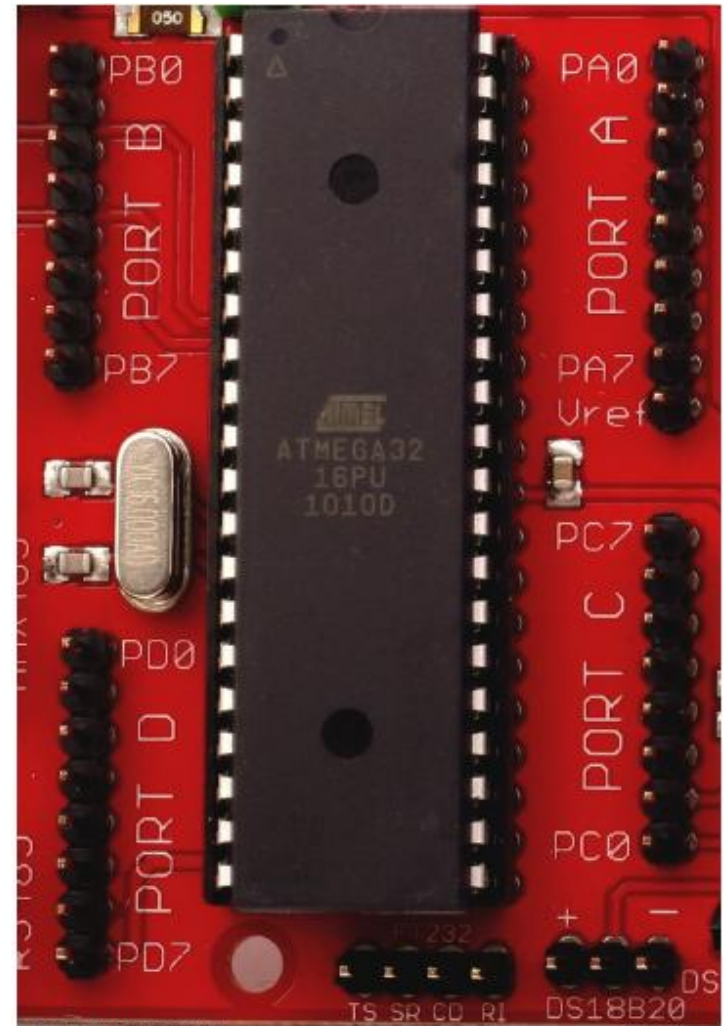




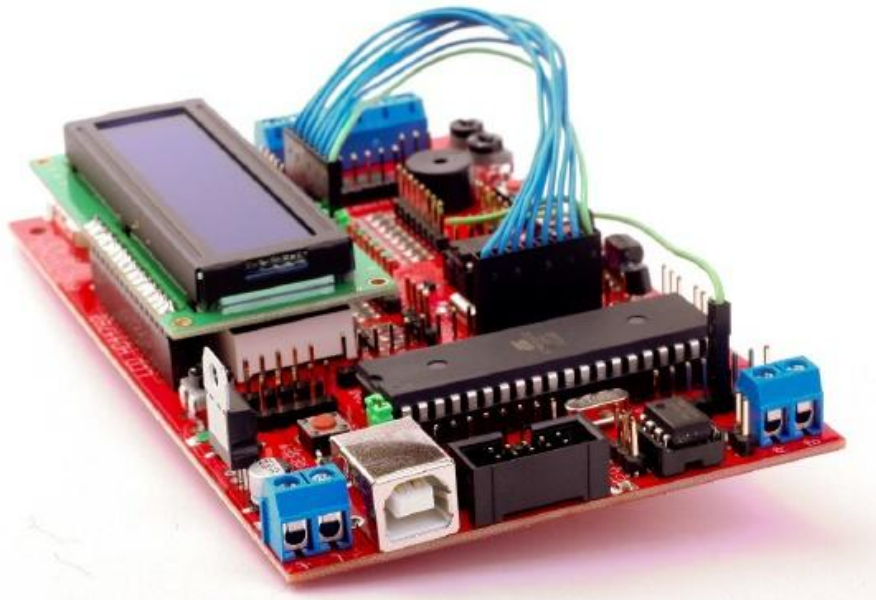
Project SW Embedded Simulator Control Clima

Scopul Proiectului

- crearea unui dispozitiv care detectează diferențele de temperatură dintre mediul înconjurător și o temperatură țintă
- acesta face anumite operații în funcție de diferența dintre cele două: răcire (COOL) sau încălzire (HEAT)
- în funcție de temperaturi, se aprind LED-uri sugestive operației



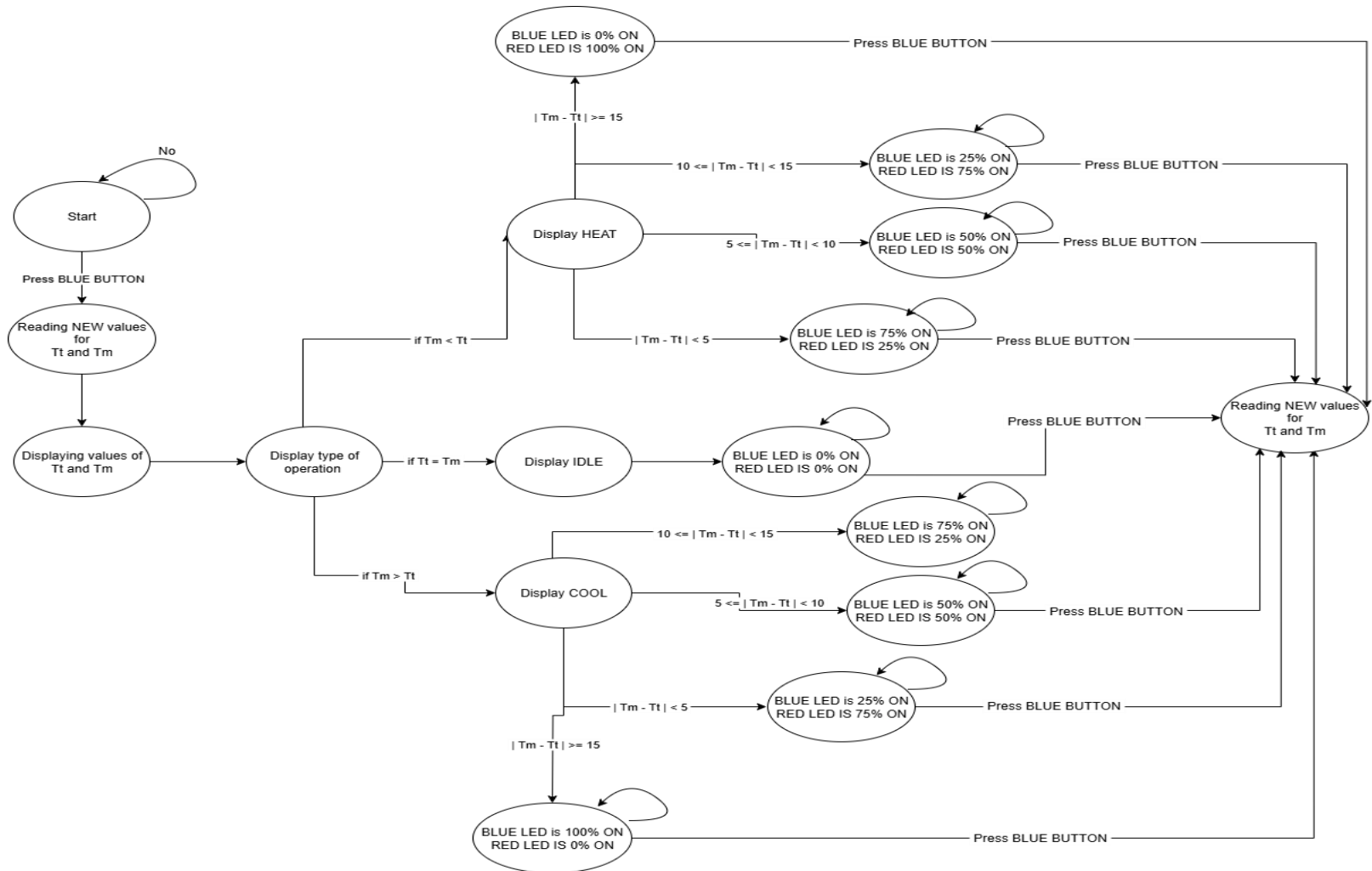
Componente



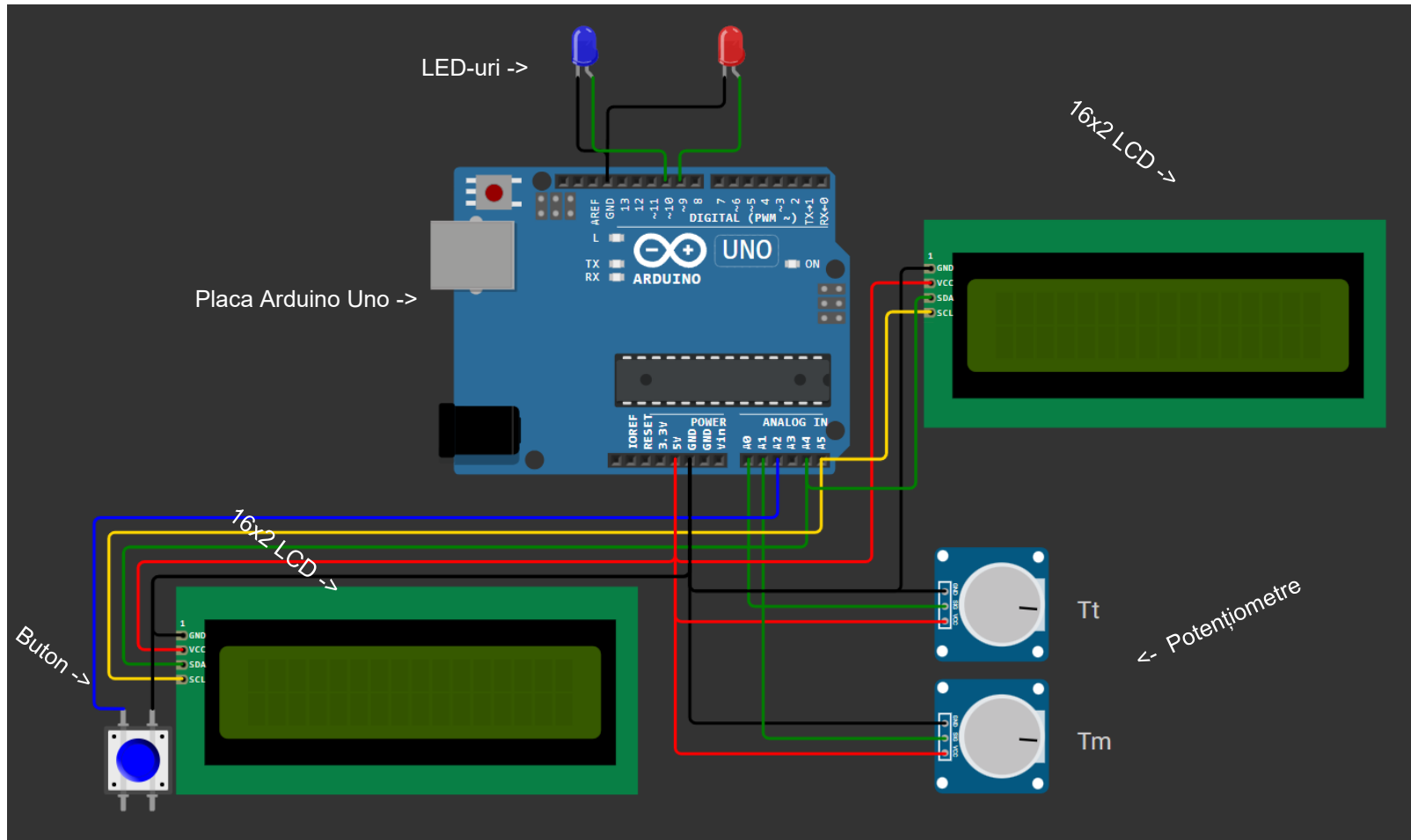
Sistemul este alcătuit din:

- › Placă Arduino Uno ce conține:
- › EvB 5.1 ATmega32
- › 2 16x2cm LCDs (I2C)
- › 2 LED-uri
- › 2 Potențiometre ADC
- › fire

Diagrama FSM a sistemului

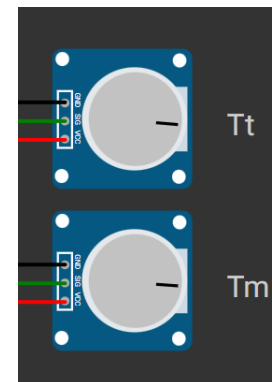


Arhitectura dispozitivului



Cum funcționează sistemul?

- › Utilizatorul generează de la cele 2 potențiometre semnale analogice ce reprezintă **Temperatura mediului** (T_m) și **Temperatura țintă** (T_t)



- › După apăsarea butonului , sunt afișate pe cele 2 ecrane:



Temperaturile, alternativ



Tipul de operațiune ce are loc

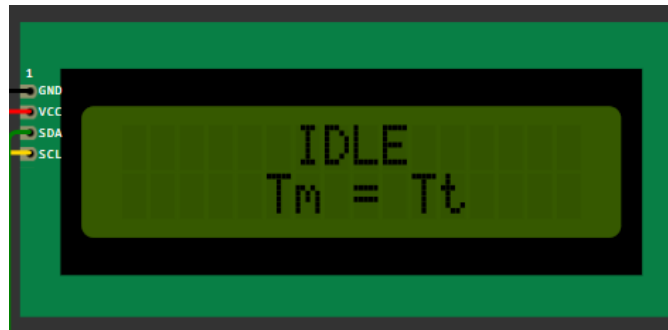
Cum funcționează sistemul?

› În funcție de temperaturi, sistemul poate face următoarele operații:

a. COOL: $T_m > T_t$;

b. IDLE: $T_m = T_t$;

c. HEAT: $T_m < T_t$.



Cum funcționează sistemul?

- › În funcție de operațiunea ce are loc și diferența numeric absolută dintre cele două valori, se vor aprinde cele 2 LED-uri după următoarele reguli:

COOL:

if $|T_m - T_t| < 5C$ then:

- BLUE_LED 25% ED (romana: ED = factor de umplere);
- RED_LED 75% ED.

2. if $5C \leq |T_m - T_t| < 10C$ then:

- BLUE_LED 50% ED;
- RED_LED 50% ED;

3. if $-10C \leq |T_m - T_t| < 15C$ then:

- BLUE_LED 75% ED;
- RED_LED 25% ED;

4. if $|T_m - T_t| \geq 15C$ then:

- BLUE_LED 100% ED;
- RED_LED 0% ED;

HEAT:

if $|T_m - T_t| < 5C$ then:

- BLUE_LED 75% ED (romana: ED = factor de umplere);
- RED_LED 25% ED.

2. if $5C \leq |T_m - T_t| < 10C$ then:

- BLUE_LED 50% ED;
- RED_LED 50% ED;

3. if $10C \leq |T_m - T_t| < 15C$ then:

- BLUE_LED 25% ED;
- RED_LED 75% ED;

4. if $|T_m - T_t| \geq 15C$ then:

- BLUE_LED 0% ED;
- RED_LED 100% ED;

IDLE

- BLUE_LED 0% ED;
- RED_LED 0% ED;
- :



Teste asupra sistemului

› Testarea funcționării potențimetrelor cu anumite valori

- Acesta poate lua valori între 0 și 1023, conversia la temperaturi fiind în intervalul [-40C , 87C]

```
void test_convertADCToTemp() {  
    Serial.println("Testing convertADCToTemp:");  
    int adcVals[] = {0, 160, 320, 400, 640, 1023};  
    for(int i = 0; i < 6; i++) {  
        int val = adcVals[i];  
        Serial.print("ADC = ");  
        Serial.print(val);  
        Serial.print(" -> Temp = ");  
        Serial.println(convertADCToTemp(val));  
    }  
}
```

```
Testing convertADCToTemp  
ADC = 0 -> Temp = -40  
ADC = 160 -> Temp = -20  
ADC = 320 -> Temp = 0  
ADC = 400 -> Temp = 10  
ADC = 640 -> Temp = 40  
ADC = 1023 -> Temp = 87
```

Teste asupra sistemului

› Testarea butonului

- Verificăm dacă butonul (de tip toggle) funcționează

```
void ButtonPressed(){  
    if((PINC & (1<<BLUE_BUTTON_PIN)) == 0)  
    {  
        Serial.println("Buton APASAT");  
        delay(50);  
    }  
}
```

Buton APASAT

› Verificarea valorilor și a valorii diferenței absolute dintre temperaturi

```
void testingValues(){  
    Serial.print("Tm: ");  
    Serial.print(Tm);  
    Serial.print(" | Tt: ");  
    Serial.println(Tt);  
}
```

```
int32_t difference = abs(Tm - Tt);  
Serial.print("|Tm - Tt| = ");  
Serial.println(difference);
```

Tm: 87 | Tt: 87
|Tm - Tt| = 0

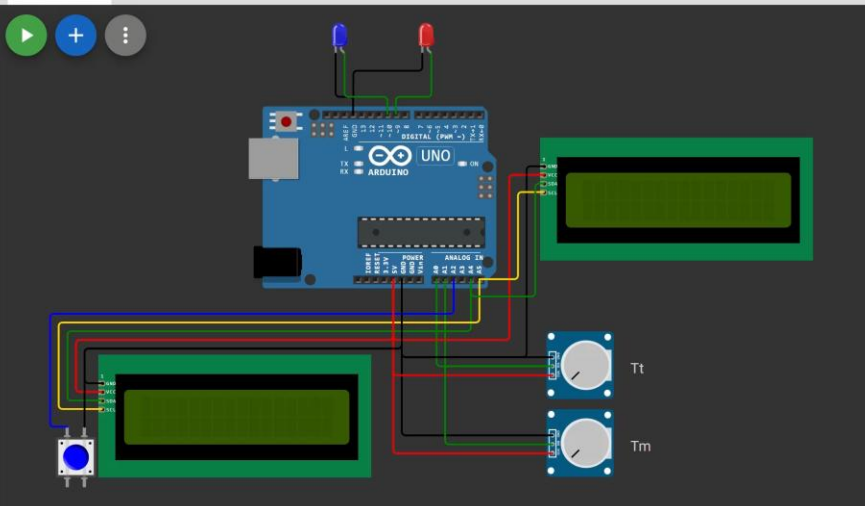
Demonstrație

WOKWI SAVE SHARE Petru Copy Docs P

sketch.ino • diagram.json libraries.txt Library Manager

```
6 #define ENABLE_PWM_BLUE() (TCCR1A |= (1 << COM1B1))
7 #define DISABLE_PWM_BLUE() (TCCR1A &= ~(1 << COM1B1))
8
9 #define BLUE_BUTTON_PIN PC2
10 #define RED_LED_PIN PB1
11 #define BLUE_LED_PIN PB2
12
13 LiquidCrystal_I2C lcd(0x27, 16, 2);
14 LiquidCrystal_I2C lcd2(0x28, 16, 2);
15
16 uint16_t channel=0;
17 int32_t Tt,Tm;
18 uint16_t toggleState=0;
19 uint16_t lastButtonState=1;
20
21 void setup() {
22
23   Serial.begin(9600);
24
25   lcd.begin(16,2);
26   lcd2.begin(16,2);
27
28   // config. the ADC
29   ADMUX = (1<<REFS0) | (0<<MUX0) | (0<<MUX1) | (0<<MUX2) | (0<<MUX3); // potentiometru A0
30   ADCSRA |= (1<<ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
31
32   // config. blue button on PORTC as INPUT
33   DDRC &= ~(1<<BLUE_BUTTON_PIN);
34   PORTC |= (1<<BLUE_BUTTON_PIN);
35
36   // config PORTB output (for LEDs)
37   DDRB |= (1<<RED_LED_PIN) | (1<<BLUE_LED_PIN);
38
39
40   // config. registers TCCR1A & TCCR1B for PWM
41   TCCR1A |= (0 << WGM11) | (1 << WGM10) | (1 << COM1A1) | (0 << COM1A0) | (1 << COM1B1) | (0 << COM1B0);
42   TCCR1B |= (0 << WGM13) | (1 << WGM12) | (1 << CS11);
43 }
```

Simulation



Buton APASAT

Tm: 87 | Tt: 87

|Tm - Tt| = 0

Buton APASAT

Tm: 87 | Tt: 87

|Tm - Tt| = 0

Tm: 87 | Tt: 87

|Tm - Tt| = 0