**Enhancing Security and Protection for ATM’s**

**ABSTRACT:**

The main theme of prevention and providing security to atm from atm robberies. Now a days these atm robberies are common and frequently occurred. For example, stolen money from atm by damaging atm machines. Now we want to deals with prevention of atm thefts and providing security to atm machines from atm robberies. Whenever robbery occurs we used sensors which helps us to produce beep sound from buzzers. The entire scenario was recorded by camera ,which is sending video continuously to pc and it will be saved in computer. By using some hardware components, the entire scenario, robbery occurred time and its location send to near by police station and corresponding bank. Whenever theft wanted to do robbery or try to break the atm machine, vibrating sensors was sends protocol to door and that door will locked automatically. After an ATM is broken and cash is taken out from it, the GPS module is immediately switched on and sends the location of the respective place to the nearby police station. In the meantime, if the thief tries to break the door or damage the camera, the ATM responds to send alert messages to the nearby police station and the corresponding bank.

**KEYWORDS:** Arduino board, USB Camera, Vibrating sensor, Servo motor, LCD display, Buzzer, embedded C.

# 

# INDEX

**Chapter Page No**

**1.Introdution 1-2**

1.1 motivation

1.2 problem statement

1.3 Objectives

1. **REQUIREMENT ANALYSIS 3-8**

2.1Software libraries

* 1. Machine learning
  2. Deep learning
  3. Open cv
  4. Computer vision
  5. NumPy
  6. python
  7. Organization of project

1. **Embedded System**  **8-11** 
   1. Embedded System architecture
   2. Central processing unit
   3. Memory
   4. Input & output

3.5use case diagram

1. **Hardware essentials 11-19**

4.1Arduino

* 1. pi camera
  2. Dc motor
  3. Buzzer
  4. Vibration sensor
  5. LCD module
  6. GPS Tracker

1. **WORKING 19-25** 
   1. Setting Up internet connection on the Arduino:
2. **IMPLEMENTATION 26-42** 
   1. SOURCE CODE
   2. GPS Tracker
   3. ARUNDIO CODE
   4. CAMERA MODULE CODE

**7.DESGIN 43-45**

**8.RESULTS AND OUTPUT 46-47**

**9.CONCLUSION 48-49**

**INTRODUCTION**

* 1. **MOTIVATION**: In today’s technically advanced world, autonomous systems are gaining rapid popularity. As the social computerization and automation has been increased and the ATM and credit card has been installed and spread out to simplify the activity for financial activity, the banking activity has been simplified, however the crime related with financial organization has been increased in proportion to the ratio of spread out of automation and devices. Those crimes for the financial organization have been increased gradually from year 1999 to 2003, little bit decreased in 2004, and then increased again from year 2005.

In the year of 2007, 212,530 of theft and 4,439 of robber cases are happened, and 269,410 of theft and 4,409 of robber cases are happened in year 2010 and also in the year 2011, 270,109 of theft and 4,509 of robber cases are happened .so that the cases of theft and robber have been increased gradually during past 12 years. Among the crime for financial organization, the cases of theft and robber have very high proportion of over 90% and the crime for the ATM has been increased because the external ATM has been increased and it is always exposed to the crime.

Therefore, this study is going to suggest the method of rapid reaction and minimization of loss by detecting the ATM machine at real-time.

. Pgno: 2

* 1. **PROBLEM STATEMENT:**

Today robberies are increasing day by day in many fields, so to reduce the theft that are happening in the public sector. The main aim of the project is to stop the robberies by detecting the theft using vibrating sensor and locking the main doors. And also, it immediately sends a SMS to the police and sounds a buzzer to alert the surroundings.

* 1. **OBJECTIVES:**

The main objective of this project is to reduce the ATM robberies using An Advanced technologies like raspberry pi. The objectives of the proposed system are

* To overcome the ATM theft.
* Restrict the entry of any unauthorized person.

* Provide more security.
* Formal step towards smart city.

The proposed setup can be installed at public places like schools, colleges, shopping malls, supermarkets, parks. ATM etc., where the chances occurring of robberies.

Pgno:3

1. **REQUIREMENT ANALYSIS:**

**2.1Software libraries:**

* 1. **Machine learning:**

Machine learning (ML) is a subset of artificial intelligence (AI) that involves the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. These algorithms learn from and make predictions or decisions based on data. The primary goal of ML is to enable computers to learn from experience and improve their performance over time.

**Key Concepts in Machine Learning**

* + 1. **Supervised Learning**: Algorithms are trained on labeled data, which means the input comes with the correct output. The model makes predictions and is corrected by the known labels.
    2. **Unsupervised Learning**: Algorithms are trained on data without labels. The model tries to learn the patterns and the structure from the data.
    3. **Reinforcement Learning**: Algorithms learn by interacting with an environment. They make a sequence of decisions to maximize some notion of cumulative reward.

* **Deep learning:** Deep learning is a subset of machine learning that involves neural networks with many layers (hence the term "deep").
* These neural networks are designed to simulate the way the human brain processes information, allowing the model to learn and make complex decisions based on vast amounts of data.

**Key Concepts in Deep Learning**

* + 1. **Neural Networks**: The fundamental building blocks of deep learning. These networks consist of layers of interconnected nodes (neurons) that process and transmit information.
    2. **Layers**: Deep neural networks have multiple layers, including an input layer, several hidden layers, and an output layer. Each layer extracts different features from the data.
    3. **Training**: Deep learning models are trained using large datasets and computational power to adjust the weights of the connections between neurons, minimizing the difference between predicted and actual outcomes.
    4. **Activation Functions**: Functions applied to the output of each neuron to introduce non-linearity, enabling the network to learn complex patterns.
    5. **Backpropagation**: The process of adjusting weights in the network by propagating errors backward through the layers during training.

**OPENCV:**Open-Source Computer Vision Library (OpenCV) is the open-source deep learning and computer vision software library. It was built to offer some mutual structure computer vision systems and for acceleration of machine perception usage on commercial products. OpenCV makes business operations easy by utilizing and modification of the code as it is based on BSD-licensed product. There are over 2500 algorithms that are optimized in the library, that includes a complete set of classic and state of art deep learning and computer vision. Pgno:5

* 1. **Computer vision:**

Computer vision involves with how computers programs can obtain higher level of comprehending of digital videos or images. In other words, the computer systems seek to comprehend and automate the responsibilities that human visual programs do. The following are sample task of computer vision; i. Extraction of high dimensional data from actual world environment as inputs to output symbolic or numerical information e.g., in the forms of decisions 9 ii. Procedures of acquiring, processing, analyzing and comprehending digital videos or images. This means that the visual images are transformed into descriptive nature of the actual world which makes logic to thought

procedures and can cause applicable action. The image comprehending cab be seen as the unravelling of representational information from the acquired image data using the designed models by help of geometry, statistics, physics and theory of learning. Computer vision discipline is concerned with artificial systems theory which extracts data from digital images. The acquired image information is presented in various forms such as; in sequences of videos, multiple views from the cameras, 3D scanners multi-dimensional data or scanning devices from medical. The computer vision technological discipline strives for application of its own theories and models to aid the construction of systems with computer vision. Also, computer vision will entail with automated extraction, analyzing and comprehending of essential information acquired from a particular image or the sequences of multiple image.

* 1. **NumPy:**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations and random simulation. As it is known that mathematics is the foundation of machine learning, most of the mathematical tasks can be performed using NumPy.

Pgno:7

* 1. **python:**

Python is an interpreter, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. We have used this software to write our facial detection program and all the necessary above said libraries are installed here.

* 1. **ORGANIZATION OF PROJECT:**

The project can be organized as follows.

Chapter1: Provides the introduction, objectives and scope of the project, existing and proposing systems.

Chapter 2: Discuss various literatures regarding the project and the best methods have been studied. This survey helped a lot in getting complete description of the project in various angles.

Chapter 3: Hardware and software requirements includes pin description of Raspberry pi and other required components.

Chapter 4: Design and Implementation which includes block diagram and working.

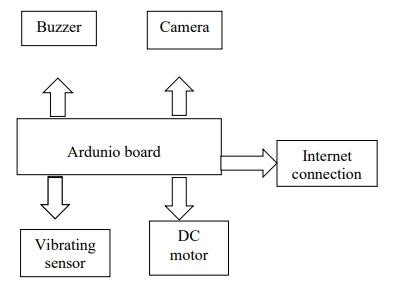
Chapter 5: Gives the results.Chapter 6: Gives the conclusion and future scope of advance anti-theft ATM security using Arduino

Pgno:8

## 3. Embedded System

**3.1 Embedded System architecture:**

An embedded system is a specialized computing system designed to perform dedicated functions or tasks within a larger system. Unlike general-purpose computers, embedded systems are typically optimized for specific operations and are integrated into hardware and software. They often operate with real-time constraints and are designed to be highly reliable, efficient, and compact. Embedded systems can be found in a wide range of applications, from consumer electronics like smartphones, washing machines, and televisions to industrial machines, automotive systems, medical devices, and more. They consist of a microcontroller or microprocessor, memory, input/output interfaces, and software, all working together to perform specific functions. These systems are usually designed to run continuously and autonomously, providing critical functionality in various applications. The integration of embedded systems with IoT technologies has further expanded their capabilities, enabling advanced features like remote monitoring, data collection, and automation.



Pageno.9

**3.2 Central processing unit (CPU):**

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A microcontroller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, Analog-to digital converter etc.

**3.3 MEMORY:**

The memory is categorized as Random Access Memory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is executed.

**3.4 INTERNET OF THINGS:**

The major concept using in this project is the Internet of Things. The Internet of Things (IoT) can be connecting various types of objects like smart phones, personal computer and tablets to the internet, which brings new-fangled type of communication between things and things, and things and people. Any man-made objects that can be assigned an IP address and it has the ability to transfer data successfully over a network, the interaction through a network is called IoT. The internet helps us to bring immediate solutions for many problems and able to connect from any of the remote places. The Internets of Things technology is used to come in with innovative idea and large development space for smart homes to improve the living standards of life.

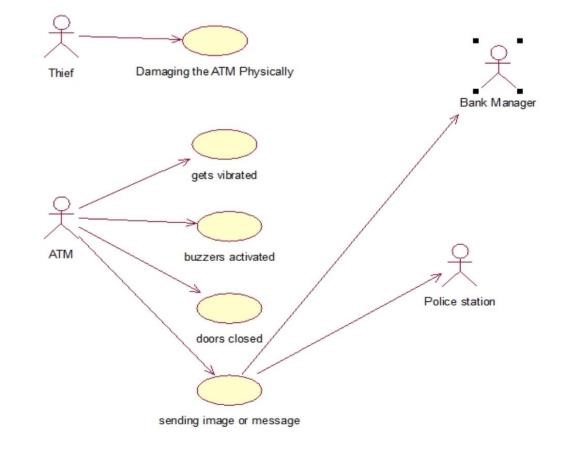
The growth of the Internet of Things will reform a number of sectors, like healthcare, automation energy, transportation, etc. The cloud computing can be used in such case to implement the IoT infrastructure that augmented with sensors and actuators to monitor and control “things” from anywhere.

**Pgno:10**

**3.5 Use case diagram:**

A utilization case chart is a sort of conduct diagram characterized by and created from a Use-case examination in the Unified Modeling Language UML (UML). It will probably get a graphical portrayal of cutting-edge materials regarding entertainers, targets (addressed as Pgno:11

use cases), and any incongruencies between those utilization cases. A utilization case chart's primary objective is to show which framework exercises are led for which entertainer. The jobs of the framework's entertainers can be highlighted



Pgno:12

## 4.Hardware essentials

**4.1Arduino:**

**Pin description:**

GPIO are the standard pins that simply be used to turn devices on and off.I2C (Inter Integrated Circuit) pins allow connecting and talking to hardware modules that support this protocol (I2C Protocol). This will typically take up 2 pins. SPI (Serial Peripheral Interface Bus) pins can be used to connect and talk to SPI devices. UART (Universal Asynchronous Receiver/Transmitter) are the serial pins used to communicate with other devices. DNC stands for do not connect. The power pins pull power directly from the Raspberry Pi.[4] GND are the pins used to ground the devices.



Pgno:13

**Key Applications:**

* Low-cost PC/tablet/laptop
* IoT applications 16
* Media centre
* Robotics
* Industrial/Home automation

**4.2 Pi camera:**

This 5 mega pixels sensor with OV5647 camera module is capable of 1080p video and still images that connect directly to your Raspberry Pi. This is the plug-and play-compatible latest version of the Raspbian operating system, making it perfect for time-lapse photography, recording video, motion detection and security applications. Connect the included ribbon cable to the CSI (Camera Serial Interface) port on your Raspberry Pi, and you are good to go!



FEATURES:

* Compatible with Raspberry Pi 4 Model B/3B+/3B/2B/Zero Wireless

Pgno:14 ❖ 5 Megapixel OV5647 Camera

* Camera specifications
* Static Images Resolution: 2592×1944
* Supported Video Resolution: 1080p/30 fps, 720p/ 60fps and 640 x480p 60/90 ❖ video recording
* Aperture (F): 1.8
* Visual Angle: 65 degrees
* Dimension: 24mmx23.5mmx8mm, Weight: 3g
* Interface: CSI connector
* Supported OS: Raspbian (latest version recommended)

**4.3 DC MOTOR:**

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. One of the easiest and inexpensive way to control DC motors is to interface L293D Motor Driver IC with Raspberry pi. It can control both speed and spinning direction of two DC motors. In order to have a complete control over DC motor, we have to control its speed and rotation direction. This can be achieved by combining these two techniques.

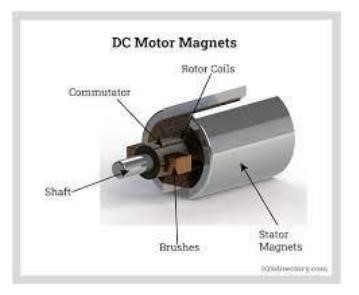
* PWM – For controlling speed
* H-Bridge – For controlling rotation direction

Pgno:15

**PWM – For controlling speed:**

The speed of a DC motor can be controlled by varying its input voltage. A common technique for doing this is to use PWM (Pulse Width Modulation), is a technique where average value of the input voltage is adjusted by sending a series of ON-OFF pulses. **H-Bridge For controlling rotation direction:**

The DC motors spinning direction can be controlled by changing polarity of its input voltage. A common technique for doing this is to use an H-Bridge. An H-Bridge circuit contains four switches with the motor at the center forming an H-like arrangement. Closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This causes change in spinning direction of the motor



**4.4 BUZZER:**

If the person has a facemask and the body temperature is below the protocol value then the buzzer will buzz for a single time, if not

Pgno:16

then it will buzz continuously for 5 times which will be helpful in alerting the respected authorities.



**Specifications:**

The specifications of the buzzer include the following.

* Color is black
* The frequency range is 3,300Hz
* Operating Temperature ranges from – 20° C to +60°C
* Operating voltage ranges from 3V to 24V DC
* The sound pressure level is 85dBA or 10cm
* The supply current is below 15mA.

**4.5. Vibrating sensor:**

A vibration sensor is a device that measures the amount and frequency of vibration in a given system, machine, or piece of equipment. Vibration sensors can be used to give maintenance teams insight into conditions within key assets that might lead to equipment failure, allowing them to predict the maintenance of the

Pgno:17 machinery, to reduce overall costs and increase the performance of the machinery.



**4.6 LCD MODULE:**

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day-to-day life, either at PCO’s or calculators. The appearance and the pinouts have already been visualized above now let us get a bit technical. 16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.



**4.7 GPS TRACKER:**

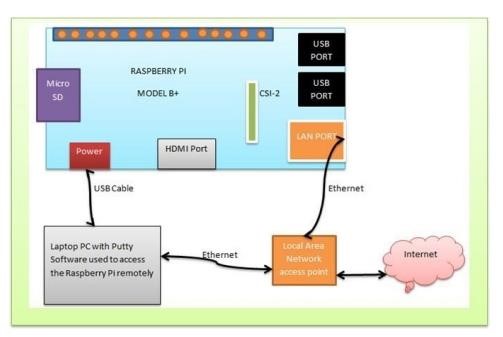
A GPS tracker in the context of IoT (Internet of Things) is a device that uses the Global Positioning System (GPS) to determine and transmit its precise location. Integrated with IoT technology, GPS trackers offer enhanced functionalities by connecting to the internet, allowing real-time location tracking, data analysis, and seamless communication with other devices and systems.

## 5. WORKING

This project deals with prevention of ATM theft from robbery. Whenever robbery occurs, Vibration sensor is used here which senses vibration produced from ATM machine. Once the vibration is sensed the beep sound will occur from the buzzer. DC Motor is used for closing the door of ATM. Stepper motor is used to leak the gas inside the ATM to bring the thief into unconscious stage. Camera is always in processing and sending video continuous to the PC and it will be saved in computer. RTC used to capture the robber occur time and send the robbery occur time with the message to the nearby police station and Pgno:19

corresponding bank through the GSM. Hear LCD display board using showing the output of the message continuously. This will prevent the robbery and the person involving in robbery can be easily caught. Here, Keil tools are used to implement the idea and results are obtained. Keil tools is used for run the DC motor and stepper motor for automatic door lock and also leak the gas inside the ATM.

**5.1.Setting Up internet connection on the Arduino:** Internet was necessary in so that the Pi can communicate over network protocols and thus allow for installation of necessary Python packages. The architecture below was used to achieve that.



Pgno:20

**6.** **IMPLEMENTATION**

**6.1 SOURCE CODE:**

**6.2 GPS Tracker**

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include <ESP\_Mail\_Client.h>

#include <WiFi.h>

static const int RXPin = 4, TXPin = 3; static const uint32\_t GPSBaud = 9600;

// The TinyGPS++ object

TinyGPSPlus gps;

// The serial connection to the GPS device

SoftwareSerial GPS(RXPin, TXPin);

// ESP32 Wi-Fi credentials

Pgno:21 const char\* ssid = "fruitu"; const char\* password = "12345678";

// Email credentials

#define emailSenderAccount "cameramodule95@gmail.com"

#define emailSenderPassword "\*\*\*\* \*\*\*\* \*\*\*\* \*\*\*\*"

#define smtpServer "smtp.gmail.com"

#define smtpServerPort 465

#define emailSubject "ESP32-CAM Photo Captured"

#define emailRecipient "pavantejveesam26@gmail.com"

// Pins connected to ESP32 const int pinA0 = A0; const int pinA1 = 8; const int pin13 = 13; const int pin14 = 14;

void setup() {

WRITE\_PERI\_REG(RTC\_CNTL\_BROWN\_OUT\_REG, 0); //disable brownout detector

Pgno:22

Serial.begin(115200);

Serial.println();

// Connect to Wi-Fi

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi..."); while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.print(".");

}

Serial.println();

// Print ESP32 Local IP Address

Serial.print("IP Address: http://");

Serial.println(WiFi.localIP());

// Initialize GPS

GPS.begin(GPSBaud);

// Set pin modes for Arduino pinMode(pinA0, INPUT);

Pgno:23 pinMode(pinA1, INPUT); pinMode(pin13, OUTPUT); pinMode(pin14, OUTPUT);

}

void loop() {

// Read the values of A0 and A1 pins int valueA0 = digitalRead(pinA0); int valueA1 = digitalRead(pinA1);

// Check if either A0 or A1 has a value of 1 if (valueA0 == 1 || valueA1 == 1) {

// Set the values of pins 13 and 14 to HIGH digitalWrite(pin13, HIGH); digitalWrite(pin14, HIGH);

// Additional actions can be performed here based on the condition sendEmail(); // Send email

sendGPSDetails(); // Send GPS details

} else {

// Reset pins 13 and 14 to LOW if both A0 and A1 are 0

Pgno:24 digitalWrite(pin13, LOW); digitalWrite(pin14, LOW);

}

// Add other functionalities as needed

delay(1000); // Adjust delay as needed

}

void sendEmail() {

// SMTP session config SMTPSession smtp; smtp.debug(1); smtp.callback(smtpCallback);

Session\_Config config;

config.time.ntp\_server = F("pool.ntp.org,time.nist.gov"); config.time.gmt\_offset = 0;

config.time.day\_light\_offset = 1;

config.server.host\_name = smtpServer;

Pgno:25

config.server.port = smtpServerPort; config.login.email = emailSenderAccount; config.login.password = emailSenderPassword; config.login.user\_domain = "";

// Message

SMTP\_Message message; message.sender.name = "ESP32-CAM"; message.sender.email = emailSenderAccount; message.subject = emailSubject;

message.addRecipient("Recipient", emailRecipient);

String htmlMsg = "<h2>Photo captured with ESP32-CAM and attached in this email.</h2>";

message.html.content = htmlMsg.c\_str(); message.html.charSet = "utf-8"; message.html.transfer\_encoding = Content\_Transfer\_Encoding::enc\_qp; SMTP\_Attachment att; att.descr.filename = "photo.jpg";

26 att.descr.mime = "image/png"; att.file.path = "/photo.jpg";

att.file.storage\_type = esp\_mail\_file\_storage\_type\_flash; att.descr.transfer\_encoding =

Content\_Transfer\_Encoding::enc\_base64; message.addAttachment(att);

// Connect to server and send email if (!smtp.connect(&config)) return;

if (!MailClient.sendMail(&smtp, &message, true))

Serial.println("Error sending Email, " + smtp.errorReason());

}

void sendGPSDetails() { while (GPS.available() > 0) { gps.encode(GPS.read());

if (gps.location.isUpdated()) {

Serial.print("Latitude= ");

Serial.print(gps.location.lat(), 6);

Serial.print(" Longitude= ");

27

Serial.println(gps.location.lng(), 6);

}

}

}

void smtpCallback(SMTP\_Status status) {

Serial.println(status.info());

if (status.success()) {

Serial.println("----------------");

Serial.printf("Message sent success: %d\n", status.completedCount());

Serial.printf("Message sent failed: %d\n", status.failedCount()); Serial.println("----------------\n"); smtp.sendingResult.clear();

}

}

**6.3 Arundio source code:**

const int pinA0 = A0; const int pinA1 = 8;

28 const int pin13 = 13; const int pin14 = 14;

void setup() { Serial.begin(115200); pinMode(pinA0, INPUT); pinMode(pinA1, INPUT); pinMode(pin13, OUTPUT); pinMode(pin14, OUTPUT);

}

void loop() {

// Read the values of A0 and A1 pins int valueA0 = digitalRead(pinA0); int valueA1 = digitalRead(pinA1);

// Check if either A0 or A1 has a value of 1 if (valueA0 == 1 || valueA1 == 1) {

// Set the values of pins 13 and 14 to HIGH digitalWrite(pin13, HIGH);

29 digitalWrite(pin14, HIGH)

// Trigger ESP32-CAM to run email sending code

Serial.println("RUN\_EMAIL\_SENDING");

// Additional actions can be performed here based on the condition

} else {

// Reset pins 13 and 14 to LOW if both A0 and A1 are 0 digitalWrite(pin13, LOW); digitalWrite(pin14, LOW);

}

// Add other functionalities as needed

delay(1000); // Adjust delay as needed

}

**6.4 Camera module code:**

#include "esp\_camera.h"

#include "SPI.h"

#include "driver/rtc\_io.h"

30

#include <ESP\_Mail\_Client.h>

#include <FS.h>

#include <WiFi.h>

// Your network, email, and camera configuration here...

void setup() {

// Your setup code here...

Serial.begin(9600); // Start serial communication at 9600 baud rate

}

void loop() {

if (Serial.available()) { // If data is available to read String data = Serial.readString(); // Read the data if (data == "HIGH") {

capturePhotoSaveLittleFS(); sendPhoto();

}

}

}// REPLACE WITH YOUR NETWORK CREDENTIALS

31 const char\* ssid = "fruitu"; const char\* password = "12345678";

// To send Email using Gmail use port 465 (SSL) and SMTP Server smtp.gmail.com

// You need to create an email app password

#define emailSenderAccount "cameramodule95@gmail.com"

#define emailSenderPassword "\*\*\*\* \*\*\*\* \*\*\*\* \*\*\*\*"

#define smtpServer "smtp.gmail.com"

#define smtpServerPort 465

#define emailSubject "ESP32-CAM Photo Captured"

#define emailRecipient "pavantejveesam26@gmail.com"

#define CAMERA\_MODEL\_AI\_THINKER

#if defined(CAMERA\_MODEL\_AI\_THINKER)

#define PWDN\_GPIO\_NUM 32

#define RESET\_GPIO\_NUM -1

#define XCLK\_GPIO\_NUM 0

#define SIOD\_GPIO\_NUM 26

#define SIOC\_GPIO\_NUM 27

#define Y9\_GPIO\_NUM 35

32

#define Y8\_GPIO\_NUM 34

#define Y7\_GPIO\_NUM 39

#define Y6\_GPIO\_NUM 36

#define Y5\_GPIO\_NUM 21

#define Y4\_GPIO\_NUM 19

#define Y3\_GPIO\_NUM 18

#define Y2\_GPIO\_NUM 5

#define VSYNC\_GPIO\_NUM 25

#define HREF\_GPIO\_NUM 23

#define PCLK\_GPIO\_NUM 22

#else

#error "Camera model not selected"

#endif

/\* The SMTP Session object used for Email sending \*/

SMTPSession smtp;

/\* Callback function to get the Email sending status \*/ void smtpCallback(SMTP\_Status status);

33

// Photo File Name to save in LittleFS

#define FILE\_PHOTO "photo.jpg"

#define FILE\_PHOTO\_PATH "/photo.jpg"

void setup() {

WRITE\_PERI\_REG(RTC\_CNTL\_BROWN\_OUT\_REG, 0); //disable brownout detector

Serial.begin(115200);

Serial.println();

// Connect to Wi-Fi

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi...");

while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.print(".");

}

Serial.println();

// Print ESP32 Local IP Address

34

Serial.print("IP Address: http://");

Serial.println(WiFi.localIP());

// Init filesystem

ESP\_MAIL\_DEFAULT\_FLASH\_FS.begin();

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;

35

config.pin\_sccb\_sda = SIOD\_GPIO\_NUM; config.pin\_sccb\_scl = SIOC\_GPIO\_NUM; config.pin\_pwdn = PWDN\_GPIO\_NUM; config.pin\_reset = RESET\_GPIO\_NUM; config.xclk\_freq\_hz = 20000000; config.pixel\_format = PIXFORMAT\_JPEG; config.grab\_mode = CAMERA\_GRAB\_LATEST;

if(psramFound()){

config.frame\_size = FRAMESIZE\_UXGA; config.jpeg\_quality = 10; config.fb\_count = 1;

} else {

config.frame\_size = FRAMESIZE\_SVGA; config.jpeg\_quality = 12; config.fb\_count = 1;

}

// Initialize camera

esp\_err\_t err = esp\_camera\_init(&config);

36 if (err != ESP\_OK) {

Serial.printf("Camera init failed with error 0x%x", err); return;

}

capturePhotoSaveLittleFS(); sendPhoto();

}

void loop() {

}

int valueA0 = digitalRead(A0);

int valueA1 = digitalRead(A1);

if (valueA0 == 1 || valueA1 == 1) { // Capture Photo and Save it to LittleFS void capturePhotoSaveLittleFS( void ) { // Dispose first pictures because of bad quality camera\_fb\_t\* fb = NULL;

// Skip first 3 frames (increase/decrease number as needed).

37 for (int i = 0; i < 3; i++) { fb = esp\_camera\_fb\_get(); esp\_camera\_fb\_return(fb); fb = NULL;

}

// Take a new photo fb = NULL;

fb = esp\_camera\_fb\_get();

if(!fb) {

Serial.println("Camera capture failed"); delay(1000);

ESP.restart();

}

// Photo file name

Serial.printf("Picture file name: %s\n", FILE\_PHOTO\_PATH);

File file = LittleFS.open(FILE\_PHOTO\_PATH, FILE\_WRITE);

// Insert the data in the photo file

38 if (!file) {

Serial.println("Failed to open file in writing mode");

}

else {

file.write(fb->buf, fb->len); // payload (image), payload length

Serial.print("The picture has been saved in ");

Serial.print(FILE\_PHOTO\_PATH);

Serial.print(" - Size: ");

Serial.print(fb->len);

Serial.println(" bytes");

}

// Close the file file.close(); esp\_camera\_fb\_return(fb);

}

void sendPhoto( void ) {

/\*\* Enable the debug via Serial port

* none debug or 0

39

* basic debug or 1

\*/

smtp.debug(1);

/\* Set the callback function to get the sending results \*/ smtp.callback(smtpCallback);

/\* Declare the session config data \*/

Session\_Config config;

/\*Set the NTP config time

For times east of the Prime Meridian use 0-12

For times west of the Prime Meridian add 12 to the offset.

Ex. American/Denver GMT would be -6. 6 + 12 = 18

See https://en.wikipedia.org/wiki/Time\_zone for a list of the GMT/UTC timezone offsets

\*/

config.time.ntp\_server = F("pool.ntp.org,time.nist.gov"); config.time.gmt\_offset = 0; config.time.day\_light\_offset = 1;

40

/\* Set the session config \*/ config.server.host\_name = smtpServer; config.server.port = smtpServerPort; config.login.email = emailSenderAccount; config.login.password = emailSenderPassword; config.login.user\_domain = "";

/\* Declare the message class \*/

SMTP\_Message message;

/\* Enable the chunked data transfer with pipelining for large message if server supported \*/

message.enable.chunking = true;

/\* Set the message headers \*/ message.sender.name = "ESP32-CAM"; message.sender.email = emailSenderAccount;

message.subject = emailSubject;

message.addRecipient("Sara", emailRecipient);

41

String htmlMsg = "<h2>Photo captured with ESP32-CAM and attached in this email.</h2>";

message.html.content = htmlMsg.c\_str(); message.html.charSet = "utf-8"; message.html.transfer\_encoding = Content\_Transfer\_Encoding::enc\_qp;

message.priority = esp\_mail\_smtp\_priority::esp\_mail\_smtp\_priority\_normal; message.response.notify = esp\_mail\_smtp\_notify\_success | esp\_mail\_smtp\_notify\_failure | esp\_mail\_smtp\_notify\_delay; /\* The attachment data item \*/

SMTP\_Attachment att;

/\*\* Set the attachment info e.g.

* file name, MIME type, file path, file storage type,
* transfer encoding and content encoding

\*/

att.descr.filename = FILE\_PHOTO; att.descr.mime = "image/png";

42 att.file.path = FILE\_PHOTO\_PATH;

att.file.storage\_type = esp\_mail\_file\_storage\_type\_flash; att.descr.transfer\_encoding =

Content\_Transfer\_Encoding::enc\_base64;

/\* Add attachment to the message \*/ message.addAttachment(att);

/\* Connect to server with the session config \*/ if (!smtp.connect(&config))

return;

/\* Start sending the Email and close the session \*/ if (!MailClient.sendMail(&smtp, &message, true))

Serial.println("Error sending Email, " + smtp.errorReason());

}

// Callback function to get the Email sending status void smtpCallback(SMTP\_Status status){

/\* Print the current status \*/

Serial.println(status.info());

43

/\* Print the sending result \*/ if (status.success())

{

Serial.println("----------------");

Serial.printf("Message sent success: %d\n", status.completedCount());

Serial.printf("Message sent failled: %d\n", status.failedCount()); Serial.println("----------------\n"); struct tm dt;

for (size\_t i = 0; i < smtp.sendingResult.size(); i++){

/\* Get the result item \*/

SMTP\_Result result = smtp.sendingResult.getItem(i); time\_t ts = (time\_t)result.timestamp; localtime\_r(&ts, &dt);

ESP\_MAIL\_PRINTF("Message No: %d\n", i + 1);

ESP\_MAIL\_PRINTF("Status: %s\n", result.completed ? "success" : "failed");

44

ESP\_MAIL\_PRINTF("Date/Time: %d/%d/%d %d:%d:%d\n", dt.tm\_year +

1900, dt.tm\_mon + 1, dt.tm\_mday, dt.tm\_hour, dt.tm\_min, dt.tm\_sec);

ESP\_MAIL\_PRINTF("Recipient: %s\n", result.recipients.c\_str());

ESP\_MAIL\_PRINTF("Subject: %s\n", result.subject.c\_str());

}

Serial.println("----------------\n");

// You need to clear sending result as the memory usage will grow up.

smtp.sendingResult.clear();

}

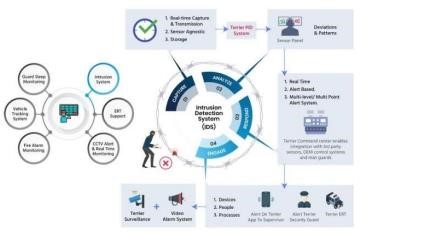
}}

// Your functions for capturing photo, saving to LittleFS, and sending email here...

## 45 7.DESIGN

**7.1 BLOCK DIAGRAM**

The proposed system is used whenever the robbery occurs, Vibration sensor is used here which senses vibration produced from ATM machine. This system uses Arduino controller based embedded system to process real time data collected using the vibration sensor. Once the vibration is sensed, information is passed to channel where alarm sound starts from the buzzer. DC Motor is used for closing the door of ATM. Camera is always in processing and sending video continuous to the PC and it will be saved in computer. RTC used to capture the robber occur time and send the robbery occur time with the message to the nearby police station and corresponding bank through the GSM. This will prevent the robbery and the person involving in robbery can be easily carried.



46

The main controlling device of the project is Raspberry pi3 processor. The input devices of the system are:

* Pi camera: when the code is executed, picture capturing is started with the help of pi camera. The output devices of the proposed system are
* Dc motor: After the person satisfies the required criteria, the motor driver activates the dc motor, which is considered as the automatic door in the prototype.
* Buzzer: Buzzer is another output device of the system. It activates when the person is not satisfying both or any one of the required criteria.
* LCD module: This is used to display the status of the monitoring. There are 4 types of cases given in the code and the LCD is used to display the result of the respective cases. At one time of execution, it displays two statuses (one for mask and other for temperature)
* Vibrating sensor: It is an output device of the system. It predicts the vibrations when the vibrations occurs.

47

## 8. RESULT AND OUTPUT

The effective functioning of an Advance ATM anti-theft using raspberry pi is one of the outcome. This technology may identify the robbery and notify the nearest police station. All the components were connected according to the block diagram. The hardware connections and project prototype are shown below.

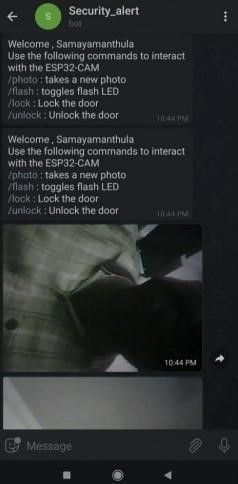
**Step 1**: Project prototype



**Step 2:** When robbery occurs the dc motor close the door and GSM sends the message to the nearby police station and to the bank and buzzer will make sound for theft indication and camera will captures images.



**Step 3:** Sending message



48

49

## 9.CONCLUSION AND FUTURESCOPE

**9.1 Conclusion:**

In conclusion, the integration of advanced security measures and IoT technologies in ATMs significantly enhances the prevention and response to ATM robberies. By employing sensors that detect unauthorized attempts to tamper with the ATM, the system can trigger immediate alerts through buzzers and lock doors to deter thieves. Continuous video surveillance ensures that the entire incident is recorded and transmitted to a central computer, providing crucial evidence. Furthermore, the activation of GPS modules upon detecting a breach allows for real-time location tracking, aiding law enforcement in swift response and recovery of stolen assets. Automated alert systems ensure that both the police and corresponding bank are notified instantly, enhancing the coordination and effectiveness of security efforts. These comprehensive measures not only deter potential robbers but also ensure a rapid and coordinated response in the event of an attempted theft, significantly improving the security and integrity of ATM operations.