingerprint sensor and while the user clicked the click picture button in the blynk app the esp32 clicked the picture and upon clicking on the unlock door button the solenoid lock unlocked.

CHAPTER 5 Coding of the Project

Installing Library – Arduino_ESP32

Follow the next steps to install those libraries.

1. Open your Arduino IDE and go to Sketch > Include Library > Library Manager. The Library Manager should open.



Smart_Door_Lock_ESP32CAM_Blynk_01 | Arduino 1.8.19
File Edit Sketch Tools Help
Smart_Door_Lock_ESP32CAM_Blynk_01
app_httpd.cpp
camera_index.h
camera_pins.h
Library Manager
Close
AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM4

```
/**************************************************************************
 *  TITLE: WiFi Door Lock using ESP32 CAM & Blynk
 *  Click on the following links to learn more.
 *  YouTube Video: https://youtu.be/M3dHV2NnFZo
 *  Related Blog : https://easyelectronicsproject.com/esp32-projects/wifi-door-lock-with-esp32cam-blynk/
 *  by Tech StudyCell
 ****************************************

#include "esp_camera.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_TEMPLATE_ID "TMPLomUBVNpw"
#define BLYNK_TEMPLATE_NAME "Quickstart Te
#define BLYNK_AUTH_TOKEN "YGjWJdQ8lEXXI1RL
//
// WARNING!!! PSRAM IC required for UXGA r
//            Ensure ESP32 Wrover Module o
//            Partial images will be trans
//

// Select camera model
#define CAMERA_MODEL_AI_THINKER // Has PSR

#include "camera_pins.h"

#define BUTTON 13
#define PHOTO 14
#define LOCK 12
#define LED 4
```

Type: All   Topic: All   esp32

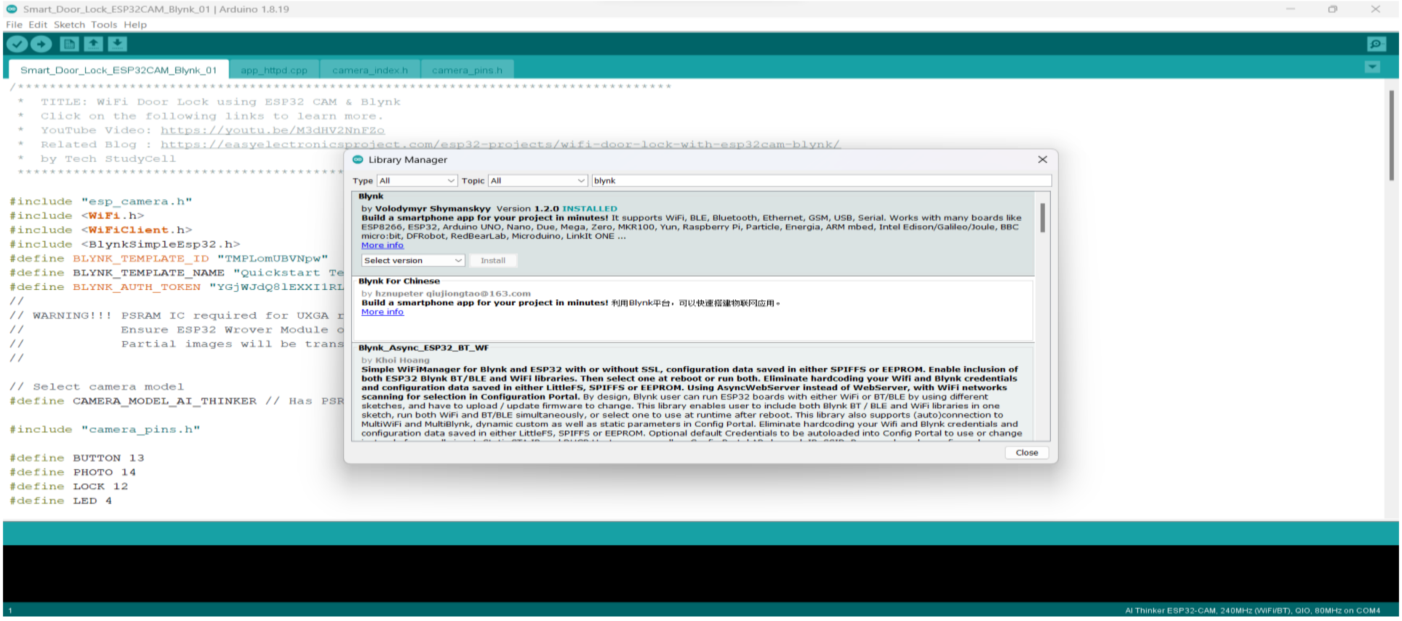
Arduino_ESP32_OTA
by **Arduino** Version **0.1.0** INSTALLED
Firmware update for ESP32. This library allows performing a firmware update on ESP32.
More info

ABB PowerOne Aurora inverter communication protocol
by Renzo Mischianti Version 1.0.3 INSTALLED
Arduino ABB PowerOne Aurora inverter communication protocol, esp8266 and esp32 Library. Library that implements the complete communication protocol of ABB (ex PowerOne) Aurora Inverter suitable with Arduino, esp8266 and esp32.
More info

ACAN2515
by Pierre Molinaro
Driver for MCP2515 CAN Controller Arduino CAN network driver for the MCP2515 CAN Controller. Compatible with ACAN, ACAN2515Tiny, ACAN2517, ACAN2517FD libraries. The default configuration enables to receive all the frames. User can easily defines reception filters. Runs on ESP32 from version 1.1.2, on Raspberry Pi Pico.
More info

ACAN2517
by Pierre Molinaro

2. After installing the Arduino library. In ribbon type "blynk" in the search box and install the library.



Smart_Door_Lock_ESP32CAM_Blynk_01 | Arduino 1.8.19
File Edit Sketch Tools Help
Smart_Door_Lock_ESP32CAM_Blynk_01
app_httpd.cpp
camera_index.h
camera_pins.h
Library Manager
Select version
Install
Close
AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM4

```
/**************************************************************************
 *  TITLE: WiFi Door Lock using ESP32 CAM & Blynk
 *  Click on the following links to learn more.
 *  YouTube Video: https://youtu.be/M3dHV2NnFZo
 *  Related Blog : https://easyelectronicsproject.com/esp32-projects/wifi-door-lock-with-esp32cam-blynk/
 *  by Tech StudyCell
 ****************************************

#include "esp_camera.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_TEMPLATE_ID "TMPLomUBVNpw"
#define BLYNK_TEMPLATE_NAME "Quickstart Te
#define BLYNK_AUTH_TOKEN "YGjWJdQ8lEXXI1RL
//
// WARNING!!! PSRAM IC required for UXGA r
//            Ensure ESP32 Wrover Module o
//            Partial images will be trans
//

// Select camera model
#define CAMERA_MODEL_AI_THINKER // Has PSR

#include "camera_pins.h"

#define BUTTON 13
#define PHOTO 14
#define LOCK 12
#define LED 4
```

Type: All   Topic: All   blynk

Blynk
by **Volodymyr Shymanskyy** Version **1.2.0** INSTALLED
Build a smartphone app for your project in minutes! It supports WiFi, BLE, Bluetooth, Ethernet, GSM, USB, Serial. Works with many boards like ESP8266, ESP32, Arduino UNO, Nano, Due, Mega, Zero, MKR100, Yun, Raspberry Pi, Particle, Energia, ARM mbed, Intel Edison/Galileo/Joule, BBC micro:bit, DFRobot, RedBearLab, Microduino, LinkIt ONE ...
More info

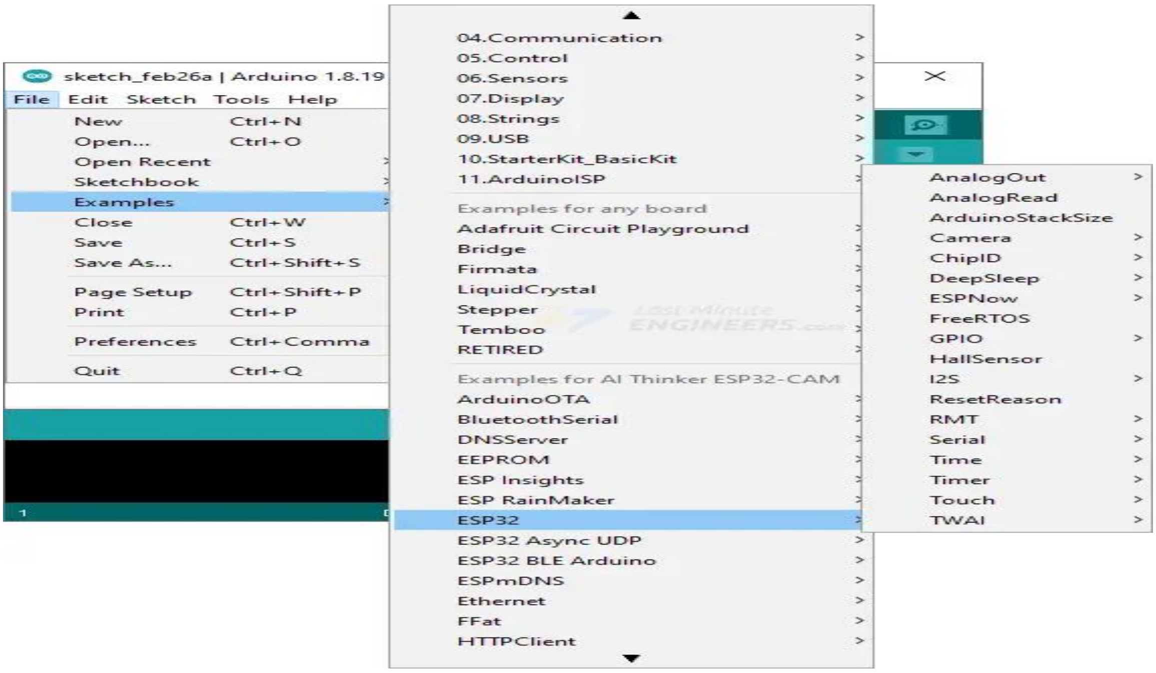
Blynk For Chinese
by hznupeter qiujiongtao@163.com
Build a smartphone app for your project in minutes! 利用Blynk平台，可以快速搭建物联网应用。
More info

Blynk_Async_ESP32_BT_WF
by Khoi Hoang
Simple WiFiManager for Blynk and ESP32 with or without SSL, configuration data saved in either SPIFFS or EEPROM. Enable inclusion of both ESP32 Blynk BT/BLE and WiFi libraries. Then select one at reboot or run both. Eliminate hardcoding your Wifi and Blynk credentials and configuration data saved in either LittleFS, SPIFFS or EEPROM. Using AsyncWebServer instead of WebServer, with WiFi networks scanning for selection in Configuration Portal. By design, Blynk user can run ESP32 boards with either WiFi or BT/BLE by using different sketches, and have to upload / update firmware to change. This library enables user to include both Blynk BT / BLE and WiFi libraries in one sketch, run both WiFi and BT/BLE simultaneously, or select one to use at runtime after reboot. This library also supports (auto)connection to MultiWiFi and MultiBlynk, dynamic custom as well as static parameters in Config Portal. Eliminate hardcoding your Wifi and Blynk credentials and configuration data saved in either LittleFS, SPIFFS or EEPROM. Optional default Credentials to be autoloaded into Config Portal to use or change

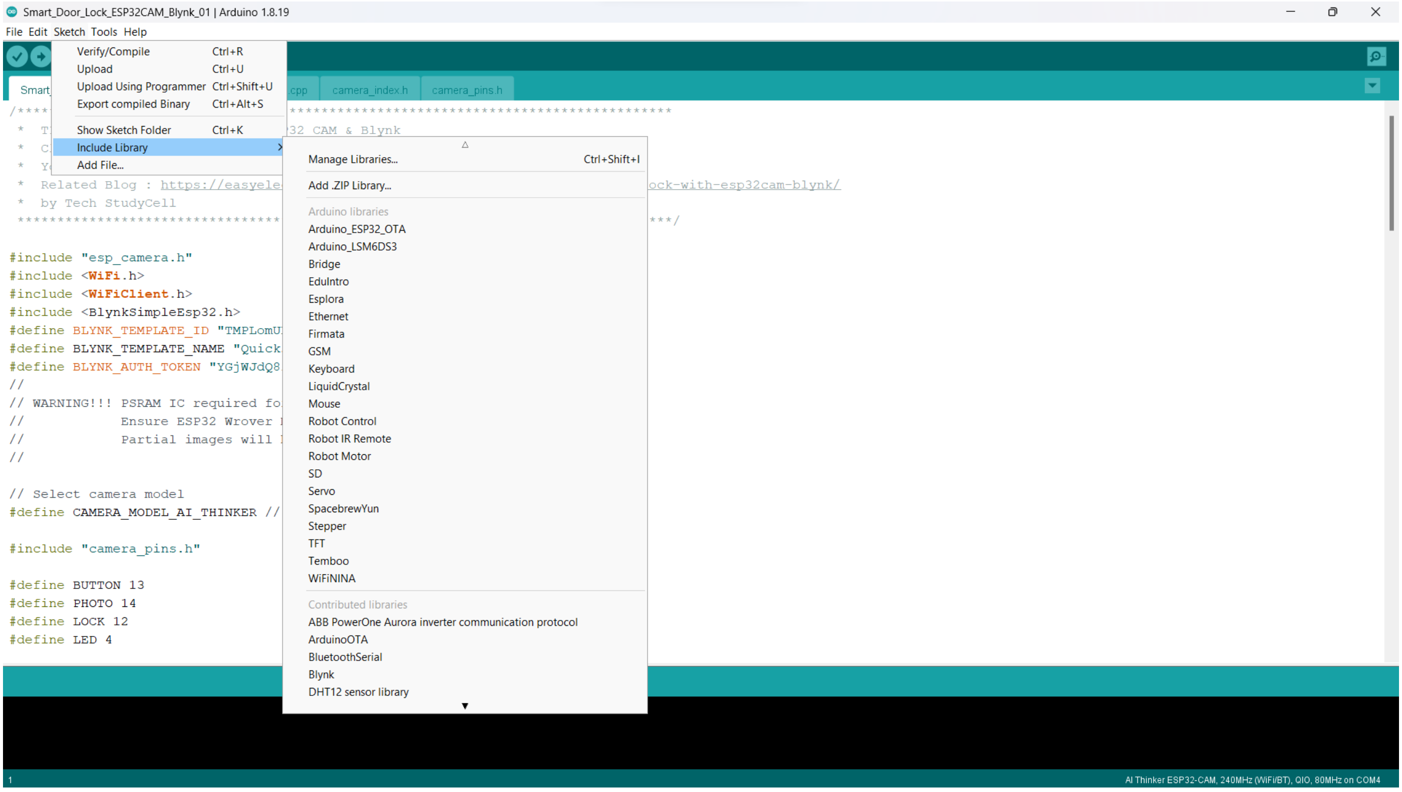
3. After installing the libraries, restart your Arduino IDE.

After installation of blynk 1.2.0 version all required libraries, you can use one example from the library to see if everything is working properly.

To access the example sketches, navigate to File > Examples > ESP32.



4. After the check, start Arduino IDE and navigate to Tools > Include Library



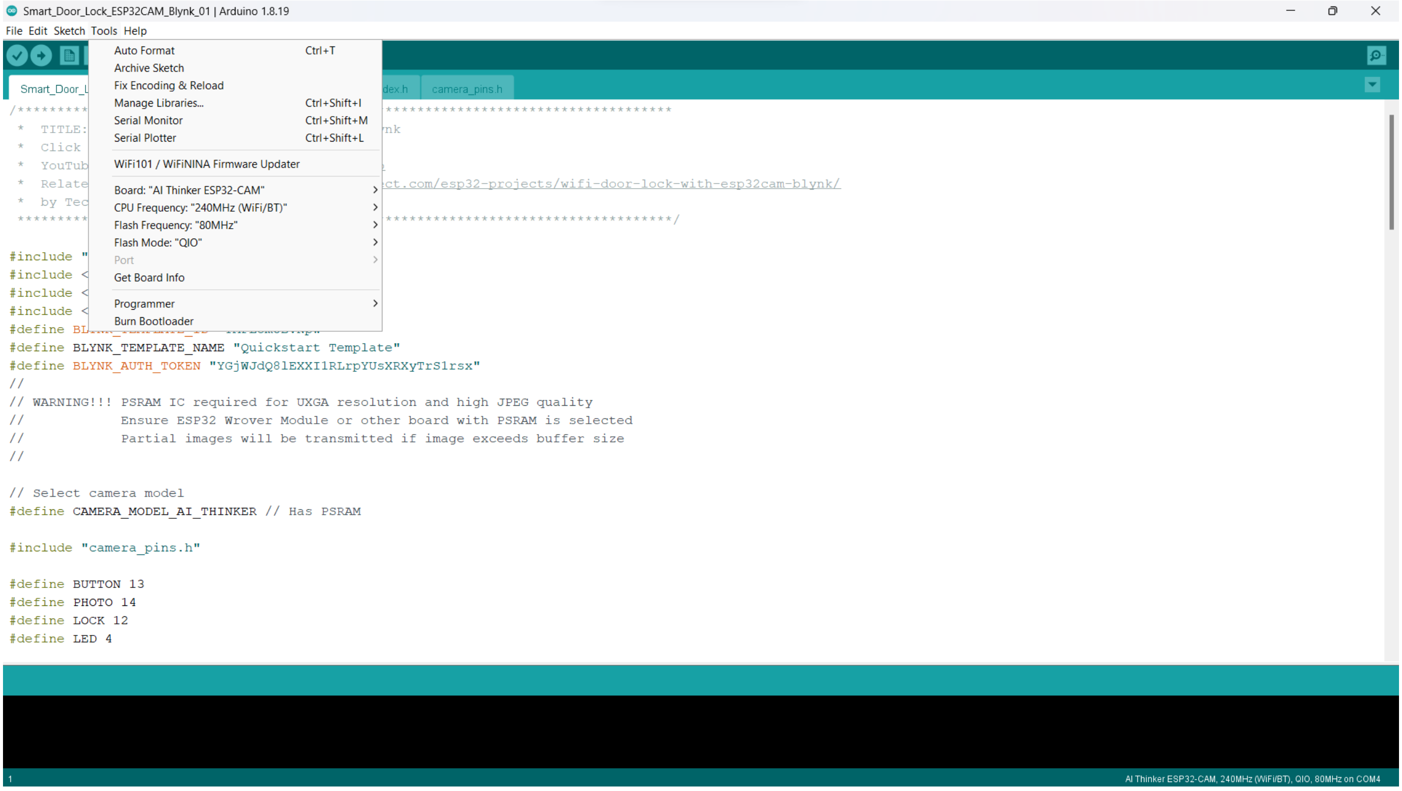
```
/****                                   ***********************************************
 *  T                                   32 CAM & Blynk
 *  C
 *  Y
 *  Related Blog : https://easyele                                ock-with-esp32cam-blynk/
 *  by Tech StudyCell
 *********************************                                ***/

#include "esp_camera.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_TEMPLATE_ID "TMPLomU
#define BLYNK_TEMPLATE_NAME "Quick
#define BLYNK_AUTH_TOKEN "YGjWJdQ8
//
// WARNING!!! PSRAM IC required fo
//            Ensure ESP32 Wrover 
//            Partial images will 
//

// Select camera model
#define CAMERA_MODEL_AI_THINKER //

#include "camera_pins.h"

#define BUTTON 13
#define PHOTO 14
#define LOCK 12
#define LED 4
```


Smart_Door_Lock_ESP32CAM_Blynk_01 | Arduino 1.8.19
File Edit Sketch Tools Help
Smart_Door_L
dex.h
camera_pins.h
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Port
Get Board Info
Programmer
Burn Bootloader
1

Board: "AI Thinker ESP32-CAM"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"

```
/********                                    ***********************************
 *  TITLE:                                   ynk
 *  Click
 *  YouTub
 *  Relate                                   ct.com/esp32-projects/wifi-door-lock-with-esp32cam-blynk/
 *  by Tec
 ********                                    ***********************************/

#include "
#include <
#include <
#include <
#define BL
#define BLYNK_TEMPLATE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "YGjWJdQ8lEXXI1RLrpYUsXRXyTrS1rsx"
//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#include "camera_pins.h"

#define BUTTON 13
#define PHOTO 14
#define LOCK 12
#define LED 4
```

AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM4

5. After all the installation we starts the fingerprint Enrollment Code

```
#include <Adafruit_Fingerprint.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); //you can change them if it is not working on 2 or 3



Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);



uint8_t id;



void setup()

{

Serial.begin(9600);

while (!Serial); // For Yun/Leo/Micro/Zero/…
```

```
delay(100);

Serial.println("\n\nFingerprint sensor enrollment");


// set the data rate for the sensor serial port

finger.begin(57600);


if (finger.verifyPassword()) {

Serial.println("Found fingerprint sensor!");

} else {

Serial.println("Did not find fingerprint sensor :(");

while (1) { delay(1); }

}

}


uint8_t readnumber(void) {

uint8_t num = 0;


while (num == 0) {

while (! Serial.available());

num = Serial.parseInt();

}

return num;

}


void loop() // program wil repeat this part (loop here)

{
```

```
Serial.println("Ready to enroll a fingerprint!");
Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger as…");
id = readnumber();
if (id == 0) {// ID #0 not allowed, try again!
return;
}
Serial.print("Enrolling ID #");
Serial.println(id);

while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

int p = -1;
Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println(".");
break;
case FINGERPRINT_PACKETRECIEVEERR:
```

```
Serial.println(“Communication error”);
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println(“Imaging error”);
break;
default:
Serial.println(“Unknown error”);
break;
}
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
Serial.println(“Image converted”);
break;
case FINGERPRINT_IMAGEMESS:
Serial.println(“Image too messy”);
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println(“Communication error”);
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println(“Could not find fingerprint features”);
```

```
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println(“Could not find fingerprint features”);
return p;
default:
Serial.println(“Unknown error”);
return p;
}


Serial.println(“Remove finger”);
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
Serial.print(“ID “); Serial.println(id);
p = -1;
Serial.println(“Place same finger again”);
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.pr…
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println(“Communication error”);
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println(“Imaging error”);
break;
default:
Serial.println(“Unknown error”);
break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
Serial.println(“Image converted”);
break;
case FINGERPRINT_IMAGEMESS:
Serial.println(“Image too messy”);
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println(“Communication error”);
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println(“Could not find fingerprint features”);
return p;
case FINGERPRINT_INVALIDIMAGE:
```

```
Serial.println(“Could not find fingerprint features”);

return p;

default:

Serial.println(“Unknown error”);

return p;

}


// OK converted!

Serial.print(“Creating model for #”); Serial.println(id);

p = finger.createModel();

if (p == FINGERPRINT_OK) {

Serial.println(“Prints matched!”);

} else if (p == FINGERPRINT_PACKETRECIEVEERR) {

Serial.println(“Communication error”);

return p;

} else if (p == FINGERPRINT_ENROLLMISMATCH) {

Serial.println(“Fingerprints did not match”);

return p;

} else {

Serial.println(“Unknown error”);

return p;

}


Serial.print(“ID “); Serial.println(id);

p = finger.storeModel(id);

if (p == FINGERPRINT_OK) {
```

```
Serial.println(“Stored!”);
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println(“Communication error”);
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial.println(“Could not store in that location”);
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial.println(“Error writing to flash”);
return p;
} else {
Serial.println(“Unknown error”);
return p;
}
}
```

6.After the enrolment code here is Fingerprint Unlock Code

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
Serial.begin(9600);
```

```
while (!Serial); // For Yun/Leo/Micro/Zero/…
delay(100);
Serial.println(“fingertest”);
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword()) {
Serial.println(“Found fingerprint sensor!”);
} else {
Serial.println(“Did not find fingerprint sensor :(“);
while (1) {
delay(1);
}
}

finger.getTemplateCount();
Serial.print(“Sensor contains “); Serial.print(finger.templateCount); Serial.println(” templates”);
Serial.println(“Waiting for valid finger…”);
}

void loop() // run over and over again

{
getFingerprintIDez();
```

```
delay(50); //don’t ned to run this at full speed.

digitalWrite(12, LOW);

digitalWrite(11, LOW);

}


uint8_t getFingerprintID() {

uint8_t p = finger.getImage();

switch (p) {

case FINGERPRINT_OK:

Serial.println(“Image taken”);

break;

case FINGERPRINT_NOFINGER:

Serial.println(“No finger detected”);

return p;

case FINGERPRINT_PACKETRECIEVEERR:

Serial.println(“Communication error”);

return p;

case FINGERPRINT_IMAGEFAIL:

Serial.println(“Imaging error”);

return p;

default:

Serial.println(“Unknown error”);

return p;

}


// OK success!
```

```
p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
Serial.println(“Image converted”);
break;
case FINGERPRINT_IMAGEMESS:
Serial.println(“Image too messy”);
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println(“Communication error”);
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println(“Could not find fingerprint features”);
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println(“Could not find fingerprint features”);
return p;
default:
Serial.println(“Unknown error”);
return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
```

```
Serial.println(“Found a print match!”);
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println(“Communication error”);
return p;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println(“Did not find a match”);
return p;
} else {
Serial.println(“Unknown error”);
return p;
}
{digitalWrite(11, HIGH);
delay(3000);
digitalWrite(11, LOW);
Serial.print(“Not Found”);
Serial.print(“Error”);
return finger.fingerID;
}
 // found a match!
Serial.print(“Found ID #”); Serial.print(finger.fingerID);
Serial.print(” with confidence of “); Serial.println(finger.confidence);

return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
```

```
int getFingerprintIDez() {

uint8_t p = finger.getImage();

if (p != FINGERPRINT_OK) return -1;


p = finger.image2Tz();

if (p != FINGERPRINT_OK) return -1;


p = finger.fingerFastSearch();

if (p != FINGERPRINT_OK) return -1;


// found a match!


{

digitalWrite(12, HIGH);

delay(3000);

digitalWrite(12, LOW);

Serial.print(“Found ID #”); Serial.print(finger.fingerID);

Serial.print(” with confidence of “); Serial.println(finger.confidence);


} }
```

7. Then we start the ESP32CAM code

```
#include "esp_camera.h"

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

//
```

```
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#include "camera_pins.h"

#define BUTTON 13
#define PHOTO 14
#define LOCK 12
#define LED 4

const char* ssid = "YOUR SSID";
const char* password = "PASSWORD";
char auth[] = "AUTH TOKEN";  //sent by Blynk

String local_IP;

void startCameraServer();

void takePhoto()
{
  digitalWrite(LED, HIGH);
  delay(200);
  uint32_t randomNum = random(50000);
  Serial.println("http://"+local_IP+"/capture?_cb="+ (String)randomNum);
  Blynk.setProperty(V1, "urls", "http://"+local_IP+"/capture?_cb="+(String)randomNum);
  digitalWrite(LED, LOW);
  delay(1000);
}

void setup() {
  Serial.begin(115200);
  pinMode(LOCK,OUTPUT);
  pinMode(LED,OUTPUT);
  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_sscb_sda = SIOD_GPIO_NUM;

  config.pin_sscb_scl = SIOC_GPIO_NUM;

  config.pin_pwdn = PWDN_GPIO_NUM;

  config.pin_reset = RESET_GPIO_NUM;

  config.xclk_freq_hz = 20000000;

  config.pixel_format = PIXFORMAT_JPEG;


  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality

  //                      for larger pre-allocated frame buffer.

  if(psramFound()){

    config.frame_size = FRAMESIZE_UXGA;

    config.jpeg_quality = 10;

    config.fb_count = 2;

  } else {

    config.frame_size = FRAMESIZE_SVGA;

    config.jpeg_quality = 12;

    config.fb_count = 1;

  }


  // camera init

  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();
  // initial sensors are flipped vertically and colors are a bit saturated
  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
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

  WiFi.begin(ssid, password);

  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());

  local_IP = WiFi.localIP().toString();

  Serial.println("' to connect");

  Blynk.begin(auth, ssid, password);

}


void loop() {

  // put your main code here, to run repeatedly:

  Blynk.run();

  if(digitalRead(BUTTON) == HIGH){

  Serial.println("Send Notification");

  Blynk.notify("Someone is at the door...");

  }

  if(digitalRead(PHOTO) == HIGH){

  Serial.println("Capture Photo");

  takePhoto();

  delay(1000);

  }

  if(digitalRead(LOCK) == HIGH){

  Serial.println("Unlock Door");

  }

}
```

CHAPTER 5

Logic and Operation of Hardware Model

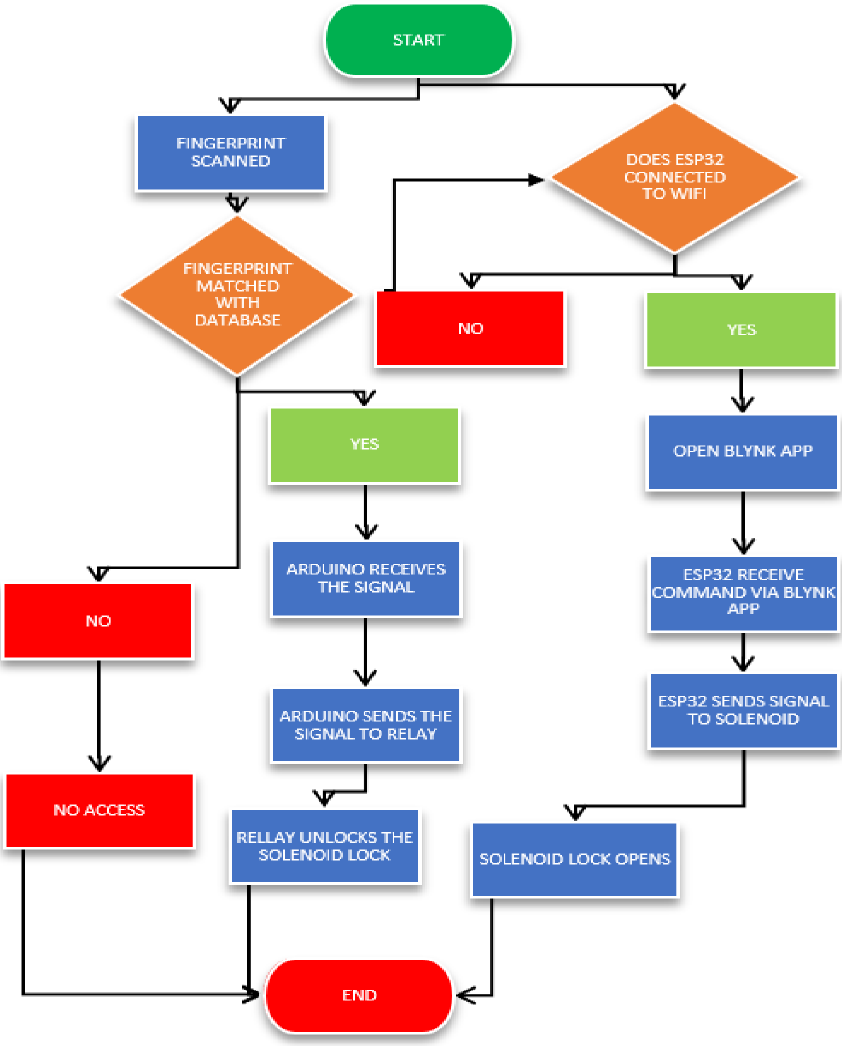
INTRODUCTION-

After assembling the system, what remains is to observe its operation and efficiency of the system. Thetotal system is divided in several sub systems, like

- Fingerprint Sensor section
- Arduino Section
- ESP32 Cam section
- Blynk App section
- Relay Section
- Solenoid Section

The operation of the whole circuit is depending on every section's performance.

5.2 FLOW CHART


START
FINGERPRINT SCANNED
DOES ESP32 CONNECTED TO WIFI
FINGERPRINT MATCHED WITH DATABASE
NO
YES
YES
OPEN BLYNK APP
ARDUINO RECEIVES THE SIGNAL
NO
ESP32 RECEIVE COMMAND VIA BLYNK APP
ARDUINO SENDS THE SIGNAL TO RELAY
ESP32 SENDS SIGNAL TO SOLENOID
NO ACCESS
RELLAY UNLOCKS THE SOLENOID LOCK
SOLENOID LOCK OPENS
END

5.3 PRINCIPLE AND OPERATION

According to the flow chart there is two ways to unlock the solenoid lock system that is by biometric system and wirelessly by Blynk app. To use the wireless system, we need to ensure that the ESP32 is connected to a wi-fi network. If the user clicks the unlock button in the Blynk app then the ESP32 send signal to the solenoid lock and the lock opens. The lock can also be opened by the fingerprint sensor. The user needs to put the finger in the sensor and if the database matches with the user's fingerprint, then the relay is triggered and the solenoid lock opens. Below is the complete flow chat of the above discussed matter.

Blynk app uses token to confirm the identity of the network as well as the device because we burn the code which consists the token from Blynk app to be able to run properly.

```
#include <BlynkSimpleEsp32.h>

#define BLYNK_TEMPLATE_ID "TMPLomUBVNpw"

#define BLYNK_TEMPLATE_NAME "Quickstart Template"

#define BLYNK_AUTH_TOKEN "YGjWJdQ8lEXXI1RLrpYUsXRXyTrS1rsx"
```

This is an example of the token and other necessary lines of code to run the Blynk environment properly.

5.4 ADVANTAGES OF SMART DOOR LOCK

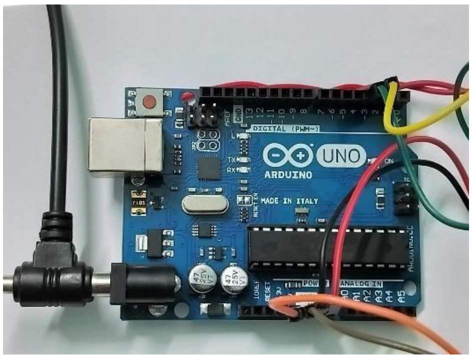
- **Fingerprint Sensor:** It consists a fingerprint sensor to unlock the door so only the registered fingerprints have access to open the door, so it is a very secure solution.
- **Wireless connectivity through Blynk:** It uses esp32 cam which can be remotely connected through Wi-Fi with Blynk app to unlock the door. It is the most easy and effective way and also very safe because the Blynk app uses
- **12-Volt Operation:** This system requires very less power to operate so any power outlet can handle the power that needs to be delivered.
- **Cost Effective solution:** This door lock requires very less power to operate and the maintenance cost is very low and provides maximum level of security and is also future proof solution.

5.5 COST ESTIMATION OF THE PROJECT

SL. NO.	COMPONENTS	QUANTITY	COST(INR)
1	ESP32 Microcontroller	1	250
2	Relay Module	1	40
3	BC547 NPN Transistor	1	5
4	220-ohm Resistor	1	5
5	1 K ohm Resistor	1	2
6	10 k ohm Resistor	1	2
7	LED	1	2
8	FTDI 232 USB to serial interface board	1	120
9	12 Volt DC Supply	1	100
10	Arduino Uno	1	600
11	Arduino Cable	1	30
12	Finger print Sensor	1	900
13	Jumper wire	As Required	50
14	Solenoid Lock	1	300
15	Micro SD Card	1	200
16	Bread Board	1	60
TOTAL			2666/-

Table: Cost of the Project

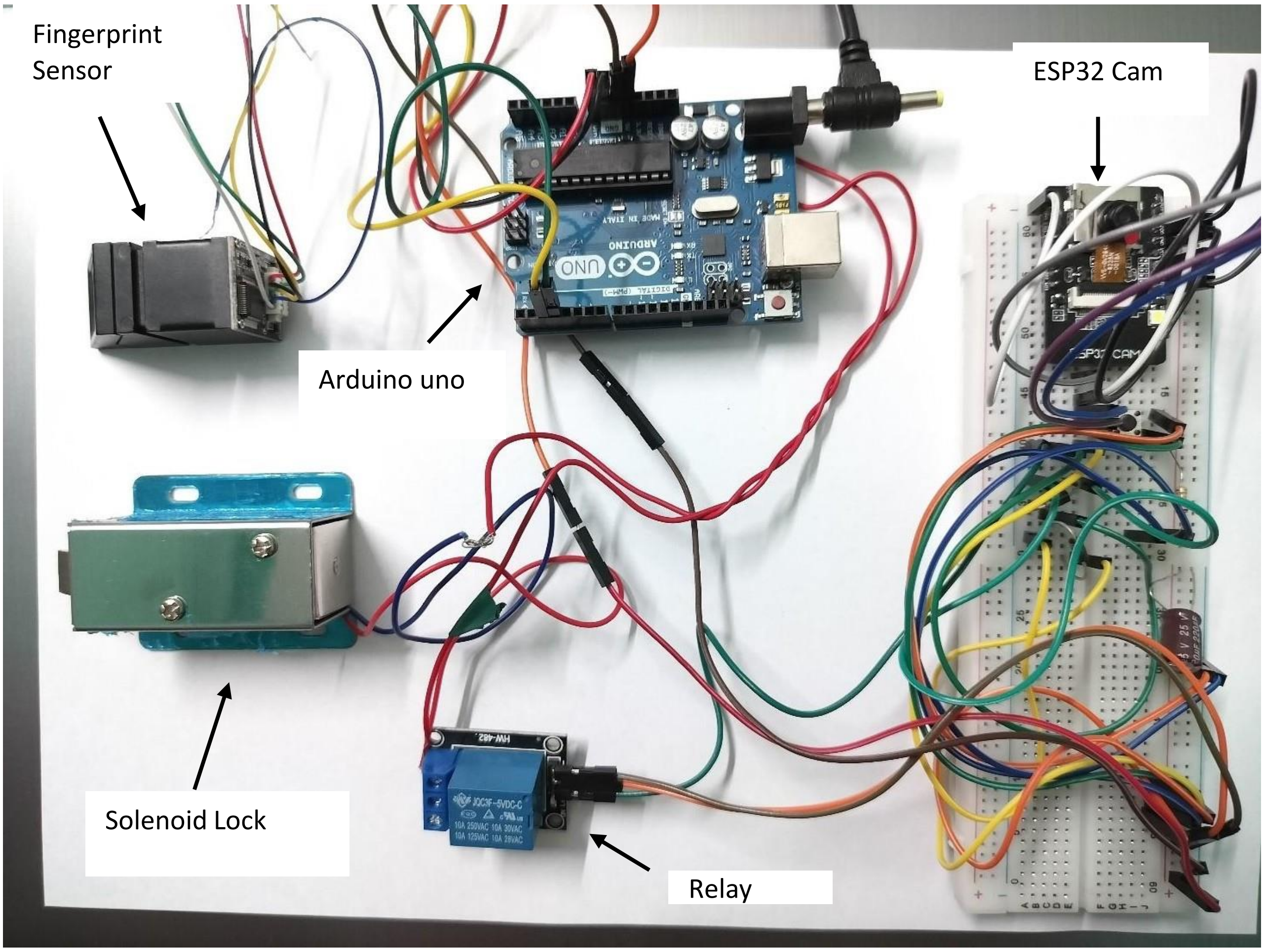
5.5 PHOTOGHAPHS OF THE PROTOTYPE



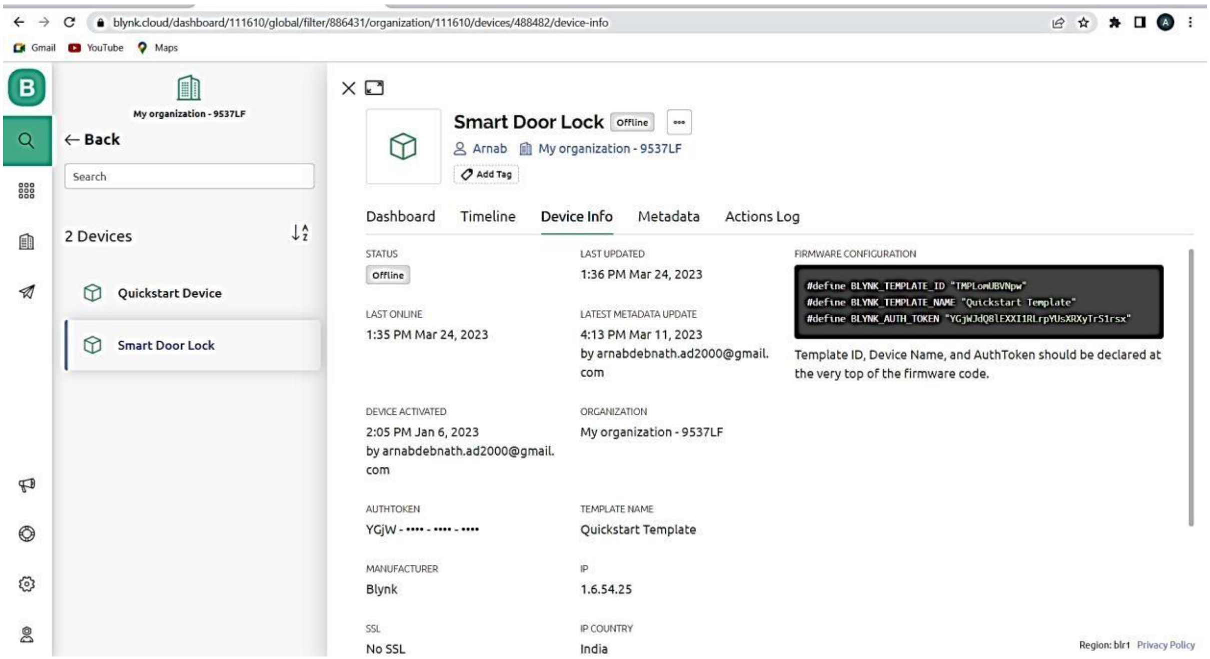
Arduino UNO

ESP32 cam

Fingerprint Sensor



Pic: Main Prototype



blynk.cloud/dashboard/111610/global/filter/886431/organization/111610/devices/488482/device-info
Gmail
YouTube
Maps
My organization - 9537LF
Back
Search
2 Devices
Quickstart Device
Smart Door Lock

Smart Door Lock Offline

Arnab · My organization - 9537LF

STATUS
Offline

LAST UPDATED
1:36 PM Mar 24, 2023

LAST ONLINE
1:35 PM Mar 24, 2023

LATEST METADATA UPDATE
4:13 PM Mar 11, 2023
by arnabdebnath.ad2000@gmail.com

DEVICE ACTIVATED
2:05 PM Jan 6, 2023
by arnabdebnath.ad2000@gmail.com

ORGANIZATION
My organization - 9537LF

AUTHTOKEN
YGjW - •••• - •••• - ••••

TEMPLATE NAME
Quickstart Template

MANUFACTURER
Blynk

IP
1.6.54.25

SSL
No SSL

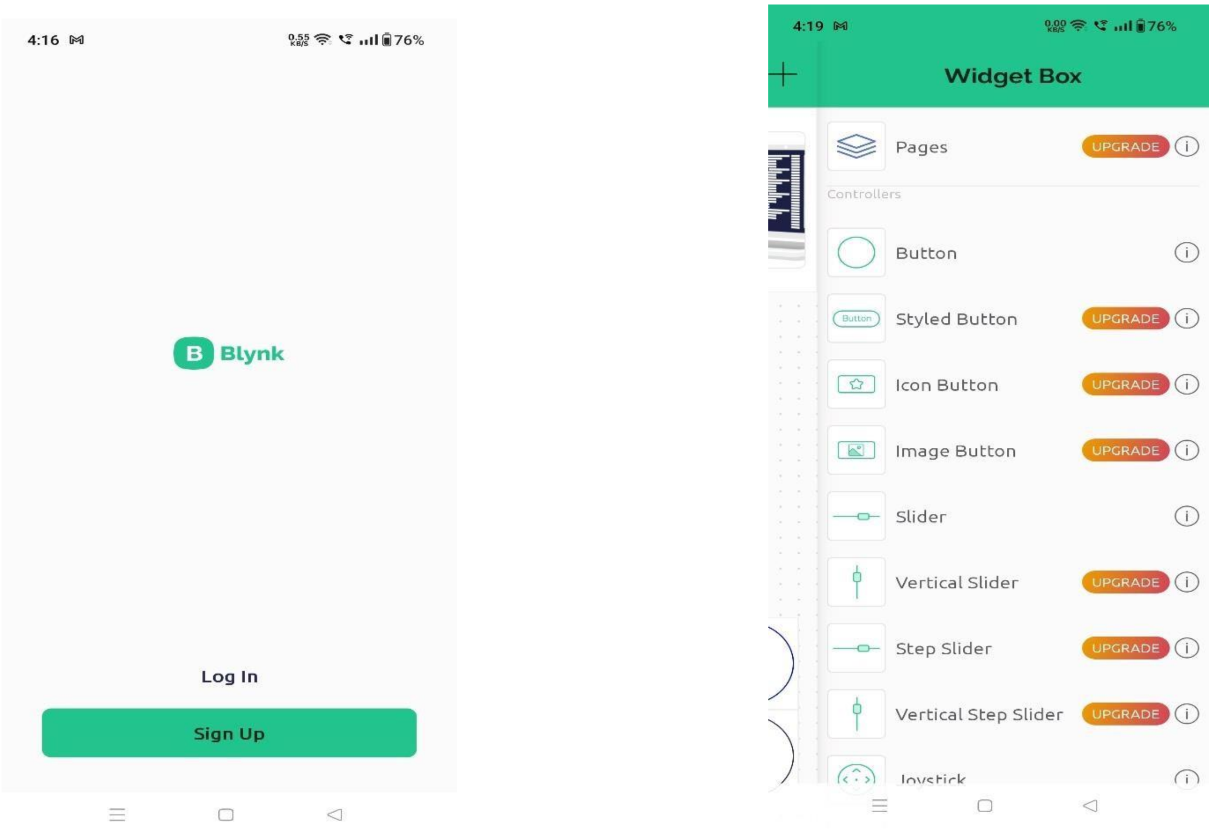
IP COUNTRY
India

FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPLomUBVNpw"
#define BLYNK_TEMPLATE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "YGjWJdQ8lEXXI1RLrpYUsXRXyTrS1rsx"
```

Template ID, Device Name, and AuthToken should be declared at the very top of the firmware code.

Pic: Configuring the Blynk App



Pic: Blynk Mobile App configuration

CHAPTER 6

Conclusion and Future Scope

6.1 CONCLUSION

The proposed system allows remote access to lock or unlock the door without physical user interaction. The system fulfils the requirements of supporting autonomous locking device and easy fingerprint sensing compared to physical keys. The system has minimum requirements for hardware and supports customization of keys. The prototype-built shows that the design consumes minimal power and the locking/unlocking of the door happens in 2 seconds on an average. Thus, the system proposed is feasible.

6.2 FUTURE SCOPE

This project can also be modified with RFID and artificial intelligence to recognize the person to open the door lock hands free. Smart door lock has huge potential in the future market. It is getting popular every day. Over the coming decade, predictions range on how exactly common mechanical devices will change. WiFi connectivity and Bluetooth are among the foremost technologies we may see. The next iteration of Ultra-Wideband (UWB) is another technology to look for. “UWB will enable hands-free access to entry and exit points,” says Broiskin “It can decipher whether a credential holder is simply walking by a door or is actually walking toward the door to enter.”

AI is another surprising prediction for the coming years. While AI today resides mostly in the space of video and analytics, there are exciting possibilities for the future of AI with door security and locks, including knowing how many times a door was accessed at what part of the day and how a user typically behaves. This type of technology can go a long way into securing your home or business in a smart, hassle-free way.

APPENDIX A

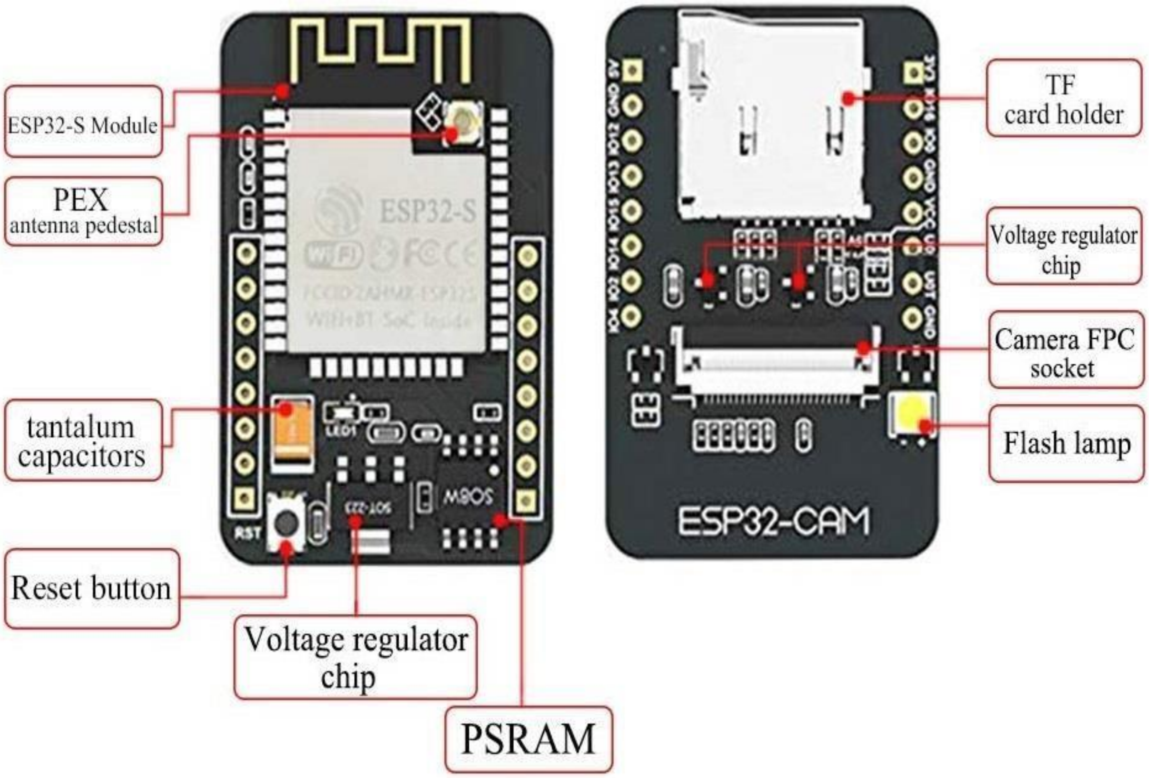
Hardware Despriction

1. ESP32 CAMERA MODULE

The ESP32-CAM is a small size, low power consumption camera modulebased on ESP32. It comes with an OV2640 camera and provides onboardTF card slot.
The ESP32-CAM can be widely used in intelligent IoT applications such as wireless video monitoring, WiFi image upload, QR identification, andso on.

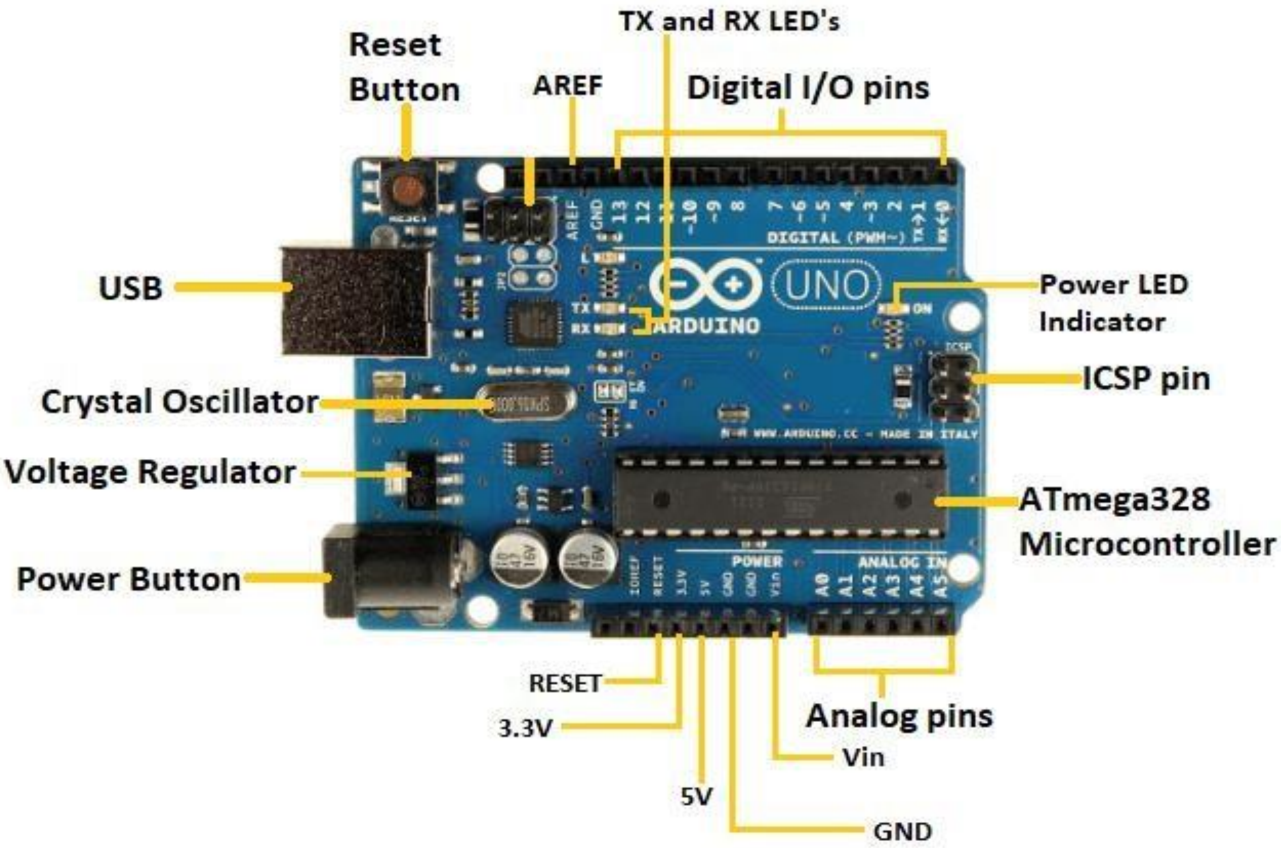
The ESP32-CAM suit for IOT applications such as:

- Smart home devices image upload
- Wireless monitoring
- Intelligent agriculture
- QR wireless identification
- facial recogniti



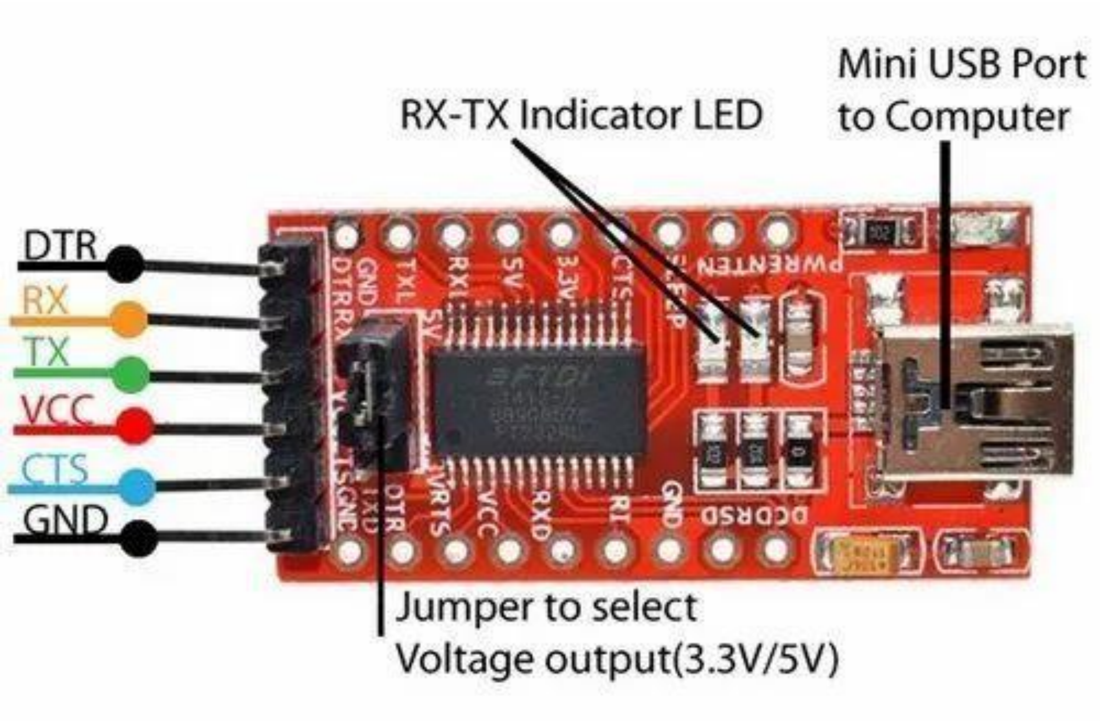
ARDUINO UNO

The Arduino UNO SMD is frequently used microcontroller board in thefamily of an Arduino. This is the latest third version of an Arduino boardand released in the year 2011. The main advantage of this board is if we make a mistake, we can change the microcontroller on the board. The main features of this board mainly include, it is available in DIP (dual- inline-package), detachable and ATmega328 microcontroller. The programming of this board can easily be loaded by using an Arduino computer program. This board has huge support from the Arduino community, which will make a very simple way to start working in embedded electronics, and many more applications Fig 1: ARDUINO UNO.



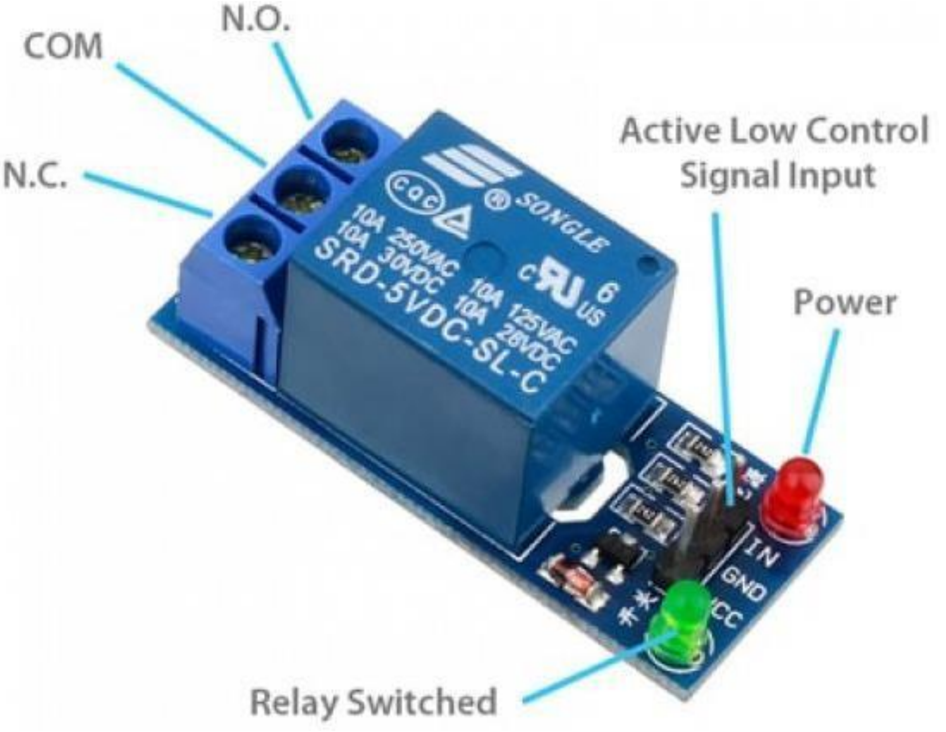
FTDI CHIP-

FTDI CHIIP Original FT232R chips are one of the newer chips manufactured by FTDI (Future Technology Devices International). Apartfrom being an USB to serial UART, it has an integrated EEPROM and optional clock generator output. The chip also incorporates FTDI Chip- ID functionality (giving each chip a unique identifier for security) and USB termination resistors. Cloned boards (with a cloned chip) will likelyexclude the unique ID functionality and does not include an EEPROM, meaning that signals cannot be inverted.



RELAY MODULE-

A Relay is a device that opens or closes an auxiliary circuit under some pre-determined condition in the Main circuit. The object of a Relay is generally to act as a sort of electric magnifier, that is to say, it enables a comparatively week current to bring in to operation on a much stronger current. It also provides complete electrical isolation between the controlling circuit and the controlled circuit. Relays are the switches which aim at closing and opening the circuits electromechanically.



PUSH BUTTONS-

A Push Button is a type of switch work on a simple mechanism called “Push-to-make”. Initially, it remains in off state or normally open state butwhen it is pressed, it allows the current to pass through it or we can say itmakes the circuit when pressed. Normally their body is made up of plastic or metal in some types.

R307 Fingerprint Module -

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 templatestorage and other functions.

- Perfect function: independent fingerprint collection, fingerprint registration, fingerprint comparison (1: 1) and fingerprint search (1: N)function.

- Small size: small size, no external DSP chip algorithm, has beenintegrated, easy to install, less fault.

- Ultra-low power consumption: low power consumption of the productas a whole, suitable for low-power requirements of the occasion.

- Anti-static ability: a strong anti-static ability, anti-static index reached15KV above.

- Application development is simple: developers can provide control instructions, self fingerprint application product development, withoutthe need for professional knowledge of fingerprinting.

- Adjustable security level: suitablefor different applications, security levels can be set by the user to adjust.

- Finger touch sensing signal output, low effective, sensing circuit standbycurrent is very low, less than 5uA

Solenoid Electric Door Lock

12V Solenoid lock are basically electromagnets: they are made of a bigcoil of copper wire with an armature (a slug of metal) in the middle.
When the coil is energized, the slug is pulled into the center of the coil. This makes the solenoid able to pull from one end. This solenoid lock inparticular is nice and strong, and has a slug with a slanted cut and a goodmounting bracket. It's basically an electronic lock, designed for a basic cabinet or safe or door. Normally the lock is active so you can't open thedoor because the solenoid slug is in the way. It does not use any power
in this state. When 9-12VDC is applied, the slug pulls in so it doesn't stick out anymore and the door can be opened. The solenoid lock comewith the slanted slug as shown above, but you can open it with the twoPhillips-head screws and turn it around so its rotated 90, 180 or 270 degrees so that it matches the door you want to use it with. To drive a solenoid lock with an Arduino you will need a relay module fairly goodpower supply, as a lot of current will rush into the solenoid to charge up the electro-magnet, about 500mA, so don't try to power it with a 9V battery.

CHAPTER 8

Reference

Reference

[1]K.Rajesh, ASST. PROFESSOR, B.VenkataRao, P.AV.S.K.Chaitanya, A.Ruchitha Reddy, “SMART DOOR UNLOCK SYSTEM USING FINGERPRINT” Pramana SL. NO. COMPONENTS QUANTITY 1 ESP32 Microcontroller 1 2 Relay Module 1 3 BC547 NPN Transistor 1 4 220-ohm Resistor 1 5 1 K ohm Resistor 1 6 10 k ohm Resistor 1 7 LED 1 8 FTDI 232 USB to serial interface board 1 9 12 Volt DC Supply 1 10 Arduino Uno 1 11 Arduino Cable 1 12 Finger print Sensor 1 13 Jumper wire As Required 14 Solenoid Lock 1 15 Micro SD Card 1 16 Bread Board 1 Research Journal Volume 9, Issue 3, 2019 ISSN NO: 2249-2976 doi: 10.1120/ICECCO.2019.22492976

[2] LiaKamelia, AlfinNoorhassan S.R, MadaSanjaya and W.S., Edi Mulyana, “DOOR-AUTOMATION SYSTEM USING BLUETOOTH-BASED ANDROID FOR MOBILE PHONE” VOL. 9, NO. 10, OCTOBER 2014 ISSN 1819-6608 doi: 12.34/RRTIFN.2014.2345678 ARPN Journal of Engineering and Applied Sciences.

[3] Adarsh V Patil, SreevarshaPrakash, Akshay S, Mahadevaswamy, ChandanBPatgar, Sharath Kumar A J, “Android Based Smart Door Locking System” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 doi: 56.90/JUVKL.2018.3489076 Published by, www.ijert.org NCESC - 2018 Conference Proceedings.

[4] Dr.M.SivaSangari, Dhivakar.E, Gowthaam.K, “Secret Knock Detecting Door Lock” Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 5, 2021, Pages. 406-410 Received 15 April 2021; Accepted 05 May 2021 doi: 22.89/ASXK/2021.265908

[5] Badamasi, Y.A., "The working principle of an Arduino," in Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on , vol., no., pp.1-4, Sept. 29 2014-Oct. 1 2014 doi: 10.1109/ICECCO.2014.6997578.

[6] Galadima, A.A., "Arduino as a learning tool," in Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on,vol.,no.,pp.1-4, Sept. 29 2014-Oct. 1 2014 doi: 10.1109/ICECCO.2014.6997577.

[7] Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things Conference Paper · September 2017 DOI: 10.1109/ITECHA.2017.8101926.

[8] Using the ESP32 Microcontroller for Data Processing Conference Paper · May 2019 DOI: 10.1109/CarpathianCC.2019.876594

[9] Sebastian, S., Ray, P.P., 2015. Development of IoT invasive architecture for complying with health of home. In: Proceedings of I3CS, Shillong, pp.

[10] G. Yang, X. Li, M. Mäntysalo, X. Zhou, Z. Pang, L.D. Xu , S.K. Walter, Q. Chen, L. ZhengA,2014 health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor and intelligent

medicine box.

[11] Justin Lahart (27 November 2009). “Taking an Open- Source Approach to Hardware”. The Wall Street Journal. https://en.m.wikipedia.org/wiki/.

[12] C. Floerkemeier, C. Roduner, M. LampeRFID application development with the Accada middleware platform

[13] J.-i. Jeong,Department of Law, Kyonggi University, Iui- Dong, Yeongtong-Gu.

[14]https://www.tutorialspoint.com/php/index.htm [2

[15] https://harshsharmatechnicals.com/2021/03/27/keypad-lock/

[16]https://www.youtube.com/watch?v=XLfgRNciGDc

[17]] R. Collobert, F. Sinz, J. Weston, and Bottou.Largescaletransductivesvms.JMLR, 2006

[18] A data mining approach to characterize road accident locations”, Sachin Kumar, Durga Toshniwal.

[19] Road traffic accidents prediction modelling: An analysis of Anambra State, Nigeria” Chukwutoo C. Ihueze, Uchendu O. Onwurah, Department of Industrial and Production.Engineering, Nnamdi Azikiwe University,

[20] An Integrated Approach for Weather Forecasting based on Data Mining and Forecasting Analysis”, G. Vamsi Krishna Research scholar, Department of CSE, GITAM University.

[21] Shewta Chanda, Deepak Rasaily, PrernaKhulal, â€œDesign and Implementation of a Digital Code Lock using Arduinoâ€

[22] Adamu Murtala Zungeru, â€œAn Electronic Digital Combination Lock: A precise and reliable security system.â€

[23] Janaki Venukumar, Naveen. S, â€œArduino based Door Access Controlâ€

[24] Atzori, L., Irea, A., Morabito, G.: The Internet of Things.

[25] Kim, H.S., Park, D.R., Cha, J.W., Kim, Y.C.: â€œDesign of Data Encryption Module using AES/SEED and Implementation of Multimedia Security Systemâ€