

DIY RC Car

Introducere

- Proiectul este o mașină RC controlată cu telefonul, echipată cu mai multe trepte de viteză reglabile atât din aplicația de pe telefon, cât și dintr-un buton fizic situat pe mașină. De asemenea, mașina dispune de un LED RGB care indică modul de viteză în care se află și de un buzzer care poate fi controlat din aplicația de telefon.
- Scopul proiectului este de a face controlul mașinii RC să fie distractiv și interactiv pentru utilizatori, oferindu-le posibilitatea de a o manevra cu telefonul mobil
- Ideea de la care am pornit a fost conectarea telefonului la placa de dezvoltare prin Bluetooth
- Utilitatea proiectului stă în aplicarea notiunilor teoretice în practică

Descriere generală

Diagramă bloc



Descriere mod de funcționare

Telefonul comunică cu modulul Bluetooth, care, la rândul său, trimite comenzile către placa de control. În funcție de comanda primită, placa va efectua următoarele acțiuni:

- Schimbarea modului de viteză și schimbarea culorii LED-ului RGB
- Mers înainte/înapoi
- Rotație la stânga/dreapta
- Pornire/oprire a buzzerului

La apăsarea butonului switch, va avea loc schimbarea modului de viteză și a culorii LED-ului RGB.

Hardware Design

Componentele folosite

- Pulsivo cu ATmega328p
- Modul Bluetooth HC-05
- Driver de Motoare Dual L298N
- Kit Robot cu 2 Motoare
- Switch On/Off
- Buzzer
- Led RGB
- Fire

Schema electrică



Software Design

- **Mediu de dezvoltare:** Arduino IDE
- **Surse și funcții implementate:**

```
void switchNitro() {
  if (PIND & (1 << NITRO_PIN)) {
    nitro = false;
  } else {
    nitro = true;
  }
}
```

```
void setup() {
  Serial.begin(9600);
  // Input pullup
  DDRD &= ~(1 << NITRO_PIN);
  PORTD |= (1 << NITRO_PIN);
  DDRD |= (1 << HONK_PIN);
  DDRD |= (1 << LED_GREEN_PIN);
  DDRD |= (1 << LED_RED_PIN);
  DDRB |= (1 << UP_LEFT_PIN);
  DDRB |= (1 << DOWN_LEFT_PIN);
  DDRB |= (1 << DOWN_RIGHT_PIN);
  DDRB |= (1 << UP_RIGHT_PIN);
  attachInterrupt(digitalPinToInterrupt(NITRO_PIN), switchNitro, CHANGE);
  // Init nitro
  nitro = false;
  // Init RGB LED
  PORTD &= ~(1 << LED_RED_PIN);
}
```

```
PORTD |= (1 << LED_GREEN_PIN);  
// Init wheels pins  
PORTB &= ~(1 << UP_LEFT_PIN);  
PORTB &= ~(1 << DOWN_LEFT_PIN);  
PORTB &= ~(1 << DOWN_RIGHT_PIN);  
PORTB &= ~(1 << UP_RIGHT_PIN);  
}
```

```
void loop() {  
  if (Serial.available() > 0) {  
    cmd = Serial.read();  
  } else {  
    cmd = None;  
  }  
  if (cmd == Start_Honk) {  
    tone(HONK_PIN, 300);  
  } else if (cmd == Stop_Honk){  
    noTone(HONK_PIN);  
  }  
  if (cmd == Nitro_On) {  
    nitro = true;  
  } else if (cmd == Nitro_Off) {  
    nitro = false;  
  }  
  if (cmd == Up) {  
    if (nitro) {  
      PORTB |= (1 << UP_LEFT_PIN);  
      PORTB |= (1 << UP_RIGHT_PIN);  
    } else {  
      analogWrite(UP_LEFT_PIN_ARDUINO, 180);  
      analogWrite(UP_RIGHT_PIN_ARDUINO, 180);  
    }  
    PORTB &= ~(1 << DOWN_LEFT_PIN);  
    PORTB &= ~(1 << DOWN_RIGHT_PIN);  
  } else if (cmd == Down) {  
    digitalWrite(UP_LEFT_PIN_ARDUINO, LOW);  
    digitalWrite(UP_RIGHT_PIN_ARDUINO, LOW);  
    PORTB |= (1 << DOWN_LEFT_PIN);  
    PORTB |= (1 << DOWN_RIGHT_PIN);  
  } else if (cmd == Left) {  
    digitalWrite(UP_LEFT_PIN_ARDUINO, LOW);  
    if (nitro) {  
      PORTB |= (1 << UP_RIGHT_PIN);  
    } else {  
      analogWrite(UP_RIGHT_PIN_ARDUINO, 180);  
    }  
    PORTB &= ~(1 << DOWN_LEFT_PIN);  
    PORTB &= ~(1 << DOWN_RIGHT_PIN);  
  } else if (cmd == Right) {  
    if (nitro) {  
      PORTB |= (1 << UP_LEFT_PIN);  
    }  
  }  
}
```

```
} else {
  analogWrite(UP_LEFT_PIN_ARDUINO, 180);
}
digitalWrite(UP_RIGHT_PIN_ARDUINO, LOW);
PORTB &= ~(1 << DOWN_LEFT_PIN);
PORTB &= ~(1 << DOWN_RIGHT_PIN);
} else if(cmd == Center) {
  digitalWrite(UP_LEFT_PIN_ARDUINO, LOW);
  digitalWrite(UP_RIGHT_PIN_ARDUINO, LOW);
  PORTB &= ~(1 << DOWN_LEFT_PIN);
  PORTB &= ~(1 << DOWN_RIGHT_PIN);
}
if (nitro) {
  PORTD |= (1 << LED_RED_PIN);
  PORTD &= ~(1 << LED_GREEN_PIN);
} else {
  PORTD &= ~(1 << LED_RED_PIN);
  PORTD |= (1 << LED_GREEN_PIN);
}
}
```

Download

[sorin_petrut.barbu.zip](#)

[Export to PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/sorin_petrut.barbu



Last update: **2024/05/18 15:57**