Sistem de muzica ambientala in functie de factorii externi

Introducere

Sistemul de muzica ambientala in functie de factorii externi este un dispozitiv care se foloseste de un senzor DHT11 de temperatura si umiditate care preia datele din mediul extern si le afiseaza pe un ecran LCD 1602. In functie de datele colectate de la senzor se vor putea aprinde 4 LED-uri: un LED rosu pentru temperatura ridicata, un LED verde pentru temperatura scazuta/normala, un LED albastru pentru umiditate crescuta si un LED alb pentru umiditate scazuta/normala. Voi folosi un card microSD pe care voi incarca 4 melodii pentru muzica ambientala, iar pentru a citi de pe cardul microSD voi folosi un modul microSD. Melodiile vor fi redate prin intermediul unei boxe 40mm 3W. Cele 2 butoane le voi folosi pentru a selecta daca muzica ambientala va fi in functie de temperatura sau in functie de umiditate.

Ideea de la care am pornit in dezvoltarea proiectului a fost cardul microSD, mi s-a parut o componenta interesanta de integrat in proiect, iar la scurt timp mi-a venit ideea de a face un sistem de muzica ambientala.

Consider ca acest dispozitiv este unul util in casa sau masina oricarei persoane, muzica ambientala fiind o modalitate prin care ne putem relaxa, dar ne prin care ne putem creste si capacitatea de concentrare.

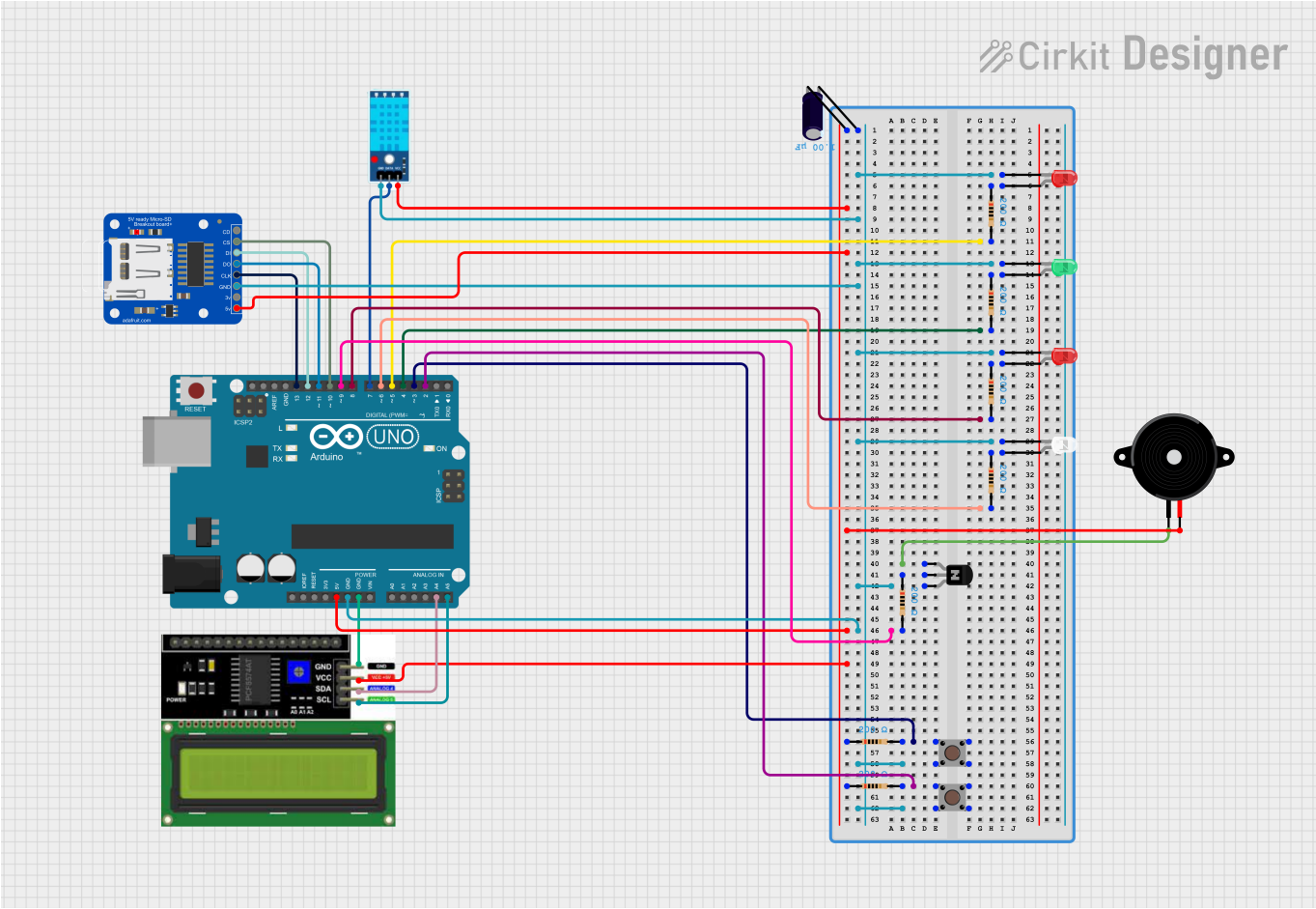
Descriere generală

Senzorul DHT11 transmite informatia colectata catre Arduino, iar acesta se va folosi de ea pentru a o transmite la modulul LCD si LED-uri. Apoi modulul microSD transmite catre Arduino datele sale si in functie de datele de la DHT11 si de starea butoanelor le va transmite catre speaker.

O schemă bloc cu toate modulele proiectului vostru, atât software cât şi hardware însoţită de o descriere a acestora precum şi a modului în care interacţionează.

Exemplu de schemă bloc: http://www.robs-projects.com/mp3proj/newplayer.html

Hardware Design



Componente utilizate:

- Arduino UNO ATMega328P
- senzor DHT11
- ecran LCD + modul I2C
- card microSD
- modul microSD
- LED-uri
- butoane
- breadboard

Aici puneţi tot ce ţine de hardware design:

- listă de piese
- scheme electrice (se pot lua şi de pe Internet şi din datasheet-uri, e.g. http://www.captain.at/electronic-atmega16-mmc-schematic.png)
- diagrame de semnal
- rezultatele simulării

Software Design

Pentru partea software initial am facut proiectul in 2 parti, prima data am facut partea am facut codul pentru a canta o melodie, componentele utilizate au fost boxa, sd card-ul si modulul pentru micro sd, butoane si placuta de arduino uno. Dupa ce am reusit sa fac aceasta parte a codului sa mearga am salvat-o si int-un alt fisier am realizat codul pentru ecranul lcd si senzorul dht11. Dupa ce am reusit sa afisez ceva pe ecran, am afisat si datele obtinute de la senzor. La final am facut un singur fisier in care am imbinat ambele coduri.

Bibliotecile utilizate sunt: #include <SD.h> *include SD module library #include <TMRpcm.h>* include speaker control library https://github.com/TMRh20/TMRpcm/archive/master.zip #include <LiquidCrystal_I2C.h> *include I2C LCD library #include <DHT.h>* include DHT sensor library

Pnetru butoane am folosit intreruperi: void button_humi(){

```
curr_time = millis();
if ((curr_time - last_updated) > debounce) {
  last_updated = curr_time;
  noInterrupts(); // disable interrupts temporarily
  if (humidity > humidity_treashold) {
    sound.stopPlayback();
    sound.play("third.wav");     // 8-bit, mono, 16-24khz at best
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Playing sound..3");
  } else {
    sound.stopPlayback();
    sound.play("fourth.wav");     // 8-bit, mono, 16-24khz at best
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Playing sound..4");
  }
  interrupts(); // re-enable interrupts
}
```

}

in setup() fac setez pinii si volumul boxei si sd card-ul.

void setup(){

```
//interrupt setup
pinMode(led_on_high_temp, OUTPUT);
pinMode(led_on_low_temp, OUTPUT);
pinMode(led_on_high_humi, OUTPUT);
pinMode(led_on_low_humi, OUTPUT);
pinMode(button_tem, INPUT_PULLUP);
pinMode(button_hum, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(button_tem), button_temp, FALLING);
attachInterrupt(digitalPinToInterrupt(button_hum), button_humi, FALLING);
```

```
// LCD setup
lcd.init();
```

```
lcd.backlight();
lcd.print("Initializing SD");
```

```
// Open serial communications and wait for port to open:
Serial.begin(9600);
while (!Serial) {
  ; // wait for serial port to connect. Needed for native USB port only
}
Serial.println("Initializing SD card...");
if (!SD.begin(SD_ChipSelectPin)) {      //see if the card is present and can
be initialized
  Serial.println("Initialization failed...");
  lcd.clear();
  lcd.print("SD init failed");
  return;                               //don't do anything more if not
}
else {
    Serial.println("Initialization done...");
    lcd.clear();
    lcd.print("SD init done");
}
```

```
dht.begin();
sound.speakerPin = 9;
sound.quality(1);
sound.setVolume(5);
```

}

Functia de loop():

void loop(){

```
if (!SD.begin(SD_ChipSelectPin)) {      //see if the card is present and can
be initialized
  lcd.clear();
  lcd.print("SD init failed");
  return;                               //don't do anything more if not
}
else {
    lcd.clear();
    lcd.print("SD init done");
}

if(!sound.isPlaying()) {
  // Read temperature and humidity from DHT sensor
  humidity = dht.readHumidity();
  temperature = dht.readTemperature();
```

```
  // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperature)) {
```

```
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
```

```
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print(" C");
  lcd.setCursor(0,1);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print(" %");
```

```
  if (temperature > temperature_threshold) {
    digitalWrite(led_on_high_temp, HIGH); // Turn on the LED on pin 5
    digitalWrite(led_on_low_temp, LOW);
  } else {
    digitalWrite(led_on_high_temp, LOW);
    digitalWrite(led_on_low_temp, HIGH); //Turn on the LED on pin 4
  }
```

```
  if (humidity > humidity_treashold) {
    digitalWrite(led_on_high_humi, HIGH); // Turn on the LED on pin 8
    digitalWrite(led_on_low_humi, LOW);
  } else {
    digitalWrite(led_on_high_humi, LOW);
    digitalWrite(led_on_low_humi, HIGH); //Turn on the LED on pin 6
  }
```

```
  delay(5000);
```

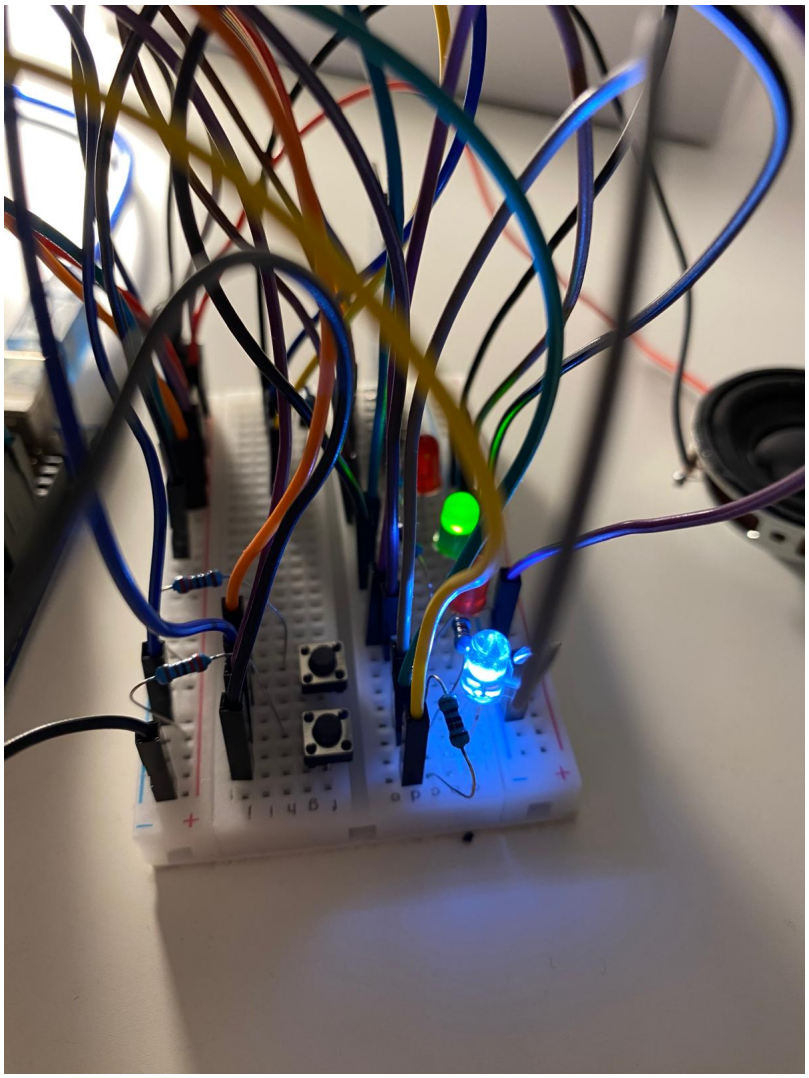
```
  // Display temperature and humidity in the Serial Monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" C");
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");
}
else {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Playing sound...");
}
```

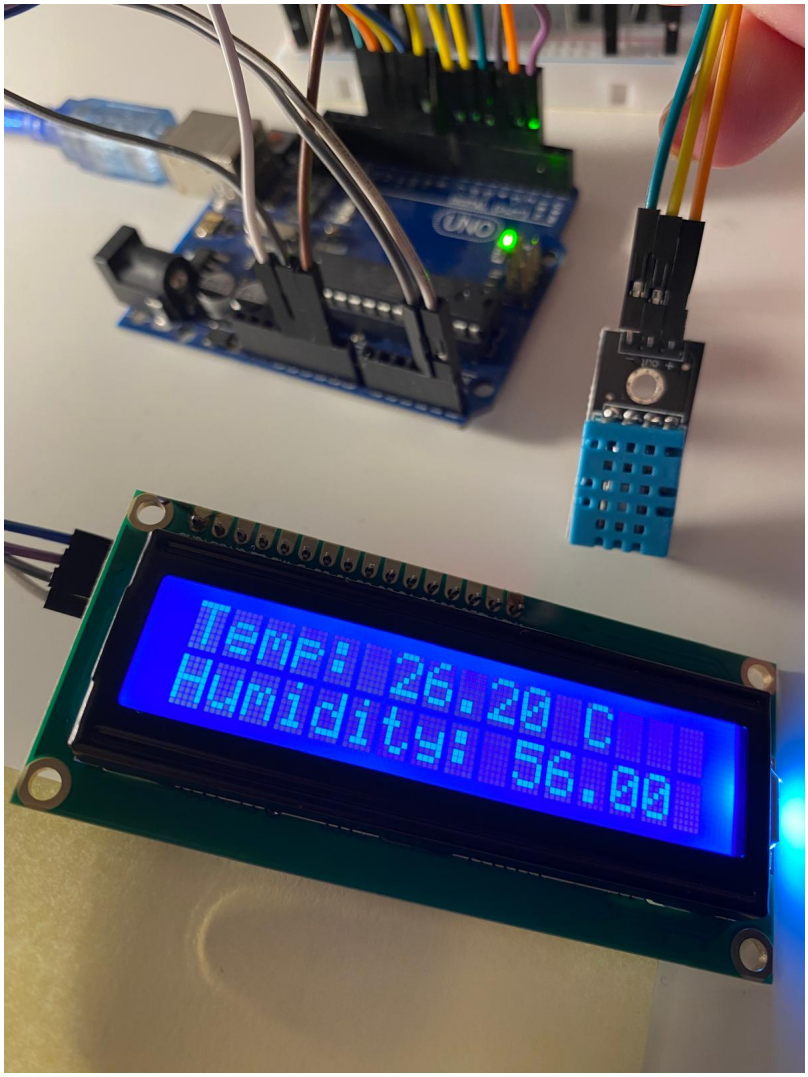
```
}
```

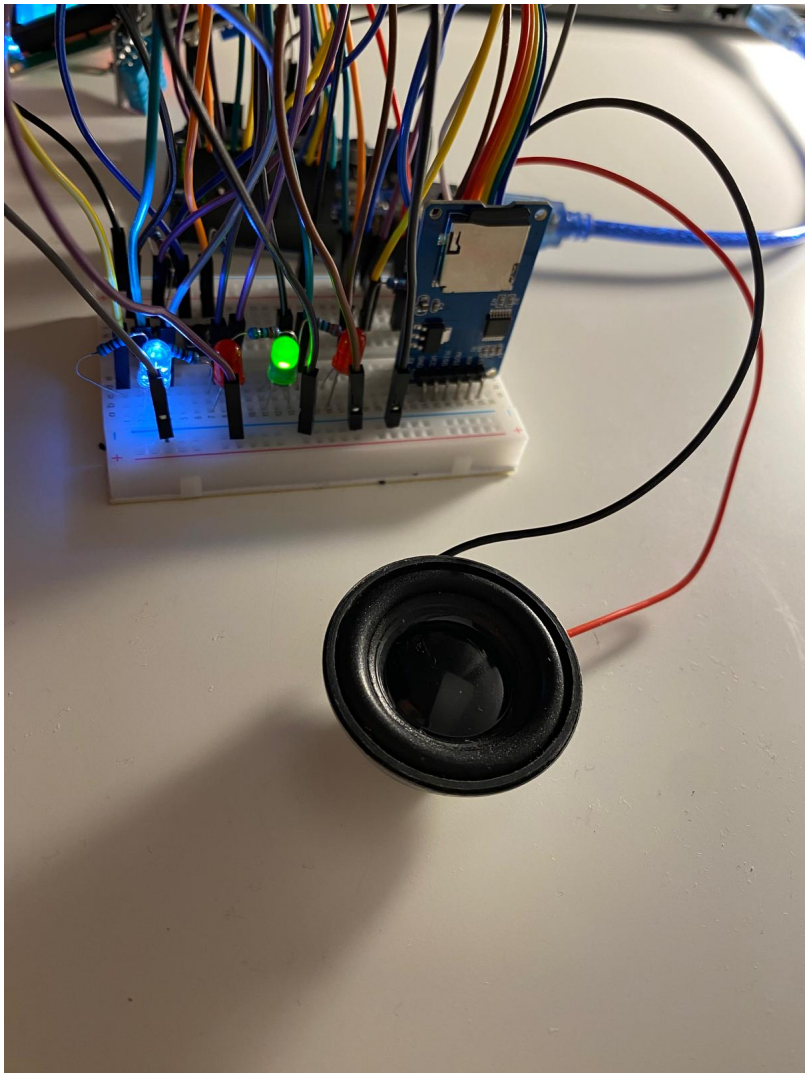
Rezultate Obţinute

Rezultatele obţinute în urma realizării proiectului.

https://www.youtube.com/watch?v=ALfjnH8IBiQ&ab_channel=StoianAmalia




Temp: 26.20 C
Humidity: 56.00



Download

stoian_georgiana-amalia_proiect_pm.zip

Bibliografie/Resurse

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

https://www.youtube.com/watch?v=tNNycZpvg-Y&ab_channel=AndrobotTECH

https://www.youtube.com/watch?v=pOFFsiolsGU&ab_channel=AndrobotTECH

https://www.youtube.com/watch?v=CvqHkXeXN3M&t=481s&ab_channel=Tronicslk

Export to PDF