Visvesvaraya National Institute of Technology (VNIT), Nagpur

Embedded System (ECP 403)

Endsem Submission Report

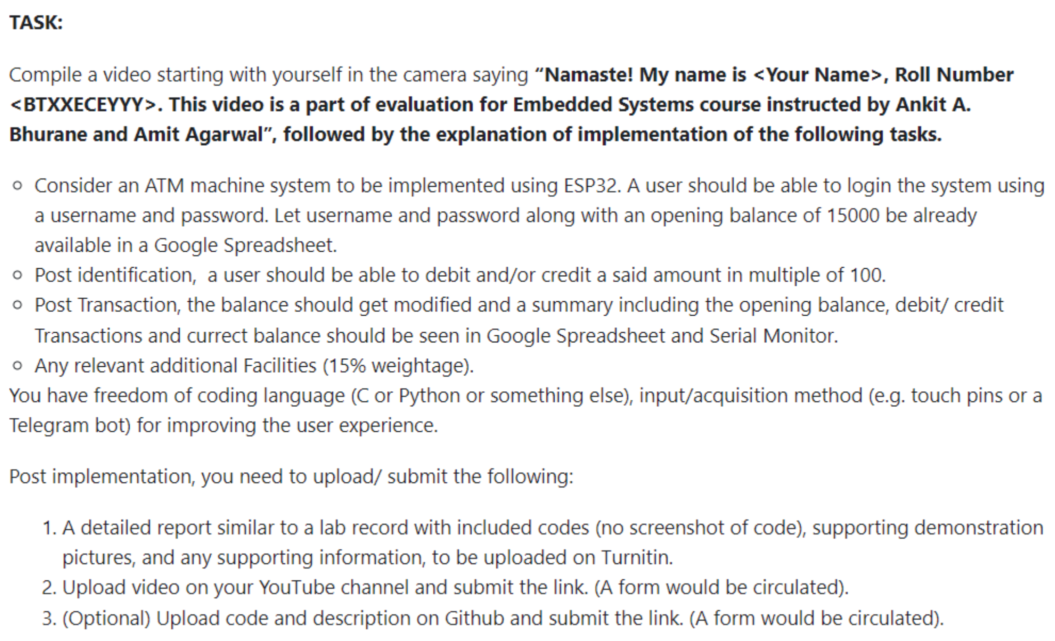
Submitted by :
Nikhil Vanjari (BT20ECE109)
Semester 5

Submitted to :
Dr. Ankit A. Bhurane
(Course Constructor)
Department of Electronics and Communication Engineering,
VNIT Nagpur

Contents

Problem statement: To make atm-like operations using ESP32

Aim: To make an esp based atm system for atm-like operations. The question is as follows:



TASK:

Compile a video starting with yourself in the camera saying **"Namaste! My name is <Your Name>, Roll Number <BTXXECEYYY>. This video is a part of evaluation for Embedded Systems course instructed by Ankit A. Bhurane and Amit Agarwal", followed by the explanation of implementation of the following tasks.**

- Consider an ATM machine system to be implemented using ESP32. A user should be able to login the system using a username and password. Let username and password along with an opening balance of 15000 be already available in a Google Spreadsheet.
- Post identification, a user should be able to debit and/or credit a said amount in multiple of 100.
- Post Transaction, the balance should get modified and a summary including the opening balance, debit/ credit Transactions and currect balance should be seen in Google Spreadsheet and Serial Monitor.
- Any relevant additional Facilities (15% weightage).

You have freedom of coding language (C or Python or something else), input/acquisition method (e.g. touch pins or a Telegram bot) for improving the user experience.

Post implementation, you need to upload/ submit the following:

1. A detailed report similar to a lab record with included codes (no screenshot of code), supporting demonstration pictures, and any supporting information, to be uploaded on Turnitin.
2. Upload video on your YouTube channel and submit the link. (A form would be circulated).
3. (Optional) Upload code and description on Github and submit the link. (A form would be circulated).

Figure 1:

Appartus: Arduino software, telegram, ESP32.

Theory:

We use the GPIO pins to provide the touch inputs to the ESP-32 The Pinout of the same is as follows:

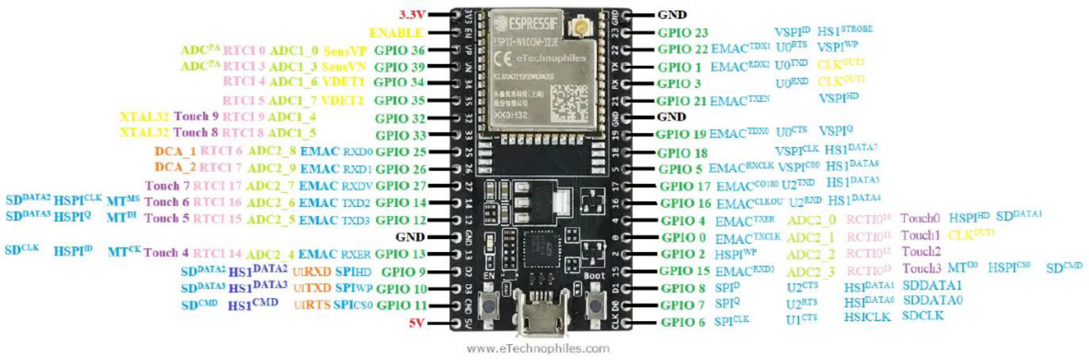

3.3V
ENABLE
ADCPA RTCI 0 ADC1_0 SensVP GPIO 36
ADCPA RTCI 3 ADC1_3 SensVN GPIO 39
RTCI 4 ADC1_6 VDET1 GPIO 34
RTCI 5 ADC1_7 VDET2 GPIO 35
XTAL32 Touch 9 RTCI 9 ADC1_4 GPIO 32
XTAL32 Touch 8 RTCI 8 ADC1_5 GPIO 33
DCA_1 RTCI 6 ADC2_8 EMAC RXD0 GPIO 25
DCA_2 RTCI 7 ADC2_9 EMAC RXD1 GPIO 26
Touch 7 RTCI 17 ADC2_7 EMAC RXDV GPIO 27
SDDATA2 HSPICLK MTMS Touch 6 RTCI 16 ADC2_6 EMAC TXD2 GPIO 14
SDDATA3 HSPIQ MTDI Touch 5 RTCI 15 ADC2_5 EMAC TXD3 GPIO 12
GND
SDCLK HSPIID MTCK Touch 4 RTCI 14 ADC2_4 EMAC RXER GPIO 13
SDDATA2 HS1DATA2 U1RXD SPIHD GPIO 9
SDDATA3 HS1DATA3 U1TXD SPIWP GPIO 10
SDCMD HS1CMD U1RTS SPICS0 GPIO 11
5V
GND
GPIO 23 VSPIID HS1STROBE
GPIO 22 EMACTDX1 U0RTS VSPIWP
GPIO 1 EMACRDX2 U0TXD CLKOUT3
GPIO 3 U0RXD CLKOUT2
GPIO 21 EMACTXEN VSPIHD
GND
GPIO 19 EMACTDX0 U0CTS VSPIQ
GPIO 18 VSPICLK HS1DATA7
GPIO 5 EMACRXCLK VSPICS0 HS1DATA6
GPIO 17 EMACCO180 U2TXD HS1DATA5
GPIO 16 EMACCLKOUT U2RXD HS1DATA4
GPIO 4 EMACTXER ADC2_0 RCTIO10 Touch0 HSPIHD SDDATA1
GPIO 0 EMACTXCLK ADC2_1 RCTIO11 Touch1 CLKOUT1
GPIO 2 HSPIWP ADC2_2 RCTIO12 Touch2
GPIO 15 EMACRXD3 ADC2_3 RCTIO13 Touch3 MTDO HSPICS0 SDCMD
GPIO 8 SPID U2CTS HS1DATA1 SDDATA1
GPIO 7 SPIQ U2RTS HS1DATA0 SDDATA0
GPIO 6 SPICLK U1CTS HS1CLK SDCLK
ESPRESSIF
eTechnophiles
Boot
www.eTechnophiles.com

Figure 2:

We use Telegram Bot generator, i.e., the BotFather and generate the bots. We also get the Telegram Chat ID from IDBot so that we are the sole users of the telegram bot. The well commented code with explanation of each and every line is as follows:

Simulation Code in Assembly Language::

```
//diployment id= ...
    AKfycbwWj_VRevNcSE6cxagBXivfz9BqTE29rTC2fNrH1Q2yjs5DJl−07nkBkpNxqR4EIrcn
//diployment web_app = ...
    https://script.google.com/macros/s/AKfycbwWj_VRevNcSE6cxagBXivfz9BqTE
//29rTC2fNrH1Q2yjs5DJl−07nkBkpNxqR4EIrcn/exec

#include<stdio.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include "HTTPClient.h"
#include <EEPROM.h>
#define EEPROM_SIZE 1

// Wifi network station credentials
// esp 32 getting access via personal wifi

#define WIFI_SSID "Nick1jari"
#define WIFI_PASSWORD "nikhil22"
/************************* Telegram BOT Token (Get from ...
    Botfather)****************************/
#define BOT_TOKEN "5931724500:AAHPv9HLU2AZWHpRlsGGBoKhdLMIp32_43A"
```

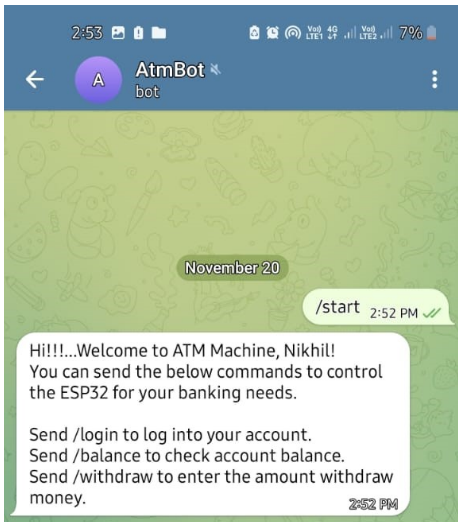
```

//connecting to google speadsheet
String GOOGLE_SCRIPT_ID = ...
    "AKfycbzF1FjO7xMMnQGfD1KJV4MCmY5q95jk51sYeTn65_2XDy7A_6I_r4q-rfJv6ZSDrUir";

// giving authority to my own chat of telegram
#define CHAT_ID "959072868"

const unsigned long BOT_MTBS = 1000; // mean time between scan ...
    messages

WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);
unsigned long bot_lasttime;          // last time messages' scan ...
    has been done
bool Start = false;



/*************************** google spreadsheet ...
    *********************/

String username; //defining username
String password; //defining password
String amount; //defining amount
void get_username(void) {
   HTTPClient http;
   String ...
       url="https://script.google.com/macros/s/"+GOOGLE_SCRIPT_ID+"/exec?read=A";//
       script for reading column no A
//   Serial.print(url);
    Serial.print("Making a request");
    http.begin(url.c_str()); //Specify the URL and certificate
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
    int httpCode = http.GET();

    if (httpCode > 0) { //Check for the returning code
        username = http.getString(); //function for getting the ...
            string
      }
    else {
      Serial.println("Error on HTTP request");//gives out error ...
          if there is error in fetching the password
    }
      http.end();
}

String get_password(void) {
   HTTPClient http;
   String ...
       url="https://script.google.com/macros/s/"+GOOGLE_SCRIPT_ID+"/exec?read=B";//c
       scrip for reading column no B
//   Serial.print(url);
    Serial.print("Making a request");
    http.begin(url.c_str()); //Specify the URL and certificate
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
    int httpCode = http.GET();

    if (httpCode > 0) { //Check for the returning code
        return http.getString(); //function for getting the string
      }
    else {
      Serial.println("Error on HTTP request password fetch");
      return "Error";//gives out error if there is error in ...
          fetching the password
    }
    http.end();
}


void get_amount(void) {
  HTTPClient http;
  String ...
      url="https://script.google.com/macros/s/"+GOOGLE_SCRIPT_ID+"/exec?read=478888
  //google script for reading column no C
  //Serial.print(url);
    Serial.print("Making a request");
    http.begin(url.c_str()); //Specify the URL and certificate
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
    int httpCode = http.GET();

    if (httpCode > 0) { //Check for the returning code
        amount = http.getString(); //function for getting the string

      }
    else {
      Serial.println("Error on HTTP request");//gives out error ...
          if there is error in fetching the password
    }

    http.end();
}

/******************************* Amount Enter, Withdrawal, ...
    ************************************/

// now, we perform the same input conditions for the Amount to ...
    be Wirhdrawn
```

```
int amountEnter()
{
  // creating the array of pin values
  int pVals[8]={4,15,13,12,14,27,33,32};
  // mapping the pin values to the required digits
  int dVals[8]={2,3,4,5,6,7,8,9};

  // initialising counter variables
  int i,j;
  j=0;
  i=0;
  // initialising the amount received variable
  int amt_Recd = 0;
  // providing sufficient delay to enter the values
  delay(15000);
  // looping
  for(j=0;j<5;j++)
  {
    for(i=0;i<8;i++)
    {
      // Since the GPIO pins corresponding to Touch 0 and Touch ...
          1 are highly sensitive,
      // they act erratic and hence disrupt the functioning of ...
          the code.
        // So, we combine two pins and read them as 1 and 0, ...
            code as follows:
      // To read 1 as input
      if(touchRead(13)<80 && touchRead(15)<80)
      {
        // printing the input received in serial monitor
        Serial.print("\nInput Received:");
        Serial.print('1');
        // sufficient delay for next value
        delay(15000);
        // storing the value
        amt_Recd=(amt_Recd*10) + 1;
        // to come out of the loop
        break;
      }


      if(touchRead(15)<80 && touchRead(4)<80)
      {
        // printing the input received in serial monitor
        Serial.print("\nInput Received:");
        Serial.print('0');
        // sufficient delay for next value
        delay(15000);
        // storing the value
```

```
          amt_Recd=(amt_Recd*10) + 0;
          // to come out of the loop
          break;
        }


        if(touchRead(pVals[i])<80)
        {
          // printing the input received in serial monitor
          Serial.print("\nInput Received:");
          Serial.print(dVals[i]);
          // sufficient delay for next value
          delay(15000);
          // storing the value
          amt_Recd=(amt_Recd*10) + dVals[i];
          // to come out of the loop
          break;
        }
      }

    }

    // printing the amount to be withdrawn in the Serial Monitor
    Serial.print("\nSo, the amount to be withdrawn is ");
    Serial.print(amt_Recd);
    Serial.print("\n");

    // returning the amount
    return amt_Recd;
  }



  // to count the number of notes of 2000
  int withdrawAmount1(int withdrawAMT, String chat_id)
  {
    // initialising the variables
    int remAMT=withdrawAMT;
    int count_two_th = 0;
    int count_one_th = 0;
    int count_five_h = 0;
    int remain_two_th = 0;
    int remain_one_th = 0;
    int remain_five_h = 0;
    // making sure that the amount to be withdrawin is valid.
      // the amount should be:
    // 1) Positive
    // 2) Less than 15,000
    // 3) Able to be represented in denomination of 2000, 1000 and 500
    if((withdrawAMT ≤ 15000) && (withdrawAMT≥0) && ...
        (withdrawAMT%500==0))
    {
      // counting the number of notes of 2000
      count_two_th=remAMT/2000;
      // finding the remaining amount to be disbursed
      remAMT%=2000;
      // since we have only 5 notes of 2000, we make sure that we ...
          aren't using more than that
      // we spill over the excess amount to the 1000 and 500 notes
      if(count_two_th>5)
      {
        // we return the excess amount value to the variable
        remAMT+=(count_two_th-5)*2000;
        // we reset the number of notes of 2000 to max, i.e. five
        count_two_th=5;
      }
    }

    // we return the count of number of notes of 2000
    return count_two_th;
  }


  int withdrawAmount2(int withdrawAMT, String chat_id)
  {
    // initialising the variables
    int remAMT=withdrawAMT;

    int count_two_th = 0;
    int count_one_th = 0;
    int count_five_h = 0;
    int remain_two_th = 0;
    int remain_one_th = 0;
    int remain_five_h = 0;
    // making sure that the amount to be withdrawin is valid.
    // the amount should be:
    // 1) Positive
    // 2) Less than 15,000
    // 3) Able to be represented in denomination of 2000, 1000 and 500
    if((withdrawAMT≤15000) && (withdrawAMT≥0) && (withdrawAMT%500==0))
    {
      // counting the number of notes of 2000
      count_two_th=remAMT/2000;
      // finding the remaining amount to be disbursed
      remAMT%=2000;
      // since we have only 5 notes of 2000, we make sure that we ...
          aren't using more than that
      // we spill over the excess amount to the 1000 and 500 notes
      if(count_two_th>5)
      {
        // we return the excess amount value to the variable
        remAMT+=(count_two_th-5)*2000;
        // we reset the number of notes of 2000 to max, i.e. five
        count_two_th=5;
      }
      // counting the number of notes of 1000
      count_one_th=remAMT/1000;
      // finding the remaining amount to be disbursed
      remAMT%=1000;
      // since we have only 10 notes of 1000, we make sure that we ...
         aren't using more than that
      // we spill over the excess amount to the 500 notes
      if(count_one_th>10)
      {
        // we return the excess amount value to the variable
        remAMT+=(count_one_th-10)*1000;
        // we reset the number of notes of 1000 to max, i.e. ten
        count_one_th=10;
      }
    }
    // returning the count of number of notes of 1000
    return count_one_th;
  }


  int withdrawAmount3(int withdrawAMT, String chat_id)
  {
    // initialising the variables
    int remAMT=withdrawAMT;
    int count_two_th = 0;
    int count_one_th = 0;
    int count_five_h = 0;
    int remain_two_th = 0;
    int remain_one_th = 0;
    int remain_five_h = 0;
    // making sure that the amount to be withdrawin is valid.
    // the amount should be:
      // 1) Positive
    // 2) Less than 15,000
    // 3) Able to be represented in denomination of 2000, 1000 and 500
    if((withdrawAMT≤15000) && (withdrawAMT≥0) && (withdrawAMT%500==0))
    {
      // counting the number of notes of 2000
      count_two_th=remAMT/2000;
      // finding the remaining amount to be disbursed
      remAMT%=2000;
      // since we have only 5 notes of 2000, we make sure that we ...
          aren't using more than that
      // we spill over the excess amount to the 1000 and 500 notes
      if(count_two_th>5)
      {
        // we return the excess amount value to the variable
        remAMT+=(count_two_th-5)*2000;
        // we reset the number of notes of 2000 to max, i.e. five
        count_two_th=5;
      }
      // counting the number of notes of 1000
      count_one_th=remAMT/1000;
      // finding the remaining amount to be disbursed
      remAMT%=1000;
      // since we have only 10 notes of 1000, we make sure that we ...
          aren't using more than that
      // we spill over the excess amount to the 500 notes
      if(count_one_th>10)
      {
         // we return the excess amount value to the variable
        remAMT+=(count_one_th-10)*1000;
         // we reset the number of notes of 1000 to max, i.e. ten
        count_one_th=10;
      }
      // counting the number of notes of 500
      count_five_h=remAMT/500;
    }
    // returning the number of notes of 500
    return count_five_h;
  }



  /************************* code for telegram chatbot ...
      ***************************************/


  void handleNewMessages(int numNewMessages)
  {
    Serial.println("handleNewMessages");
    Serial.println(String(numNewMessages));

    for (int i=0; i<numNewMessages; i++)
    {
    //checking whether that it's user who is giving the commands
      String chat_id = String(bot.messages[i].chat_id);
      // code for checking chat ID
      if (chat_id != CHAT_ID)
      {
        bot.sendMessage(chat_id, "You are not the owner of credit ...
            card", "");
        // repeating the loop if user isnot the one who is ...
            accessing telegram bot
        continue;
      }

      String user_text = bot.messages[i].text;
      Serial.println(user_text);


      // extracting user's telegram Name from his/her Chat ID
      String from_name = bot.messages[i].from_name;

      if (user_text == "/start")
      {
        //Welcome message when user gives input as a /start in the ...
            telegram bot
        String welcome = "Hi!!!...Welcome to ATM Machine, " + ...
            from_name + "!\n";
        welcome += "You can send the below commands to control the ...
            ESP32 for your banking needs.\n\n";
        welcome += "Send /login to log into your account. \n";
        welcome += "Send /balance to check account balance. \n";
        welcome += "Send /withdraw to enter the amount withdraw ...
            money. \n";
        bot.sendMessage(chat_id, welcome,"");
      }



     //coding the login funciton for the chatbot
    if (user_text == "/login") {
      bot.sendMessage(chat_id,"Enter username : ","");
      delay(8000);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
      String username_tele= bot.messages[i].text;

      if(username_tele !=username)
      {
        String wrong="Sorry!...You have entered wrong ...
            username...Try again.";//printing out entered wrong ...
            statement
        // when username of spreadsheet doesnt matches with the ...
            username of telegram bot
        bot.sendMessage(chat_id,wrong,"");
        continue;
      }
      password=get_password();  //getting password from google ...
          spreadsheet using app script
      bot.sendMessage(chat_id,"Enter password : ","");
      // delay(8000);
      // numNewMessages = bot.getUpdates(bot.last_message_received ...
          + 1);
      // String password_tele=bot.messages[i].text;
      String password_tele=getInpFromUser();
      if(password_tele !=password)
      {
        String wrong_password ="Sorry!...You have entered wrong ...
            password...Try again.";//printing out entered wrong ...
            statement
        // when username of spreadsheet doesnt matches with the ...
            username of telegram bot
        bot.sendMessage(chat_id, wrong_password,"");
        continue;
      }

      bot.sendMessage(chat_id,"Congratulations!!!...You are login ...
          successfully.","");

  }


      // coding the withdraw function for the chatbot
      if (user_text == "/withdraw")
      {
        String input_withdraw = "How much amount you want to ...
            withdraw.\n kindly enter it in a 5 digit form, padded ...
            with adequate number zeroes.";
        // asking the user to enter the amount to be withdrawn
        bot.sendMessage(chat_id,input_withdraw,"");
        // storing the amount to be withdrawn
        int withdrawAMT = amountEnter();
        // receiving the notes denomination entry using the functions
        int ...
            Denomination[3]={withdrawAmount1(withdrawAMT,chat_id),withdrawAmount2(wit
        // checking the validity of the withdrawn amount
        // if the amount is invalid, send this message
        if((withdrawAMT!=0) && ...
            (withdrawAmount1(withdrawAMT,chat_id)==0) ...
            &&(withdrawAmount2(withdrawAMT,chat_id)==0) && ...
            (withdrawAmount3(withdrawAMT,chat_id)==0))
        {
          bot.sendMessage(chat_id, "Please enter a Positive Amount ...
              Value, Less than 15,000, that is a Multiple of 500. ...
              Use /withdraw to try again.","");
        }
        // else, if the amount is valid, we deduct it from the ...
           balance and then indicate the denominations of the ...
           notes to be disbursed by the bank
        else
        {
          // sending the messages
          bot.sendMessage(chat_id, "Your requested amount of "+
          String(withdrawAMT) + " has been deducted from your ...
             account. \nPlease check the tended cash to be in the ...
             given denominations. \n"+
          String(Denomination[0])+ " note(s) of 2000, \n" ...
             +String(Denomination[1]) +"notes() of 1000, and\n" + ...
             String(Denomination[2]) +" note(s) of 500." , "");
          bot.sendMessage(chat_id, "Your transaction has been ...
             completed. Have a nice day!","");
          // updating the balance on the backend using the EEPROM ...
             present on the E
          // printing the balance in the serial monitorSP-32
          int originalBalanceAmount = 15000;//EEPROM.read(0)*100;
          // deducting the withdrawn amount
          int currentBalanceAmount = originalBalanceAmount - ...
             withdrawAMT;
          Serial.print(currentBalanceAmount);
          // updating the balance in the EEPROM
          //EEPROM.write(0, currentBalanceAmount/100);
        }
      }


      // writing the code for displaying the balance in the chat ...
         when user want to see available balance
      if (user_text == "/balance")
      {
        // setting the value of balance amount 15000
        int balanceAmount = 15000;  //EEPROM.read(0)*100;
        // sending the message containing the amount balance in ...
           the chat
        bot.sendMessage(chat_id, "Your current balance is " + ...
           String(balanceAmount),"");
      }
    }
  }

  /*********************************************** code for setup ...
      *********************************************************/

  //writing the set up code
  void setup()
  {
    // providing input of the pins for mapping to the required values
    pinMode(4, INPUT);
    pinMode(15,INPUT);
    pinMode(13,INPUT);
    pinMode(12,INPUT);
    pinMode(14,INPUT);
    pinMode(27,INPUT);
    pinMode(33,INPUT);
    pinMode(32,INPUT);
  //Setting band width equals to 115200
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    Serial.println();

    // attempt to connect to Wifi network:
    Serial.print("Connecting to Wifi SSID ");
    Serial.print(WIFI_SSID);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add ...
        root certificate for api.telegram.org
    while (WiFi.status() != WL_CONNECTED)
    {
      Serial.print(".");
      delay(500);//providing delay of 0.5 sec
    }
    Serial.print("\nWiFi connected. IP address: ");
    Serial.println(WiFi.localIP());
    get_username();//taking user name fuction


    Serial.print("Retrieving time: ");
    configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP
    time_t now = time(nullptr);
    while (now < 24 * 3600)
    {
      Serial.print(".");
      delay(100);
      now = time(nullptr);
    }
    Serial.println(now);
  }

  // fuction for getting input from user from the telegram bot
  String getInpFromUser(){
    int numNewMessages = bot.getUpdates(bot.last_message_received ...
        + 1);
    while(numNewMessages<1){
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    return bot.messages[0].text;
  }

  /******************************************** loop ...
      ******************************************************************/
  //starting the loop
  void loop()
  {
    if (millis() − bot_lasttime > BOT_MTBS)
    {
      int numNewMessages = ...
          bot.getUpdates(bot.last_message_received + 1);

      while (numNewMessages)
      {
        Serial.println("got response");
        handleNewMessages(numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received ...
            + 1);
      }

      bot_lasttime = millis();
    }

    //Serial.println(password);
  }
```

Output:

We provide input to the bot, by using the /start function



AtmBot
bot

November 20

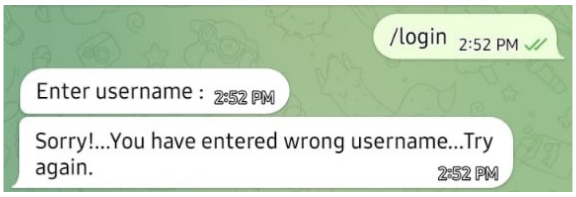
/start 2:52 PM

Hi!!!...Welcome to ATM Machine, Nikhil!
You can send the below commands to control the ESP32 for your banking needs.

Send /login to log into your account.
Send /balance to check account balance.
Send /withdraw to enter the amount withdraw money. 2:52 PM

Figure 3:

for login to the using incorrect username and password



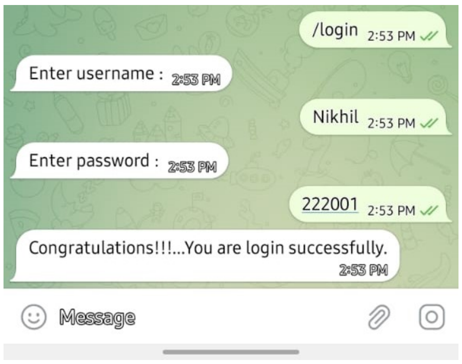
/login 2:52 PM

Enter username : 2:52 PM

Sorry!...You have entered wrong username...Try again. 2:52 PM

Figure 4:

for login to the using correct username and password



/login 2:53 PM

Enter username : 2:53 PM

Nikhil 2:53 PM

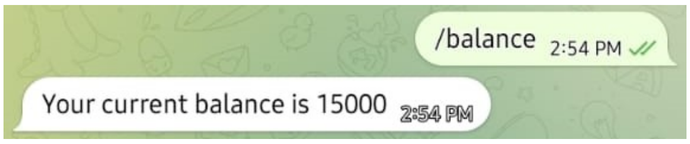
Enter password : 2:53 PM

222001 2:53 PM

Congratulations!!!...You are login successfully. 2:53 PM

Figure 5:

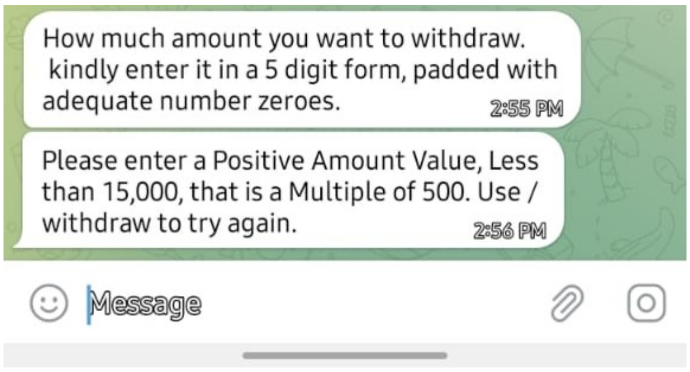
for checking balance



```
/balance 2:54 PM
Your current balance is 15000 2:54 PM
```

Figure 6:

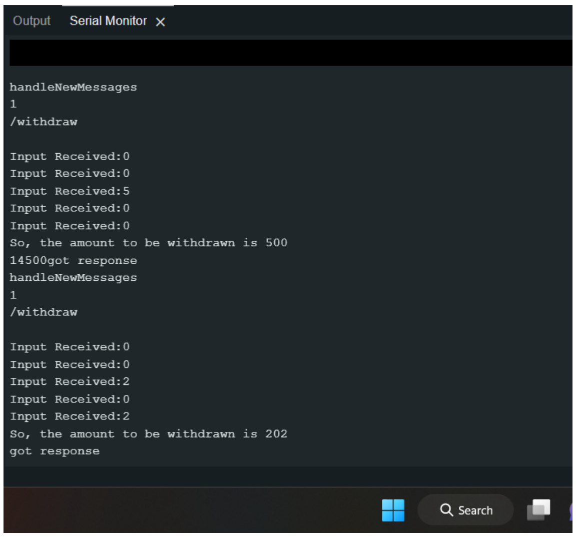
after given no which is not multiple of 100



```
How much amount you want to withdraw.
 kindly enter it in a 5 digit form, padded with
adequate number zeroes. 2:55 PM
Please enter a Positive Amount Value, Less
than 15,000, that is a Multiple of 500. Use /
withdraw to try again. 2:56 PM
```

Figure 7:

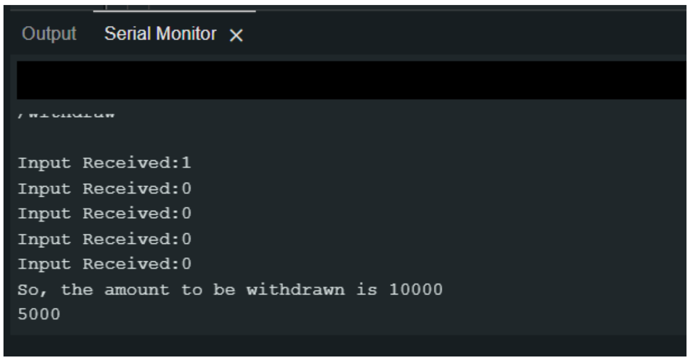
output on serial moniter



```
handleNewMessages
1
/withdraw

Input Received:0
Input Received:0
Input Received:5
Input Received:0
Input Received:0
So, the amount to be withdrawn is 500
14500got response
handleNewMessages
1
/withdraw

Input Received:0
Input Received:0
Input Received:2
Input Received:0
Input Received:2
So, the amount to be withdrawn is 202
got response
```

Figure 8:

for withdrawing 10000 rupees



```
/withdraw

Input Received:1
Input Received:0
Input Received:0
Input Received:0
Input Received:0
So, the amount to be withdrawn is 10000
5000
```

Figure 9:

<u>Conclusion</u>: Therefore, we researched Touch Input OTP Generation and Validation of ESP-32. We interface the balance, withdrawal, and login via Telegram.