RC Car

Introducere

Proiectul urmareste implementarea unei masini controlate prin telecomanda.

Descriere generală

Telecomanda Infra Rosu va transmite semnale catre un senzor, aflat pe masina. Senzorul va trasnmite mai departe datele catre Arduino, unde acestea vor fi decodificate. In urma decodificarii, masini va merge in una dintre cele 4 directii: FRONT, LEFT, RIGHT, BACK; sau va putea claxona. Deplasarea se va face prin intermediul a 4 motoare DC (2 pt fata, 2 pt spate). Vor fi si LED-uri pentru a afisa directia de mers/claxon. Claxonul va fi constituit dintr-un Piezo.

Hardware Design

Componente folosite

- Microcontroller Arduino
- 2x DC-Motor 3-6V
- Mini-telecomanda IR 38 kHz
- Modul de recepție infraroșu (senzor)
- Modul L298N - Punte H dubla
- fire tata-tata, mama-tata
- 2x 9V Battery
- Cablu alimentare jack/normal pentru Arduino/Punte H

Schema Electrica

Stadiul Actual

VARTA
LONGLIFE
POWER
VARTA
LONGLIFE
POWER

Software Design

Receptie si dreciptare date

Datele de la telcomanda se primesc printr-un sensor IR. Acestea vin sub forma de cod, iar noi le vom translata in comenzi. Folosim biblioteca **IRremote** pentru citirea datelor.

```
#define FRONT 3877175040
#define LEFT 4144561920
#define RIGHT 2774204160
#define BACK 2907897600
#define BUZZ 3927293696
```

Cum primim datele? Pentru a simplifica loop-ul principal, am decis ca primirea datelor sa se faca printr-o intrerupere ISR.

```
#define RECV_PIN 2

void setup()
{
  cli();
  USART0_init();
  ...
  EICRA |= (1 << ISC01);
  EIMSK |= (1 << INT0);

  DDRB |= (1 << DDB4);
  TCCR1A = (1 << COM1A1) | (1 << COM1B1) | (1 << WGM10);
  OCR1B = 255;

  DDRD |= (1 << DDD5);
  TCCR0A = (1 << COM0B1) | (1 << WGM00) | (1 << WGM01);
  TCCR0B = (1 << CS01) | (1 << CS00);
  OCR0B = 255;
  ...
  IrReceiver.begin(RECV_PIN, ENABLE_LED_FEEDBACK);
  EICRA |= (1 << ISC01);
  EIMSK |= (1 << INT0);
  t_reset = millis();
  ...
  sei();
}

ISR(INT0_vect)
{
  if (IrReceiver.decode())
```

```
  {
    switch(IrReceiver.decodedIRData.decodedRawData)
    {
      case FRONT:
      last_command = 2;
      t_reset = millis();
      break;
      ...
    }
    IrReceiver.resume();
  }
}
```

Tot aici configuram si timerele in regim fast PWM, alaturi de protocolul USART pentru o interfata seriala de debug.

Loop principal

Ce se intampla cu t_reset ? Avem si o variabila t_now. In loop-ul principal, asociam cu last_command directia in care se vor invarti motoarele:

```
void loop() {
  t_now = millis();
  switch (last_command) {
    case 2:
    directionA = HIGH;
    directionB = HIGH;
    break;
    ...
  }
}
```

Detectia de IR se face mai rar decat loop-ul principal. Noi vom decide daca miscam sau nu rotile pe baza last_command. O versiune mai veche seta last_command pe -1 daca nu se primea nimic. Astfel, desi noi tineam apasat butonul pe telecomanda, motoarele se invarteau putin si dupa se opreau, repetat. O versiune noua a codului, implmenetata printr-un spatiu de timp, permite motoarelor sa se invarte timp de 200ms de la ultima apasare → timp suficient pentru a detecta o alta comanda si a reseta timer-ul fara a opri motoarele. Last command va fi -1 daca se apasa alt buton decat cele alocate.

```
if (t_now - t_reset < 200 && last_command != -1) {
  USART0_transmit('O');
  USART0_transmit('n');
  USART0_transmit(decode[last_command]);
  USART0_transmit('\n');
  doMove();
} else {
  MotorOff();
```

```
}
```

Manipularea motoarelor

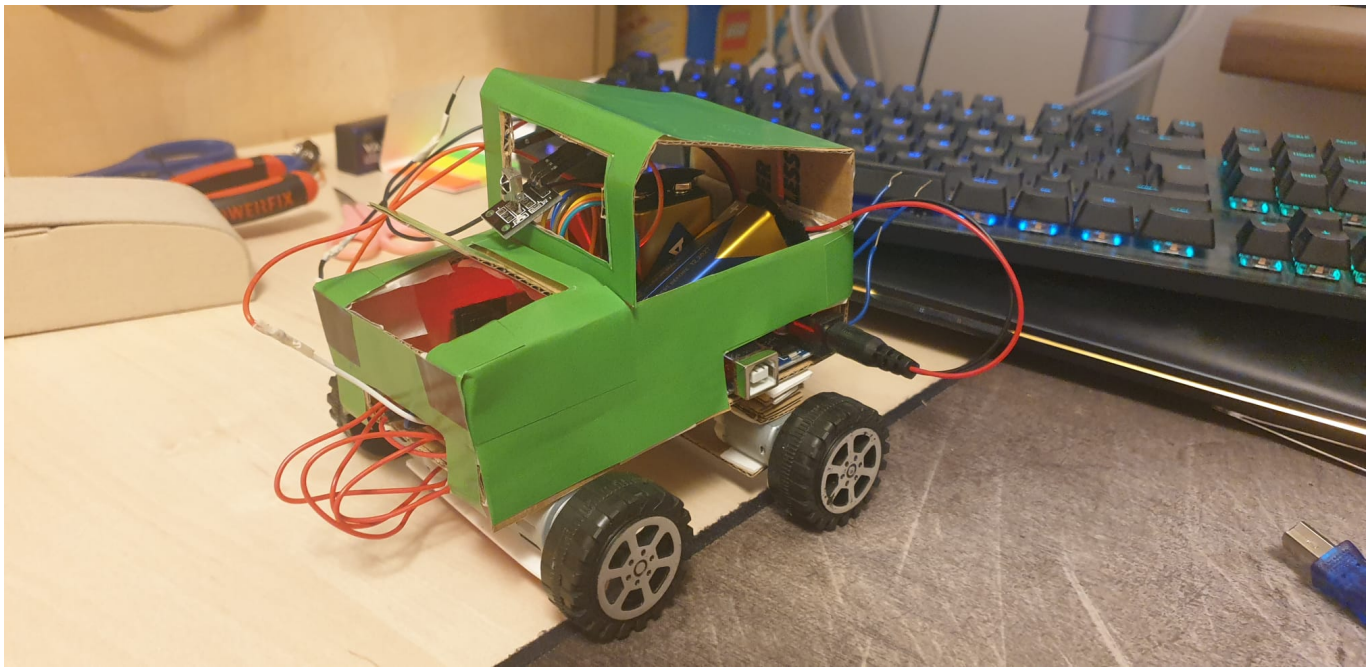
Cum se executa miscarea ? Prima oara setam viteza dorita pentru motor. Vom folosi configurarea fast PWM, ca mai sus. **TO-DO** configurare fast pwm prin metoda clasica.

```
void doMove()
{
  digitalWrite(IN1, directionA);
  digitalWrite(IN2, !directionA);
  digitalWrite(IN3, directionB);
  digitalWrite(IN4, !directionB);
}
```

Pentru MotorOff, se seteaza IN1-4 pe LOW pentru a opri miscarea rotilor.

Rezultate Obţinute

Masina functioneaza intr-un mediu ideal, dar acesta din pacate nu exista. Pe scurt, masina functioneaza optim in aer, iar pe pamant nu se va misca din lipsa torque-ului sau a rotilor.



Concluzii

Investiti bani in motoare performante si roti, altfel munca voastra va fi in zadar.

Download

rc-car.zip

Jurnal

Pana acum

- Prezentare generala finalizata
- Hardware Design finalizat
- Montaj initial finalizat
- Software scris si testat
- Prezentare Software

Bibliografie/Resurse

Resurse Hardware

- Datasheet Atmega 328
- Datasheet L298N

Resurse Software

- Arduino IDE
- IRemote.h + Docs

Export to PDF

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

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

Last update: **2023/05/29 22:36**