Parking Management System - Pavăl Bogdan-Costin

Introducere

Proiectul presupune implementarea unui sistem de parcare, asemănător celor din toate mall-urile. Când ajunge o mașină în dreptul barierei, în funcție de numărul de locuri disponibile în parcare, se ridică bariera pentru a permite intrarea (și se dirijează conducătorul către locul pe care trebuie să parcheze), iar dacă parcarea este plină, se informează conducătorul printr-un mesaj sugestiv că parcarea este plină și trebuie să aștepte până se eliberează un loc.

Scopul proiectului este de a oferi o implementare în miniatură a sistemului de gestiune a unei parcări, pentru a se putea vedea în ansamblu viitoarea formă finală (când va fi pus în aplicare). Astfel, pot spune că este foarte util acest proiect, pentru că fiecare dintre noi interacționează cu ceva similar când mergem la cumpărături și ne lăsăm mașina în parcarea magazinelor. Am pornit cu această idee după ce am fost la un magazin, unde sistemul barierei de la intrarea parcării nu funcționa și mi-am

propus, ca pentru acest proiect, să implementez și eu ceva asemănător.

Descriere generală

În momentul în care ajunge o mașină în dreptul barierei, un senzor infraroșu (care este înaintea barierei) detectează mașina și comandă servo motorului să ridice bariera (dacă există cel puțin un loc) și afișează pe ecranul LCD un mesaj cu locul pe care poate parca sau îl informează că trebuie să mai aștepte. La ieșirea din parcare, este alt senzor infraroșu, care detectează când o mașină vrea să plece și comandă servo motorului să ridice bariera.

Fiecare loc de parcare are un led verde, care este aprins când locul nu este ocupat, iar când o mașină intră în parcare și trebuie să parcheze pe locul indicat de sistem, led-ul corespunzător se stinge. La ieșirea din parcare se întâmplă procesul invers, adică se aprinde led-ul locului proaspăt eliberat.

Schema bloc

Hardware Design

Listă de piese

- **Arduino** UNO
- 2 **senzori infraroșu**
- 1 **LCD** 1602 I2C
- 1 **servo motor** SG90 9g
- 7 **rezistențe** de 330 Ω
- 6 **LED**-uri verzi
- 1 **LED** roșu
- **fire**

Conexiuni

- **LCD**: SDA → **A4**, SCL → **A5**
- **IR înainte de barieră**: OUT → **D2**
- **IR după barieră**: OUT → **D3**
- **Servo motor**: IN → **D5**
- **LED** semafor **verde** → **D6**

- **LED** semafor **roșu** → **D7**
- **LED** loc **1** → **D13**
- **LED** loc **2** → **D12**
- **LED** loc **3** → **D11**
- **LED** loc **4** → **D10**
- **LED** loc **5** → **D9**

Schema electrică

Schema Tinkercad

Software Design

Întreruperi

Pentru cei doi senzori infraroșu am folosit întreruperile externe INT0 și INT1 (fiind conectate la pinii PD2 și PD3). În registrul EICRA am setat întreruperea pe front descrescător:

```
ISR(INT0_vect)
{ // Rutina de tratare a intreruperii externe pentru ir 1 (PD2)
  my_int0 = true;
}


ISR(INT1_vect)
{ // Rutina de tratare a intreruperii externe pentru ir 2 (PD3)
  my_int1 = true;
}


void setup_interrupts()
{
  cli();

  // Configurare ir1 1 (PD2) - intrerupere externa
  DDRD &= ~(1 << PD2);         // Seteaza pinul PD2 ca intrare
  PORTD |= (1 << PD2);         // Activeaza rezistenta de pull-up pentru PD2
```

```
  EICRA |= (1 << ISC01);       // Configureaza intreruperea externa pe front
descrescator (falling edge)
  EIMSK |= (1 << INT0);        // Activeaza intreruperea externa INT0

  // Configurare ir 2 (PD3) - intrerupere externa
  DDRD &= ~(1 << PD3);         // Seteaza pinul PD3 ca intrare
  PORTD |= (1 << PD3);         // Activeaza rezistenta de pull-up pentru PD3

  EICRA |= (1 << ISC11);       // Configureaza intreruperea externa pe front
descrescator (falling edge)
  EIMSK |= (1 << INT1);        // Activeaza intreruperea externa INT1

  sei();
}
```

Setup

În partea de setup am apelat funcția **setup_interrupts()** pentru configurarea întreruperilor, apoi am configurat pinii asociați LED-urilor (ieșire) și senzorilor (intrare):

```
  DDRB |= (1 << PB5) | (1 << PB4) | (1 << PB3) | (1 << PB2) | (1 << PB1);
  DDRD |= (1 << PD6) | (1 << PD7);
  /* echivalentul cu functii:
  pinMode(LED_SLOT1, OUTPUT);
  pinMode(LED_SLOT2, OUTPUT);
  pinMode(LED_SLOT3, OUTPUT);
  pinMode(LED_SLOT4, OUTPUT);
  pinMode(LED_SLOT5, OUTPUT);
  pinMode(LED_SEM_GREEN, OUTPUT);
  pinMode(LED_SEM_RED, OUTPUT);
  */

  DDRD &= ~(1 << PD2) & ~(1 << PD3);
  /* echivalentul cu functii:
  pinMode(IR_BEFORE, INPUT);
  pinMode(IR_AFTER, INPUT);
  */
```

După configurare, se activează (aprind) LED-urile verzi și se configurează servomotorul și LCD-ul (împreună cu mesajul de întâmpinare):

```
  PORTB |= (1 << PB5) | (1 << PB4) | (1 << PB3) | (1 << PB2) | (1 << PB1);
  PORTD |= (1 << PD6);
  PORTD &= ~(1 << PD7);
  /* echivalentul cu functii:
  digitalWrite(LED_SLOT1, HIGH);
  digitalWrite(LED_SLOT2, HIGH);
```

```
  digitalWrite(LED_SLOT3, HIGH);
  digitalWrite(LED_SLOT4, HIGH);
  digitalWrite(LED_SLOT5, HIGH);
  digitalWrite(LED_SEM_GREEN, HIGH);
  digitalWrite(LED_SEM_RED, LOW);
  */

  servo.attach(SM);
  servo.write(110);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("Am PoliPark-at"); // mesajul initial
  lcd.setCursor(2, 1);
  lcd.print("Pret: 10 lei");
  delay(7000);
  lcd.clear();
```

Loop

Prima dată este tratat cazul primului senzor (cel de dinainte de barieră), unde dacă variabila **my_int0** a fost setată pe **true** în întrerupere înseamnă că a fost detectată o mașină la intrare. Se ridică bariera dacă mai sunt locuri libere și menține semaforul pe verde, iar dacă nu, se afișează un mesaj de informare și se modifică semaforul pe roșu:

```
  if (my_int0 == true && just_entered == 0) { // este masina la intrare
    if (total_slots > 0) {
      if (just_exited == 0) { // nu vine din interior
        servo.write(15);
        total_slots--;
        come = 1;
      }

      just_entered = 1;
      PORTD &= ~(1 << PD7); //digitalWrite(LED_SEM_RED, LOW);
      PORTD |= (1 << PD6); //digitalWrite(LED_SEM_GREEN, HIGH); -> semafor
verde
    } else {
      PORTD |= (1 << PD7); //digitalWrite(LED_SEM_RED, HIGH); -> semafor
rosu
      PORTD &= ~(1 << PD6); //digitalWrite(LED_SEM_GREEN, LOW);

      lcd.clear();
      lcd.setCursor(2, 0);
      lcd.print("Parcare plina!");
      lcd.setCursor(0, 1);
      lcd.print("Mai incearca!");
```

```
      delay(3000);
      lcd.clear();
      my_int0 = false;
    }
  }
```

Apoi este tratat în mod asemănător cazul senzorului de după barieră:

```
  if (my_int1 == true && just_exited == 0) { // o masina pleaca
    if (total_slots != 0) { // este cel putin un loc liber
      PORTD &= ~(1 << PD7); //digitalWrite(LED_SEM_RED, LOW);
      PORTD |= (1 << PD6); //digitalWrite(LED_SEM_GREEN, HIGH); -> semafor
verde
    }

    just_exited = 1;
    if (just_entered == 0) { // ridica bariera
      servo.write(15);
      total_slots++;
      leave = 1;
    }
  }
```

Următorul pas este să se verifice dacă ambii senzori au detectat că a trecut mașina prin fața lor, adică intrarea/ieșirea a fost făcută cu succes.

Dacă mașina pleacă, se afișează un mesaj sugestiv, iar dacă mașina intră atunci șoferul este informat pe ce loc trebuie să parcheze. Tot în acest bloc sunt și condițiile pentru aprinderea/stingerea LED-ului corespunzător locului de parcare:

```
  if (just_entered == 1 && just_exited == 1) { // a trecut prin fata ambilor
senzori
    if (leave == 1 && total_slots > 0) { // pleaca o masina
        lcd.setCursor(0, 0);
        lcd.print("Va mai asteptam!");
        lcd.setCursor(0, 1);
        lcd.print("                ");
      }

      if (come == 1 && total_slots < 5) { // intra o masina
        lcd.setCursor(1, 0);
        lcd.print("Parcheaza pe ");
        lcd.print(MAX_SLOTS - total_slots);
        lcd.print(".");
        lcd.setCursor(0, 1);
        lcd.print("                ");
      }

    delay(2000);
    servo.write(110); // coboara bariera
```

```
    /* aici sunt conditiile pentru aprinderea/stingerea LED-urilor */

    my_int0 = false;
    my_int1 = false;
    come = 0;
    leave = 0;
    just_entered = 0;
    just_exited = 0;
  }
```

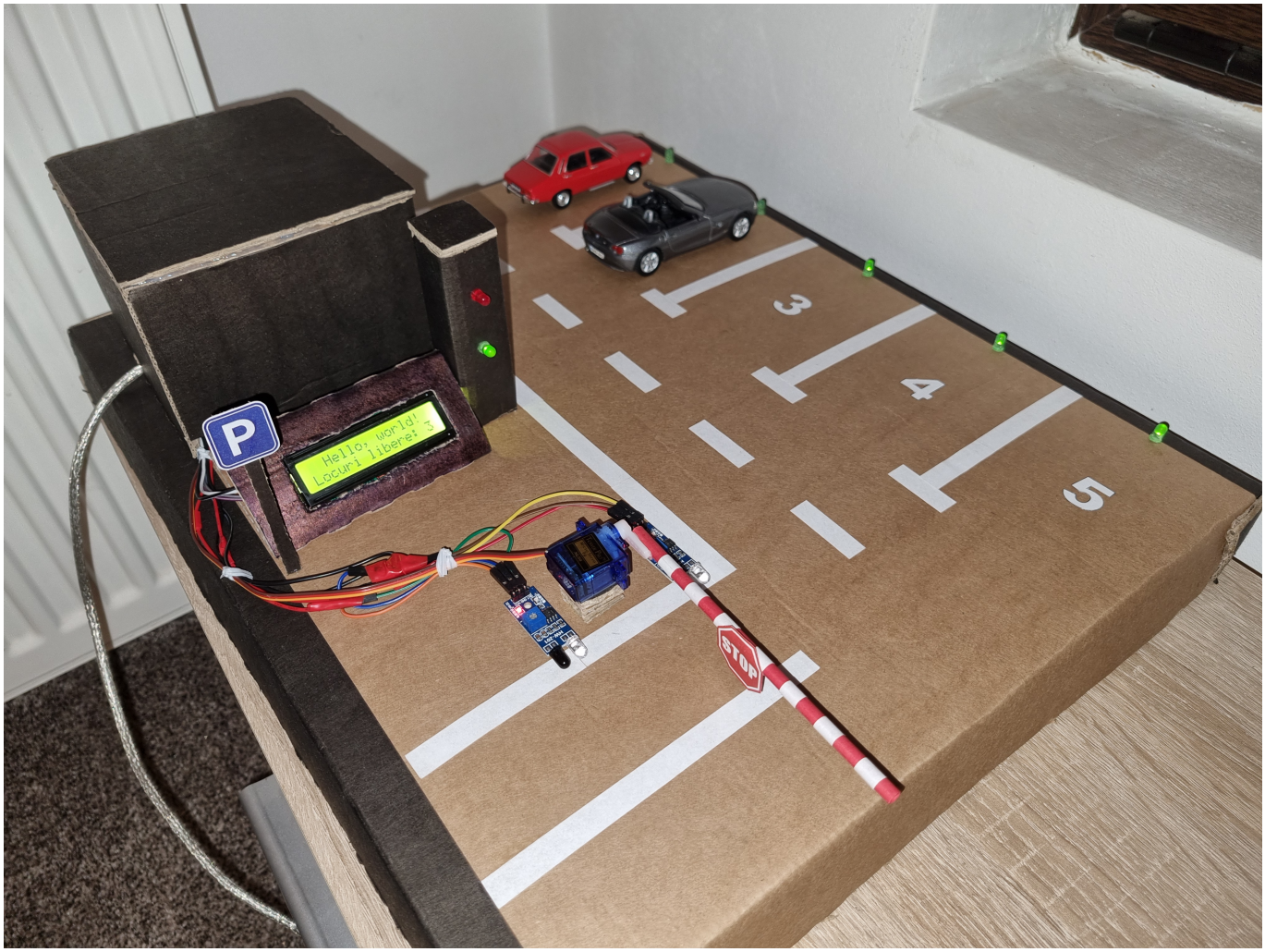
La finalul loop-ului, se verifică dacă au fost ocupate toate locurile și în acest caz se aprinde LED-ul roșu al semaforului (în restul timpului fiind pe verde), urmând să fie afișat un mesaj cu numărul de locuri disponibile în parcare:

```
  if (total_slots == 0) { // parcarea este plina
    PORTD |= (1 << PD7); //digitalWrite(LED_SEM_RED, HIGH); -> semafor rosu
    PORTD &= ~(1 << PD6); //digitalWrite(LED_SEM_GREEN, LOW);
  }

  lcd.setCursor(0, 0);
  lcd.print("  Hello, world! ");
  lcd.setCursor(0, 1);
  lcd.print("Locuri libere: ");
  lcd.print(total_slots);
```

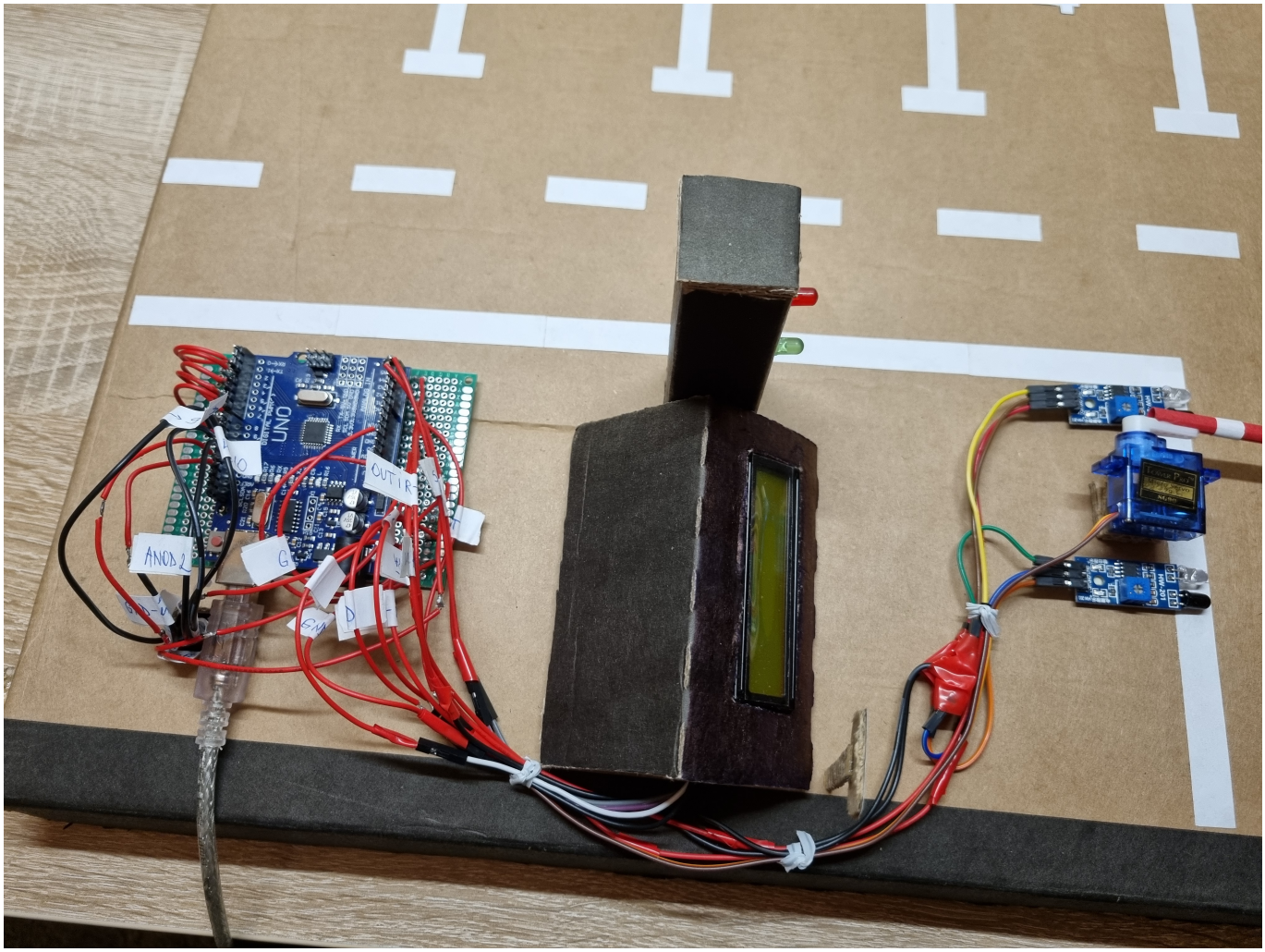
Rezultate Obţinute

Video YouTube: https://www.youtube.com/watch?v=zYuN_764P6E

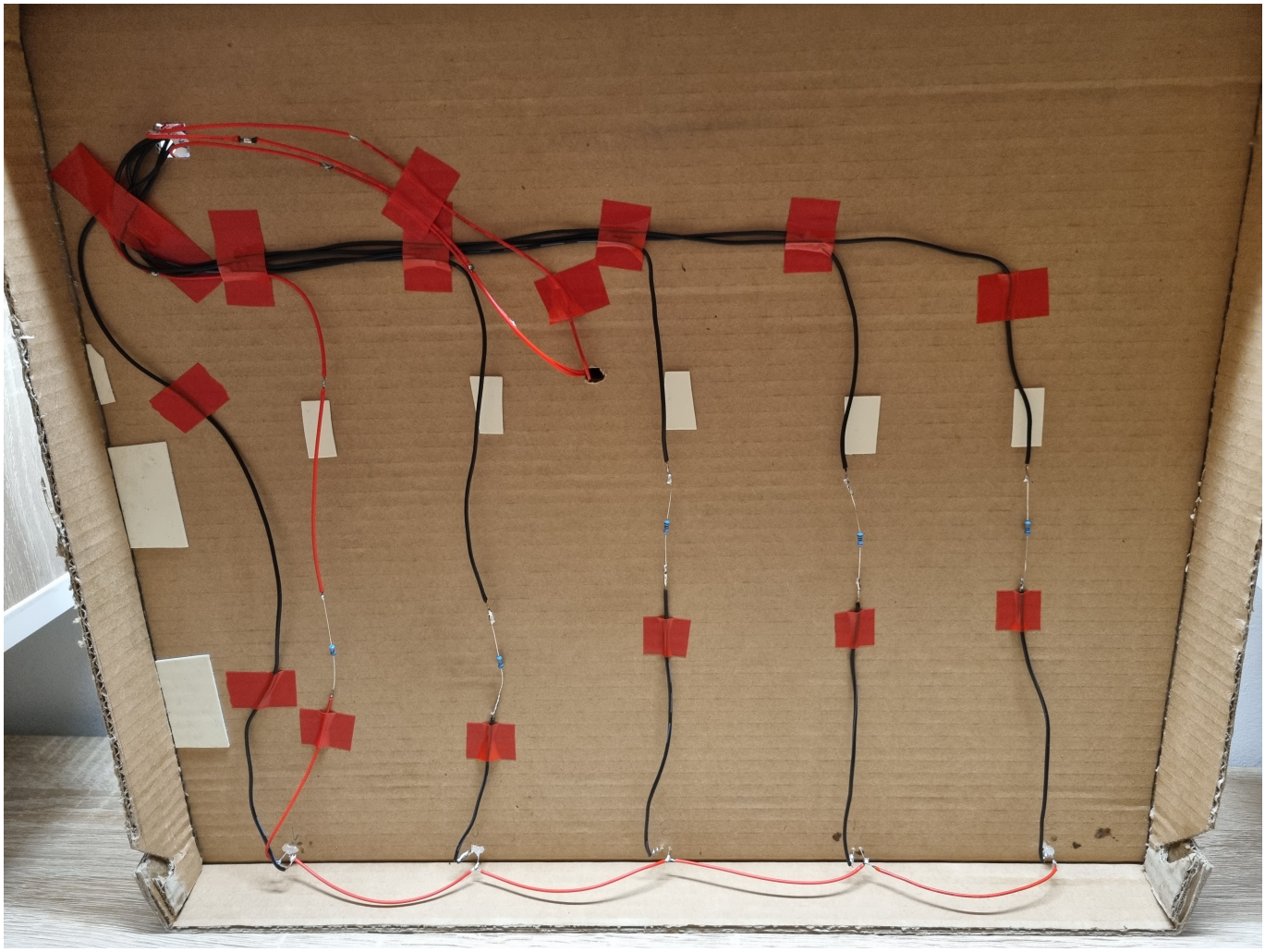


Când parcarea este plină, se afișează pe ecran:

În partea stângă este Arduino UNO pus pe plăcuța de prototipare (are trasee pe partea din spate), iar în partea dreaptă sunt lipiți pe machetă senzorii și servomotorul:



Partea din spate, cu firele pentru LED-uri și semafor:



Concluzii

Pentru că am vrut să concentrez firele pe plăcuța de prototipare, a fost mai complicat până m-am obișnuit cu lipitul în locuri așa de mici, dar m-am descurcat.

Mai mult a durat partea de construire a machetei și proiectarea legăturilor în cea mai bună formă, decât a durat conectarea efectivă a componentelor.

Download

paval_bogdan-costin_332ca.zip

Jurnal

- 22.04 Alegerea temei
- 07.05 Documentația
- 13.05, 14.05 Macheta din carton
- 15.05, 16.05 Lipire fire între componente și Arduino + testare componente
- 19.05 Fixare componente pe machetă
- 20.05 Început partea software
- 21.05, 22.05 Implementare software
- 28.15 Finalizare documentație

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** şi **Resurse Hardware**.

Paginile de specificații ale componentelor de pe site-urile comercianților, Stack Overflow, dar și:

https://www.tinkercad.com/things/luKLYUdSUZK?sharecode=c-IHSCxoYkAW3woCv-NyFcamGzlVYzZOm-233Tw6YBw

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

https://ocw.cs.pub.ro/courses/pm/lab/lab0-2022

https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023

https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023

https://ocw.cs.pub.ro/courses/pm/lab/lab6-2022

https://forum.arduino.cc/

Export to PDF