



# VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY (VNIT), NAGPUR

---

## Embedded Systems (ECL403)

### End Semester exam

---

*Submitted by :*

Simma Jaswanth (BT19ECE105)

Semester 5

*Submitted to :*

Dr. Ankit A. Bhurane

(Course Instructor)

Department of Electronics and Communication Engineering,  
VNIT Nagpur

# Contents

1	ATM system using ESP32 . . . . .	2
1.1	Aim . . . . .	2
1.2	Requirements . . . . .	2
1.3	Concept . . . . .	2
1.4	Code . . . . .	2
1.5	Hardware connections . . . . .	15
1.6	Procedure followed . . . . .	16
1.7	Observations and conclusion . . . . .	16

## ATM system using ESP32

**1.1 Aim:** Designing a ATM system using ESP32

**1.2 Requirements:** Arduino, ESP32 micro-controller, jump wires

**1.3 Concept:** Touch sensors are used as buttons of ATM. It should first verify using OTP and withdrawal the money. The commands should be come from telegram bot. Balance and remaining notes should be updated regularly.

### 1.4 Code:

```

1
2
3 #include <WiFi.h>
4 #include <WiFiClientSecure.h>
5 #include <UniversalTelegramBot.h>
6 #include <HTTPClient.h>
7 // Wifi network station credentials
8 #define WIFI_SSID "Jaswanth"
9 #define WIFI_PASSWORD "Vasu@Jashu"
10 // Telegram BOT Token (Get from Botfather)
11 #define BOT_TOKEN "2057741416:AAFFrhGoVNSc_c6Phv2T2kvSy8-paJOFYIE"
12 const char* server = "http://api.thingspeak.com/update";
13 String apiKey = "DBRUVDHXXBN4AEYQ"; // Enter your Write API ...
    key from ThingSpeak
14 const unsigned long BOT_MTBS = 1000; // mean time between scan ...
    messages
15 WiFiClient client;
16 WiFiClientSecure securedClient;
17 UniversalTelegramBot bot(BOT_TOKEN, securedClient);
18 unsigned long bot_lasttime; // last time messages' scan has been ...
    done
19
20 int bal_amount = 25000;
21 int notesnum[3] = {5,10,10}; //2000,1000,500 rupees notes
22 int pins_touch[8] = {4,15,13,12,14,27,33,32};
23 int vals[8] = {50,35,30,60,35,30,20,20};
24 void handleNewMessages(int numNewMessages)
25 {
26     Serial.print("handleNewMessages ");
27     Serial.println(numNewMessages);
28
29     for (int i = 0; i < numNewMessages; i++)
30     {
31         String chat_id = bot.messages[i].chat_id;

```

```
32     String text = bot.messages[i].text;
33
34     String from_name = bot.messages[i].from_name;
35     if (from_name == "")
36         from_name = "Guest";
37
38     if (text == "/login")
39     {
40         int v;
41
42         v = random(10,99);
43         bot.sendMessage(chat_id, String(v)+" is your OTP"+" ...
44             "+"Enter it", "");
45         int c = 0;
46         int sum = 0;
47
48         while (c<2)
49         {
50             digitalWrite(2,HIGH);
51             delay(5000);
52             int i;
53             if (touchRead(pins_touch[0])<vals[0])
54             {
55                 i=0;
56                 c++;
57             }
58             if (touchRead(pins_touch[1])<vals[1])
59             {
60                 i=1;
61                 c++;
62             }
63             if (touchRead(pins_touch[2])<vals[2])
64             {
65                 i=2;
66                 c++;
67             }
68             if (touchRead(pins_touch[3])<vals[3])
69             {
70                 i=3;
71                 c++;
72             }
73             if (touchRead(pins_touch[4])<vals[4])
74             {
75                 i=4;
76                 c++;
77             }
78             if (touchRead(pins_touch[5])<vals[5])
79             {
80                 i=5;
```

```
80         c++;
81     }
82     if(touchRead(pins_touch[6])<vals[6])
83     {
84         i=6;
85         c++;
86     }
87     if(touchRead(pins_touch[7])<vals[7])
88     {
89         i=7;
90         c++;
91     }
92     if(touchRead(pins_touch[5])<vals[5] && ...
        touchRead(pins_touch[6])<vals[6])
93     {
94         i=8;
95         c++;
96     }
97     if(touchRead(pins_touch[7])<vals[7] && ...
        touchRead(pins_touch[6])<vals[6])
98     {
99         i=9;
100        c++;
101    }
102    Serial.println(i);
103    digitalWrite(2,LOW);
104    sum = sum*10 + i;
105 }
106 bot.sendMessage(chat_id, "Verifying.....", "");
107 if(sum == v)
108 {
109     bot.sendMessage(chat_id, "Verified", "");
110     bot.sendMessage(chat_id, "/step1", "");
111 }
112 else
113 {
114     bot.sendMessage(chat_id, "Wrong OTP, please /login ...
        again", "");
115 }
116 }
117 if(text == "/balance")
118 {
119     Serial.print("Balance = ");
120     Serial.println(bal_amount);
121     Serial.print("2000 notes remained = ");
122     Serial.println(notesnum[0]);
123     Serial.print("1000 notes remained = ");
124     Serial.println(notesnum[1]);
125     Serial.print("500 notes remained = ");
```

```
126     Serial.println(notesnum[2]);
127     bot.sendMessage(chat_id, "Balance = "+String(bal_amount), "");
128     bot.sendMessage(chat_id, "2000 notes remained = ...
        "+String(notesnum[0]), "");
129     bot.sendMessage(chat_id, "1000 notes remained = ...
        "+String(notesnum[1]), "");
130     bot.sendMessage(chat_id, "500 notes remained = ...
        "+String(notesnum[2]), "");
131     bot.sendMessage(chat_id, "/step1", "");
132 }
133 if(text == "/WithdrawlMoney")
134 {
135     bot.sendMessage(chat_id, "Enter the amount required in ...
        multiples of 500 or 1000 or 2000 " , "");
136
137     bot.sendMessage(chat_id, "pin 1 5000 \n pin2 10000 \n pin ...
        3 15000 \n pin 4 20000 \n pin 5 500 \n pin 6 1000\n pin ...
        7 2000 \n pin 8 ENTER" , "");
138
139     int sum = 0;
140
141     while(touchRead(pins.touch[7])>vals[7])
142     {
143         digitalWrite(2,HIGH);
144         delay(5000);
145         int i;
146         if(touchRead(pins.touch[0])<vals[0])
147         {
148             i=5000;
149
150         }
151         if(touchRead(pins.touch[1])<vals[1])
152         {
153             i=10000;
154
155         }
156         if(touchRead(pins.touch[2])<vals[2])
157         {
158             i=15000;
159
160         }
161         if(touchRead(pins.touch[3])<vals[3])
162         {
163             i=20000;
164
165         }
166         if(touchRead(pins.touch[4])<vals[4])
167         {
168             i=500;
```

```

169     }
170     if (touchRead(pins_touch[5]) < vals[5])
171     {
172         i=1000;
173     }
174     if (touchRead(pins_touch[6]) < vals[6])
175     {
176         i=2000;
177     }
178
179     digitalWrite(2, LOW);
180     sum = sum + i;
181     Serial.println(i);
182     Serial.println("touch pin 8 if completes");
183     delay(1000);
184 }
185 bot.sendMessage(chat_id, "Processing....", "");
186 if (bal_amount >= sum)
187 {
188     int notes[3] = {0, 0, 0};
189     processing(sum, notesnum, notes);
190     bal_amount = bal_amount - sum;
191     Serial.print("Balance = ");
192     Serial.println(bal_amount);
193     Serial.print("2000 notes remained = ");
194     Serial.println(notesnum[0]);
195     Serial.print("1000 notes remained = ");
196     Serial.println(notesnum[1]);
197     Serial.print("500 notes remained = ");
198     Serial.println(notesnum[2]);
199     bot.sendMessage(chat_id, "2000 notes = " + ...
200         String(notes[0]) + "\n" + "1000 notes = " + ...
201         String(notes[1]) + "\n" + "500 notes = " + ...
202         String(notes[2]) + "\n", "");
203     bot.sendMessage(chat_id, "Withdrawal amount = " + ...
204         String(sum), "");
205     bot.sendMessage(chat_id, "Balance = " + ...
206         String(bal_amount), "");
207     HTTPClient http;
208     http.begin(server);
209     String DataSent = "api-key=" + apiKey + "&field1=" + ...
210         String(bal_amount) + "&field2=" + String(notesnum[0]) + ...
211         + "&field3=" + String(notesnum[1]) + "&field4=" + ...
212         String(notesnum[2]);
213     int Response = http.POST(DataSent);
214     http.end();
215     client.stop();
216     delay(1000);
217     bot.sendMessage(chat_id, "/step1", "");

```

```
210     }
211     else
212     {
213         bot.sendMessage(chat_id,"Exceeded, please ...
                /WithdrawlMoney again", "");
214     }
215 }
216 if(text == "/Exit")
217 {
218     bot.sendMessage(chat_id,"THANK YOU VISIT AGAIN :)", "");
219 }
220
221
222 if (text == "/step1")
223 {
224     String welcome = "Welcome to Our Bank, " + from_name + ".\n";
225     welcome += "Select your purpose\n";
226     welcome += "/balance : to check the balance\n";
227     welcome += "/WithdrawlMoney : to withdraw the money\n";
228     welcome += "/Exit : exit from services\n";
229     bot.sendMessage(chat_id, welcome, "Markdown");
230 }
231 }
232 }
233
234 void processing(int sum,int notes[],int note[])
235 {
236     int n1,n2,n3;
237     n1 = sum/2000;
238     if(n1>notes[0])
239     {
240         n1 = notes[0];
241     }
242     sum = sum - (n1*2000);
243     n2 = sum/1000;
244     if(n2>notes[1])
245     {
246         n2 = notes[1];
247     }
248     sum = sum - (n2*1000);
249     n3 = sum/500;
250     if(n3>notes[2])
251     {
252         n3 = notes[2];
253     }
254     sum = sum - (n3*500);
255     notes[0] = notes[0]-n1;
256     notes[1] = notes[1]-n2;
257     notes[2] = notes[2]-n3;
```



```

258   note[0] = n1;
259   note[1] = n2;
260   note[2] = n3;
261 }
262
263 void setup()
264 {
265   Serial.begin(115200);
266   Serial.println();
267   pinMode(2, INPUT);
268   pinMode(pins.touch[0], INPUT);
269   pinMode(pins.touch[1], INPUT);
270   pinMode(pins.touch[2], INPUT);
271   pinMode(pins.touch[3], INPUT);
272   pinMode(pins.touch[4], INPUT);
273   pinMode(pins.touch[5], INPUT);
274   pinMode(pins.touch[6], INPUT);
275   pinMode(pins.touch[7], INPUT);
276
277   // attempt to connect to Wifi network:
278   Serial.print("Connecting to Wifi SSID ");
279   Serial.print(WIFI_SSID);
280   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
281   secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add ...
                root certificate for api.telegram.org
282   while (WiFi.status() != WL_CONNECTED)
283   {
284     Serial.print(".");
285     delay(500);
286   }
287   Serial.print("\nWiFi connected. IP address: ");
288   Serial.println(WiFi.localIP());
289
290   Serial.print("Retrieving time: ");
291   configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP
292   time_t now = time(nullptr);
293   while (now < 24 * 3600)
294   {
295     Serial.print(".");
296     delay(100);
297     now = time(nullptr);
298   }
299   Serial.println(now);
300   HTTPClient http;
301   http.begin(server);
302   String DataSent = "api-key=" + apiKey + "&field1=" + ...
                String(bal.amount) + "&field2=" + String(notesnum[0]) + ...
                "&field3=" + String(notesnum[1]) + "&field4=" + ...
                String(notesnum[2]);

```

```
303   int Response = http.POST(DataSent);
304   http.end();
305   client.stop();
306   delay(1000);
307 }
308
309 void loop()
310 {
311   if (millis() - bot_lasttime > BOT_MTBS)
312   {
313     int numNewMessages = ...
        bot.getUpdates(bot.last_message_received + 1);
314
315     while (numNewMessages)
316     {
317       Serial.println("got response");
318       handleNewMessages(numNewMessages);
319       numNewMessages = bot.getUpdates(bot.last_message_received ...
        + 1);
320     }
321
322     bot_lasttime = millis();
323   }
324 }
```

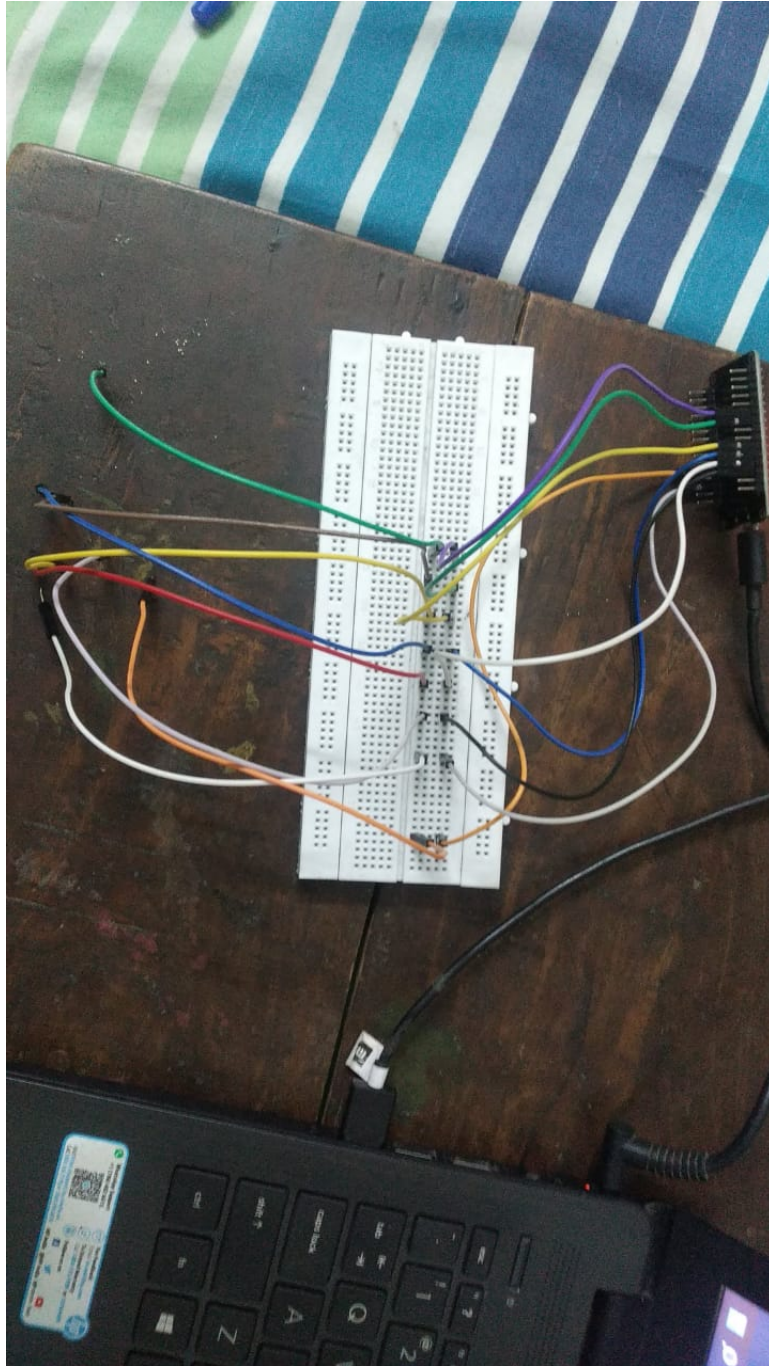


Figure 1: Hardware

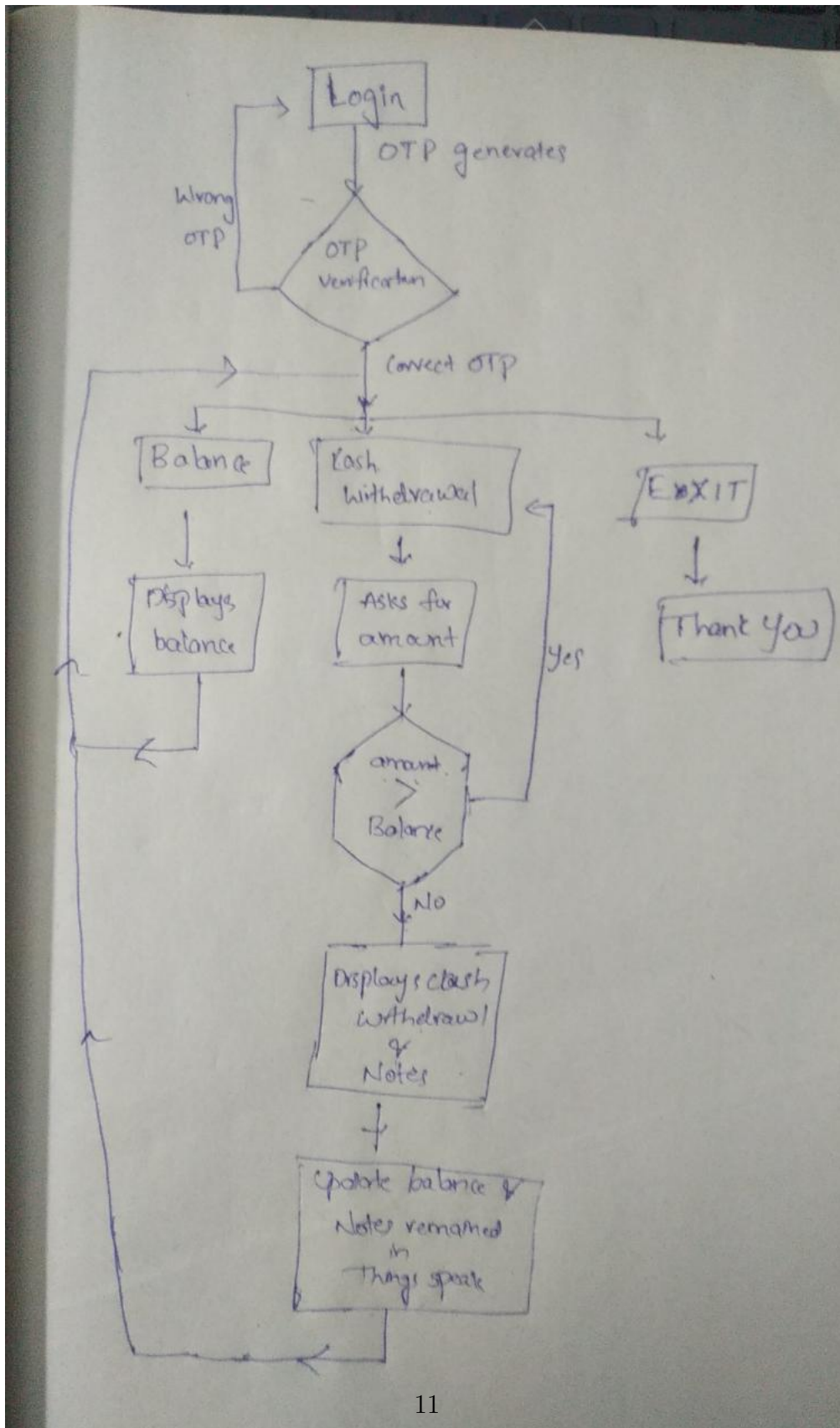


Figure 2: Block diagram of system

<u>Number</u>	<u>Pin</u>
0	4
1	15
2	13
3	12
4	14
5	27
6	33
7	32
8	<del>27</del> + 32
9	33 + 32

<u>Amount</u>	<u>Pin</u>
5000	4
10000	15
15000	13
20000	12
500	14
1000	27
2000	33

Enter pin  $\Rightarrow$  (32)

Figure 3: Pins configuration

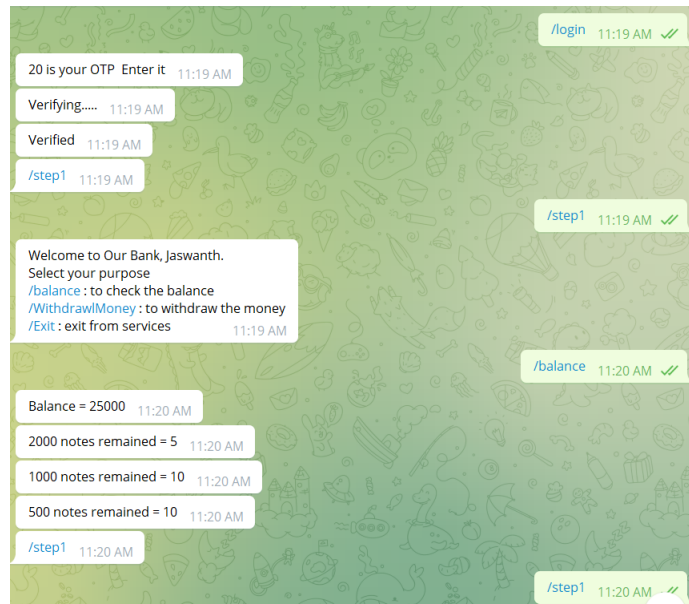


Figure 4: Bot login and balance check

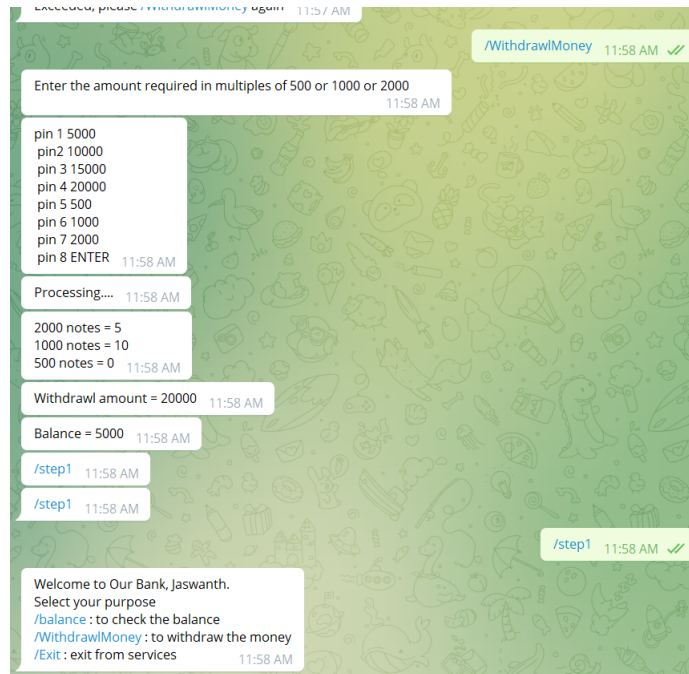


Figure 5: Bot cash withdrawal command

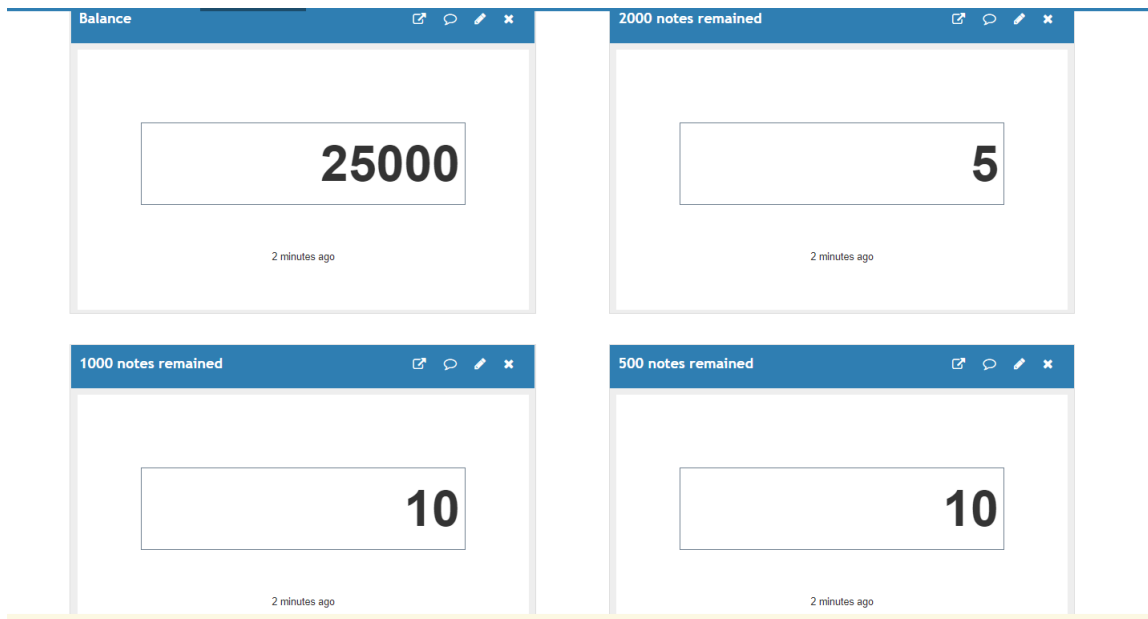


Figure 6: Thingspeak cloud intial values



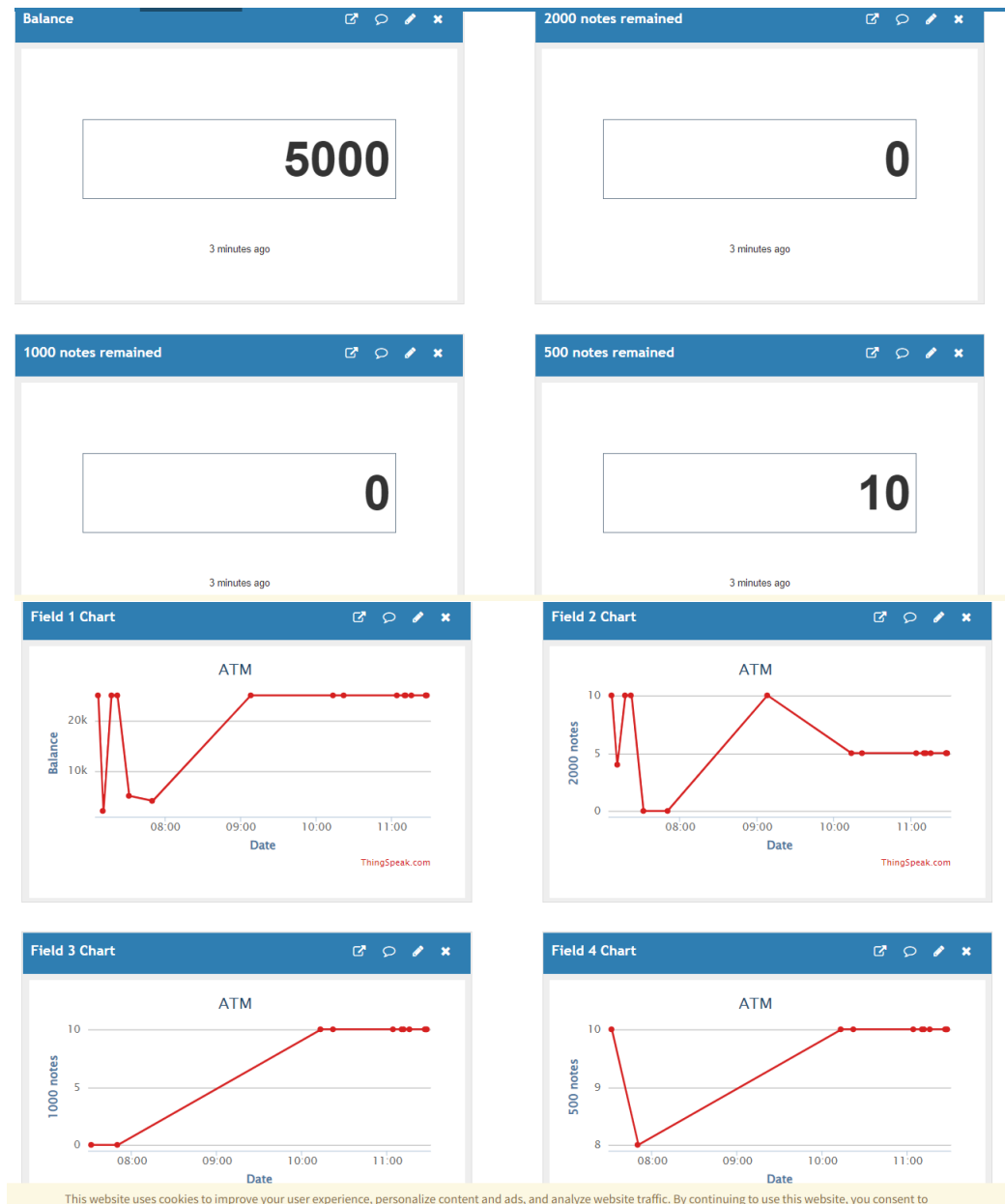


Figure 7: Thingspeak cloud updated after cash withdrawal

**1.5 Hardware connections:** ESP32 touch pins 4,15,12,1,21,27,33,32 are taken and those are connected with jump wires so that each pin can be flexibly used.

**Problems Faced (if any):** Touch sensor pins 0,2 are not working. So, remaining pins are encoded.



**1.6 Procedure followed:**

1. First, a bot is created in telegram and its token is included in code to interact with that bot.
2. Thingspeak channel is created with four fields. They are balance, 2000 , 1000, 500 notes.
3. The api key for write in channel is included.
4. When login command comes from bot, two digit otp is generated and that otp should be entered using touch sensors.
5. OTP verification is done. If wrong otp entered then it will direct to login again.
6. If correct otp entered then we get three commands. balance, withdrawlMoney and Exit.
7. If balance is send then balance and remaining notes are notified.
8. If cash withdrawal is selected, it will ask for amount required.
9. Pin confirguration also notified to bot.
10. We should enter the amount using touch sensors.
11. If amount is greater than balance then it will direct to cash withdrwal again.
12. If it is less then it will notify withdrawal money, balance and notes for entered amount.
13. Denomination of notes is done in processing function.
14. The balance and remaining notes are updated in THingspeak cloud.
15. In all the cases, both bot and serial monitor gets the same message.

**1.7 Observations and conclusion:**

- When login, otp is notified as 22.
- After touching the pins, the otp is verified and directed to next step.
- When balance command is sent, appropriate messages are notified.
- When cash withdrawal selected, it asked for amount.
- Entered two 10000 which means 20000.

- Now after processing, five 2000 notes, ten 1000 notes are issued. Also, balance and cash withdrawn is notified.
- Thingspeak cloud also updated as 5000 balance and 0,0,10 notes of 2000,1000,500.