VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY (VNIT), NAGPUR

Embedded System (ECL403)

EndSem Solution Report

Submitted by :
Rutwij Kamble (BT19ECE052)
Semester 5

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

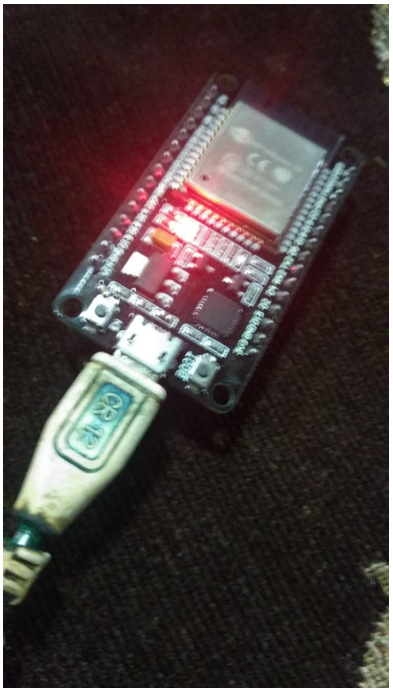
Contents

Question-1: ESP32 based ATMlike System for digital transactions.

Problem Statement:: In the given task ,We have to design and come up with an ATM-like transaction system based on ESp32 hardware.The following parameters are considered in it

- We have to design and come up with a telegram bot system for the authentication and verification process.Once connected ,this Telegram bot sends up an OTP and using commands on this bot we complete the digital transaction
- we use thingspeak platform to monitor the denominations and balance remaining in the atm. if the atm runs empty, it triggers up an alert on thingspeak IOT platform

Requirements:: Espressif's ESP32 arduino IDE Telegram Thingspeak Internet connection

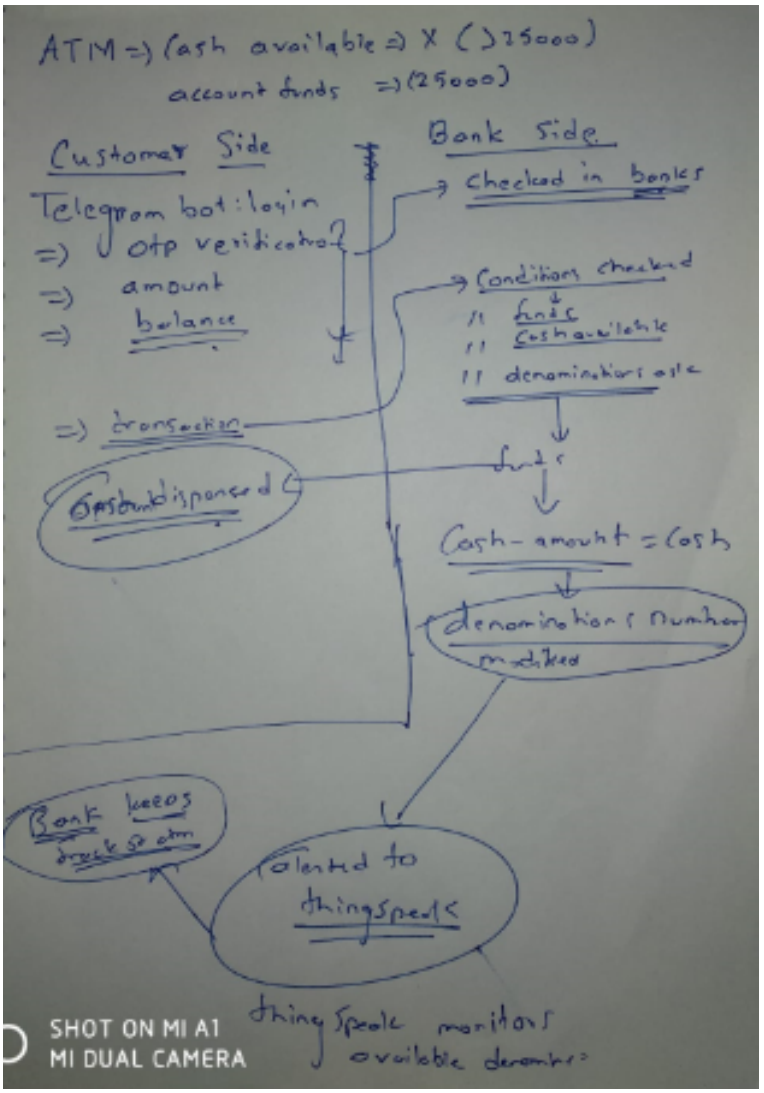


Solution:: We start up developing a process architecture for project.we include what processes we have to do and arrange them in sequence in which the project is going to reiterate. In the above project we combinely use both telegram bot and thingspeak platform.while both work on different interfaces. telegram bot works

for the customer side while the thingspeak platform is used to indicate how much denominations are left inside the atm basically from the bank side. In any realtime atm, these denominations are present in form of cassettes .so we have different cassettes for different denominations. the first atm was installed by John Shepherd-Barron for State Bank of India in 1987. although real atms use rfid chips in credit and debit cards for authentication and pin for 2 factor verification,Nowadays there are atms which transact digitally using only otp and online app. One such example is YONO SBI 's ATM where it uses it's App for dispensing amount while otp is sent to the mobile number verified to the bank account.

We try to simulate and design such a Similar system using ESP32 and its feature. Now,there are many ways to modify and increase efficiency of this project. Our project Follows Below Processes,

Architecture:


ATM => Cash available => X ( )25000)
account funds =>(25000)
Customer Side
Telegram bot: login
=> otp verification
=> amount
=> balance
=> transaction
Bank Side
Checked in banks
Conditions checked
funds
cash available
denominations are
funds
Cash disbursed
Cash - amount = Cash
denomination & number
modified
Bank keeps
track of atm
related to
thingspeak
thingspeak monitors
available denomin
SHOT ON MI A1
MI DUAL CAMERA

Clearly ,The ESP32 or in this case ATM checks conditions with the bank and then processess the transactoins.both these transactions and atm dispensing is monitored by bank(Us).

Project Source Code:

```
#ifdef ESP32
  #include <WiFi.h>
#else
  #include <ESP32WiFi.h>
#endif
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <HTTPClient.h>
const char* ssid = "";
const char* password = "";
const char* serverName = ...
    "https://api.thingspeak.com/update?api_key=64A121SOW112X84M&field1=0";
String apiKey = "64A121SOW112X84M";
#define BOTtoken "2147382245:AAG_BjSMEsrBPZeq1cVdqJYh6ddG859h4V0"
#define CHAT_ID ""
#ifdef ESP8266
  X509List cert(TELEGRAM_CERTIFICATE_ROOT);
#endif
int v,r,n;
int a ,b ,c ;
int cashremaining;
int cashdispensed;
int x = 5,y = 10,z = 10;
//x is 2000
//y is 1000
//z is 500
int atm_val= 2000*x+1000*y+500*z ;
int bal = 22000;
int amount ;
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);


int botRequestDelay = 1000;
unsigned long lastTimeBotRan;



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

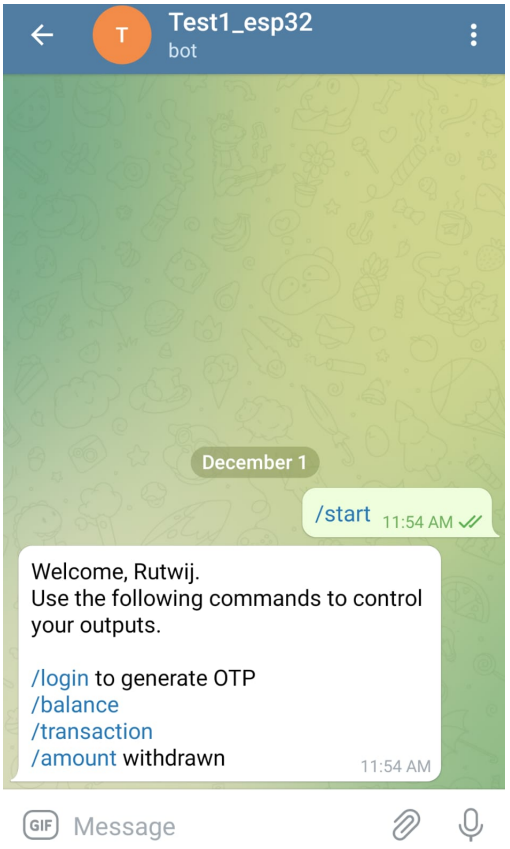
```
42
43    for (int i=0; i<numNewMessages; i++) {
44      // Chat id of the requester
45      String chat_id = String(bot.messages[i].chat_id);
46      if (chat_id != CHAT_ID){
47        bot.sendMessage(chat_id, "Unauthorized user", "");
48        continue;
49      }
50
51      // Print the received message
52      String text = bot.messages[i].text;
53      Serial.println(text);
54
55      String from_name = bot.messages[i].from_name;
56
57      if (text == "/start") {
58        String welcome = "Welcome, " + from_name + ".\n";
59        welcome += "Use the following commands to control your ...
              outputs.\n\n";
60        welcome += "/login to generate OTP \n";
61        welcome += "/balance\n";
62        welcome += "/transaction \n";
63        welcome += "/amount withdrawn \n";
64        bot.sendMessage(chat_id, welcome, "");
65      }
66      v = random(10,99);
67      if (text == "/login")
68      {
69        bot.sendMessage(chat_id, "OTP is \n" + String(v) ,"");
70      }\
71
72      if (text == "/balance")
73      {
74        bot.sendMessage(chat_id, "Your Account Balance is\n" + ...
              String(bal) ,"");
75      }
76
77       if (text == "/transaction")
78      {
79        Serial.println("Enter OTP \n");
80        Serial.write(n);
81        Serial.println(n);
82        if(n == v)
83        {
84          Serial.println("Transaction Authorised");
85          Serial.write(amount);
86          Serial.println(amount);
87            if(amount > atm_val)
88          Serial.println("sufficient cash not in the device");
```

```
89            if(amount> bal)
90          Serial.println("Sufficient Funds not available");
91
92          Serial.println("Enter which notes you want");
93          Serial.write(a);
94          Serial.println(a);
95          Serial.write(b);
96          Serial.println(b);
97
98          Serial.write(c);
99          Serial.println(c);
100         cashdispensed = 2000*x + 1000*y + 500*z;
101         if(cashdispensed = amount)
102         Serial.println("amount is dispensed");
103         x = x - a ;
104         y = y - b ;
105         z = z - c;
106         bal = bal - amount;
107         atm_val = atm_val - amount;
108         Serial.print("atm_val");
109         Serial.print(atm_val);
110         Serial.print("2000 denomination ");
111         Serial.print(x);
112         Serial.print("1000 denomination ");
113         Serial.print(y);
114         Serial.print("500 denomination ");
115         Serial.print(z);
116         Serial.println("balance available is");
117         Serial.println(bal);
118
119       }
120
121       bot.sendMessage(chat_id, "\n" + String(bal) ,"");
122     }
123     if (text == "/amount")
124     {
125
126       bot.sendMessage(chat_id, "\n" + String(amount) ,"");
127     }
128   }
129 }
130 void setup() {
131
132
133   Serial.begin(115200);
134
135   #ifdef ESP8266
136     configTime(0, 0, "pool.ntp.org");
137     client.setTrustAnchors(&cert);
```

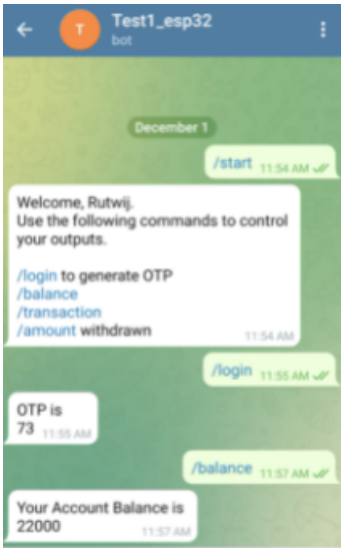
```
138   #endif
139
140
141
142   // Connect to Wi-Fi
143   WiFi.mode(WIFI_STA);
144   WiFi.begin(ssid, password);
145   #ifdef ESP32
146     client.setCACert(TELEGRAM_CERTIFICATE_ROOT);
147   #endif
148   while (WiFi.status() != WL_CONNECTED) {
149     delay(1000);
150     Serial.println("Connecting to WiFi..");
151   }
152   // Print ESP32 Local IP Address
153   Serial.println(WiFi.localIP());
154   //Serial.println("Enter OTP \n");
155
156   //Serial.write(n);
157   //if(n == v)
158     //Serial.println("Transaction Authorised");
159   //else
160     //Serial.println("Wrong OTP");
161   //Serial.println("Enter amount");
162   //Serial.write(amount);
163   //if(amount > atm_val)
164     //Serial.println("sufficient cash not in the device");
165   //if(amount> bal)
166     //Serial.println("Sufficient Funds not available");
167
168   //Serial.println("Enter which notes you want");
169   //Serial.write(a);
170   //Serial.write(b);
171   //Serial.write(c);
172   //cashdispensed = 2000*x + 1000*y + 500*z;
173   //if(cashdispensed = amount)
174     //Serial.println("amount is dispensed");
175     //x = x - a ;
176     //y = y - b ;
177     //z = z - c;
178     //bal = bal - amount;
179     //atm_val = atm_val - amount;
180   //Serial.println("balance available is");
181   //Serial.println(bal);
182
183 }
184
185 void loop()
186 {
```

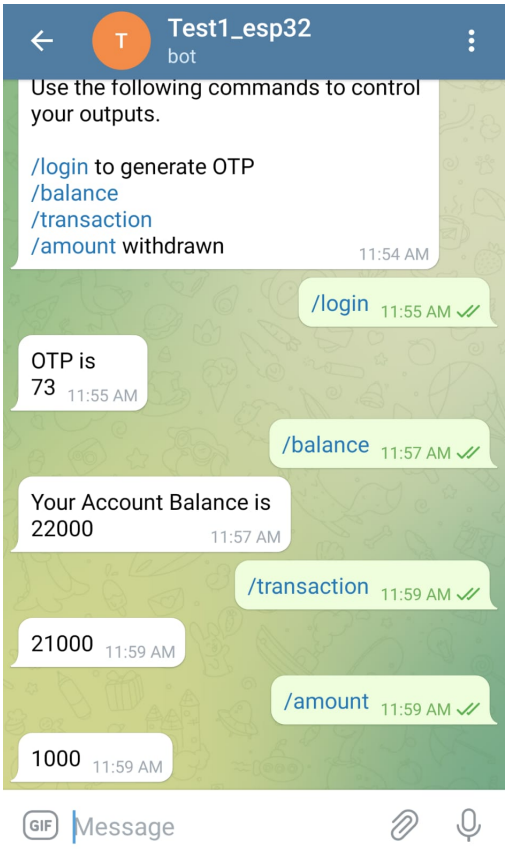
```
187    if (millis() > lastTimeBotRan + botRequestDelay)
188    {
189      int numNewMessages = ...
           bot.getUpdates(bot.last_message_received + 1);
190
191      while(numNewMessages)
192      {
193        Serial.println("got response");
194        handleNewMessages(numNewMessages);
195        numNewMessages = bot.getUpdates(bot.last_message_received ...
             + 1);
196      }
197      lastTimeBotRan = millis();
198    }
199     if(WiFi.status() == WL_CONNECTED )
200    {
201      HTTPClient http;
202      http.begin(serverName);
203      String DataSent = "api_key=" +apiKey + "&field1=" + ...
           String(atm_val) + "&field2=" + String(x) +  "&field3=" + ...
           String(y) + "&field4=" + String(z);
204      int Response = http.POST(DataSent);
205      //int Response = http.POST(DataSent);
206
207
208      //String DataSent = "api_key=" +apiKey + "&field1=" + ...
           String(v) + "&field2=" + String((temprature_sens_read() - ...
           32) / 1.8));
209      //int Response = http.POST(DataSent);
210      http.end();
211    }
212
213
214  }
```

Telegram bot : we use telegram bot to generate and send otp as well as to check amount and funds available. for purpose of authentication a particular chat id only can acces the telegram bot and the transaction command runs the transaction digitally.


Test1_esp32
bot
December 1
/start 11:54 AM
Welcome, Rutwij.
Use the following commands to control your outputs.
/login to generate OTP
/balance
/transaction
/amount withdrawn
11:54 AM
Message


Test1_esp32
bot
December 1
/start 11:54 AM
Welcome, Rutwij.
Use the following commands to control your outputs.
/login to generate OTP
/balance
/transaction
/amount withdrawn
11:54 AM
/login 11:55 AM
OTP is 73 11:55 AM
Message

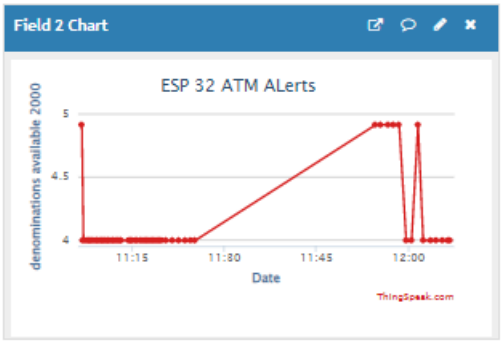

Test1_esp32
bot
December 1
/start 11:54 AM
Welcome, Rutwij.
Use the following commands to control your outputs.
/login to generate OTP
/balance
/transaction
/amount withdrawn
11:54 AM
/login 11:55 AM
OTP is 73 11:55 AM
/balance 11:57 AM
Your Account Balance is 22000 11:57 AM

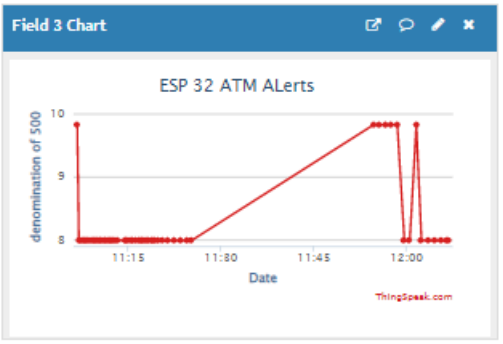

Test1_esp32
bot
Use the following commands to control your outputs.
/login to generate OTP
/balance
/transaction
/amount withdrawn
11:54 AM
/login 11:55 AM
OTP is 73 11:55 AM
/balance 11:57 AM
Your Account Balance is 22000 11:57 AM
/transaction 11:59 AM
21000 11:59 AM
/amount 11:59 AM
1000 11:59 AM
GIF
Message

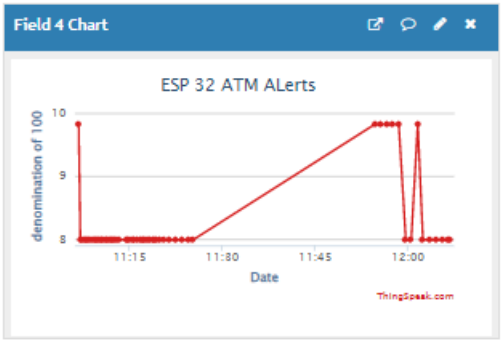
ATM interface: The ATM shows this values when we input the data using serial.write() commands.we control atm by both serial.write and telegram bot

```
12:01:30.842 -> 1
12:01:30.842 -> /login
12:01:36.685 -> got response
12:01:36.685 -> handleNewMessages
12:01:36.685 -> 1
12:01:36.685 -> /transaction
12:01:36.685 -> Enter OTP
12:01:36.685 ->
12:01:36.685 ->  0
12:01:36.685 -> Transaction Authorised
12:01:36.685 -> ₹1000
12:01:36.685 -> Enter which notes you want
12:01:36.685 ->  1
12:01:36.685 ->  2
12:01:36.685 ->  2
12:01:36.685 -> amount is dispensed
12:01:36.685 -> atm_val240002000 denomination 41000 denomination 8500 denomination 8balance availa
12:01:36.685 -> 21000
12:01:49.801 -> got response
12:01:49.801 -> handleNewMessages
12:01:49.801 -> 1
12:01:49.801 -> /amount
```

Thingspeak Interface: this interface updates the bank side of transaction and the cassetes or denominations in atm We denoted 3 different fields for 3 different denominations just like cassettes in real atm's case so that we can monitor the denominations seperately.hence three graphs for 3 denominations so in this way we can track the denominaitons as to when the transaction took place.


Field 2 Chart
ESP 32 ATM ALerts
denominations available 2000
5
4.5
4
11:15
11:30
11:45
12:00
Date
ThingSpeak.com


Field 3 Chart
ESP 32 ATM ALerts
denomination of 500
10
9
8
11:15
11:30
11:45
12:00
Date
ThingSpeak.com


Field 4 Chart
ESP 32 ATM ALerts
denomination of 100
10
9
8
11:15
11:30
11:45
12:00
Date
ThingSpeak.com

Modifications done by me in the project: In the given Problem statement, it is asked for inputs to be taken using touch sensor on esp32, however this could be also done by using serial.write command hence i tried to use it .So the authentication is done on serial monitor using the otp sent to the telegram bot. the Above project also uses commands "/transaction and /amount for making transaction happen the code checks if the otp is correct or not and if it is correct then only the transaction happens

Value addition to the above project: The above project could be made more efficient and faster if we use the RTOS approach dedicating one core for telegram bot and other for thingspeak and transaction. also we can use the hall sensor inside esp32 and magnet simulating it as card used in real atms. also we can interface esp32 with keypad for input and lcd screen for output and display transaction on it.We ca also use whatsapp/signal bot instead of telegram bot