

EEE 416 (July 2023)

Microprocessor and Embedded Systems Laboratory

Final Project Report

Section: C2 Group: 06

Development of IoT based Bangla Calendar Clock 3.0

Course Instructors:

Dr. Sajid Muhaimin Choudhury, Assistant Professor
Ayan Biswas Pranta, Part-Time Lecturer

Signature of Instructor: _____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. 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."

Signature: _____



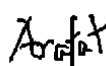
Full Name: Ameer Hamja Ibne Jamal
Student ID: 1906177

Signature: _____



Full Name: Fahim Shahriar Anim
Student ID: 1906178

Signature: _____



Full Name: Yeasir Arafat Prodhan
Student ID: 1906191

Signature: _____



Full Name: U Mong Sain Chak
Student ID: 1906195

Table of Contents

| | | |
|----------|--|-----------|
| 1 | Abstract..... | 1 |
| 2 | Introduction | 1 |
| 3 | Design | 2 |
| 3.1 | Problem Formulation | 2 |
| 3.1.1 | Formulation of Problem..... | 2 |
| 3.1.2 | Analysis | 2 |
| 3.2 | Design Method (PO(a))..... | 3 |
| 3.3 | Circuit Diagram | 5 |
| 3.4 | Simulation..... | 6 |
| 3.5 | Full Source Code of Firmware..... | 8 |
| 4 | Implementation | 26 |
| 4.1 | Description..... | 26 |
| 5 | Design Analysis and Evaluation | 27 |
| 5.1 | Novelty..... | 27 |
| 5.2 | Design Considerations | 28 |
| 5.2.1 | Considerations to public health and safety | 28 |
| 5.2.2 | Considerations to environment | 28 |
| 5.2.3 | Considerations to cultural and societal needs | 28 |
| 5.3 | Limitations of Tools..... | 28 |
| 5.4 | Impact Assessment (PO(f))..... | 29 |
| 5.4.1 | Assessment of Societal and Cultural Issues | 29 |
| 5.4.2 | Assessment of Health and Safety Issues | 29 |
| 5.5 | Sustainability Evaluation (PO(g))..... | 29 |
| 5.5.1 | Assessment of Societal and Cultural Issues | 29 |
| 5.5.2 | Assessment of Health and Safety Issues | 29 |
| 6 | Reflection on Individual and Team work (PO(i))..... | 30 |
| 6.1 | Individual Contribution of Each Member | 30 |
| 6.2 | Mode of TeamWork and Diversity | 30 |
| 7 | Communication to External Stakeholders | 31 |
| 7.1 | Executive Summary..... | 31 |
| 7.2 | User Manual..... | 31 |
| 7.3 | Github Link..... | 31 |

| | | |
|-----------|--|-----------|
| 8 | Project Management and Cost Analysis (PO(k))..... | 31 |
| 8.1 | Bill of Materials | 31 |
| 8.2 | Timeline of Project Implementation | 32 |
| 9 | Future Work..... | 32 |
| 10 | References | 33 |

1 Abstract

The basic goal of our project is to incorporate *Time, Weekdays and Bangla Dates* into a unique hardware clock with efficient power management system. The time and dates will be automatically synchronized with the standard time over the internet at regular intervals with the help of a Wi-fi module, since the time synchronization is based on the idea of the Internet of Things. Multiple LED matrices are used as the calendar clock display and a microcontroller is used as the main processing unit. Additionally, a Real Time Clock module ensures that the clock operates in offline mode.

2 Introduction

The Internet of Things (IoT) has transformed various sectors, including agriculture, smart homes, and urban management, by enabling seamless connectivity and data exchange among devices. In agriculture, IoT devices monitor and manage environmental conditions for improved crop yields and resource efficiency. In smart homes, IoT technology allows for remote control of household devices, enhancing comfort and energy efficiency. In urban management, IoT sensors gather data for optimizing city services and infrastructure. Accurate time synchronization is essential not only for IoT ecosystems to function effectively, but also for human users. In large networks, it's important to provide timekeeping devices in native languages to ensure accessibility for all users. As technology advances and expands globally, there's a growing demand for localized interfaces and device compatibility with native languages.

Bengali language has a heritage of following its own calendar since ancient times. There have been several attempts to design unique Bangla fonts for digital clock displays. Several eminent researchers attempted to incorporate a font into segmented displays; nevertheless, the displays were far too complicated. Their attempt to enhance the design by utilizing an 8x8 dot matrix display turned out to be successful to depict Bangla digits but lacked the adaptability of displaying the complete Bangla date.

In this project, we present a further enhanced cost-effective device large enough to accommodate-

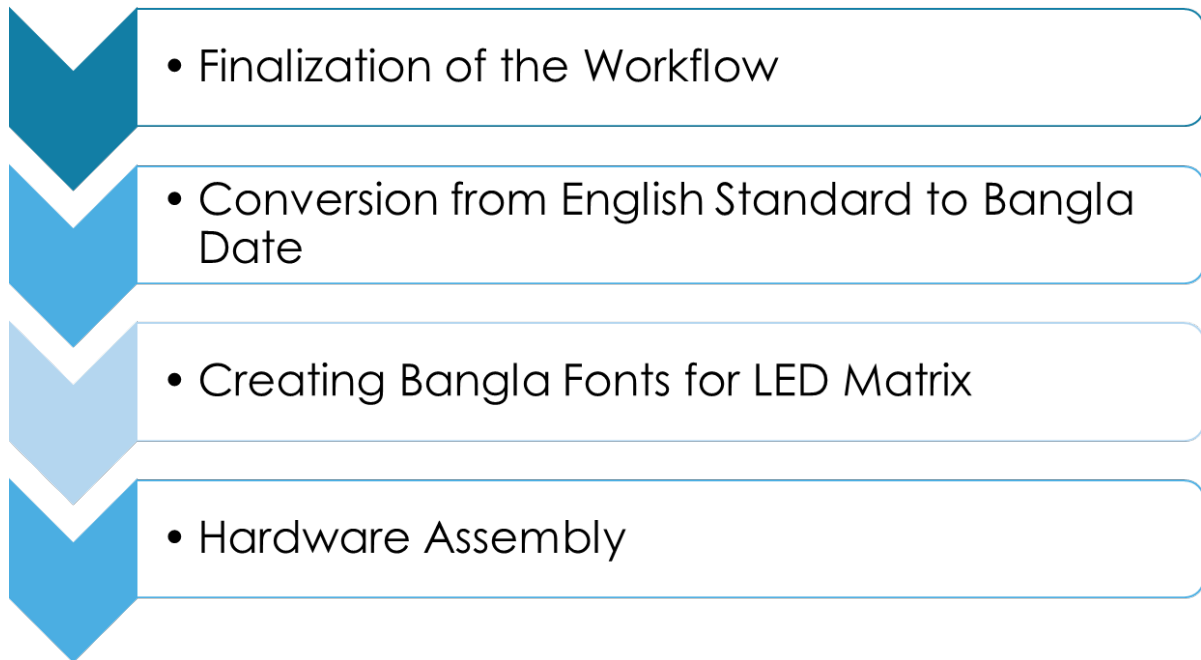


without using any scroll / fade with advanced power management system with a personalized Bangla font.

3 Design

3.1 Problem Formulation

3.1.1 Formulation of Problem



3.1.2 Analysis

Working Principle:

Scans and connects to an available open Wi-Fi network or connect to a secure network with appropriate credentials.

Sends API request for the English Time and Date and records the data.

Displays the Appropriate Datas on the LED Matrix.

Periodically refreshes the display with new time and date from the RTC module and syncs with the Internet Provided Date and Time.

In case of a Failure to Connect with the Wi-Fi Signal or to receive the requested data due to network unavailability, the controller fetches the previously recorded data from the RTC module and displays it on the LED matrix until the next sync event.

3.2 Design Method (PO(a))

Microcontroller:



Arduino Uno R3

The Arduino Uno R3 Standard-Quality Edition

- Micro-controller : ATmega328.
- Operating Voltage : 5V.
- Input Voltage (recommended) : 7-12V.
- Digital I/O Pins : 14 (of which 6 provide PWM output).
- Analog Input Pins : 6

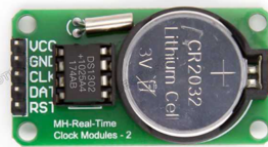
The microcontroller, Arduino UNO is the main component of the project communicating with the Wi-fi and RTC module, ensuring the Calendar and Clock data is synchronized.

Other Components:



MAX7219 4 Channel Dot Matrix Display Panel

- Driver Chip: MAX7219
- Input Voltage: 3.7 to 5.3 V
- Input Current: 320 mA
- Display Dimensions: 32x32x6 (LxWxH) mm



DS1302 Real Time Clock Module

- The DS1302 trickle-charge timekeeping chip module with battery backup
- A real-time clock/calendar and 31 bytes of static RAM.
- Communicates with a microprocessor via a simple serial interface.
- Real-time clock/calendar provides seconds, minutes, hours, day, date, month, and year information.

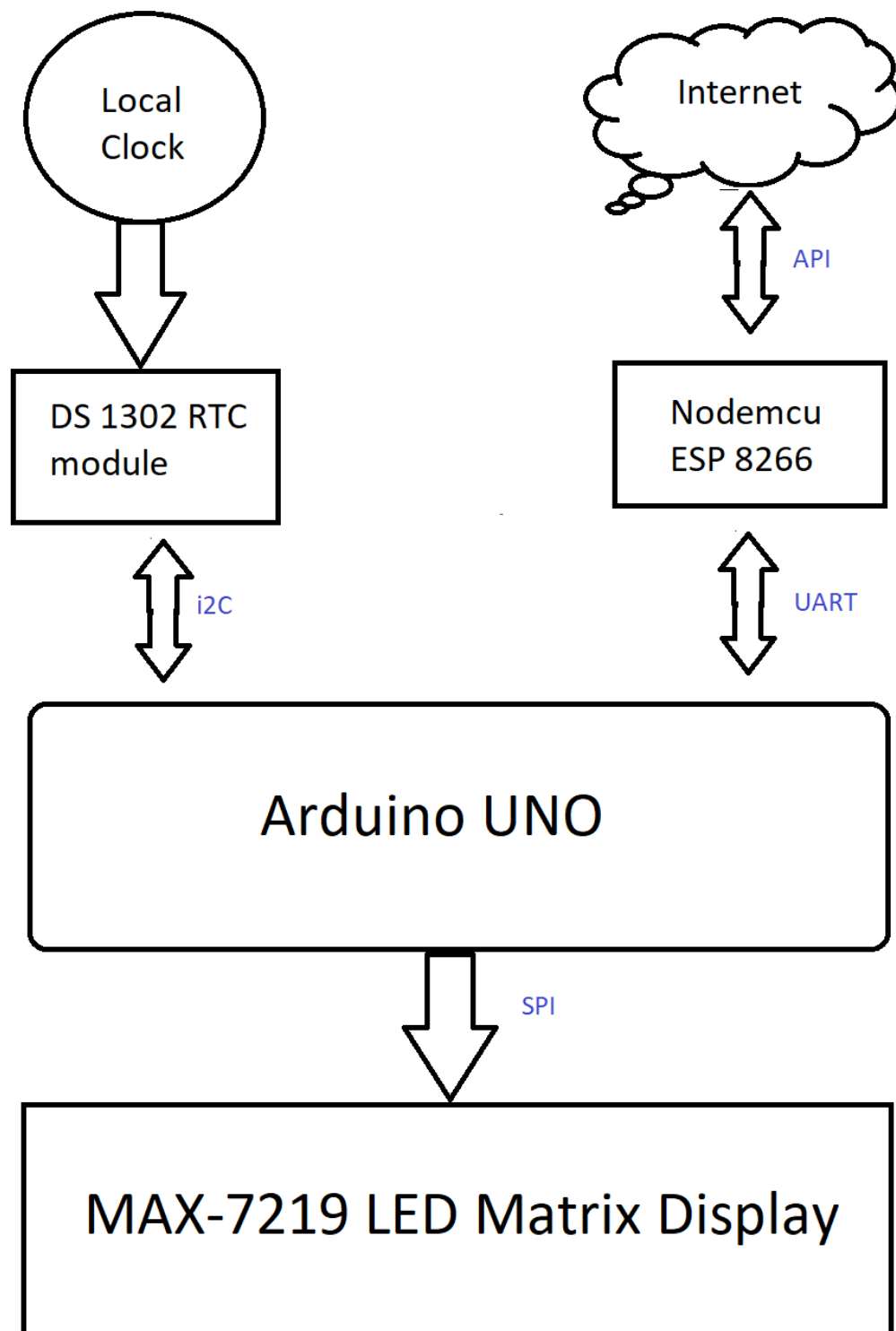


ESP8266 CH340 NodeMCU Wifi Module Lua V3

- Power input: 4.5V - 9V (10VMAX), USB-powered
- Transfer rate: 110-460800bps
- Support UART / GPIO data communication interface
- Support Smart Link Smart Networking
- Working temperature: -40°C ~ +125°C
- Drive Type: Dual high-power H-bridge
- Don't need to download resetting
- A great set of tools to develop ESP8266
- Flash size: 4MByte
- Lowest cost Wi-Fi

- The MAX 7219 dot matrix display panel is used as our main display. Here four 8x8 dot matrices are cascaded together as a single unit.
- The DS1302 is the Real Time Clock module, which is used to show real time when Wi-fi module is idle.
- The ESP 8266 NodeMCU module is the Wi-fi module which synchronizes the time and updates the RTC module at regular intervals (in our case 1 hour).

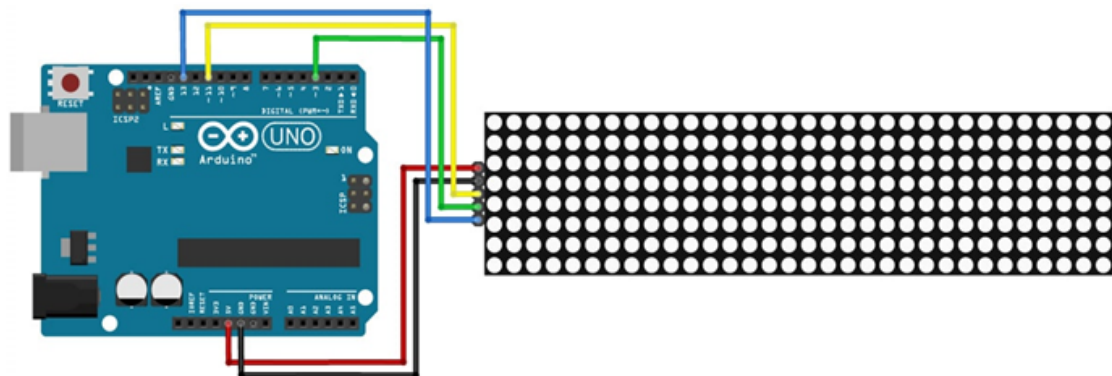
Block Diagram:



3.3 Circuit Diagram

MAX7219 LED dot matrix display connections

| MAX7219 Display | Arduino |
|-----------------|-----------|
| VCC | 5 V |
| GND | GND |
| DIN | 11 (MOSI) |
| CS | 3 (SS) |
| CLK | 13 (SCK) |



DS 1302 Real Time Clock Module

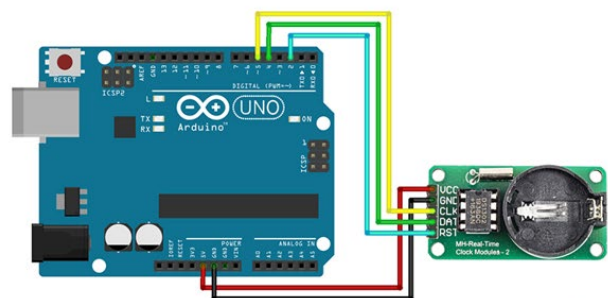
POWER
GND
Clock
Data
Reset



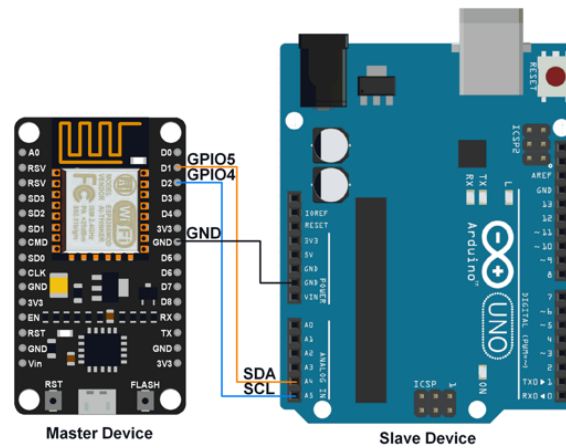
DS1302 RTC Module Pinout

This module has 5 pins:

- VCC: Module power supply – 5V
- GND: Ground
- CLK: Clock pin
- DAT: Data pin
- RST: Reset (Must be HIGH for active mode / Active High)



NODEMCU ESP 8266 with Arduino UNO

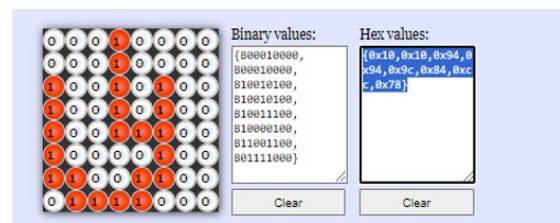
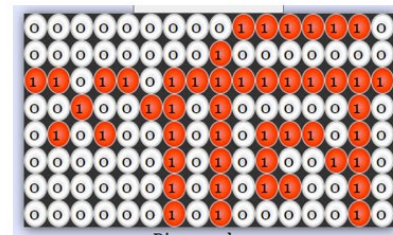
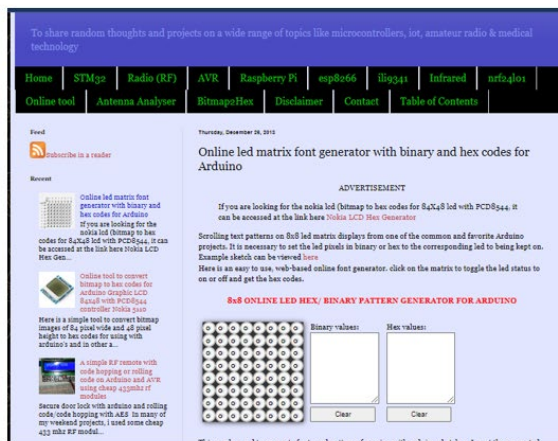


Finally, we combined the whole system as a package.

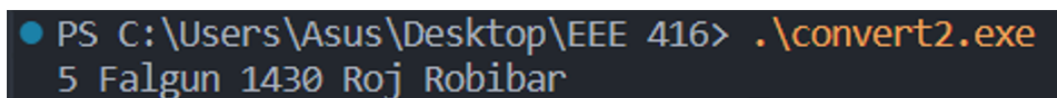
3.4 Simulation

We created customized Bangla digits and phrases using a website. In the website we designed them by turning on/off suitable LED's for desired characters/digits and with the help of HEX values provided, we successfully demonstrated the time and date.

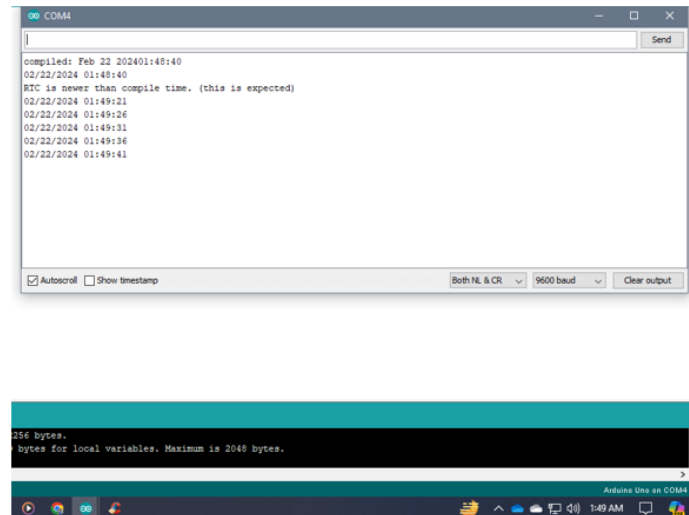
Online LED Matrix Font Generator



Mapping to Bangla Year System



Interfacing DS 1302 Module



Inside conv.h: A Custom Header File for Date Conversion

```
22
23 int EnToBnYear(int year, int month, int date) {
24     int banglaYear = year - 594; // 2017(Gregorian Year) - 594 = 1423(Bangla Year)
25     if ((month > lastMonthIndex) || (month == lastMonthIndex && date > 13)) {
26         banglaYear += 1;
27     }
28     return banglaYear;
29 }
30
31 struct Bdate EnglishToBangla(int gDay, int gMonth, int gYear)
32 {
33     gMonth--;
34     struct Bdate retval;
35     retval.banglaYear = EnToBnYear(gYear, gMonth, gDay);
36
37     int monthDays = totalMonthDays[gMonth];
38     if (gDay <= midMonthDate[gMonth])
39     {
40         if (gMonth == leapYearIndex && isLeapYear(gYear))
41         {
42             monthDays = totalMonthDays[gMonth] + 1;
43         }
44         retval.banglaDate = monthDays + gDay - midMonthDate[gMonth];
45         retval.banglaMonthIndex = gMonth;
46     }
47     else
48     {
49         retval.banglaDate = gDay - midMonthDate[gMonth];
50         retval.banglaMonthIndex = (gMonth + 1) % 12; // banglaMonthsList is 0-based indexed
51     }
52 }
```

This file was made keeping in mind regarding the complications in Bangla Year System due to the Leap Year. The extra day in a Leap year was added in the month of Falgun (according to the recent updates provided by Bangla Academy).

3.5 Full Source Code of Firmware

```
#include <MD_MAX72xx.h>
#include <SPI.h>
#include <RtcDS1302.h>
#include "conv.h"

#include <SoftwareSerial.h>

const byte rxPin = 4;
const byte txPin = 5;

// Set up a new SoftwareSerial object
SoftwareSerial mySerial (rxPin, txPin);

// RTC CONNECTIONS:
// DS1302 CLK/SCLK --> 7
// DS1302 DAT/IO --> 6
// DS1302 RST/CE --> 8
// DS1302 VCC --> 3.3v - 5v
// DS1302 GND --> GND

ThreeWire myWire(6,7,8); // IO, SCLK, CE
RtcDS1302<ThreeWire> Rtc(myWire);

//LED_Matrix SPI Arduino UNO
// CLK Pin > 13 SCK
// Data Pin > 11 MOSI
#define CS_PIN 3
#define delay_t 50 // in milliseconds

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 16
// Hardware SPI connection
MD_MAX72XX mx = MD_MAX72XX(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);

//defining Bangla Characters

//byte zero[8] = {0x30,0x78,0xcc,0xcc,0xcc,0xcc,0x78,0x30};
byte zero[8] = {0x30,0x48,0x84,0x84,0x84,0x84,0x48,0x30};
//byte one[8] = {0x40,0x60,0x30,0x18,0x0c,0xcc,0xd8,0x70};
byte one[8] = {0x20,0x10,0x08,0x04,0x04,0x64,0x68,0x30};
byte two[8] = {0x10,0x10,0x08,0x04,0x44,0x38,0x10,0x0c};
//byte three[8] = {0x30,0xb8,0xb4,0x84,0x84,0xc4,0x6c,0x38};
byte three[8] = {0x10,0x38,0xb4,0x84,0x84,0x84,0x48,0x30};
byte four[8] = {0x38,0x44,0x44,0x38,0x44,0x44,0x44,0x38};
byte five[8] = {0x44,0x7c,0x48,0x48,0x4a,0x4c,0x24,0x18};
byte six[8] = {0x10,0x10,0x94,0x94,0x9c,0x84,0xcc,0x78};
byte seven[8] = {0x30,0x78,0xc8,0xc8,0x78,0x08,0x08,0x08};
byte eight[8] = {0xc4,0x44,0x48,0x70,0x58,0x58,0x58,0x70};
byte nine[8] = {0x60,0x30,0x1c,0x04,0x74,0xd4,0x5c,0x10};

byte shoni_0[8]={0x00,0x00,0xdb,0x26,0x52,0x02,0x02,0x02};
byte shoni_1[8]={0x7e,0x80,0xff,0x82,0xba,0xa6,0xb2,0x82};
byte robi_0[8]={0x00,0x00,0xff,0x1c,0x24,0x64,0x14,0x2c};
byte robi_1[8]={0x7f,0x80,0xff,0x8e,0x92,0xb2,0x8a,0x86};
byte shom_0[8]={0xff,0x51,0x89,0x85,0x95,0x9b,0x81,0x61};
byte shom_1[8]={0xff,0x51,0x49,0x45,0x5d,0x5d,0x5b,0x41};
byte mongol_0[8]={0xf9,0x8a,0x4a,0x28,0xe9,0xaa,0xd8,0x08};
byte mongol_1[8]={0x11,0xa1,0xc1,0x95,0xab,0xa1,0x81,0x71};
byte budh_0[8]={0x00,0xff,0x0e,0x12,0x22,0x12,0x0f,0x06};
byte budh_1[8]={0x18,0x20,0x1e,0x34,0x44,0x64,0x14,0x8c};
byte brihoshpoti_0[8]={0xff,0x1c,0x24,0x44,0x34,0x0c,0x10,0x0e};
byte brihoshpoti_1[8]={0xfc,0x70,0x8b,0x8b,0x30,0xc3,0x23,0x18};
byte shukro_0[8]={0x0f,0xa8,0x58,0xa8,0x08,0x08,0x1c,0x1a};
byte shukro_1[8]={0xff,0x1c,0x2a,0x49,0xcb,0x28,0x98,0x78};

byte boishakh_0[8]={0x04,0x06,0x01,0x01,0x02,0x02,0x02,0x01};
byte boishakh_1[8]={0x00,0xfd,0x0c,0x15,0x24,0x44,0x34,0x8c};
byte boishakh_2[8]={0x00,0x5d,0xb5,0x54,0x15,0x15,0x15,0x14};
byte boishakh_3[8]={0x00,0x5c,0xd0,0x50,0xd0,0x10,0x90,0xf0};
byte joishtho_0[8]={0x08,0x07,0x04,0x08,0x0a,0x0a,0x09,0x04};
byte joishtho_1[8]={0x00,0xff,0x30,0x48,0x46,0x74,0x28,0xc8};
byte joishtho_2[8]={0x08,0xe7,0x48,0x8f,0x91,0x5e,0x40,0xc0};
byte joishtho_3[8]={0x00,0xf0,0x80,0x40,0x20,0x90,0x90,0x60};

byte ashar_0[8]={0x3f,0x00,0x01,0x13,0x13,0x10,0x08,0x07};
byte ashar_1[8]={0xff,0x15,0x94,0x54,0x54,0x55,0xf4,0x94};
byte ashar_2[8]={0xff,0x14,0x94,0x54,0xb4,0x14,0x94,0x74};
byte ashar_3[8]={0xfe,0x80,0x90,0xb8,0x98,0x90,0xa0,0xd0};
byte srabon_0[8]={0x1b,0x04,0x0e,0x00,0x10,0x0f,0x00,0x00};
byte srabon_1[8]={0x7f,0xd0,0x51,0x52,0x54,0x56,0xd1,0x00};
byte srabon_2[8]={0xcd,0xd3,0x59,0x59,0x41,0x41,0xc1,0x00};
```

```

byte srabon_3[8]={0xc0,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
byte vadro_0[8]={0x03,0x00,0x02,0x02,0x02,0x02,0x01,0x00};
byte vadro_1[8]={0xff,0x20,0x74,0x46,0x3a,0x02,0x04,0xf8};
byte vadro_2[8]={0xff,0x44,0x44,0x45,0x46,0x50,0x4f,0x40};
byte vadro_3[8]={0xf0,0x00,0xc0,0x40,0x40,0x40,0x40,0xc0};
byte ashwin_0[8]={0x00,0x00,0x07,0x14,0x16,0x10,0x1f,0x00};
byte ashwin_1[8]={0x03,0x02,0xaa,0xaa,0xaa,0xea,0xaa,0x2a};
byte ashwin_2[8]={0xf0,0x00,0xb5,0x4c,0xa5,0x0d,0x15,0x1c};
byte ashwin_3[8]={0x00,0x00,0xf8,0x08,0xe8,0x18,0x88,0x08};
byte kartik_0[8]={0x00,0x00,0x07,0x00,0x01,0x02,0x03,0x00};
byte kartik_1[8]={0x00,0x00,0xff,0xf2,0x4a,0x5a,0x42,0xc2};
byte kartik_2[8]={0x7e,0x81,0xff,0x86,0xad,0xa1,0x91,0x8e};
byte kartik_3[8]={0x80,0x00,0xff,0x1e,0x29,0x4b,0x68,0x18};
byte ogrohayon_0[8]={0xfe,0x33,0xba,0x8b,0x8a,0x8b,0x5e,0x22};
byte ogrohayon_1[8]={0xcf,0xa9,0x5b,0xcb,0x08,0x0b,0xe8,0x18};
byte ogrohayon_2[8]={0xff,0x4c,0x2a,0x29,0x4a,0x8c,0x4b,0x2a};
byte ogrohayon_3[8]={0xe1,0x4d,0x53,0x59,0x59,0x41,0x41,0xc1};
byte poush_0[8] = {0x00,0x0f,0x04,0x08,0x08,0x08,0x0e,0x06};
byte poush_1[8] = {0x02,0x77,0x8d,0x95,0xe5,0x05,0x05,0x05};
byte poush_2[8] = {0x00,0xff,0x21,0x11,0x0d,0x13,0x21,0x1f};
byte poush_3[8]={0x00,0xe0,0x00,0x00,0x00,0x00,0x00,0x00};
byte magh_0[8]={0x00,0x01,0x01,0x00,0x00,0x01,0x01,0x00};
byte magh_1[8]={0xfa,0x0a,0x0a,0x8a,0xca,0xea,0xda,0x00};
byte magh_2[8]={0xfc,0x84,0x84,0x74,0x24,0x44,0x7c,0x00};
byte magh_3[8]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
byte falgun_0[8]={0xf0,0x04,0x02,0x02,0x04,0x08,0x06,0x01};
byte falgun_1[8]={0xfc,0x05,0xe4,0x94,0xb5,0x84,0x85,0x84};
byte falgun_2[8]={0xaf,0x58,0x08,0xe8,0x99,0x48,0x9c,0x1a};
byte falgun_3[8]={0xfc,0x08,0x08,0xe8,0x98,0x88,0x08,0x08};
byte choitro_0[8]={0x04,0x03,0x01,0x02,0x02,0x02,0x02,0x01};
byte choitro_1[8]={0x00,0xff,0x40,0x4c,0x74,0x45,0x48,0x70};
byte choitro_2[8]={0x00,0xfe,0x30,0x68,0x68,0x08,0xe8,0x18};
byte choitro_3[8]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

// fetched data array
byte fetched_time[6]={2,0,3,3,8,7}; //timeformat: hhmms
int bn_year_arr[4] = {0, 0, 0, 0};
int bn_date_arr[4] = {0,0};
int dayOfTheWeek = 0;
int bn_month = 0;

byte hour_0[8];
byte hour_1[8];
byte mins_0[8];
byte mins_1[8];

byte year_0[8];
byte year_1[8];
byte year_2[8];
byte year_3[8];

byte date_0[8];
byte date_1[8];

byte dayotw_0[8];
byte dayotw_1[8];

byte monthBn_0[8];
byte monthBn_1[8];
byte monthBn_2[8];
byte monthBn_3[8];

// blinking time seperator
int blink_yes = 0;
byte add_dots[8] = {0x00,0x01,0x01,0x00,0x00,0x01,0x01,0x00};

void setup_rtc ()
{
    Serial.print("compiled: ");
    Serial.print(__DATE__);
    Serial.println(__TIME__);

    Rtc.Begin();

    RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
    Serial.println();

    if (!Rtc.IsDateTimeValid())
    {
        // Common Causes:
        // 1) first time you ran and the device wasn't running yet
        // 2) the battery on the device is low or even missing
    }
}

```

```

        Serial.println("RTC lost confidence in the DateTime!");
        Rtc.SetDateTime(compiled);
    }

    if (Rtc.GetIsWriteProtected())
    {
        Serial.println("RTC was write protected, enabling writing now");
        Rtc.SetIsWriteProtected(false);
    }

    if (!Rtc.GetIsRunning())
    {
        Serial.println("RTC was not actively running, starting now");
        Rtc.SetIsRunning(true);
    }

    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled)
    {
        Serial.println("RTC is older than compile time! (Updating DateTime)");
        Rtc.SetDateTime(compiled);
    }
    else if (now > compiled)
    {
        Serial.println("RTC is newer than compile time. (this is expected)");
    }
    else if (now == compiled)
    {
        Serial.println("RTC is the same as compile time! (not expected but all is fine)");
    }
}

char c;

int year, month, day, dayOfWeek, hour, min, sec;
int deltaHour=0, deltaMin=0, deltaSec=0;

void updateNTPTime()
{
    mySerial.println("U");
    RtcDateTime now = Rtc.GetDateTime();
    String res = "";
    c = ' ';
    // check if data is available
    int count = 0;

    while (1) {
        if (mySerial.available() > 0) {
            if (mySerial.read() == '#') break;
        }
        if (count > 500) return;
        Serial.println(count);
        count += 1;
    }

    while (c != '\n') {
        if (mySerial.available() > 0) {
            if (c == '.') return;
            c = mySerial.read();
            res += c;
        }
        if (count > 500) return;
        Serial.println(count);
        count += 1;
    }
    sscanf(res.c_str(), "%d %d %d %d %d %d %d", &year, &month, &day, &dayOfWeek, &hour, &min, &sec);
    deltaHour = hour - now.Hour();
    deltaMin = min - now.Minute();
    deltaSec = sec - now.Second();
}

void setup_esp8266()
{
    mySerial.begin(9600);
    Serial.begin(9600);
    updateNTPTime();
}

void setup() {
    // Serial.begin(115200);
    mx.begin();
    mx.control(MD_MAX72XX::INTENSITY, 0);
    mx.clear();
    setup_rtc();
    delay(12000);
    setup_esp8266();
}

```

```

}

void loop() {
    RtcDateTime now = Rtc.GetDateTime();
    updateTime(now);
    updateDate(now);
    updateMonth(now);
    updateYear(now);
    updateDayOfWeek(now);
    setTime();
    drawShape();
}

int myDiv(int a, int b)
{
    if(a<0) return -1;
    else return a/b;
}

void updateTime(const RtcDateTime& dt)
{
    // int sec = (dt.Second() + deltaSec) % 60;
    // int min = (dt.Minute() + deltaMin + (dt.Second() + deltaSec) / 60) % 60;
    // int hour = (dt.Hour() + deltaHour + (dt.Minute() + deltaMin + (dt.Second() + deltaSec) / 60) / 60) % 24;

    int sec = dt.Second() + deltaSec;
    sec = (sec + 60) % 60;
    int min = dt.Minute() + deltaMin + myDiv(dt.Second() + deltaSec, 60) ;
    min = (min + 60) % 60;
    int hour = dt.Hour() + deltaHour + myDiv(dt.Minute() + deltaMin + myDiv(dt.Second() + deltaSec, 60), 60) ;
    hour = (hour + 24) % 24;

    fetched_time[0] = hour / 10;
    fetched_time[1] = hour % 10;
    fetched_time[2] = min / 10;
    fetched_time[3] = min % 10;
    fetched_time[4] = sec / 10;
    fetched_time[5] = sec % 10;

    blink_yes = fetched_time[5] % 2 ;

    if ((min + sec) == 0) updateNTPTime();
}

void updateDate(const RtcDateTime& dt)
{
    Bdate date = EnglishToBangla(dt.Day(),dt.Month(),dt.Year());
    for (int i=1; i >= 0; i--)
    {
        bn_date_arr[i] = date.banglaDate % 10;
        date.banglaDate = date.banglaDate / 10;
    }
}

void updateMonth(const RtcDateTime& dt)
{
    Bdate date = EnglishToBangla(dt.Day(),dt.Month(),dt.Year());
    bn_month = date.banglaMonthIndex;
}

void updateYear(const RtcDateTime& dt)
{
    int x = EnToBnYear(dt.Year(), dt.Month(), dt.Day());
    for (int i=3; i >= 0; i--)
    {
        bn_year_arr[i] = x % 10;
        x = x / 10;
    }
}

void updateDayOfWeek(const RtcDateTime& dt){
    dayOfTheWeek = dt.DayOfWeek();
}

void setTime(){
    setHour();
    setMinute();
    setDateBn();
    setMonth();
    setYear();
    setDayOTW();
}

```

```

}

void setMonth(){

    if(bn_month == 1)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = boishakh_0[i];
            monthBn_1[i] = boishakh_1[i];
            monthBn_2[i] = boishakh_2[i];
            monthBn_3[i] = boishakh_3[i];
        }

    else if(bn_month == 2)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = joishtho_0[i];
            monthBn_1[i] = joishtho_1[i];
            monthBn_2[i] = joishtho_2[i];
            monthBn_3[i] = joishtho_3[i];
        }

    else if(bn_month == 3)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = ashwar_0[i];
            monthBn_1[i] = ashwar_1[i];
            monthBn_2[i] = ashwar_2[i];
            monthBn_3[i] = ashwar_3[i];
        }

    else if(bn_month == 4)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = srabon_0[i];
            monthBn_1[i] = srabon_1[i];
            monthBn_2[i] = srabon_2[i];
            monthBn_3[i] = srabon_3[i];
        }

    else if(bn_month == 5)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = vadro_0[i];
            monthBn_1[i] = vadro_1[i];
            monthBn_2[i] = vadro_2[i];
            monthBn_3[i] = vadro_3[i];
        }

    else if(bn_month == 6)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = ashwin_0[i];
            monthBn_1[i] = ashwin_1[i];
            monthBn_2[i] = ashwin_2[i];
            monthBn_3[i] = ashwin_3[i];
        }

    else if(bn_month == 7)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = kartik_0[i];
            monthBn_1[i] = kartik_1[i];
            monthBn_2[i] = kartik_2[i];
            monthBn_3[i] = kartik_3[i];
        }

    else if(bn_month == 9)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = ogrohayon_0[i];
            monthBn_1[i] = ogrohayon_1[i];
            monthBn_2[i] = ogrohayon_2[i];
            monthBn_3[i] = ogrohayon_3[i];
        }

    else if(bn_month == 9)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = poush_0[i];
            monthBn_1[i] = poush_1[i];
            monthBn_2[i] = poush_2[i];
            monthBn_3[i] = poush_3[i];
        }

    else if(bn_month == 10)
        for (int i=0; i<=7; i++){
            monthBn_0[i] = magh_0[i];
            monthBn_1[i] = magh_1[i];
            monthBn_2[i] = magh_2[i];

```

```

        monthBn_3[i] = magh_3[i];
    }

    else if(bn_month == 11)
    for (int i=0; i<=7; i++){
        monthBn_0[i] = falgun_0[i];
        monthBn_1[i] = falgun_1[i];
        monthBn_2[i] = falgun_2[i];
        monthBn_3[i] = falgun_3[i];
    }

    else if(bn_month == 12)
    for (int i=0; i<=7; i++){
        monthBn_0[i] = choitro_0[i];
        monthBn_1[i] = choitro_1[i];
        monthBn_2[i] = choitro_2[i];
        monthBn_3[i] = choitro_3[i];
    }

    else
    for (int i=0; i<=7; i++){
        monthBn_0[i] = zero[i];
        monthBn_1[i] = zero[i];
        monthBn_2[i] = zero[i];
        monthBn_3[i] = zero[i];
    }
}

void setDayOTW(){
    if(dayOfTheWeek == 6){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = shoni_0[i];
            dayotw_1[i] = shoni_1[i];
        }
    }
    if(dayOfTheWeek == 0){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = robi_0[i];
            dayotw_1[i] = robi_1[i];
        }
    }
    if(dayOfTheWeek == 1){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = shom_0[i];
            dayotw_1[i] = shom_1[i];
        }
    }
    if(dayOfTheWeek == 2){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = mongol_0[i];
            dayotw_1[i] = mongol_1[i];
        }
    }
    if(dayOfTheWeek == 3){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = budh_0[i];
            dayotw_1[i] = budh_1[i];
        }
    }
    if(dayOfTheWeek == 4){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = brihoshpoti_0[i];
            dayotw_1[i] = brihoshpoti_1[i];
        }
    }
    if(dayOfTheWeek == 5){
        for (int i=0; i<=7; i++){
            dayotw_0[i] = shukro_0[i];
            dayotw_1[i] = shukro_1[i];
        }
    }
}

void setDateBn(){
    // first digit of date
    if (bn_date_arr[0] == 0){

```



```

    for (int i=0; i<=7; i++){
        date_0[i] = zero[i];
    }
}
if (bn_date_arr[0] == 1){
    for (int i=0; i<=7; i++){
        date_0[i] = one[i];
    }
}
if (bn_date_arr[0] == 2){
    for (int i=0; i<=7; i++){
        date_0[i] = two[i];
    }
}
if (bn_date_arr[0] == 3){
    for (int i=0; i<=7; i++){
        date_0[i] = three[i];
    }
}
if (bn_date_arr[0] == 4){
    for (int i=0; i<=7; i++){
        date_0[i] = four[i];
    }
}
if (bn_date_arr[0] == 5){
    for (int i=0; i<=7; i++){
        date_0[i] = five[i];
    }
}
if (bn_date_arr[0] == 6){
    for (int i=0; i<=7; i++){
        date_0[i] = six[i];
    }
}
if (bn_date_arr[0] == 7){
    for (int i=0; i<=7; i++){
        date_0[i] = seven[i];
    }
}
if (bn_date_arr[0] == 8){
    for (int i=0; i<=7; i++){
        date_0[i] = eight[i];
    }
}
if (bn_date_arr[0] == 9){
    for (int i=0; i<=7; i++){
        date_0[i] = nine[i];
    }
}
// 2nd digit date
if (bn_date_arr[1] == 0){

```

```

    for (int i=0; i<=7; i++){
        date_1[i] = zero[i];
    }
}
if (bn_date_arr[1] == 1){
    for (int i=0; i<=7; i++){
        date_1[i] = one[i];
    }
}
if (bn_date_arr[1] == 2){
    for (int i=0; i<=7; i++){
        date_1[i] = two[i];
    }
}
if (bn_date_arr[1] == 3){
    for (int i=0; i<=7; i++){
        date_1[i] = three[i];
    }
}
if (bn_date_arr[1] == 4){
    for (int i=0; i<=7; i++){
        date_1[i] = four[i];
    }
}
if (bn_date_arr[1] == 5){
    for (int i=0; i<=7; i++){
        date_1[i] = five[i];
    }
}
if (bn_date_arr[1] == 6){
    for (int i=0; i<=7; i++){
        date_1[i] = six[i];
    }
}
if (bn_date_arr[1] == 7){
    for (int i=0; i<=7; i++){
        date_1[i] = seven[i];
    }
}
if (bn_date_arr[1] == 8){
    for (int i=0; i<=7; i++){
        date_1[i] = eight[i];
    }
}
if (bn_date_arr[1] == 9){
    for (int i=0; i<=7; i++){
        date_1[i] = nine[i];
    }
}
}

```

```

void setYear(){
    // first digit of year
    if (bn_year_arr[0] == 0){

        for (int i=0; i<=7; i++){

            year_0[i] = zero[i];

        }

    }

    if (bn_year_arr[0] == 1){

        for (int i=0; i<=7; i++){

            year_0[i] = one[i];

        }

    }

    if (bn_year_arr[0] == 2){

        for (int i=0; i<=7; i++){

            year_0[i] = two[i];

        }

    }

    if (bn_year_arr[0] == 3){

        for (int i=0; i<=7; i++){

            year_0[i] = three[i];

        }

    }

    if (bn_year_arr[0] == 4){

        for (int i=0; i<=7; i++){

            year_0[i] = four[i];

        }

    }

    if (bn_year_arr[0] == 5){

        for (int i=0; i<=7; i++){

            year_0[i] = five[i];

        }

    }

    if (bn_year_arr[0] == 6){

        for (int i=0; i<=7; i++){

            year_0[i] = six[i];

        }

    }

    if (bn_year_arr[0] == 7){

        for (int i=0; i<=7; i++){

            year_0[i] = seven[i];

        }

    }

    if (bn_year_arr[0] == 8){

        for (int i=0; i<=7; i++){

            year_0[i] = eight[i];

        }

    }

    if (bn_year_arr[0] == 9){

        for (int i=0; i<=7; i++){

            year_0[i] = nine[i];

        }

    }

}

```

```

// 2nd digit year
if (bn_year_arr[1] == 0){
    for (int i=0; i<=7; i++){
        year_1[i] = zero[i];
    }
}
if (bn_year_arr[1] == 1){
    for (int i=0; i<=7; i++){
        year_1[i] = one[i];
    }
}
if (bn_year_arr[1] == 2){
    for (int i=0; i<=7; i++){
        year_1[i] = two[i];
    }
}
if (bn_year_arr[1] == 3){
    for (int i=0; i<=7; i++){
        year_1[i] = three[i];
    }
}
if (bn_year_arr[1] == 4){
    for (int i=0; i<=7; i++){
        year_1[i] = four[i];
    }
}
if (bn_year_arr[1] == 5){
    for (int i=0; i<=7; i++){
        year_1[i] = five[i];
    }
}
if (bn_year_arr[1] == 6){
    for (int i=0; i<=7; i++){
        year_1[i] = six[i];
    }
}
if (bn_year_arr[1] == 7){
    for (int i=0; i<=7; i++){
        year_1[i] = seven[i];
    }
}
if (bn_year_arr[1] == 8){
    for (int i=0; i<=7; i++){
        year_1[i] = eight[i];
    }
}
if (bn_year_arr[1] == 9){
    for (int i=0; i<=7; i++){
        year_1[i] = nine[i];
    }
}
}

```

```

//3rd digit
if (bn_year_arr[2] == 0){
    for (int i=0; i<=7; i++){
        year_2[i] = zero[i];
    }
}
if (bn_year_arr[2] == 1){
    for (int i=0; i<=7; i++){
        year_2[i] = one[i];
    }
}
if (bn_year_arr[2] == 2){
    for (int i=0; i<=7; i++){
        year_2[i] = two[i];
    }
}
if (bn_year_arr[2] == 3){
    for (int i=0; i<=7; i++){
        year_2[i] = three[i];
    }
}
if (bn_year_arr[2] == 4){
    for (int i=0; i<=7; i++){
        year_2[i] = four[i];
    }
}
if (bn_year_arr[2] == 5){
    for (int i=0; i<=7; i++){
        year_2[i] = five[i];
    }
}
if (bn_year_arr[2] == 6){
    for (int i=0; i<=7; i++){
        year_2[i] = six[i];
    }
}
if (bn_year_arr[2] == 7){
    for (int i=0; i<=7; i++){
        year_2[i] = seven[i];
    }
}
if (bn_year_arr[2] == 8){
    for (int i=0; i<=7; i++){
        year_2[i] = eight[i];
    }
}
if (bn_year_arr[2] == 9){
    for (int i=0; i<=7; i++){
        year_2[i] = nine[i];
    }
}
}

```

```

// 4th digit
if (bn_year_arr[3] == 0){
    for (int i=0; i<=7; i++){
        year_3[i] = zero[i];
    }
}
if (bn_year_arr[3] == 1){
    for (int i=0; i<=7; i++){
        year_3[i] = one[i];
    }
}
if (bn_year_arr[3] == 2){
    for (int i=0; i<=7; i++){
        year_3[i] = two[i];
    }
}
if (bn_year_arr[3] == 3){
    for (int i=0; i<=7; i++){
        year_3[i] = three[i];
    }
}
if (bn_year_arr[3] == 4){
    for (int i=0; i<=7; i++){
        year_3[i] = four[i];
    }
}
if (bn_year_arr[3] == 5){
    for (int i=0; i<=7; i++){
        year_3[i] = five[i];
    }
}
if (bn_year_arr[3] == 6){
    for (int i=0; i<=7; i++){
        year_3[i] = six[i];
    }
}
if (bn_year_arr[3] == 7){
    for (int i=0; i<=7; i++){
        year_3[i] = seven[i];
    }
}
if (bn_year_arr[3] == 8){
    for (int i=0; i<=7; i++){
        year_3[i] = eight[i];
    }
}
if (bn_year_arr[3] == 9){
    for (int i=0; i<=7; i++){
        year_3[i] = nine[i];
    }
}
}

```

```

}

void setHour(){
    // setting first digit
    if (fetched_time[0] == 0){
        for (int i=0; i<=7; i++){
            hour_0[i] = zero[i];
        }
    }
    if (fetched_time[0] == 1){
        for (int i=0; i<=7; i++){
            hour_0[i] = one[i];
        }
    }
    if (fetched_time[0] == 2){
        for (int i=0; i<=7; i++){
            hour_0[i] = two[i];
        }
    }
    if (fetched_time[0] == 3){
        for (int i=0; i<=7; i++){
            hour_0[i] = three[i];
        }
    }
    if (fetched_time[0] == 4){
        for (int i=0; i<=7; i++){
            hour_0[i] = four[i];
        }
    }
    if (fetched_time[0] == 5){
        for (int i=0; i<=7; i++){
            hour_0[i] = five[i];
        }
    }
    if (fetched_time[0] == 6){
        for (int i=0; i<=7; i++){
            hour_0[i] = six[i];
        }
    }
    if (fetched_time[0] == 7){
        for (int i=0; i<=7; i++){
            hour_0[i] = seven[i];
        }
    }
    if (fetched_time[0] == 8){
        for (int i=0; i<=7; i++){
            hour_0[i] = eight[i];
        }
    }
    if (fetched_time[0] == 9){
        for (int i=0; i<=7; i++){
            hour_0[i] = nine[i];
        }
    }
}

```

```

    }
}
// 2nd digit hour
if (fetched_time[1] == 0){
    for (int i=0; i<=7; i++){
        hour_1[i] = zero[i];
    }
}
if (fetched_time[1] == 1){
    for (int i=0; i<=7; i++){
        hour_1[i] = one[i];
    }
}
if (fetched_time[1] == 2){
    for (int i=0; i<=7; i++){
        hour_1[i] = two[i];
    }
}
if (fetched_time[1] == 3){
    for (int i=0; i<=7; i++){
        hour_1[i] = three[i];
    }
}
if (fetched_time[1] == 4){
    for (int i=0; i<=7; i++){
        hour_1[i] = four[i];
    }
}
if (fetched_time[1] == 5){
    for (int i=0; i<=7; i++){
        hour_1[i] = five[i];
    }
}
if (fetched_time[1] == 6){
    for (int i=0; i<=7; i++){
        hour_1[i] = six[i];
    }
}
if (fetched_time[1] == 7){
    for (int i=0; i<=7; i++){
        hour_1[i] = seven[i];
    }
}
if (fetched_time[1] == 8){
    for (int i=0; i<=7; i++){
        hour_1[i] = eight[i];
    }
}
if (fetched_time[1] == 9){
    for (int i=0; i<=7; i++){
        hour_1[i] = nine[i];
    }
}

```



```

    }
}

    if (blink_yes == 1){
        for (int i=0; i<=7; i++){

            hour_1[i] = hour_1[i] + add_dots[i];

        }
    }

    //Serial.println("Hour printed");
}

void setMinute(){

    //setting third digit
    if (fetched_time[2] == 0){

        for (int i=0; i<=7; i++){

            mins_0[i] = zero[i];

        }
    }

    if (fetched_time[2] == 1){

        for (int i=0; i<=7; i++){

            mins_0[i] = one[i];

        }
    }

    if (fetched_time[2] == 2){

        for (int i=0; i<=7; i++){

            mins_0[i] = two[i];

        }
    }

    if (fetched_time[2] == 3){

        for (int i=0; i<=7; i++){

            mins_0[i] = three[i];

        }
    }

    if (fetched_time[2] == 4){

        for (int i=0; i<=7; i++){

            mins_0[i] = four[i];

        }
    }

    if (fetched_time[2] == 5){

        for (int i=0; i<=7; i++){

            mins_0[i] = five[i];

        }
    }

    if (fetched_time[2] == 6){

        for (int i=0; i<=7; i++){

            mins_0[i] = six[i];

        }
    }

    if (fetched_time[2] == 7){

        for (int i=0; i<=7; i++){

            mins_0[i] = seven[i];

        }
    }
}

```

```

}
if (fetched_time[2] == 8){
    for (int i=0; i<=7; i++){
        mins_0[i] = eight[i];
    }
}
if (fetched_time[2] == 9){
    for (int i=0; i<=7; i++){
        mins_0[i] = nine[i];
    }
}
//setting fourth digit
if (fetched_time[3] == 0){
    for (int i=0; i<=7; i++){
        mins_1[i] = zero[i];
    }
}
if (fetched_time[3] == 1){
    for (int i=0; i<=7; i++){
        mins_1[i] = one[i];
    }
}
if (fetched_time[3] == 2){
    for (int i=0; i<=7; i++){
        mins_1[i] = two[i];
    }
}
if (fetched_time[3] == 3){
    for (int i=0; i<=7; i++){
        mins_1[i] = three[i];
    }
}
if (fetched_time[3] == 4){
    for (int i=0; i<=7; i++){
        mins_1[i] = four[i];
    }
}
if (fetched_time[3] == 5){
    for (int i=0; i<=7; i++){
        mins_1[i] = five[i];
    }
}
if (fetched_time[3] == 6){
    for (int i=0; i<=7; i++){
        mins_1[i] = six[i];
    }
}
if (fetched_time[3] == 7){
    for (int i=0; i<=7; i++){
        mins_1[i] = seven[i];
    }
}

```

```

}
if (fetched_time[3] == 8){
    for (int i=0; i<=7; i++){
        mins_1[i] = eight[i];
    }
}
if (fetched_time[3] == 9){
    for (int i=0; i<=7; i++){
        mins_1[i] = nine[i];
    }
}
//Serial.println("Mins printed");
}

void drawShape() {

    /*setRow(startDev, endDev, r, value)
    startDev - from which division you want to display the character.
    endDev - at which division you want to stop.
    r - no of the row which is to be set.
    value - bit map of that row.
    */

    // time showing display
    for (int i = 0; i <= 7; i++) {
        //mx.setRow(3, 3, i, hour_0[i]);
        mx.setRow(3, 3, i, hour_0[i]);
    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(2, 2, i, hour_1[i]);
    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(1, 1, i, mins_0[i]);
    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(0, 0, i, mins_1[i]);
    }
    delay(delay_t);

    // day,date showing display
    for (int i = 0; i <= 7; i++) {

        mx.setRow(15, 15, i, dayotw_0[i]);

    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(14, 14, i, dayotw_1[i]);
    }
    delay(delay_t);

    for (int i = 0; i <= 7; i++) {
        mx.setRow(13, 13, i, date_0[i]);
    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(12, 12, i, date_1[i]);
    }
    delay(delay_t);

    // month showing display
    for (int i = 0; i <= 7; i++) {

        mx.setRow(11, 11, i, monthBn_0[i]);

    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(10, 10, i, monthBn_1[i]);
    }
    delay(delay_t);
    for (int i = 0; i <= 7; i++) {
        mx.setRow(9, 9, i, monthBn_2[i]);
    }
    delay(delay_t);
}

```

```

for (int i = 0; i <= 7; i++) {
    mx.setRow(8, 8, i, monthBn_3[i]);
}
delay(delay_t);

//Year Showing Display

for (int i = 0; i <= 7; i++) {
    //mx.setRow(3, 3, i, hour_0[i]);
    mx.setRow(7, 7, i, year_0[i]);
}
delay(delay_t);
for (int i = 0; i <= 7; i++) {
    mx.setRow(6, 6, i, year_1[i]);
}
delay(delay_t);
for (int i = 0; i <= 7; i++) {
    mx.setRow(5, 5, i, year_2[i]);
}
delay(delay_t);
for (int i = 0; i <= 7; i++) {
    mx.setRow(4, 4, i, year_3[i]);
}
delay(delay_t);
}

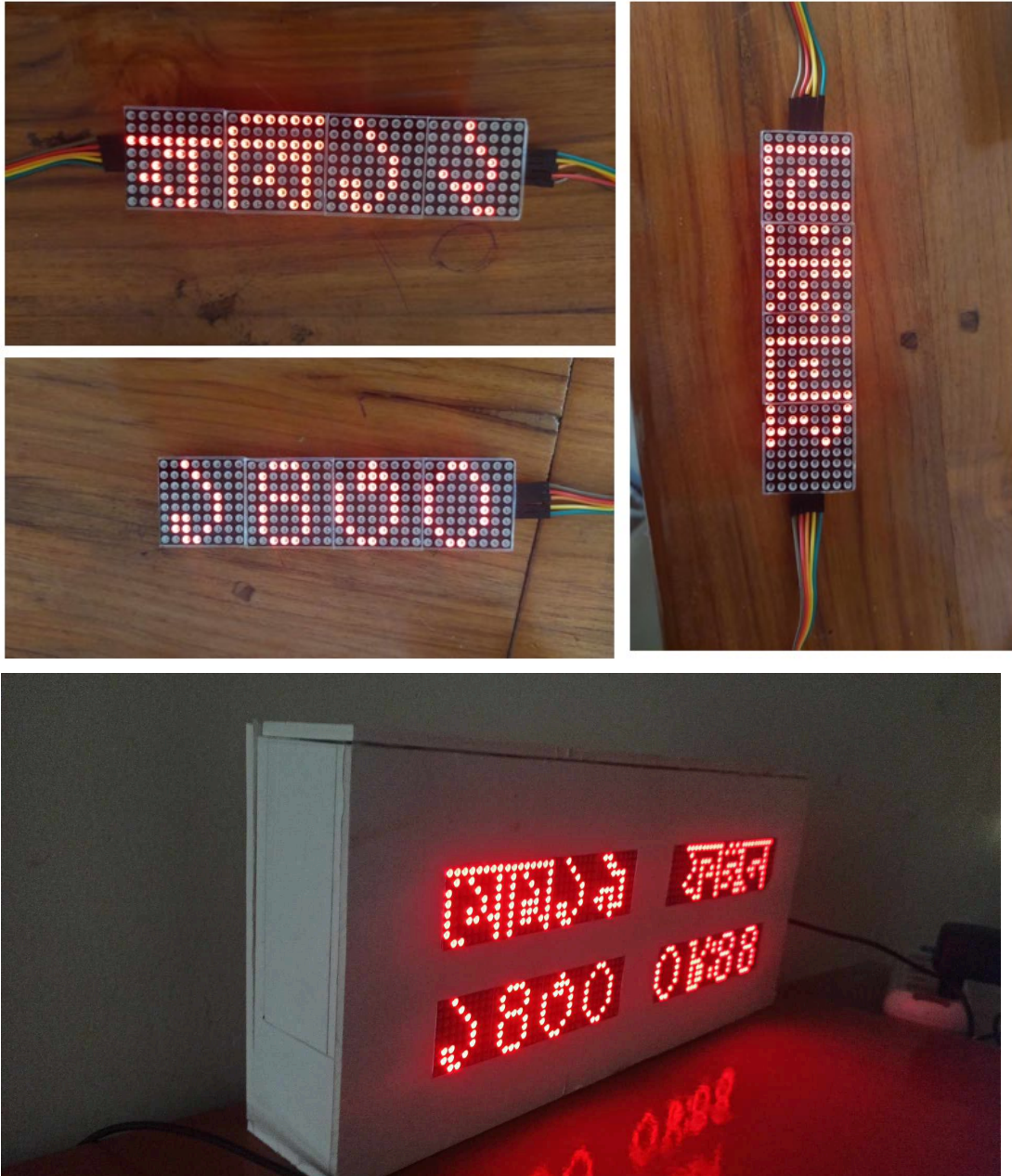
```

Table: Source Code for the main program

4 Implementation

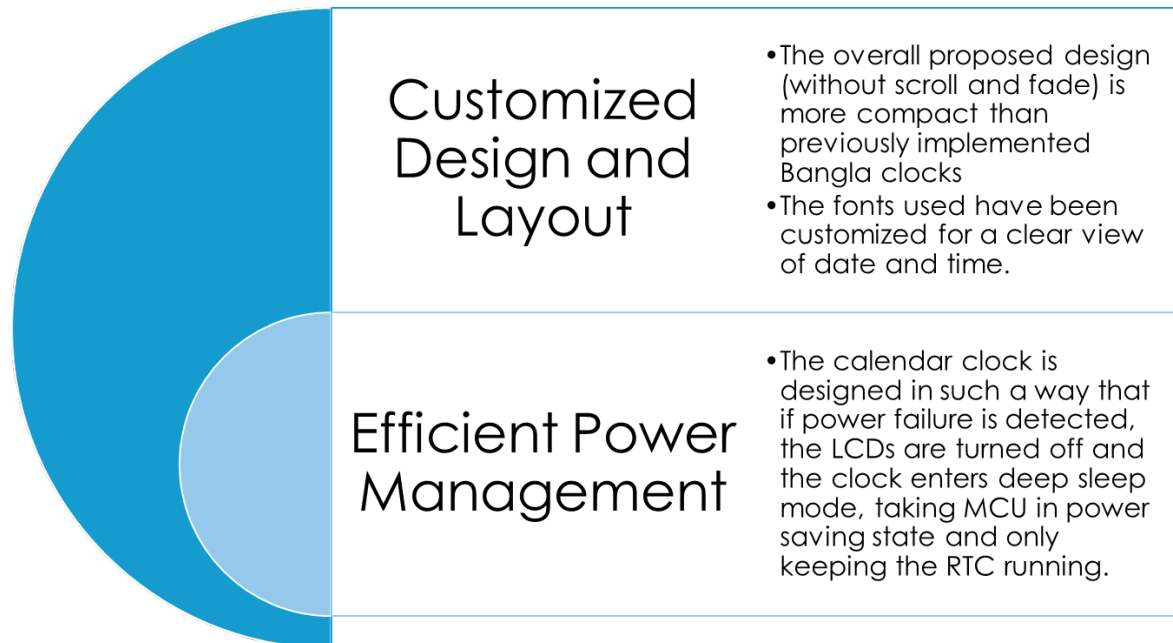
4.1 Description

We have used 4 displays. The top left one shows Day of the Week and Day of the Bangla Month. The top right one shows Bangla month, whereas bottom left one shows Bangla Year and Bottom right one shows time.



5 Design Analysis and Evaluation

5.1 Novelty



5.2 Design Considerations

5.2.1 Considerations to public health and safety

- Ergonomic design for comfortable use.
- Electrical safety standards adherence.
- Minimization of radiation exposure.
- Clear user instructions for safe operation.
- Durable construction to prevent hazards.

5.2.2 Considerations to environment

- Energy-efficient design.
- End-of-life recycling considerations.
- Minimal, recyclable packaging.
- Lifecycle analysis for eco-friendly practices.

5.2.3 Considerations to cultural and societal needs

- Cultural sensitivity in design.
- Localization for regional variations.
- Accessibility for diverse users.
- Community engagement in development.
- Educational value for cultural awareness.

5.3 Limitations of Tools

- The calendar clock could be improvised more with an addition of two or more display modules. Due to budget restrictions, keeping in mind that our aim was to make a cost-effective device, we could not add more modules.
- The ESP module needs adequate power all the time to work properly. It also shows some discrepancy in poor network regions. Hence, it sometimes acts as a faulty module.

5.4 Impact Assessment (PO(f))

5.4.1 Assessment of Societal and Cultural Issues

- Global manufacturing of Bangla Calendar Clock can increase cultural recognition of Bangla Language across the globe.

5.4.2 Assessment of Health and Safety Issues

- In hospitals and medical centers, accurate timekeeping is vital for coordinating patient care, administering medications, and conducting medical procedures. Any discrepancies in time synchronization among medical devices or systems could lead to errors in treatment, potentially jeopardizing patient safety. Therefore, ensuring that clocks in healthcare settings are synchronized and reliable is essential for maintaining patient health and safety.
- During natural disasters or other emergencies, synchronized clocks and timekeeping systems are crucial for coordinating evacuation procedures, managing resources, and disseminating timely information to the public. Ensuring that clocks remain synchronized, even in the event of power outages or network disruptions, is essential for maintaining public safety and facilitating effective disaster response efforts.

5.5 Sustainability Evaluation (PO(g))

5.5.1 Assessment of Societal and Cultural Issues

- Cultural Sensitivity: Consideration of cultural norms and values to ensure the Digital Bangla Clock and Calendar aligns with user preferences and respects cultural heritage.
- Language Localization: Assessment of language requirements to provide a user interface in Bangla, accommodating linguistic diversity and enhancing accessibility for Bangla-speaking users.

5.5.2 Assessment of Health and Safety Issues

- Electrical Safety: Assessment of electrical components and design to ensure compliance with safety standards, minimizing risks of electric shocks or fire hazards for user protection.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

| Teammate ID | Task |
|-------------|---|
| 1906177 | <ul style="list-style-type: none">• Conversion of English to Bangla Dates• Hardware Assembly |
| 1906178 | <ul style="list-style-type: none">• LED mapping of Bangla Fonts• Hardware Assembly |
| 1906191 | <ul style="list-style-type: none">• Fetching Date/Time Using DS 1302• Hardware Assembly |
| 1906195 | <ul style="list-style-type: none">• Fetching Date/Time Using ESP8266• Hardware Assembly |

6.2 Mode of TeamWork and Diversity

- **Inclusive Collaboration:** Valuing every member's input and celebrating diversity of ideas and backgrounds.
- **Cross-Functional Teams:** Leveraging diverse skill sets for comprehensive problem-solving.
- **Collaborative Technology:** Utilizing tools for seamless communication and inclusivity across locations.
- **Flexible Work Arrangements:** Allowing remote work and flexible hours to accommodate diverse needs.
- **Training and Development:** Investing in diversity training to promote understanding and respect.
- **Accountability and Feedback:** Establishing clear expectations and providing regular feedback for all members.
- **Celebration of Diversity:** Recognizing and celebrating unique contributions from diverse team members.

7 Communication to External Stakeholders

7.1 Executive Summary

The Bangla Calendar Clock is a digital timepiece designed to incorporate the traditional Bangla calendar which caters primarily to Bengali communities and enthusiasts worldwide, providing a seamless blend of modern timekeeping with cultural heritage. By integrating the Bangla calendar, the clock acknowledges and celebrates the cultural identity of Bengali communities, fostering a sense of connection to tradition and heritage. The clock employs precise timekeeping mechanisms to ensure accurate tracking of Bangla dates, catering to the practical needs of users while honoring cultural significance. Beyond its utility as a timepiece, the Bangla Calendar Clock serves as an educational resource, promoting awareness and understanding of the Bangla calendar system among users of diverse backgrounds. The Bangla Calendar Clock targets a niche market of Bengali communities worldwide, as well as individuals with an interest in cultural diversity and heritage. With an estimated global Bengali population of over 250 million, the clock holds significant potential for widespread adoption and cultural impact. Overall, the Bangla Calendar Clock represents a harmonious fusion of tradition and modernity, catering to the cultural needs of Bengali communities while offering practical utility in daily life. Its innovative design, educational value, and market potential position it as a unique product with considerable appeal to diverse audiences globally.

7.2 User Manual

7.3 Github Link

<https://github.com/arafatshovon/Bangla-Calender-Clock-3.0>

8 Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

| Name Of the Product | Units ordered | Per Unit Cost | Total cost |
|---|---------------|---------------|-------------|
| Max7219 4-Channel Dot Matrix Display Panel | 4 | 545 | 2180 |
| DS1302 Real Time Clock Module | 1 | 190 | 190 |
| NodeMCU ESP8266 Wi-Fi module | 1 | 615 | 615 |
| AC-DC adapter 220V/12V | 1 | 160 | 160 |
| Arduino Uno | 1 | 1069 | 1069 |
| Casing and Others | 1 | 500 | 500 |
| 2. Calculation of Per Unit Cost of Prototype | | | 4714 |

8.2 Timeline of Project Implementation

| Topic | Date |
|-----------------------|-------------------|
| Project Selection | December 22, 2023 |
| Workflow finalization | January 10, 2024 |
| Group Discussion 1 | January 17, 2024 |
| Components Purchase | January 20, 2024 |
| Group Discussion 2 | February 10, 2024 |

9 Future Work

Adding a PIR motion sensor can further increase Power Efficiency, ensuring the clock LED's are on only when the PIR senses an object.

Introducing RTC Calibration Button for user convenience

Introduction of a voice and small scale sound system can help visually impaired people.

10 References

- [1] Sengupta, Nitish. “Land of two rivers: A history of Bengal from the Mahabharata to Mujib.” Penguin UK, 2011.
- [2] M. S. Arefin, M. A. Dewan, M. I. Khan, and M. S. Islam, “Designing a 24-segment display for Bengali numerical digits and characters,” in Proc 3rd International Conference on Electrical and Computer Engineering ICECE, 2004, pp. 549-552.
- [3] M.A. Kader, A. Al Monsur, M. Moinuddin, Sayed Allmah Iqbal, Mohammad Shah Alamgir, “Bengali character based digital clock using 8×8 dot matrix display”, in Proc of 2nd Int’l Conf on Electrical Engineering and Information & Communication Technology (ICEEICT) 2015.
- [4] M.A. Kader, Raquib Uddin, Maher Abdullah, “Bengali character based digital clock using 13 segment LED display”, in Proc of 17th International Conference on Computer and Information Technology (ICCIT) 2014.