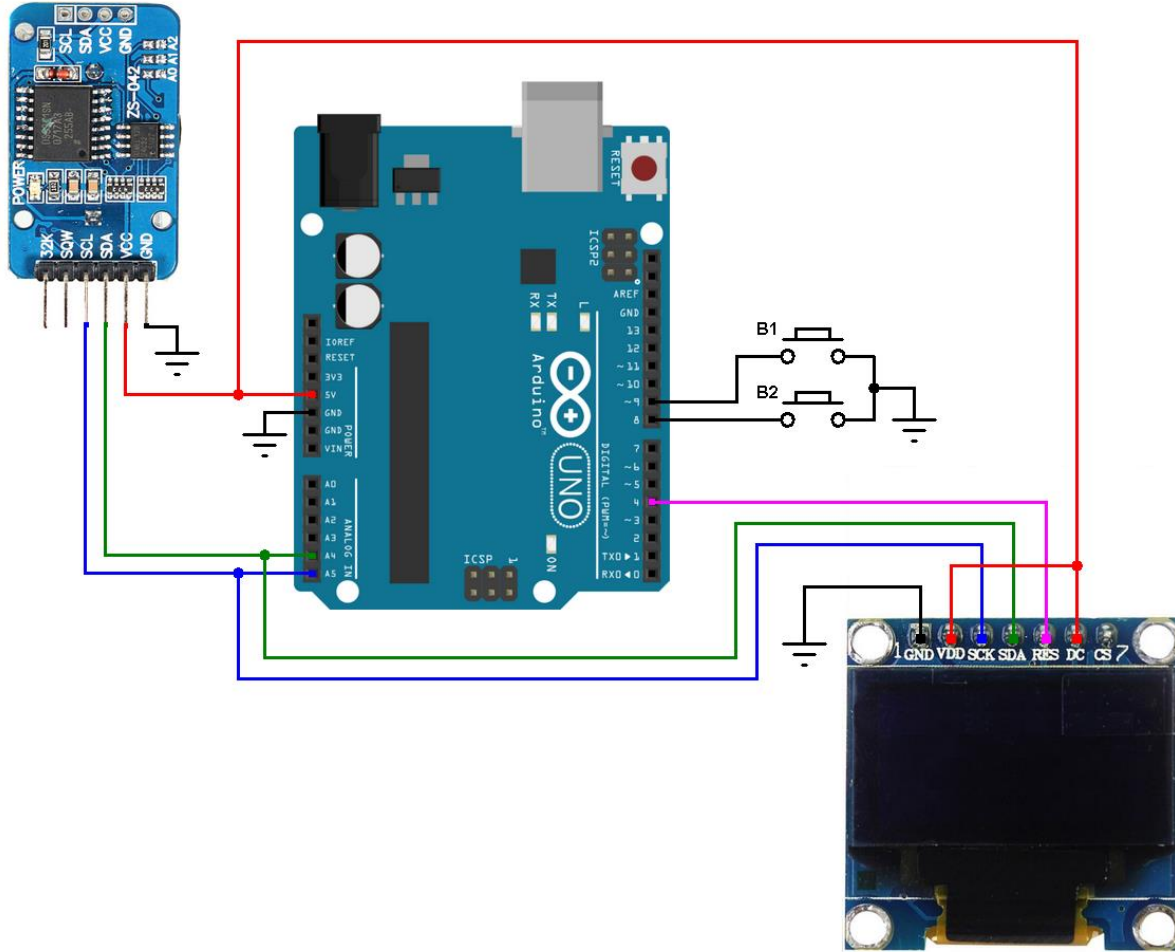
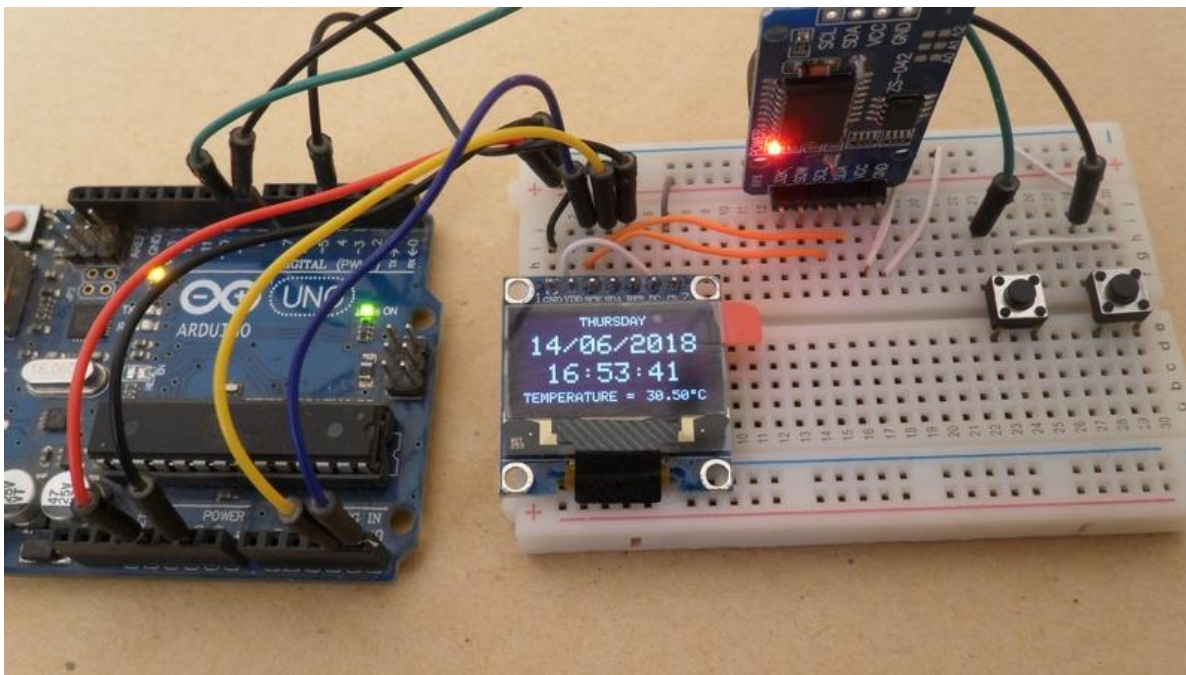


Dinesh Kumar, ISRO, Bangalore

DS3231 Board



SSD1306 OLED
128x64 Pixel



```
// Arduino real time clock and temperature monitor with DS3231 and SSD1306 OLED
```

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
#include "ds3231.h"
```

```
#define BUFF_MAX 256
```

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

```
#define OLED_RESET 4
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
#define button1 9 // Button B1 is connected to Arduino pin 9
```

```
#define button2 8 // Button B2 is connected to Arduino pin 8
```

```
void setup(void) {
```

```
  pinMode(button1, INPUT_PULLUP);
```

```
  pinMode(button2, INPUT_PULLUP);
```

```
  delay(1000);
```

```
  // by default, we'll generate the high voltage from the 3.3v line internally! (neat!)
```

```
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // initialize with the I2C addr 0x3D (for the 128x64)
```

```
  // init done
```

```
  // Clear the display buffer.
```

```
  display.clearDisplay();
```

```

display.display();

display.setTextColor(WHITE, BLACK);

display.drawRect(117, 56, 3, 3, WHITE); // Put degree symbol ( ° )

draw_text(0, 56, "TEMPERATURE =", 1);

draw_text(122, 56, "C", 1);

}

char Time[] = " : : ";

char Calendar[] = " / /20 ";

char temperature[] = " 00.00";

char temperature_msb;

byte i, second, minute, hour, day, date, month, year, temperature_lsb;

void display_day(){

    switch(day){

        case 1: draw_text(40, 0, " SUNDAY ", 1); break;

        case 2: draw_text(40, 0, " MONDAY ", 1); break;

        case 3: draw_text(40, 0, " TUESDAY ", 1); break;

        case 4: draw_text(40, 0, " WEDNESDAY", 1); break;

        case 5: draw_text(40, 0, " THURSDAY ", 1); break;

        case 6: draw_text(40, 0, " FRIDAY ", 1); break;

        default: draw_text(40, 0, " SATURDAY ", 1);

    }

}

void DS3231_display(){

    // Convert BCD to decimal

```

```
second = (second >> 4) * 10 + (second & 0x0F);
minute = (minute >> 4) * 10 + (minute & 0x0F);
hour = (hour >> 4) * 10 + (hour & 0x0F);
date = (date >> 4) * 10 + (date & 0x0F);
month = (month >> 4) * 10 + (month & 0x0F);
year = (year >> 4) * 10 + (year & 0x0F);
// End conversion
```

```
Time[7] = second % 10 + 48;
```

```
Time[6] = second / 10 + 48;
```

```
Time[4] = minute % 10 + 48;
```

```
Time[3] = minute / 10 + 48;
```

```
Time[1] = hour % 10 + 48;
```

```
Time[0] = hour / 10 + 48;
```

```
Calendar[9] = year % 10 + 48;
```

```
Calendar[8] = year / 10 + 48;
```

```
Calendar[4] = month % 10 + 48;
```

```
Calendar[3] = month / 10 + 48;
```

```
Calendar[1] = date % 10 + 48;
```

```
Calendar[0] = date / 10 + 48;
```

```
if(temperature_msb < 0){
```

```
    temperature_msb = abs(temperature_msb);
```

```
    temperature[0] = '-';
```

```
}
```

```
else
```

```
    temperature[0] = ' ';
```

```
temperature_lsb >>= 6;
```

```
temperature[2] = temperature_msb % 10 + 48;
```

```

temperature[1] = temperature_msb / 10 + 48;

if(temperature_lsb == 0 || temperature_lsb == 2){

    temperature[5] = '0';

    if(temperature_lsb == 0) temperature[4] = '0';

    else          temperature[4] = '5';

}

if(temperature_lsb == 1 || temperature_lsb == 3){

    temperature[5] = '5';

    if(temperature_lsb == 1) temperature[4] = '2';

    else          temperature[4] = '7';

}

draw_text(4, 14, Calendar, 2);          // Display the date (format: dd/mm/yyyy)

draw_text(16, 35, Time, 2);             // Display the time

draw_text(80, 56, temperature, 1);      // Display the temperature

}

```

```

void blink_parameter(){

    byte j = 0;

    while(j < 10 && digitalRead(button1) && digitalRead(button2)){

        j++;

        delay(25);

    }

}

```

```

byte edit(byte x_pos, byte y_pos, byte parameter){

    char text[3];

    sprintf(text,"%02u", parameter);

```

```

while(!digitalRead(button1));           // Wait until button B1 released

while(true){

  while(!digitalRead(button2)){         // If button B2 is pressed

    parameter++;

    if(i == 0 && parameter > 31)         // If date > 31 ==> date = 1

      parameter = 1;

    if(i == 1 && parameter > 12)         // If month > 12 ==> month = 1

      parameter = 1;

    if(i == 2 && parameter > 99)         // If year > 99 ==> year = 0

      parameter = 0;

    if(i == 3 && parameter > 23)         // If hours > 23 ==> hours = 0

      parameter = 0;

    if(i == 4 && parameter > 59)         // If minutes > 59 ==> minutes = 0

      parameter = 0;

    sprintf(text,"%02u", parameter);

    draw_text(x_pos, y_pos, text, 2);

    delay(200);                          // Wait 200ms

  }

  draw_text(x_pos, y_pos, " ", 2);

  blink_parameter();

  draw_text(x_pos, y_pos, text, 2);

  blink_parameter();

  if(!digitalRead(button1)){             // If button B1 is pressed

    i++;                                  // Increment 'i' for the next parameter

    return parameter;                     // Return parameter value and exit

  }

}

}

```

```

void draw_text(byte x_pos, byte y_pos, char *text, byte text_size) {
    display.setCursor(x_pos, y_pos);
    display.setTextSize(text_size);
    display.print(text);
    display.display();
}

```

```

void loop() {

    if(!digitalRead(button1)){           // If button B1 is pressed

        i = 0;

        while(!digitalRead(button1));    // Wait for button B1 release

        while(true){

            while(!digitalRead(button2)){ // While button B2 pressed

                day++;                    // Increment day

                if(day > 7) day = 1;

                display_day();           // Call display_day function

                delay(200);              // Wait 200 ms

            }

            draw_text(40, 0, "    ", 1);

            blink_parameter();           // Call blink_parameter function

            display_day();               // Call display_day function

            blink_parameter();           // Call blink_parameter function

            if(!digitalRead(button1))    // If button B1 is pressed

                break;

        }
    }
}

```

```

date = edit(4, 14, date);          // Edit date
month = edit(40, 14, month);       // Edit month
year = edit(100, 14, year);        // Edit year
hour = edit(16, 35, hour);         // Edit hours
minute = edit(52, 35, minute);     // Edit minutes

// Convert decimal to BCD
minute = ((minute / 10) << 4) + (minute % 10);
hour = ((hour / 10) << 4) + (hour % 10);
date = ((date / 10) << 4) + (date % 10);
month = ((month / 10) << 4) + (month % 10);
year = ((year / 10) << 4) + (year % 10);

// End conversion

// Write data to DS3231 RTC
Wire.beginTransmission(0x68);      // Start I2C protocol with DS3231 address
Wire.write(0);                     // Send register address
Wire.write(0);                     // Reset seconds and start oscillator
Wire.write(minute);                // Write minute
Wire.write(hour);                  // Write hour
Wire.write(day);                   // Write day
Wire.write(date);                  // Write date
Wire.write(month);                 // Write month
Wire.write(year);                  // Write year
Wire.endTransmission();            // Stop transmission and release the I2C bus
delay(200);                        // Wait 200ms
}

```