BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416 (January 2022) **A2**
Microprocessor and Embedded Systems Laboratory

Final Project Report

IoT Based Bangla Calendar Clock

Evaluation Form:

STEP	DESCRIPTION	MAX	SCORE
1	Report (Format, Reference)	10	
2	Design Method and Complete Design (Hardware Implementation)	15	
3	Video Demonstration	10	
4	Novelty of Design	15	
5	Project Management and Cost Analysis	10	
6	Considerations to Public Health and Safety, Environment and Cultural and Societal Needs	10	
7	Assessment of Societal, Health, Safety, Legal and Cultural issues relevant to the solution	10	
8	Evaluation of the sustainability and impact of designed solution in societal and environmental contexts	10	
9	Individual Contribution (Viva)	20	
10	Team work and Diversity	10	
	TOTAL	120	

Signature of Evaluator: ______________________________

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and Write your name in your own handwriting. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Full Name: Saleh Ahmed Khan **Student ID:** 1706053	**Full Name:** Tiasa Mondal **Student ID:** 1706054
Full Name: Abdullah Jubair Bin Iqbal **Student ID:** 1706058	**Full Name:** Sadat Tahmeed Azad **Student ID:** 1706064

Table of Contents

1 Abstract

To address the negligence towards the Bangla calendar and necessity to address the lack of Bangla friendly devices, a digital clock has been made that can show the time in Bangla letters and digits and keep track of both the Bangla and the English calendars. The clock was based on an Arduino microcontroller. An RTC module is used to keep track of the time and ESP8266-ESP01 module have been used to establish Wi-Fi connectivity to synchronize the time twice a day. The time and date are displayed on dual P10 LED matrices. The time is kept static, while the scrolling date, day, and month (both Bangla and English) were displayed in a scrolling marquee style.

2 Introduction

Despite many digital wall clocks being available in the market, there is a scarcity of the same that keeps track of the Bangla calendar as well as the English one. Keeping track of the Bangla calendar is important in more than one aspect- firstly it helps us to keep in touch with our culture and tradition and track the important days and festivities. Secondly, in rural areas, where people are mostly dependent on agriculture for their livelihood, they still need to keep track of Bangla months for crop cutting, harvesting, sowing etc. Therefore, to address these issues, we have designed a digital clock that shows the time and date in Bangla letters, and keeps track of both Bangla and English calendars. This makes the clock accessible to the mass people, since, the huge majority of the people of this country have Bangla literacy. Besides all this, in order to make the clock more self-sustaining, we have added Wi-Fi connectivity, so that the time can be synced twice a day to keep the correct time. We believe that this project will ignite a spark of interest in the engineering community to address and solve more problems in Bangla.

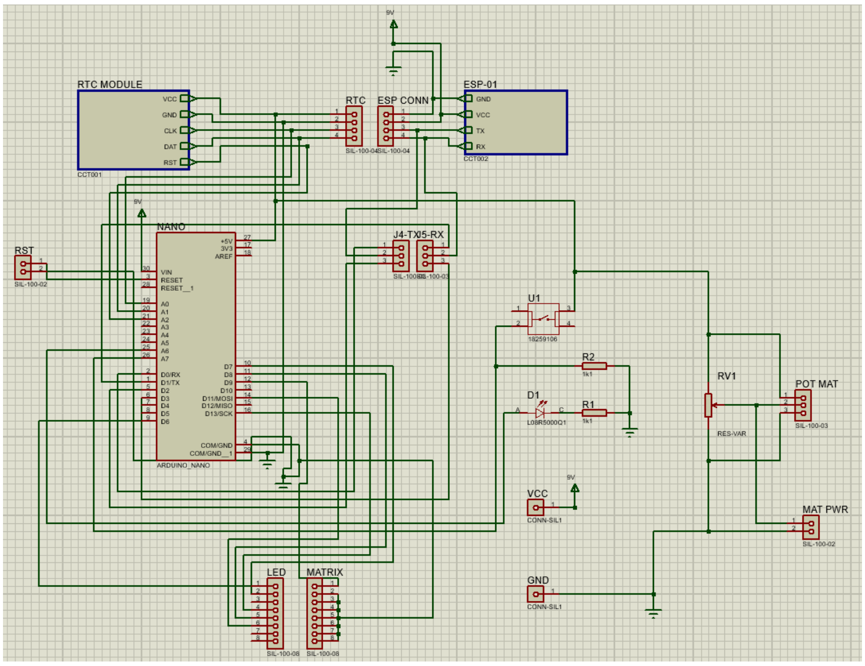
3 Design

3.1 Design Method

The following modules were implemented in out final project
Microcontroller:
Arduino Nano (ATMega328p)
Peripheral Components:
DS1307 Real Time Clock (RTC) Module
ESP8266 ESP-01 Wi-Fi Module
Display:
2 P10 32x16 LED Matrix Boards
Key Algorithms:
API Request, Serial and SPI Communications,
Matrix Manipulation, String Parsing

Choosing LED P10 boards as the suitable choice for wall mountable clock, our first task was to ensure that we can display any value of date and month in our assembly of two matrices. Separately, date and time were taken from the web and later, the two parts were brought together, implementing compact and judicial coding. PCB design was implemented to make the whole design compact and ready for commercialization.

3.2 Circuit Diagram



The circuit was generated using Proteus, from where the PCB design was completed

3.3 Full Source Code of Firmware

```
#include <virtuabotixRTC.h>
#include <EEPROM.h>
#include <SPI.h>
#include <DMD2.h>

SoftDMD dmd(2, 1);
virtuabotixRTC myRTC(A0, A1, A2); //
#define serialCommunicationSpeed 115200
#define time_out 1000
#define DEBUG 1

#define ban_date_0_addr 4
#define ban_date_1_addr 5
#define ban_mn_addr 6
#define ban_year_0_addr 7
#define ban_year_1_addr 8
#define ban_year_2_addr 9
#define ban_year_3_addr 10

uint8_t en_m_size[13] = {0, 43, 49, 19, 31, 13, 19, 30, 36, 47, 44, 38, 42};
uint8_t bn_m_size[13] = {0, 33, 23, 34, 28, 23, 35, 38, 48, 25, 20, 33, 22};
uint8_t wk_size[8]  = {0, 23, 21, 27, 30, 18, 24, 27};
uint8_t d_size = 19; //8+1+8+2
uint8_t y_size = 37; //2 + (4*8 + 3)

//uint8_t M1 = 12;  //sizeof(disp1) / sizeof(disp1[0]);
uint8_t N1 = 8; //sizeof(disp1[0]) / sizeof(disp1[0][0]);
uint32_t lo_op=0; //selects which column from the disp matrix will appear first
uint8_t second_loop = 0;

//time variables

uint8_t sync_time[6];
uint8_t tim_e[6];
uint8_t ampm=0; //0 if AM,1 if PM

uint8_t en_date[2];
uint8_t en_year[4];
uint8_t en_m; //month

uint8_t bn_date[2];
uint8_t bn_year[4];
uint8_t bn_m; //month

uint8_t week;

uint8_t count = 1;
uint8_t scroll = 1;

char sc = '"';
String rec_command;
String rec_data;
//String response = "";
uint8_t prev;

uint8_t dmd_flag = 0; // 0 = DMD is OFF, 1 = DMD is off

long int elapsed_time = 0;
uint8_t prev_min = 0;
uint8_t prev_sec = 0;
uint8_t trans_flag = 1;
uint8_t wifi_timeout = 0;

void(* resetFunc) (void) = 0; //declare reset function @ address 0
```

```
void setup()
{

Serial.begin(serialCommunicationSpeed);
dmd.setBrightness(255);

load_rtc();
delay(100);

initwifi();
delay(100);
}

void loop()
{
//Serial.println("Loading and Showing RTC Time...");
if(count == 5)
{
if(dmd_flag == 0)
{
//load_rtc();
Serial.end();
load_rtc();
prev_min = myRTC.minutes;
prev_sec = myRTC.seconds;
elapsed_time = millis();
dmd.begin();
dmd_flag = 1;
//populate_cells();
}
long int current_time = millis();
//if(((abs(elapsed_time - current_time)) >= (60000 - (prev_sec)*1000)) || (trans_flag == 1) ) //((abs(34 - prev_sec))*1000))
if(((abs(elapsed_time - current_time)) >= (60000 - (prev_sec)*1000)))    //////((abs(34 - prev_sec))*1000))
{
elapsed_time = current_time;s
dmd.end();
load_rtc();
if((prev_min != myRTC.minutes) && (trans_flag == 1))
{
  elapsed_time = current_time;
  //elapsed_time = current_time - 3500;
  trans_flag = 0;
}

prev_min = myRTC.minutes;
prev_sec = myRTC.seconds;
dmd.begin();

}
populate_cells();

}
else //// If count != 5
{
dmd.end();
dmd_flag = 0;
Serial.begin(serialCommunicationSpeed);
}

//load_rtc();

if(count == 1)
{
get_time();  //// Fetch data from ESP as string
set_time();  //// Converts the string into integer values to be used in populate_calls()
wifi_timeout += 1;  //// Number of tries to fetch data from online
if(wifi_timeout >= 15)
count = 4;
}
```

```
if(count == 2)
{
get_eng_date();
set_eng_date();
wifi_timeout += 1;
if(wifi_timeout >= 15)
count = 4;
}
if(count == 3)
{
get_ban_date();
set_ban_date();
wifi_timeout += 1;
if(wifi_timeout >= 15)
count = 4;
}
if(count == 4)
{
myRTC.updateTime();
prev = myRTC.hours;
count = count + 1;
//reads from EERPROM in case wifi is not available
if(wifi_timeout >= 15)
{
bn_date[0] = EEPROM.read(ban_date_0_addr);
bn_date[1] = EEPROM.read(ban_date_1_addr);
bn_m = EEPROM.read(ban_mn_addr);

bn_year[0] = EEPROM.read(ban_year_0_addr);
bn_year[1] = EEPROM.read(ban_year_1_addr);
bn_year[2] = EEPROM.read(ban_year_2_addr);
bn_year[3] = EEPROM.read(ban_year_3_addr);
}

}
if(count == 5)
{
wifi_timeout = 0;
//delay(250);
}
if((count == 5) && (tim_e[0] == 0) && (tim_e[1] == 6) && (tim_e[2] == 0) && (tim_e[3] == 0))
{
resetFunc();  //call reset
}

}
////// void loop ends ///////
////// Sends initiation commands to ESP module //////////
void sendData(String command, const int timeout, boolean debug)
{
rec_data = "";
Serial.print(command);
long int time = millis();
while( (time+timeout) > millis())
{
while(Serial.available())
{
char c = Serial.read();
rec_data+=c;
}
}
if(debug)
{
//lcd.setCursor(0,0);
//lcd.print(rec_data);
delay(1000);
}
}

///// Initialize Wifi //////
void initwifi(void)
{
//Serial.println("Closing Previous Connection");
delay(1);
rec_command = "AT+CIPCLOSE";
rec_command += "\r\n";
sendData(rec_command, 2000, 1);
delay(100);
{
//Serial.println("Initiating WiFi Module, Connecting to Specified WiFi");
delay(1);

rec_command = "AT+RST";
rec_command += "\r\n";
sendData(rec_command, 1000, 0);

rec_command = "AT+CWJAP=";
rec_command += sc;
rec_command += "alu wifi"; //SSID
rec_command += sc;
rec_command += ",";
rec_command += sc;
rec_command += "12345678"; //PASS
rec_command += sc;
rec_command += "\r\n";
sendData(rec_command, 2000, 1);
delay (1000);

rec_command = "\r\n";
sendData(rec_command, 1000, 1);

}


/// Get time as string from wifi /////
void get_time(void)
{
//Serial.println("Getting Online Time...");
//   lcd.setCursor(0,0);
//  lcd.print("Getting Time...");
//  delay(1);

rec_command = "AT+CIPSTART=";
rec_command += sc;
rec_command += "TCP";
rec_command += sc;
rec_command += ",";
rec_command += sc;
rec_command += "api.thingspeak.com";
rec_command += sc;
rec_command += ",80";
rec_command += "\r\n";
sendData(rec_command, 1, DEBUG);

delay(1);

rec_command = "AT+CIPSEND=90";
rec_command += "\r\n";
sendData(rec_command, 1, DEBUG);
delay(1);

rec_command = "GET /apps/thinghttp/send_request?api_key=Q1ZBWRBEXXO44K68";
rec_command += "\r\n";
rec_command += "Host:api.thingspeak.com";
rec_command += "\r\n\r\n\r\n\r\n";
sendData(rec_command, 1000, DEBUG);

delay(1);
}

//// Converts string to relevant integers for display function populate_cells //////
void set_time(void)
{
if(rec_data.indexOf("+IPD,53") > 0)
{
//Serial.println("Error 53");
}
else if(rec_data.indexOf("+IPD,132") > 0)
{
//Serial.println("Error 132");
}
else if(rec_data.indexOf("+IPD,") > 0)
{
//parsing the string
rec_data = rec_data.substring(rec_data.indexOf("+IPD")+20, rec_data.indexOf("+IPD")+32);

if(rec_data.substring(rec_data.indexOf("M")-1, rec_data.indexOf("M")) == "P")
```

```
{
sync_time[0] = ((rec_data.substring(rec_data.indexOf("M")-10, rec_data.indexOf("M")-8).toInt())/10);
sync_time[1] = ((rec_data.substring(rec_data.indexOf("M")-10, rec_data.indexOf("M")-8).toInt())%10);
if(sync_time[0] != 1 && sync_time[1] != 2) // 12 PM Special Case
{
  sync_time[0] = sync_time[0]+1;
  sync_time[1] = sync_time[1]+2;
}
}
if(rec_data.substring(rec_data.indexOf("M")-1, rec_data.indexOf("M")) == "A")
{
sync_time[0] = ((rec_data.substring(rec_data.indexOf("M")-10, rec_data.indexOf("M")-8).toInt())/10);
sync_time[1] = ((rec_data.substring(rec_data.indexOf("M")-10, rec_data.indexOf("M")-8).toInt())%10);
if(sync_time[0] == 1 && sync_time[1] == 2) // 12 AM Special Case - 0 hrs in 24 hr format
{
  sync_time[0] = 0;
  sync_time[1] = 0;
}
}
sync_time[2] = (rec_data.substring(rec_data.indexOf("M")-7, rec_data.indexOf("M")-5).toInt())/10;
sync_time[3] = (rec_data.substring(rec_data.indexOf("M")-7, rec_data.indexOf("M")-5).toInt())%10;
sync_time[4] = (rec_data.substring(rec_data.indexOf("M")-4, rec_data.indexOf("M")-2).toInt())/10;
sync_time[5] = (rec_data.substring(rec_data.indexOf("M")-4, rec_data.indexOf("M")-2).toInt())%10;
myRTC.setDS1302Time((sync_time[4]*10) + sync_time[5], (sync_time[2]*10) + sync_time[3], (sync_time[0]*10) + sync_time[1], week, (en_date[0]*10)+en_date[1], en_m, (en_year[0]*1000)+(en_year[1]*100)+(en_year[2]*10)+en_year[3]);

rec_data = "";
count = count+1;
}
}

void get_eng_date(void)
{
    //Serial.println("Getting Online English Date...");
   //     lcd.setCursor(0,0);
   // lcd.print("Getting Eng Date...");
    delay(1);

    rec_command = "AT+CIPSTART=";
    rec_command += sc;
    rec_command += "TCP";
    rec_command += sc;
    rec_command += ",";
    rec_command += sc;
    rec_command += "api.thingspeak.com";
    rec_command += sc;
    rec_command += ",80";
    rec_command += "\r\n";
    sendData(rec_command, 1, DEBUG);

    delay(1);

    rec_command = "AT+CIPSEND=90";
    rec_command += "\r\n";
    sendData(rec_command, 1, DEBUG);


    rec_command = "GET /apps/thinghttp/send_request?api_key=6EX7E9O58AOAB9DG";
    rec_command += "\r\n";
    rec_command += "Host:api.thingspeak.com";
    rec_command += "\r\n\r\n\r\n\r\n";
    sendData(rec_command, 1000, DEBUG);
}

void set_eng_date(void)
{
  if(rec_data.indexOf("+IPD,53") > 0)
  {
    //Serial.println("Error 53");
  }
  else if(rec_data.indexOf("+IPD,132") > 0)
  {
    //Serial.println("Error 132");
  }
  else if(rec_data.indexOf("+IPD,") > 0)
    {
      //parsing the string

      uint8_t dat_e = (rec_data.substring(rec_data.lastIndexOf(",")-2, rec_data.lastIndexOf(","))).toInt();
      en_date[0] = dat_e/10;
      en_date[1] = dat_e%10;
//Serial.println(rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2));
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "January")
        en_m = 1;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "February")
        en_m = 2;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "March")
        en_m = 3;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "April")
        en_m = 4;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "May")
        en_m = 5;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "June")
        en_m = 6;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "July ")
        en_m = 7;
if(((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "August") || ((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "August "))
        en_m = 8;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "September")
        en_m = 9;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "October")
        en_m = 10;
```

```
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "November")
        en_m = 11;
if((rec_data.substring(rec_data.indexOf(",",rec_data.indexOf(",")+1)+2, rec_data.lastIndexOf(",")-2)) == "December")
        en_m = 12;

      uint16_t yea_r = (rec_data.substring(rec_data.lastIndexOf(",")+2, rec_data.lastIndexOf(",")+6)).toInt();
      en_year[0] = yea_r/1000;
      en_year[1] = (yea_r%1000)/100;
      en_year[2] = ((yea_r%1000)%100)/10;
      en_year[3] = ((yea_r%1000)%100)%10;

//Serial.println(rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)));
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Saturday")
        week = 1;
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Sunday")
        week = 2;
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Monday")
        week = 3;
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Tuesday")
        week = 4;
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Wednesday")
        week = 5;
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Thursday")
        week = 6;
if((rec_data.substring(rec_data.indexOf(",")+9, rec_data.indexOf(",",rec_data.indexOf(",")+1)) == "Friday")
        week = 7;

      myRTC.setDS1302Time((sync_time[4]*10) + sync_time[5], (sync_time[2]*10) + sync_time[3], (sync_time[0]*10) + sync_time[1], week, (en_date[0]*10)+en_date[1], en_m, (en_year[0]*1000)+(en_year[1]*100)+(en_year[2]*10)+en_year[3]);

      rec_data = "";
      count = count+1;
    }
}
```

```
void get_ban_date(void)
{
//Serial.println("Getting Online Bangla Date...");
//               lcd.setCursor(0,0);
// lcd.print("Getting Ban Date...");
// delay(1);

rec_command = "AT+CIPSTART=";
rec_command += sc;
rec_command += "TCP";
rec_command += sc;
rec_command += ",";
rec_command += sc;
rec_command += "api.thingspeak.com";
rec_command += sc;
rec_command += ",80";
rec_command += "\r\n";
sendData(rec_command, 1, DEBUG);

delay(1);

rec_command = "AT+CIPSEND=90";
rec_command += "\r\n";
sendData(rec_command, 1, DEBUG);
rec_command = "GET /apps/thinghttp/send_request?api_key=5MKIPC0USLFNP3UT";
rec_command += "\r\n";
rec_command += "Host:api.thingspeak.com";
rec_command += "\r\n\r\n\r\n\r\n";
sendData(rec_command, 2000, DEBUG);
}

// Convert bangla date string to intetgers //
void set_ban_date(void)
{
  if(rec_data.indexOf("+IPD,53") > 0)
  {
    //Serial.println("Error 53");
  }
  else if(rec_data.indexOf("+IPD,132") > 0)
  {
    //Serial.println("Error 132");
  }
  else if(rec_data.indexOf("+IPD,") > 0)
    {
      //Serial.println("Setting Bangla Date");
   //                    lcd.setCursor(0,0);
   // lcd.print("Setting Bangla Date...");
//Serial.println(rec_data.substring(rec_data.indexOf("+IPD")));
      //parsing the string

      int dat_e = (rec_data.substring(rec_data.lastIndexOf("-")+1, rec_data.lastIndexOf("-")+4)).toInt();
      bn_date[0] = dat_e/10;
      bn_date[1] = dat_e%10;

//Serial.println(rec_data.substring(rec_data.indexOf(" ", rec_data.indexOf("-")+2)+1, rec_data.lastIndexOf(",")));
      String mont_h = rec_data.substring(rec_data.indexOf(" ", rec_data.indexOf("-")+2)+1, rec_data.lastIndexOf(","));
      if(mont_h == "Baisakh")
        bn_m = 1;
      if(mont_h == "Joishtho")
        bn_m = 2;
      if(mont_h == "Asadha")
        bn_m = 3;
      if(mont_h == "Srabon")
        bn_m = 4;
      if(mont_h == "Bhadra")
        bn_m = 5;
      if(mont_h == "Ashshin")
        bn_m = 4;
      if(mont_h == "Kartik")
        bn_m = 7;
      if(mont_h == "Ogrohayon")
        bn_m = 8;
      if(mont_h == "Push")
        bn_m = 9;
      if(mont_h == "Magh")
        bn_m = 10;
      if(mont_h == "Falgun")
        bn_m = 11;
      if(mont_h == "Choitro")
        bn_m = 12;

      int yea_r =
(rec_data.substring(rec_data.lastIndexOf(",")+2,
rec_data.lastIndexOf(",")+6)).toInt();
      bn_year[0] = yea_r/1000;
      bn_year[1] = (yea_r%1000)/100;
      bn_year[2] = ((yea_r%1000)%100)/10;
      bn_year[3] = ((yea_r%1000)%100)%10;

      //Store Bangla date in EEPROM
      EEPROM.write(ban_date_0_addr,bn_date[0]);
      EEPROM.write(ban_date_1_addr, bn_date[1]);

      EEPROM.write(ban_mn_addr, bn_m);

      EEPROM.write(ban_year_0_addr, bn_year[0]);
      EEPROM.write(ban_year_1_addr, bn_year[1]);
      EEPROM.write(ban_year_2_addr, bn_year[2]);
      EEPROM.write(ban_year_3_addr, bn_year[3]);

      rec_data = "";
      count = count+1;
    }
}

void load_rtc(void)
{
  myRTC.updateTime();
  //Serial.println(myRTC.hours);
  if(myRTC.hours >= 13)
  {
    tim_e[0] = (myRTC.hours-12)/10;
    tim_e[1] = (myRTC.hours-12)%10;
  }
  else if(myRTC.hours == 0)
  {
    tim_e[0] = 1;
    tim_e[1] = 2;
  }
  else
  {
    tim_e[0] = (myRTC.hours)/10;
    tim_e[1] = (myRTC.hours)%10;
  }

  tim_e[2] = (myRTC.minutes)/10;
  tim_e[3] = (myRTC.minutes)%10;

  tim_e[4] = myRTC.seconds/10;
  tim_e[5] = myRTC.seconds%10;

  if(myRTC.hours >= 12)
    ampm = 1; //0 if AM,1 if PM
  else
    ampm = 0; //0 if AM,1 if PM

  week = myRTC.dayofweek;

  en_date[0] = myRTC.dayofmonth/10;
  en_date[1] = myRTC.dayofmonth%10;

  en_m = myRTC.month;

  en_year[0] = myRTC.year/1000;
  en_year[1] = (myRTC.year%1000)/100;
  en_year[2] = ((myRTC.year%1000)%100)/10;
  en_year[3] = ((myRTC.year%1000)%100)%10;
}
```

```
/// MATRIX MANIPULATION //////////
//matrices for English months
uint64_t en_months( uint8_t en_m, uint8_t i,
uint8_t shft ){
uint64_t temp = 0;

 if (en_m==1){ // JANUARY, size = 43
    if      (i==0)  temp =
0b0000000000000000000000000000000000011111100;
    else if (i==1)  temp =
0b0000000000000000000000000000000000100000011;
    else if (i==2)  temp =
0b0000000000000000000000000000000000100000000;
    else if (i==3)  temp =
0b1111111111110111111111111111111011111111111;
    else if (i==4)  temp =
0b0000101100001000000010011000100100100000010;
    else if (i==5)  temp =
0b1001000011101000000010000100100100100001110;
    else if (i==6)  temp =
0b1001001000101001111010001110100100100110010;
    else if (i==7)  temp =
0b1001001101001001101010010000100100100100010;
    else if (i==8)  temp =
0b0100110101001001110110011100100100100011010;
    else if (i==9)  temp =
0b0100000101001000000110000010100100100000110;
    else if(i==10)  temp =
0b0010001001001000000010000001100100100010010;
    else if(i==11)  temp =
0b0001111001001000000010000100100100100010010;
    else if(i==12)  temp =
0b0000000000100000001111000100000000000000000;
    else if(i==13)  temp =
0b0000000000000000001011100000000000000000000;
    else if(i==14)  temp =
0b0000000000000000001110010000000000000000000;
    else if(i==15)  temp =
0b0000000000000000000000000000000000000000000;
 }

/// Folded due to being repetitive ///
else if (en_m==12){  // DECEMBER, size = 42
// ……………… //

  temp = temp << shft;
  return temp;
 }

//Function for AMPM data
uint64_t ampm_fn( uint8_t ampm, uint8_t i,
uint8_t shft ){
//ampm = 0 means am, purbahno
 uint64_t temp = 0;

 if (ampm==0){ //AM
    if      (i==0)  temp =
0b0000000000000000010000000000000000;
    else if (i==1)  temp =
0b0000000000000001100000000000000000;
    else if (i==2)  temp =
0b0000000000000001100000000000000000;
    else if (i==3)  temp =
0b0111100111111111101111111111100000;
    else if (i==4)  temp =
0b1000010100000001001000100000000000;
    else if (i==5)  temp =
0b1110111100001110010011110000000000;
    else if (i==6)  temp =
0b0011001100110010010110001111000000;
    else if (i==7)  temp =
0b0010000100100010010000000100100000;
    else if (i==8)  temp =
0b0000000100010010010011110011000000;
    else if (i==9)  temp =
0b0000000100001010010000110000000000;
    else if(i==10)  temp =
0b0000000100000110010000000110000000;
    else if(i==11)  temp =
0b0000000100000010010000000110000000;
    else if(i==12)  temp =
0b0000001100000000000000000001011000;
    else if(i==13)  temp =
0b0000010001100000000000000000011000;

```

```
    else if(i==14)  temp = 0b000001111000000000000000000001000;
    else if(i==15)  temp = 0b000000000000000000000000000010000;

else{ //PM
    if      (i==0)  temp = 0b000000000000000000000000000000000000000000000000;
    else if (i==1)  temp = 0b000000000000000000000000000000000000000000000000;
    else if (i==2)  temp = 0b000000000000000000000000000000000000000000000000;
    else if (i==3)  temp = 0b111111111111001111001111111111110111111111100000;
    else if (i==4)  temp = 0b000011100010010000101000000010010001000000000000;
    else if (i==5)  temp = 0b010010010010011101111000001110010001110000000000;
    else if (i==6)  temp = 0b010011001010000110011000110010010010001110000000;
    else if (i==7)  temp = 0b010000001010000100001001000010010000001001000000;
    else if (i==8)  temp = 0b001000001010000000001000100010010011110011000000;
    else if (i==9)  temp = 0b000100011010000000001000011010010000110000000000;
    else if(i==10)  temp = 0b000011110110000000001001000110010000001100000000;
    else if(i==11)  temp = 0b000000000010000000001000100010010000000010000000;
    else if(i==12)  temp = 0b000000000000000000000000000000000000000001011000;
    else if(i==13)  temp = 0b000000000000000000000000000000000000000000011000;
    else if(i==14)  temp = 0b000000000000000000000000000000000000000000001000;
    else if(i==15)  temp = 0b000000000000000000000000000000000000000000010000;
 }
  temp = temp << shft;
  return temp;
}

//matrices for Bengali months
uint64_t bn_months( uint8_t bn_m, uint8_t i, uint8_t shft ){
uint64_t temp = 0;

 if (bn_m==1){ //Baishakh, size = 33
    if      (i==0)  temp = 0b000000000000000000000000000000000;
    else if (i==1)  temp = 0b111100000000000000000000000000000;
    else if (i==2)  temp = 0b000100000000000000000000000000000;
    else if (i==3)  temp = 0b001111111111111001101110101010010011;
    else if (i==4)  temp = 0b001000000001000110111001001111110;
    else if (i==5)  temp = 0b010000000111001111011001000000110;
    else if (i==6)  temp = 0b010000110010011110010010000011010;
    else if (i==7)  temp = 0b010000100010011110010010000100010;
    else if (i==8)  temp = 0b010000001101000000001001000011010;
    else if (i==9)  temp = 0b010000000011000000001001000001110;
    else if(i==10)  temp = 0b011110000001000000001001000000010;
    else if(i==11)  temp = 0b001110000001000000001001000000010;
    else if(i==12)  temp = 0b000000000000000000000000000000000;
    else if(i==13)  temp = 0b000000000000000000000000000000000;
    else if(i==14)  temp = 0b000000000000000000000000000000000;
    else if(i==15)  temp = 0b000000000000000000000000000000000;
 }
/// Folded due to being repititive ///
else if (bn_m==12){ // CHOITRO, size = 22
// ---------------------- //

  temp = temp << shft;
  return temp;
}

//The function to display digits
uint64_t digits_fn( uint8_t dig_it, uint8_t i, uint8_t shft ){
uint64_t temp=0;

 if (dig_it==0){
  // ZERO
    if      (i==2)  temp = 0b00000000;
    else if (i==3)  temp = 0b00000000;
    else if (i==4)  temp = 0b00111100;
    else if (i==5)  temp = 0b01000010;
    else if (i==6)  temp = 0b10000001;
    else if (i==7)  temp = 0b10000001;
    else if (i==8)  temp = 0b10000001;
    else if (i==9)  temp = 0b10000001;
    else if(i==10)  temp = 0b01000010;
    else if(i==11)  temp = 0b00111100;
    else if(i==12)  temp = 0b00000000;
    else if(i==13)  temp = 0b00000000;
}

/// Folded due to being repetitive ///
else if (dig_it==9){ // NINE
// -------------- //

else if (dig_it==10){
  //SPECIAL CONDITION FOR PRINTING COMMA
    if      (i==12)  temp = 0b011;
    else if (i==13)  temp = 0b011;
    else if (i==14)  temp = 0b001;
    else if (i==15)  temp = 0b010;}

  temp = temp << shft;
  return temp;
}

//The function to dispay week days
uint32_t week_fn( uint8_t wk_day, uint8_t i, uint8_t shft ){

 uint32_t temp = 0;

 if (wk_day==1){
  // SAT, size = 23
    if      (i==0)  temp = 0b00000000000111111110000;
    else if (i==1)  temp = 0b00000000001000000001000;
    else if (i==2)  temp = 0b00000000001000000000000;
    else if (i==3)  temp = 0b11011101111111111111100;
    else if (i==4)  temp = 0b00110111001000000001000;
    else if (i==5)  temp = 0b01111011001000000001000;
    else if (i==6)  temp = 0b11111001001000111001000;
    else if (i==7)  temp = 0b11011001001001110101000;
    else if (i==8)  temp = 0b00000001001000110011000;
    else if (i==9)  temp = 0b00000001001000000011000;
    else if(i==10)  temp = 0b00000001001000000001000;
    else if(i==11)  temp = 0b00000001001000000001000;
    else if(i==12)  temp = 0b00000000000000000000011;
    else if(i==13)  temp = 0b00000000000000000000001;
    else if(i==14)  temp = 0b00000000000000000000001;
    else if(i==15)  temp = 0b00000000000000000000010;
 }
```

```
/// Folded due to being repetitive ///
else if (wk_day==7){ //FRIDAY
// ------------------- //

  temp = temp << shft;
  return temp;
 }

/// /Let there be Light!!! /////
void populate_cells()
{
  //ampm size: 34 if ampm=0 (AM), 48 if ampm=1 (PM)
  uint8_t ampm_size = 34 + 14*ampm;

  //Arrays for display
  uint64_t time_mat = 0;
  uint64_t ampm_mat = 0;
  uint64_t ban_dy_mat = 0; //size 64
  uint64_t ban_m_mat = 0; //size 64
  uint64_t eng_dy_mat = 0; //size 64
  uint64_t eng_m_mat = 0; //size 64
  uint32_t week_mat = 0; //size 32

  uint8_t week_size = wk_size[week] + 3; //3 spaces after week

  /// Disp size is the length of the scrolling text with ampm, week day and dates
  uint32_t disp_size = ampm_size + week_size + d_size + bn_m_size[bn_m] + y_size + 3 + 3 + d_size + en_m_size[en_m] + y_size + 6;

  for(int i=0;i<16;i++){

      //AMPM
      ampm_mat = ampm_fn( ampm, i, (64 - ampm_size) );

      //MONTHS
      eng_m_mat = en_months( en_m, i, (64 - en_m_size[en_m] - 0));
      ban_m_mat = bn_months( bn_m, i, (64 - bn_m_size[bn_m] - 0));
      week_mat  =   week_fn( week, i, (32 - wk_size[week]) );

      eng_dy_mat=0;
      ban_dy_mat=0;
      time_mat=0;

      //TIME DOTS
      if((i==4 || i==5 || i==10 || i==11) && second_loop<6 ){
        time_mat = 1;
        time_mat = time_mat << (64 - 2*N1 - 3);
      }

      if (i>1 && i<14){

        //TIME
        time_mat = time_mat | digits_fn( tim_e[0], i, (64 - N1) );
        time_mat = time_mat | digits_fn( tim_e[1], i, (64 - 2*N1 - 1) );
        time_mat = time_mat | digits_fn( tim_e[2], i, (64 - 3*N1 - 4) );
        time_mat = time_mat | digits_fn( tim_e[3], i, (64 - 4*N1 - 5) );

        //BENGALI DATE
        ban_dy_mat = digits_fn( bn_date[0], i, (64 - (N1 + 0)) );
        ban_dy_mat = ban_dy_mat | digits_fn( bn_date[1], i, (64 - (N1 + N1 + 1)) );

        //BENGALI YEAR
        ban_dy_mat = ban_dy_mat | digits_fn( bn_year[0], i, (64 - (N1 + 2*N1 + 5)) );
        ban_dy_mat = ban_dy_mat | digits_fn( bn_year[1], i, (64 - (N1 + 3*N1 + 6)) );
        ban_dy_mat = ban_dy_mat | digits_fn( bn_year[2], i, (64 - (N1 + 4*N1 + 7)) );
        ban_dy_mat = ban_dy_mat | digits_fn( bn_year[3], i, (64 - (N1 + 5*N1 + 8)) );
        //ENGLISH DATE
        eng_dy_mat = digits_fn( en_date[0], i, (64 - (N1 + 0)) );
        eng_dy_mat = eng_dy_mat | digits_fn( en_date[1], i, (64 - (N1 + N1 + 1)) );

        //ENGLISH YEAR
        eng_dy_mat = eng_dy_mat | digits_fn( en_year[0], i, (64 - (N1 + 2*N1 + 5)) );
        eng_dy_mat = eng_dy_mat | digits_fn( en_year[1], i, (64 - (N1 + 3*N1 + 6)) );
        eng_dy_mat = eng_dy_mat | digits_fn( en_year[2], i, (64 - (N1 + 4*N1 + 7)) );
        eng_dy_mat = eng_dy_mat | digits_fn( en_year[3], i, (64 - (N1 + 5*N1 + 8)) );
      }

      //COMMA AFTER BENGALI YEAR
      if(i>10) ban_dy_mat = ban_dy_mat | digits_fn( 10, i, (64 - (N1 + 5*N1 + 8 + 3)) );


    //The seperating Line
    uint8_t offset = 39;
    dmd.setPixel(offset-1,i,GRAPHICS_ON);

    for(int j=0;j<38;j++)
    {
      //TIME
      if ( (time_mat >> (63 - j)) & 0b1 )
        dmd.setPixel(j,i,GRAPHICS_ON);
      else
        dmd.setPixel(j,i,GRAPHICS_OFF);

//// SCROLLING DATE AND TIME /////
      if (j<25){
      uint32_t col_ind = (j+lo_op)%disp_size;
      uint8_t flag = 0;

      //disp_size = ampm_size + week_size + d_size + bn_m_size[bn_m] + y_size + 3 + 3 + d_size + en_m_size[en_m] + y_size + 6;


      if (col_ind < ampm_size)
        flag = (ampm_mat >> (63-col_ind) ) & 0b1;

      //WEEK DAY DISPLAY
      else if (col_ind < ampm_size + week_size)
        flag = (week_mat >> (32 - col_ind + ampm_size)) & 0b1;

      //BANGLA DATE AND TIME
      else if (col_ind < (ampm_size + week_size + d_size))
        flag = (ban_dy_mat >> (63-col_ind +ampm_size + week_size) ) & 0b1;

      else if (col_ind < (ampm_size + week_size + d_size + bn_m_size[bn_m]))
        flag = (ban_m_mat >> (63-col_ind +ampm_size + week_size +d_size) ) & 0b1;

      else if (col_ind < (ampm_size + week_size + d_size + bn_m_size[bn_m]+y_size+3))
        flag = (ban_dy_mat >> (63-col_ind +ampm_size + week_size +bn_m_size[bn_m]) ) & 0b1;

      //ENGLISH DATE AND TIME
      else if (col_ind < (ampm_size + week_size + d_size + bn_m_size[bn_m]+y_size+6+d_size))
        flag = (eng_dy_mat >> (63-col_ind +ampm_size + week_size + d_size + bn_m_size[bn_m]+y_size+6) ) & 0b1;

      else if (col_ind < (ampm_size + week_size + d_size + bn_m_size[bn_m]+y_size+6+d_size + en_m_size[en_m]))
```

```
        flag = (eng_m_mat >> (63-col_ind +ampm_size + week_size + d_size +
bn_m_size[bn_m]+y_size+6+d_size) ) & 0b1;

      else if (col_ind < (ampm_size + week_size + d_size + bn_m_size[bn_m]+y_size+6+d_size +
en_m_size[en_m] + y_size + 6))
        flag = (eng_dy_mat >> (63-col_ind +ampm_size + week_size + d_size +
bn_m_size[bn_m]+y_size+6+en_m_size[en_m]) ) & 0b1;

      if (flag)
        dmd.setPixel(j+offset,i,GRAPHICS_ON);
      else
        dmd.setPixel(j+offset,i,GRAPHICS_OFF);
      }
    }


  }

//Now wait and scroll
delay(100); // Increase for slower scrolling
second_loop = (second_loop + 1)%10;
lo_op = (lo_op+1)%disp_size;
}
```

(Some parts of the code, such as the binary patterns of different months, are repetitive; and have been omitted for convenience. Please visit the GitHub repository for the complete code)

4 Implementation

4.1 Description

The Arduino Nano acts as the main control unit and the other three devices are connected to it via appropriate pins. The ESP module connects to the specified Wi-Fi signal mentioned in the code, then sends an API request to fetch the appropriate time and dates from the internet. Afterwards, the fetched value is stored in RTC module's clock (English time and date) and EEPROM (Bangla date).

API requests:

Time:
https://api.thingspeak.com/apps/thinghttp/send_request?api_key=Q1ZBWRBEXXO44K68
English Date
https://api.thingspeak.com/apps/thinghttp/send_request?api_key=6EX7E9O58AOAB9DG
Bangla Date
https://api.thingspeak.com/apps/thinghttp/send_request?api_key=5MKIPC0USLFNP3UT

In case there is no Wi-Fi signal, the system tries to connect to the Wi-Fi for about 1:30 minutes (adjustable), before loading the latest data from the previous sync and showing them in the LED matrix. The system will sync with the internet twice each day at 6 am and 6 pm. The logic behind 6 am is that, Bangla date changes during dawn (not midnight). An update at 12 am (midnight) is not needed, as RTC module can change the English date by itself.

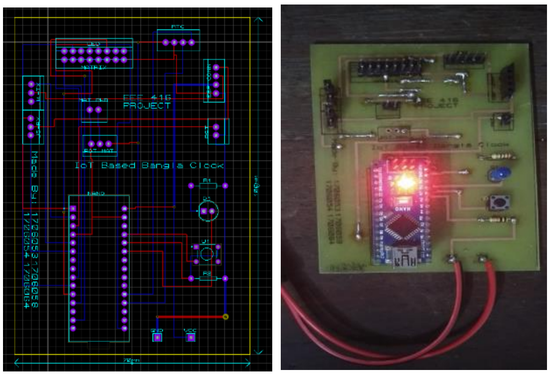


Figure 2: (Left) PCB Layout and (Right) Implementation of Design

Some challenges we had to overcome are-

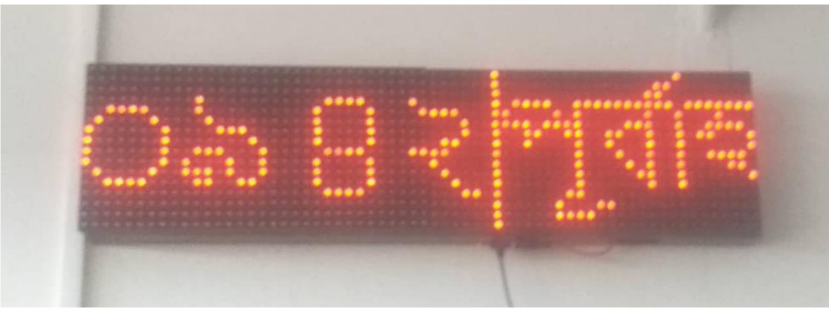
1) Initially, the binary patterns for displaying months and weeks were stored in integer matrices. However, it consumed too much of the allocated 2 kB memory for global variables, so we had to use the 32 kB program memory instead by converting the matrices into functions implementing if-else statements
2) The RTC module was conflicting with DMD library driving the LED panels. We had to turn

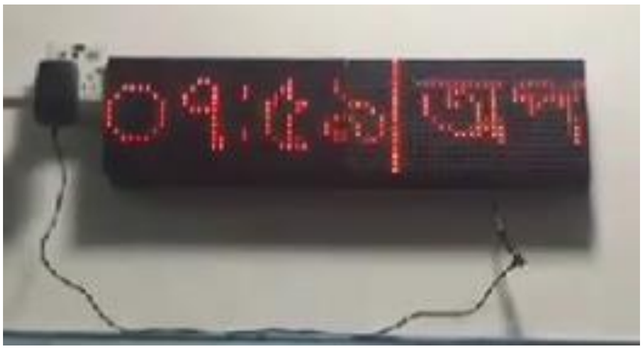
OFF DMD while taking reading from RTC, which resulted in a lot of flickering. As it is necessary for the LED panel to run continuously while we only need to take time Reading each minute, we had to work around this problem by using Arduino’s internal millisecond count to only take readings from RTC at the beginning of each minute.

4.2 Results

When first powered up, our LED display first connects to the internet, fetching the time and the appropriate dates and stores then in appropriate variables and updates the RTC module. Then, the data is shown using the LED matrix panel, while syncing the time using RTC’s onboard clock every minute.

Here, we can see two display styles of the matrix, one showing “AM” and the other “PM”





4.3 GitHub Link

https://github.com/Sadat3366/EEE-416---Bangla-clock---Group-A2.2

4.4 YouTube Link

https://youtu.be/6aywOkfDy6A

5 Design Analysis and Evaluation

5.1 Novelty

Very few attempts have been made in order to design a digital clock that keeps track of both the Bangla and English calendars simultaneously. Although there are some small-scale projects that are available on the internet, they only display the English calendar in Bangla letterings. Ours is probably the first Bangla digital clock that follow the Bangla calendar and the English calendar as well. It can also sync the time with the internet in case a power outage happens, this removes the hassle of manually adjusting the time.

5.2 Project Management and Cost Analysis

5.1.1 Bill of Materials

Name of Product	Unit Cost (Tk)	Amount	Total Cost (Tk)
P10 LED Matrix (32x16)	700	2	1400
Arduino Nano	720	1	720
RTC Module (DS1302)	190	1	190
ESP-8266 ESP-01 WiFi Module	200	1	200
ESP-01 Shield	110	1	110
9V Adapter and DC Jack	(120+30) = 150	1	150
Custom PCB Board	300	1	300
Soldering, Wires, Connectors and Headers	50	1	50
Metal Frame and Cover	(0+30) = 30	1	30
Total Cost	-	-	**3150**

5.1.2 Calculation of Per Unit Cost of Prototype

Apart from the Cost of Bulk Materials, the prototyping cost also includes the price of online delivery charges, and the price of a few other modules and components used during initial prototyping and breadboard implementation. So, per unit cost of prototype = 3600 BDT (approx.)

5.1.3 Calculation of Per Unit Cost of Mass-Produced Unit

Per unit cost of each mass produced unit = 3150 BDT (approx.)

5.1.4 Timeline of Project Implementation

We started working on our project around the beginning of June, and the final version of our completed clock was available on 20 August. So, approximately it took us two and a half months to finish the whole project. We had worked on it mainly on our weekends due to academic pressure during the other days of the week.

5.2 Practical Considerations of the Design to Address Public Health and Safety, Environment, Cultural, and Societal Needs

5.2.1 Considerations to public health and safety

The clock is enclosed safely in a metal casing that is welded in order to ensure maximum protection and a secure mounting mechanism is in place, so that the clock hangs from the wall securely without the risk of falling. Hence, making the device a very safe one.

5.2.2 Considerations to environment

No pollutant was reported

5.2.3 Considerations to cultural and societal needs

Bangla Calendar Clock represents and revives our mother language "Bangla". It will hopefully help us to keep track of Bangla month and dates that we almost forget to use. We hardly can say which month or date today is according to the Bangla calendar. Now, we can just look at the wall and stay updated with this neglected aspect of our tradition.

5.3 Assessment of the Impact of the Project on Societal, Health, Safety, Legal and Cultural Issues

5.3.1 Assessment of Societal Issues

No social issues faced

5.3.2 Assessment of Health and Safety Issues

No health issues were reported at any stage of the product development, and no health issues were identified while using the device.

5.3.3 Assessment of Legal Issues

Our project has no patent infringements or copyright violations. Hence, we are safe in this regard.

5.3.4 Assessment of Cultural Issues

Our project promotes awareness about the Bengali culture and heritage, and has no issues regarding misrepresentation or appropriation of culture.

5.4 Evaluation of the Sustainability the and Impact of the Designed Solution in the Societal and Environmental Contexts

5.4.1 Evaluation of Sustainability

We've developed the model with sustainability in mind. We wanted the clock to be reliable and strong. The metal casing is very sturdy and provides robust housing for the electrical parts inside. The module runs from the socket, therefore, there is no need for constantly changing batteries.

5.4.2 Evaluation of Impact of Design in Societal Context

The design is kept simple and compact using dark color and not too flashy. Similar clocks are very common in our local mosques and offices. So, it can be used in our societal setting with ease.

5.4.3 Evaluation of Impact of Design in Environmental Context

Since we do not have to use large external batteries and have opted for long lasting materials, detrimental effects to the environment are kept minimal.

6 Reflection on Individual and Team work:

6.1 Individual Contribution of Each Member

- **Saleh Ahmed Khan - 1706053**
    - Establishing the WiFi link between Arduino and the Internet using ESP module, generating and maintaining the API requests for date and time in ThingSpeak website.
    - Initial Wiring and testing connections using an LCD display
    - Leading PCB designer using Proteus

- **Tiasa Mondal - 1706054**
    - Establishing communication between RTC module and the LED matrix, ensuring conflict free communication between the two modules.
    - Generating binary Patterns for English months
    - Initial hardware assembly

- **Abdullah Jubair Bin Iqbal - 1706058**

    - String Parsing to convert RTC and ESP module output to date and time integers for using in the display portion of the code
    - Generating binary Patterns for Bangla months
    - EEPROM implementation for storing Bengali date

- **Sadat Tahmeed Azad - 1706064**
    - Generating binary patterns for Digits, Week days and others during the test phase
    - Using the patterns to display custom Bangla Fonts in the LED panel. Displaying static and scrolling texts in runtime and incorporating display code with the rest
    - Frame design idea and management

6.2 Mode of Teamwork

Though all of us divided our work primarily, we helped one another in other parts time to time to quicken the work and to keep the overall idea about our project. Saleh has always been the most focused member of the team, always pushing everyone to finish their tasks, after doing the heavy lifting himself. Tiasa has been immensely helpful in terms of debugging, searching for the appropriate parts, making sure that the hardware components are proper. Jubair has always been about efficiency and making the best use of everyone’s time, always finding quicker and faster ways to solve any problem. Sadat has been the most approachable person of the group. Whenever there was an issue, we could all just reach out to him, and he would help us find a solution. Overall, it has been an amazing experience working together as a productive team.

6.3 Diversity Statement of the Team

We have team members from different cultural and religious backgrounds. Our female team member Tiasa Mondal is based in Dhaka, while Sadat Tahmeed had his early education in suburban areas. The other members Abdullah Jubair and Saleh Ahmed Khan also spent their whole childhood in Dhaka. In our group, Tiasa is the extrovert, while Sadat, Saleh and Jubair were introverts. It has been quite a work experience for all of us. We have had our disagreements during the project work. Through collaboration and friendly interaction, we came to understand and settle our differences and find a common ground for successful teamwork.

6.4 Log Book of Project Implementation

Date	Milestone achieved	Individual Role	Team Role	Comments
9/6/22	Bengali & English months matrix formation	Sadat Tahmeed Azad generated custom Bangla Fonts in the LED display and displaying technique static and scrolling texts in runtime.	Months and times were divided among 4 members	Successfully done!
11/6/22	ESP-01 wifi module connection with Arduino and storage and date and time track using RTC and EEPROM in case of power cut	Saleh Ahmed Khan established the Wi-Fi link between Arduino and the Internet using ESP module. Jubair Adib converted the date and time strings obtained from online into proper integer variables to be used in the display code module designed by Sadat	Modified the code together	Successfully done
18/6/22	Working around module conflict	Tiasa Mondal figured out a way to establish communication between RTC module and the LED matrix, ensuring conflict free incorporation of the two modules.	We had to put our heads together to resolve the issues	Running well enough
10/7/22	More issues resolved	Saleh resolved some issues related to flickering	Saleh and Sadat worked on the flickering issue	Satisfactory performance

		Jubair incorporated EEPROM to use as backup storage for Bengali date in case Wi-Fi is unavailable	while Tiasa accompanied Jubair on EEPROM implementation	
15/7/22	Successful trial text display on LED display	Tiasa Mondal, Saleh Ahmed Khan, Sadat Tahmeed Azad, Jubair Adib combined their parts to run the main code	All members solved error issues together	Successfully done
5/8/22	PCB design done	PCB design was divided among all members, Saleh doing most of the work	All members checked different parts of the design and solved errors	Faced many issues but end of the day it worked out
20/8/22	Final clock framing	Everybody gathered some sort of information about wall clock framing.	Checked whether it will look proper or not	Feels good to see it all together!

7 References

YouTube tutorials

1. [ESP8266] How to start with ESP-Connection with PC and checking functionality with AT commands
2. *how to use p10 led text display with arduino / p10 board with arduino / arduino p10 scrolling text*
3. How to Fetch data from any website using ESP8266 | IoT Projects
4. How to connect Arduino UNO to a WIFI Network: ESP8266 - AT Commands