Visvesvaraya National Institute of Technology (VNIT), Nagpur

Embedded Systems (ECP403)

Endsem Report

Submitted by :
Kumar Mridul (BT20ECE060)
Semester 5

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

The Ultimate Task

Aim:: To design ATM system using ESP32

Requirements::

- To perform operations : ESP-WROOM-32 microcontroller
- User Interface : Telegram BOT
- Database : GOOGLE Spreadsheet
- Editor to access database : AppScript

Code - Arduino and Appscript::

ESP32 Code:

```
//required libraries for my code
#include <WiFi.h>
#include <WiFiClientSecure.h>              //libraries to access ... WiFi-connection based functions
#include <HTTPClient.h>                    //library to access ... HTTP functions
#include <UniversalTelegramBot.h>          //library to access ... Telegram BOT for User Interaction
#include <ArduinoJson.h>                   //to implement json ... functions
#include <bits/stdc++.h>                   //to access to CPP ... functions and data structures

//personal wifi credentials
//const char* ssid = "";
//const char* password ="";

//VNIT Library wifi credentials
const char* ssid = "VeNky" ;                          ... //"Reading Room";
const char* password = "76543210";                    ... //"Password@4321";

//Google script id & credentials
String Google_Script_ID = ... "AKfycbwdRvDjl5ysBbjskI-4h8b1ClBOFNlnB6DvQwsHpCMR7b7WSuffB1kDo-x7VfbTGs02zw";

//telegram Bot credentials--> BOTtoken and Chat_id
#define BOTtoken "5902213155:AAEOGY6LV8s7nUU2DVIuV8RH5WEXyNFmgDk"
#define Chat_ID "1484391810"
```

```
23
24 //Creating a new Wi-Fi client with WiFiClientSecure
25 WiFiClientSecure client;
26 UniversalTelegramBot bot(BOTtoken, client);
27
28 String readUser(int row,char column){
29
30     if (WiFi.status() == WL_CONNECTED){
31     HTTPClient http;
32     String url = "https://script.google.com/macros/s/" + ...
           Google_Script_ID + "/exec?row=" + row + "&column=" + ...
           column + "&request=read";   //url to access data of the cell
33     http.begin(url.c_str());
34     http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
35     int httpCode = http.GET();
36     String data = http.getString();
37     if(httpCode == 200 ){
38       return data;
39     }
40     else{
41       return "Something's Wrong!";
42     }
43     http.end();
44   }
45 }
46
47 String writeUser(String username, String password){
48   if (WiFi.status() == WL_CONNECTED){
49     HTTPClient http;
50     String url = "https://script.google.com/macros/s/" + ...
           Google_Script_ID + "/exec?username=" + String(username) + ...
           "&password=" + String(password) + "&request=write" ;
51     http.begin(url.c_str());
52     http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
53     int httpCode = http.GET();
54     String data = http.getString();
55     if(httpCode == 200){
56       return data;
57     }
58     else{
59       return "Something's Wrong!";          //tO-DO
60     }
61     http.end();
62   }
63 }
64
65 void updateBalance(int row, char column, String value){
66
67   if(WiFi.status() == WL_CONNECTED)
```

```
68    {
69      HTTPClient http;
70      String url = "https://script.google.com/macros/s/" + ...
            Google_Script_ID + "/exec?row="+ String(row) + "&column=" ...
            + column + "&value=" + value + "&request=update";
71      http.begin(url.c_str());
72      http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
73      int httpCode = http.GET();
74      String data = http.getString();
75      if(httpCode == 200){
76        return;
77      }
78      else{
79        return;           //tO-DO
80      }
81      http.end();
82    }
83  }
84
85  //function to fetch data entered at the Telegram bot UI
86  String getInput()
87  {
88    int newMsg = bot.getUpdates(bot.last_message_received+1);
89    while(newMsg <1){
90      newMsg = bot.getUpdates(bot.last_message_received+1);
91    }
92    String userInput = bot.messages[0].text;
93    return userInput;
94  }
95
96  //In the setup, we can put the code for WiFi connection & ...
        display of the execution over Serial Monitor
97  void setup(){
98    Serial.begin(115200);
99    delay(500);
100   Serial.println();
101   Serial.print("Connecting to WiFi: ");
102   Serial.println(ssid);
103   Serial.flush();
104   WiFi.mode(WIFI_STA);
105   WiFi.begin(ssid, password);
106
107   while (WiFi.status() != WL_CONNECTED) {
108     delay(500);
109     Serial.print(".");
110   }
111   Serial.println("");
112   Serial.println("WiFi connected");
```

```
113   client.setCACert(TELEGRAM_CERTIFICATE_ROOT);          // Add ...
          root certificate for api.telegram.org
114 }
115
116 void thankyouMsg(){
117   bot.sendMessage(Chat_ID, "Thankyou! Have a nice day! :)","");
118   }
119
120 void errorMsg(){
121   bot.sendMessage(Chat_ID, "Oops! Your userID or password ...
          entered is wrong! :(","");
122 }
123
124 void loop()
125 {
126     bot.sendMessage(Chat_ID, " WELCOME to the ATM60_bot! \n For ...
          new user registration Enter -> 0 \n For existing users : ...
          \n \t \t To Withdraw, Enter -> 1 \n \t \t To Credit, ...
          Enter -> 2 \n \t \t To Check Balance, Enter -> 3 ", "");
127     String userInput = getInput();
128     //CREATING NEW USER ACCOUNT and returning userID to the ...
          newly registered user
129     if(userInput=="0"){
130       bot.sendMessage(Chat_ID, "Enter a new username:","");
131       String newUsername = getInput();
132       //Serial.println(newUsername);
133
134       bot.sendMessage(Chat_ID, "Set a password for your ...
            account:","");
135       String newPassword = getInput();
136       //Serial.println(newPassword);
137
138
139       String userID = writeUser(newUsername,newPassword);
140       //Serial.println(userID);
141       bot.sendMessage(Chat_ID, " Dear " + newUsername + " your ...
            account has been successfully created! \n Your userID ...
            is " + userID + "\n Safe Transactions!", "");
142       thankyouMsg();                                          ...
            //thankyou msg post user registration
143     }
144
145     //WITHDRAWAL of amount entered in multiple of 100Rs and ...
          updating the result
146     else if(userInput=="1"){
147       bot.sendMessage(Chat_ID, "Enter your userID to ...
            continue...","");
148       int userID = getInput().toInt();
149       String savedPassword = readUser(userID+1,'C');
```

```
150       bot.sendMessage(Chat_ID, "Enter your password:","");
151       String enteredPassword = getInput();
152       if(savedPassword == enteredPassword){
153         bot.sendMessage(Chat_ID, "Enter the number of 100Rs ...
              notes you would like to withdraw...","");
154         int notes = getInput().toInt();
155         int balance = readUser(userID+1,'D').toInt() - 100*notes;
156         Serial.println(balance);
157         String balance_str = String(balance);
158         updateBalance(userID+1,'D',balance_str);
159         thankyouMsg();
160       }
161       else{
162         errorMsg();                                                ...
              //sending error msg to T-bot
163       }
164     }
165
166     //CREDIT of amount deposited by the registered user post ...
          verification
167     else if(userInput=="2"){
168       bot.sendMessage(Chat_ID, "Enter your userID to ...
            continue...","");
169       int userID = getInput().toInt();
170       String savedPassword = readUser(userID+1,'C');
171       bot.sendMessage(Chat_ID, "Enter your password:","");
172       String enteredPassword = getInput();
173       if(savedPassword == enteredPassword){
174         bot.sendMessage(Chat_ID, "Enter the amount you want to ...
              deposit...","");
175         int creditAmount = getInput().toInt();
176         int balance = readUser(userID+1,'D').toInt() + creditAmount;
177         Serial.println(balance);
178         String balance_str = String(balance);
179         updateBalance(userID+1,'D',balance_str);
180         thankyouMsg();                                             ...
              //thankyou msg at the end of transaction
181       }
182       else{
183         errorMsg();                                                ...
              //sending error msg to T-bot
184       }
185     }
186
187   //BALANCE DISPLAY of registered user
188     else if(userInput=="3"){
189       bot.sendMessage(Chat_ID, "Enter your userID to ...
            continue...","");
190       int userID = getInput().toInt();
```

```
191         String savedPassword = readUser(userID+1,'C');
192         bot.sendMessage(Chat_ID, "Enter your password:","");
193         String enteredPassword = getInput();
194         if(savedPassword == enteredPassword){
195           String balance = readUser(userID+1,'D');
196           bot.sendMessage(Chat_ID, "Your current balance is -> " + ...
                balance + "rupees.","");
197           thankyouMsg();
198         }
199         else{
200           errorMsg(); ...
                                                                    ...
                //sending error msg to T-bot
201         }
202       }
203   }
```

AppScript

```
1
2
3   var sheet_id="1tnWOXcl-w84Nvlz50ICdKTSFsi2tv4ss9kH63Q_Z9wY";
4   var sheet_name="BT20ECE060_Kumar_Mridul";
5
6   var ss = SpreadsheetApp.openById(sheet_id);
7   var sheet = ss.getSheetByName(sheet_name);
8
9   function doGet(e){
10
11    var request = e.parameter.request;
12    if (request == "read"){
13      var row = Number(e.parameter.row);
14      var column = e.parameter.column;
15      var cell = column + row.toString();
16      return ...
            ContentService.createTextOutput(sheet.getRange(cell).getValue());
17    }
18
19    else if (request == "write"){
20      var username = e.parameter.username;
21      var password = e.parameter.password;
22      var id = SpreadsheetApp.getActiveSheet().getLastRow();     ...
            //new_id = row count of last filled row
23      sheet.appendRow([id,username,password,15000]);             ...
            //appending new row of having informations of new user
24      var data = sheet.getDataRange().getValues();
25      data = data[id][0];
```

```
26      return ContentService.createTextOutput(data);
27    }
28
29    else if (request == "update"){
30      var row = Number(e.parameter.row);
31      var column = e.parameter.column;
32      var value = e.parameter.value;
33      var cell = column + row.toString();
34      //updating the current balance amount
35      SpreadsheetApp.getActiveSheet().getRange(cell).setValue(value);
36    }
37
38  }
```

Operations:: The operations performed using ATM60 bot include new user registration, withdrawl and credit Transactions post Password verification and Balance Display of Registered User.

After the Wifi-connection has been established, Welcome message is displayed at the Telegram Bot User Interface. The user may enter 0 if he/she wishes to make a new account, enter 1 or 2 to make transactions - debit or credit respectively OR enter 3 to check the current account balance.In case of any wrong userID or password entry, Error Message is displayed and since the Operations Code is in the loop() function of Arduino IDE, the operation is re-intitialized and user can access to ATM60 again.

The backend of the task involves use of Arduino and Appscript and the database is Google Spreadsheet. Through the functions of READ, WRITE and UPDATE, we read, write and update data in the Google Spreadsheet. The functions in App-Script correspond to the functions readUser() , writeUser() and updateBalance() in Arduino and are connected through url parameters.

The full functioning of the bot can be seen in the Youtube Link given:

Watch how the ATM60 bot works HERE!

Conclusion and Problems Faced: The Ultimate Task - as the title itself was quite a learning experience and it feels good to have overcome so many problems by will to explore more and the interest to debug the code. Being a coding enthusiast, I enjoyed the journey and by the end of the project, have got more interested into the subject.

However,there were a few problems that I faced while performing the task:

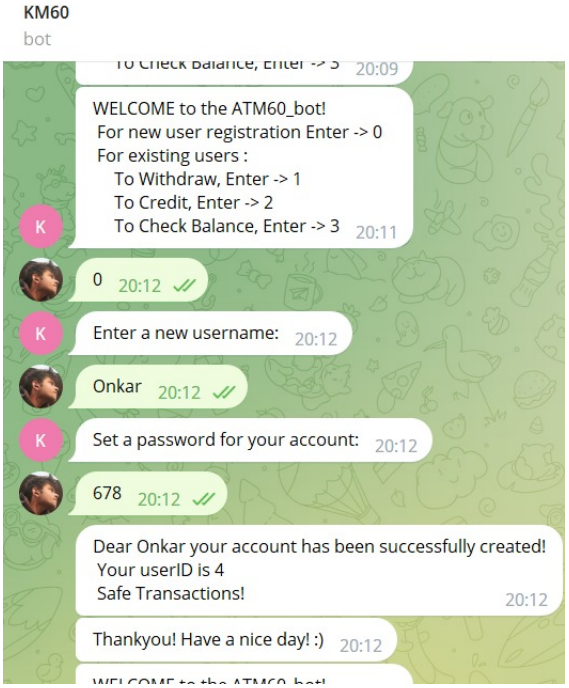

Figure 1: New Account

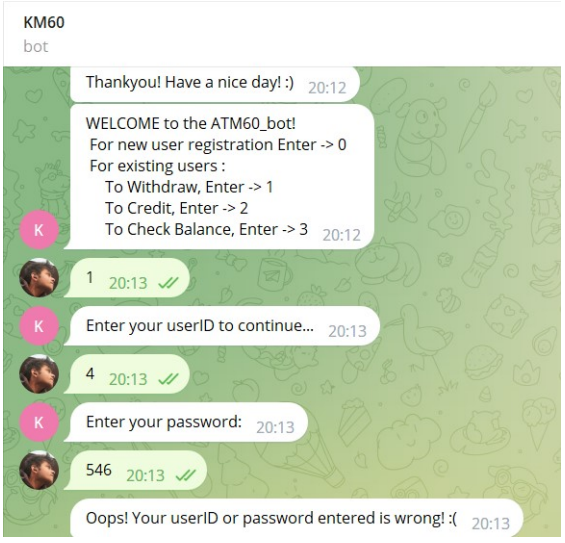

Figure 2: Incorrect userId or password entered

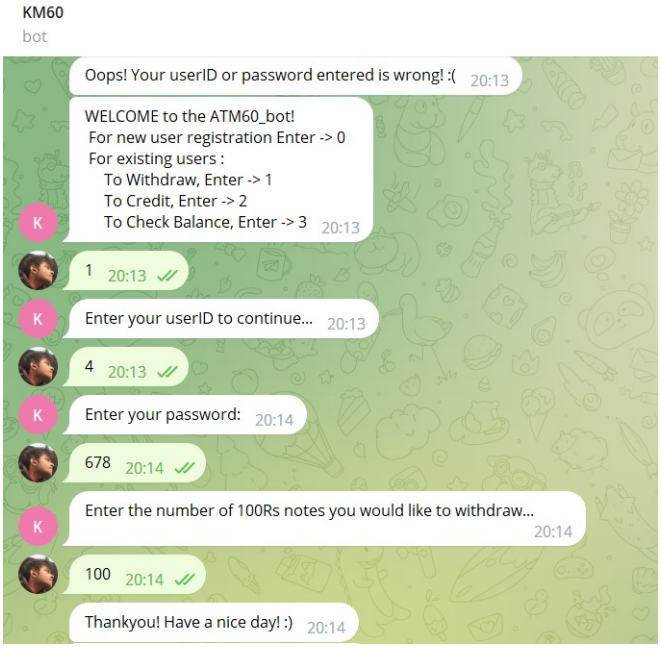

Figure 3: Withdraw money

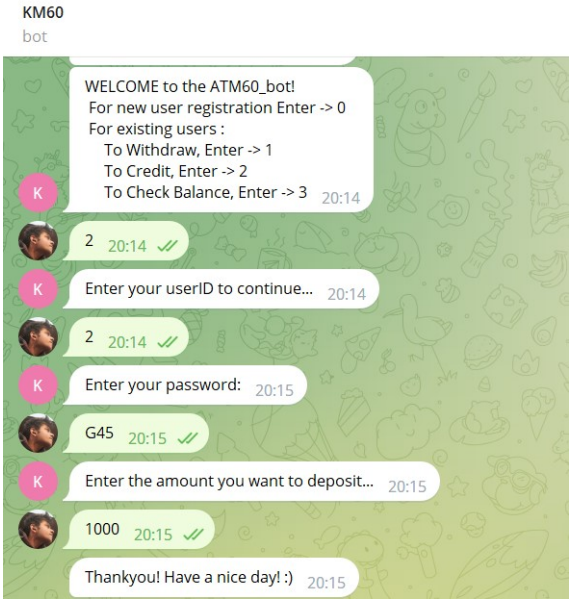

Figure 4: Deposit money

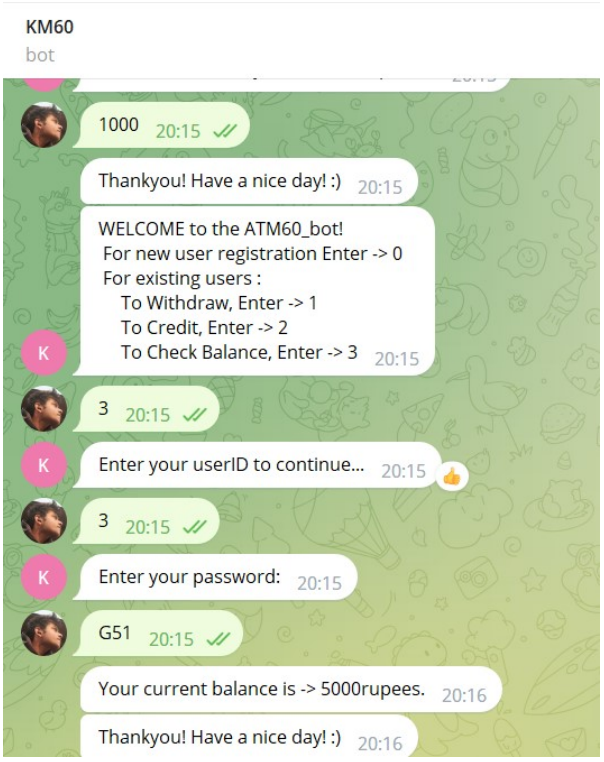

Figure 5: Display Balance

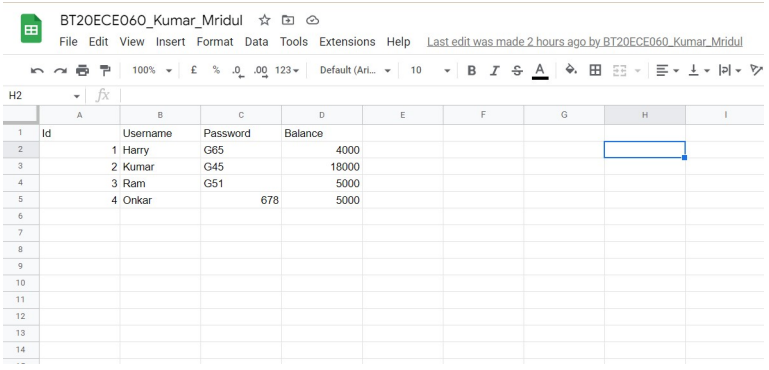

Figure 6: Display Balance on Google Sheets

- The ESP32 was not connecting with the WiFi and I had to use my fellow batchmates microcontroller after having tried so much with mine.
- Even after successful compilation, I faced difficulty in updating userBalance. The main confusion was that the Arduino function updateBalance() wasn't supposed to return anything, it being void type but the httpCode check for 200 requires to return something which I later understood can be null as well.
- The AppScript coding part wasn't easy for me because of very less web-development experience and since it is similar to javaScript.