Basketball Arcade Game

Introducere

Proiectul constă într-un joc de tip arcade în care jucătorul trebuie să marcheze cât mai multe puncte într-o anumită perioadă de timp(60s).

Proiectul utilizează un senzor ultrasonic pentru a intercepta mișcarea mingii care trece prin coșul de baschet, numără punctele marcate și le afișează pe un display digital.

Ideea acestui proiect a pornit din pasiunea mea pentru baschet și din dorința de a crea ceva cu care îmi mai pot ocupa timpul liber la cămin.

Descriere generală

Sub inelul coșului de baschet va fi atașat un senzor ultrasonic. După ce un coș va fi marcat, senzorul va înregistra mișcarea mingii de baschet și va transmite un semnal către placa Arduino Uno folosită, iar mai apoi scorul va fi actualizat pe un display digital. De asemenea după marcarea coșului, prin intermediul unui buzzer, jucătorii vor fi anunțați că au câștigat un punct prin emiterea unui sunet.

Pentru a face lucrurile mai interesante, jucătorii vor avea de marcat cât mai multe puncte într-o perioadă de 60 de secunde. Timpul rămas pentru efectuarea aruncărilor va fi afișat pe un display LCD dar va fi indicat și printr-un sistem de iluminare LED care își va schimba culoare în funcție de timpul rămas(între 60 și 40 de secunde ,verde, între 40 și 20 de secunde ,galben și între 20 și 0 secunde, roșu).

Schema:

Hardware Design

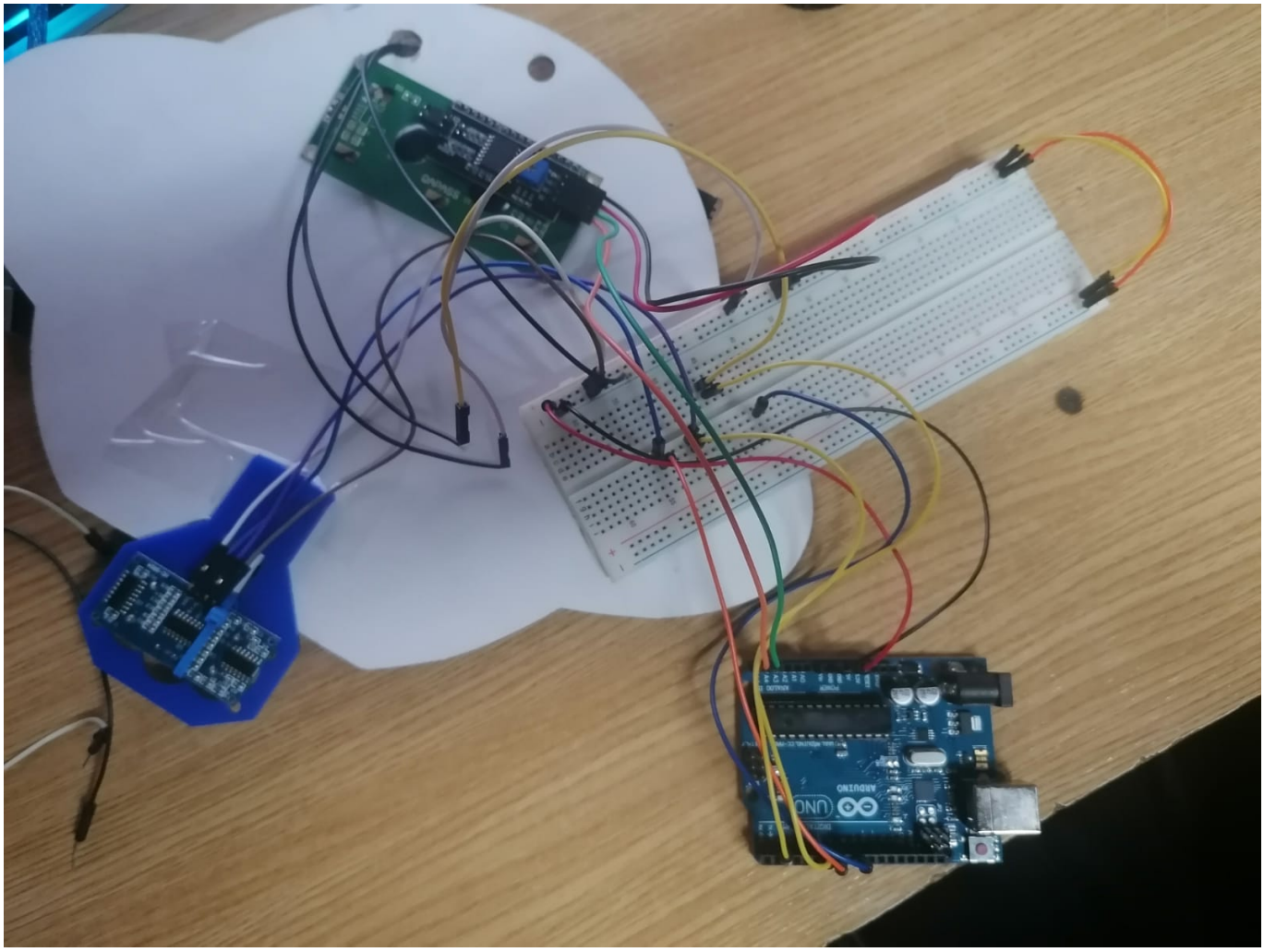
Componente folosite:

- Arduino UNO
- Modul Buzzer activ de 3V
- Senzor Ultrasonic HC-SR04

- Breadboard
- 10cm LED strip
- Wires
- LCD display

1. Hardware Scheme

2. Hardware



Software Design

Pe partea de software a proiectului, am lucrat cu 3 librării din IDE-ul Arduino, acestea fiind: “LiquidCrystal_I2C”, “Adafruit_NeoPixel” și “Wire.h”.

Sistemul înregistrează mișcarea prin intermediul senzorului ultrasonic care transimte un semnal către Arduino. După ce semnalul este interceptat, un contor numără coșurile marcate și scorul este permanent actualizat pe display-ul LCD pe care este afișat și timpul rămas pentru marcarea punctelor.

În codul folosit există și o funcție care modifică culoarea benzii LED în funcție de timpul rămas. După

ce timpul expiră, banda LED se va opri.

Algoritmi folosiți:

1.Countdown

```
long minute = 0, second = 59;
long countdown_time = minute * 60 + second;
.
.
.
void loop() {
long countdowntime_seconds = countdown_time - (millis() / 1000);
  if (countdowntime_seconds >= 0) {
    long countdown_minute = ((countdowntime_seconds / 60) % 60);
    long countdown_sec = countdowntime_seconds % 60;
    lcd.setCursor(4, 1);

    if (countdown_minute < 10) {
      lcd.print("0");
    }
    lcd.print(countdown_minute);
    lcd.print(":");
    if (countdown_sec < 10) {
      lcd.print("0");
    }
.
.
.
.
}
```

2.Control LED + LCD

```
#define LED_PIN 2
#define LED_COUNT 6
LiquidCrystal_I2C lcd(0x27, 16, 2);
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(4, 0);
  lcd.print("score:");
  lcd.setCursor(11, 0);
  lcd.print(score);

  strip.begin();
  strip.show();
}
```

```
void loop() {
.
.
.
    if (countdowntime_seconds <= 60 && countdowntime_seconds > 40) {
      setColor(strip.Color(0, 255, 0));  // Green
    } else if (countdowntime_seconds <= 40 && countdowntime_seconds > 20) {
      setColor(strip.Color(255, 255, 0));  // Yellow
    } else if (countdowntime_seconds <= 20 && countdowntime_seconds > 0) {
      setColor(strip.Color(255, 0, 0));  // Red
    } else {
      strip.clear();  // Turn off LED strip
      strip.show();
    }

    if (countdowntime_seconds == 0) {
      lcd.setCursor(4, 1);
      lcd.print("Time's up");
      strip.clear();  // Turn off LED strip
      strip.show();
    }
  }
  delay(500);
}

void setColor(uint32_t color) {
  for (int i = 0; i < LED_COUNT; i++) {
    strip.setPixelColor(i, color);
  }
  strip.show();
}
```

3. Setup Senzor Ultrasonic

```
const int triggerPin = 7;
const int echoPin = 6;
int score = 0;
.
.
.
void setup() {
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  .
  .
  .
}

void loop() {
```

```
  long duration;
  int distance;

  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);

  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);

  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.032 / 2;

  if (distance < 10) {
    score++;
    lcd.setCursor(11,0);
    lcd.print(score);
  }
}
```

4.Setup Buzzer

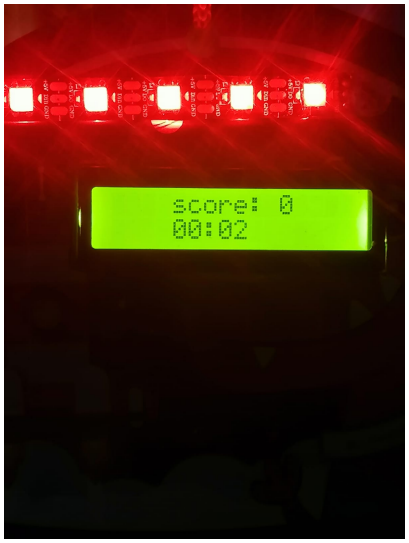
```
const int buzzerPin = 9;
.
.
.
void setup() {
  pinMode(buzzerPin, OUTPUT);
}
.
.
.
void loop {
  if (distance < 10) {
    score++;
    lcd.setCursor(11,0);
    lcd.print(score);
    playSound();
  }
  .
  .
  .
}

void playSound() {
  tone(buzzerPin, 1000, 100);
  delay(100);
  noTone(buzzerPin);
}
```

Rezultate Obţinute

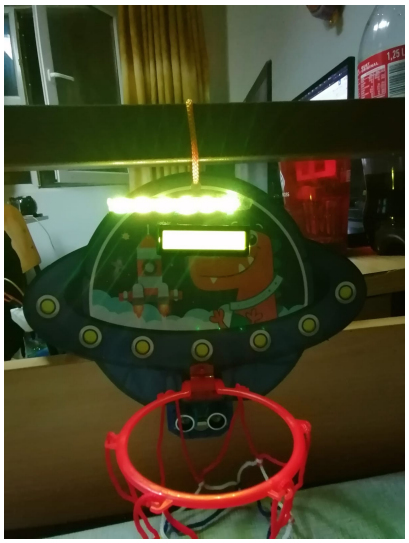
1. Coșul de baschet finalizat.

2. Sistemul de afișaj al coșului.



3. Coșul de baschet în primele 20 de secunde.

4. Coșul de baschet între 40 și 20 de secunde.



5. Coșul de baschet în ultimele 20 de secunde.

6. Afișajul după oprire + LED-urile oprite.

7. Afișajul cu puncte.

Concluzii

Tot procesul de facere a proiectului a fost o experiență interesantă și oarecum distractivă, dar care nu a fost întocmai ușoară.

Construirea proiectului a fost cu siguranță o activitate plăcută, deși pe parcurs am întâmpinat mai multe probleme cu hardware-ul folosit.

Dezvoltarea codului a fost puțin mai provocatoare decât m-aș fi așteptat, din cauza problemelor pe care le-am avut cu senzorul ultrasonic.(probleme care au o șansă să mai apară din când în când)

Una peste alta, acest proiect m-a ajutat sa învăț destul de multe și a fost probabil partea mea preferată a acestui semestru.

Download

ma.rar

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Utilizare LCD
Utilizare HC-SR04
Laboratorul 6

Export to PDF

From:
http://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2023/avaduva/basketball_score_counter

Last update: **2023/05/30 02:00**