Date, Time, Temperature and Humidity Display

Introducere

Proiectul meu este un display pe care sunt afisate in permanenta data, ora, temperatura si umiditatea, iar dupa un interval prestabilit buzzerul va incepe sa scoata un sunet.

Descriere generală

Pe ecranul OLED sunt afisate data, ora, temperatura si umiditatea pe langa o iconita cu un ceas. Pentru utilizarea functiei de alarma trebuie doar sa selectam din cod ora, minutul, si secunda la care vrem sa sune.

Hardware Design

Lista componente:

- Arduino UNO
- DHT11 Temperature & Humidity Sensor
- Maxim Integrated DS3231
- Resistor 10k ohm
- Adafruit 128×64 OLED graphic display
- Breadboard
- Jumper wires
- Buzzer

Schema electrica:

Software Design

```
//Date, Time, Temperature and Humidity Display
//////////////////////For OLED//////////////////////
#include
#include
#include
#include
// If using software SPI (the default case):
#define OLED_MOSI 9 //MOSI aka SDA
#define OLED_CLK 10 //CLK aka SCL
#define OLED_DC 11
#define OLED_CS 12
#define OLED_RESET 13
Adafruit_SSD1306 display(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);
#if (SSD1306_LCDHEIGHT != 64)
#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif
////////////////////////////////////////////////////////
//////////////////////////For DHT//////////////////////////
#include
dht DHT;
#define DHT11_PIN 3
int DHTtimer = 0;
////////////////////////////////////////////////////////
//////////////////////////For RTC (DS3231)//////////////////////////
//SDA to A4, SCL to A5
//////////////////////////For Alarm//////////////////////////////
int buzzer = 8;
int h = 21;
int m = 44;
int s = 10;
#define DS3231_I2C_ADDRESS 0x68
// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
return( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
return( (val/16*10) + (val%16) );
}
////////////////////////////////////////////////////////
void setup()
{
Wire.begin();
Serial.begin(9600);
//set the initial time here, after setting the time, comment this section
//DS3231 seconds, minutes, hours, day, date, month, year
//setDS3231time(50,45,19,3,18,5,22);
display.begin(SSD1306_SWITCHCAPVCC);
display.clearDisplay();
pinMode(buzzer, OUTPUT);
}
//////////////////////////DS3231 coding//////////////////////////
```

```
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek, byte
dayOfMonth, byte month, byte year)
{
// sets time and date data to DS3231
Wire.beginTransmission(DS3231_I2C_ADDRESS);
Wire.write(0); // set next input to start at the seconds register
Wire.write(decToBcd(second)); // set seconds
Wire.write(decToBcd(minute)); // set minutes
Wire.write(decToBcd(hour)); // set hours
Wire.write(decToBcd(dayOfWeek)); // set day of week (1=Sunday, 7=Saturday)
Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
Wire.write(decToBcd(month)); // set month
Wire.write(decToBcd(year)); // set year (0 to 99)
Wire.endTransmission();
}
void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
Wire.beginTransmission(DS3231_I2C_ADDRESS);
Wire.write(0); // set DS3231 register pointer to 00h
Wire.endTransmission();
Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
// request seven bytes of data from DS3231 starting from register 00h
*second = bcdToDec(Wire.read() & 0x7f);
*minute = bcdToDec(Wire.read());
*hour = bcdToDec(Wire.read() & 0x3f);
*dayOfWeek = bcdToDec(Wire.read());
*dayOfMonth = bcdToDec(Wire.read());
*month = bcdToDec(Wire.read());
*year = bcdToDec(Wire.read());
}
void displayTime()
{
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
// retrieve data from DS3231
readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
&year);
// send it to the serial monitor
Serial.print(hour, DEC);
// convert the byte variable to a decimal number when displayed
Serial.print(":");
if (minute<10)
{
Serial.print("0");
}
Serial.print(minute, DEC);
Serial.print(":");
if (second<10)
{
Serial.print("0");
```

```
}
Serial.print(second, DEC);
Serial.print(" ");
Serial.print(dayOfMonth, DEC);
Serial.print("/");
Serial.print(month, DEC);
Serial.print("/");
Serial.print(year, DEC);
Serial.print(" Day of week: ");
//////////////////////////////////////////////////////////////////
display.setCursor(50,0);
display.print(dayOfMonth, DEC);
display.print(",");
display.setCursor(91,0);
display.print("20");
display.print(year, DEC);
if (hour<10)
{
display.setCursor(40,10);
display.print("0");
display.print(hour, DEC);
display.print(":");
}
if (hour>9)
{
display.setCursor(40,10);
display.print(hour, DEC);
display.print(":");
}
if (minute<10)
{
display.setCursor(58,10);
display.print("0");
display.print(minute, DEC);
display.print(":");
}
if (minute>9)
{
display.setCursor(58,10);
display.print(minute, DEC);
display.print(":");
}
if (second<10)
{
display.setCursor(75,10);
display.print("0");
display.print(second, DEC);
}
if (second>9)
{
display.setCursor(75,10);
display.print(second, DEC);
}
//////////////////////////////////////////////////////////////////
```

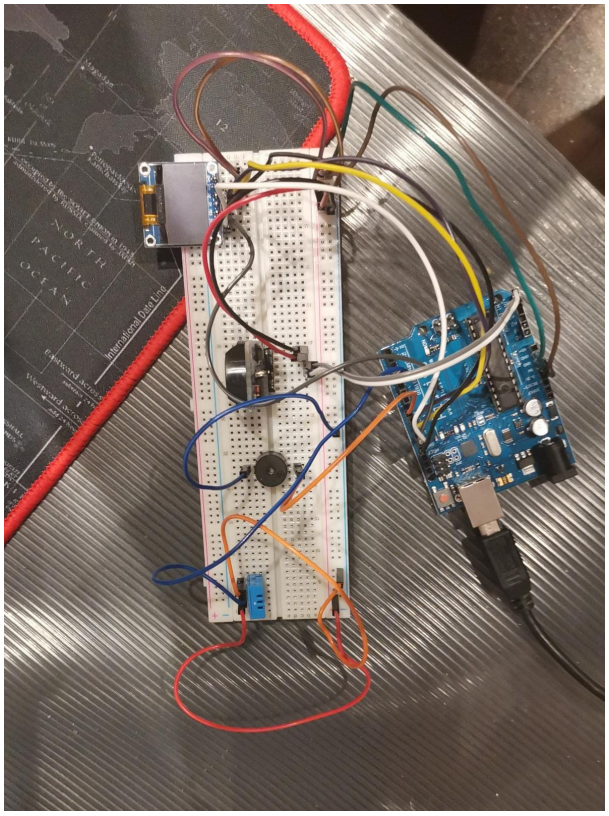
```
//////////////////////////////FOR DAY OF WEEK//////////////////////
switch(dayOfWeek){
case 1:
Serial.println("Sunday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Sun");
break;
case 2:
Serial.println("Monday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Mon");
break;
case 3:
Serial.println("Tuesday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Tue");
break;
case 4:
Serial.println("Wednesday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Wed");
break;
case 5:
Serial.println("Thursday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Thur");
break;
case 6:
Serial.println("Friday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Fri");
break;
case 7:
Serial.println("Saturday");
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(20,0);
display.print("Sat");
break;
}
//////////////////////////////FOR MONTH//////////////////////
switch(month)
{
```

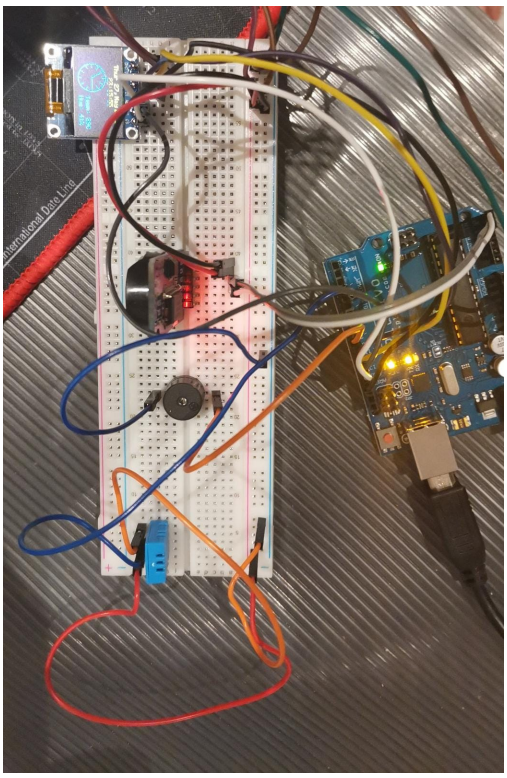
```
case 1:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Jan");
break;
case 2:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Feb");
break;
case 3:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Mar");
break;
case 4:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Apr");
break;
case 5:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("May");
break;
case 6:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Jun");
break;
case 7:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Jul");
break;
case 8:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Aug");
break;
case 9:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Sep");
break;
```

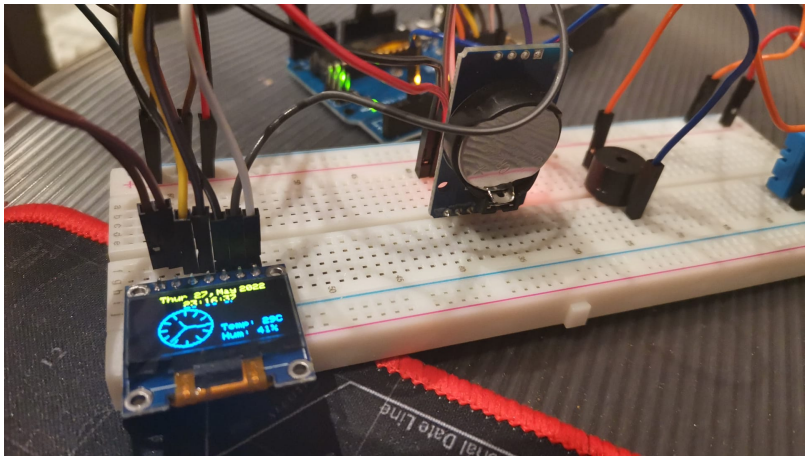
```
case 10:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Oct");
break;
case 11:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0); display.print("Nov");
break;
case 12:
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,0);
display.print("Dec");
break;
}
//////////////HOUR HAND////////////////
float x1, y1, a, b;
const float pi = 3.14;
a=((hour-15)*30);
b = (a*pi)/180;
x1=40+(9*cos(b));
y1=41+(9*sin(b));
display.drawLine(40,41, x1, y1, WHITE);
//////////////MINUTE HAND////////////////
// float x1, y1, a, b;
// const float pi = 3.14;
a=((minute-15)*6);
b = (a*pi)/180;
x1=40+(17*cos(b));
y1=41+(17*sin(b));
display.drawLine(40,41, x1, y1, WHITE);
//////////////SECOND HAND////////////////
//float x1, y1, a, b;
a=((second-15)*6);
b = (a*pi)/180;
x1=40+(19*cos(b));
y1=41+(19*sin(b));
display.drawLine(40,41, x1, y1, WHITE);
////////////////PARTS OF THE ANALOG CLOCK THAT WILL NOT BE MOVING/////////////////
display.drawCircle(40, 41, 22, WHITE);
display.drawCircle(40, 41, 1, WHITE);
display.drawLine(40, 20, 40, 25, WHITE); //12
display.drawLine(40, 63, 40, 58, WHITE); //6
display.drawLine(62, 41, 57, 41, WHITE); //3
display.drawLine(19, 41, 24, 41, WHITE); //9
display.drawLine(50, 24, 47, 28, WHITE); //1
display.drawLine(57, 31, 53, 34, WHITE); //2
display.drawLine(60, 51, 54, 48, WHITE); //4
display.drawLine(51, 58, 48, 54, WHITE); //5
display.drawLine(29, 58, 32, 54, WHITE); //7
display.drawLine(21, 51, 25, 48, WHITE); //8
display.drawLine(22, 31, 27, 33, WHITE); //10
```

```
display.drawLine(30, 23, 32, 28, WHITE); //11
}
void ReadTempAndHum ()
{
int chk = DHT.read11(DHT11_PIN);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(70,45);
display.print("Temp: ");
display.print(DHT.temperature, 0);
display.print("C");
display.setCursor(70,56);
display.print("Hum: ");
display.print(DHT.humidity, 0);
display.print("%");
Serial.print(" Humidity: " );
Serial.print(DHT.humidity, 1);
Serial.print("%");
Serial.print(" | Temperature: ");
Serial.print(DHT.temperature, 1);
Serial.println("C");
}
void loop() {
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
// retrieve data from DS3231
readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
&year);
displayTime(); // display the real-time clock data on the Serial Monitor,
DHTtimer = DHTtimer + 1;
//To ensure that the DHT is read only every 2 seconds, we make a count that goes for 2 seconds
if (DHTtimer > 1)
{
display.fillRect(70, 45, 128, 64, BLACK);
ReadTempAndHum ();
DHTtimer = 0;
}
delay(1000);
display.display();
//The portion of the screen that shows the time and date are cleared
display.fillRect(0, 0, 64, 64, BLACK);
display.fillRect(0, 0, 128, 20, BLACK);
if (hour== h && minute== m && second == s) {
tone(buzzer, 1000);
delay(1000);
}
}
```

Rezultate Obtinute







Concluzii

Proiectul a fost foarte distractiv de facut, am invatat o multime de lucruri pe parcursul acestuia si sper ca o sa mai am sansa de a lucra la alte proiecte arduino in viitor.

Download

Cod:

alarm_clock.rar

Librarii:

libraries.rar

Jurnal

10 mai - achizitionare componente

10 mai - realizare proiect

11 mai - adaptare cod, functionare partiala

15 mai - finalizare proiect, complet functional

Bibliografie/Resurse

github.com

create.arduino.cc

Export to PDF