

RAPORT TEHNIC

ROVeRe

“The word impossible is not in my dictionary”.

(Napoleon Bonaparte)

Profesor coordonator:

Coțop Carmen

Donca Radu

Membrii echipei:

Coțop Ruxandra

Lup Vasile

Maier Raul

Selicean Orlando

Tomoiață Vlad

Liceul Teoretic “Nicolae Bălcescu”, Cluj-Napoca

Romania

Majoritatea oamenilor considera ca robotii sunt niste simple instrumente care faciliteaza munca oamenilor. De fapt, cu acest scop au fost si conceputi. Marturisim ca si noi aveam aceeași percepție pana cand a trebuit sa trecem la crearea propriu-zisa a lui RoVeRe. Pe langa faptul ca procesul nu a fost usor deloc, am ajuns sa ne atasam de el.

Misiunea principala a lui constă în deplasarea teleghidată până la un punct fix și în măsurarea unor parametri: presiunea atmosferică, temperatura mediului înconjurător, compozitia atmosferica, umiditatea, cu ajutorul unor senzori aști la bord în vederea determinării habitatul sitului de catre oameni si determinarea din acesti parametri ai posibilitatii existentei vietii. Misiunea secundară constă în prelucrarea acestor date.

Capul a reprezentat partea cea mai importanta, fiindca ne-am gandit sa introducem un soi de mecanism care sa semene cu cel mai sofisticat computer pe care-l știm – creierul omenesc.

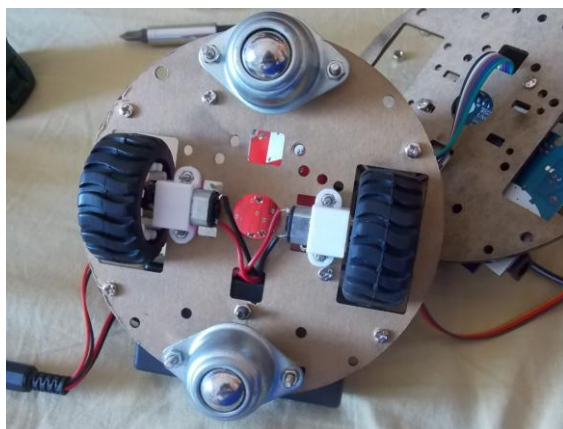
Am tot cautat unul si ne-am oprit la: Arduino Uno .

Totuși, oricât de deștept ar fi un creier, acesta nu va putea vorbi fără corzi vocale, nu se va putea deplasa fără mușchi și oase, nu va putea auzi fără urechi etc. pentru că un creier nu poate lucra decât cu semnale electrice. În mod similar, nici un microcontroller nu va putea emite lumină sau sunet fără un bec sau un difuzor, nu va putea să miște ceva fără un motor, nu va putea auzi fără un microfon etc. Asta înseamnă că pentru toate lucrurile pe care nu le poate realiza singur, microcontroller-ul are nevoie de o serie de componente auxiliare. Ansamblul format dintre un microcontroller, firmware-ul instalat în el și toate componentele auxiliare se numeste, în cazul nostru, ROVeRe.

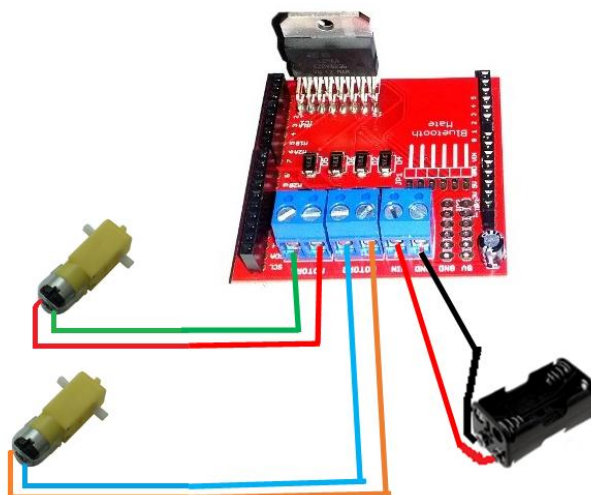
Ei, bine, probabil cea mai fascinanta parte a fost programarea. Ne-am simțit ca niste zei care au puterea de a insufla viata unei simple masinarii. Cand l-am vazut pe ROVeRe ca executa comezile, am fost tare fericiti.. Creatia noastra era, în sfarsit, vie, iar noi am finalizat procesul. De azi inainte, stim ca Rovere nu e un simplu robot, ci o parte din fiecare membru al echipei pe care ne putem bizui si care ne va rasplati pentru munca depusa prin furnizarea unor date meteo de care nu puteam sa facem rost altfel. El poate sa patrunda în locurile mai greu accesibile pentru noi, dar pe care noi dorim sa le vizitam.

Partea mecanica

ROVeRe se deplaseaza prin intermediul celor 2 roti. Datorita rotilor mari, abordeaza foarte bine orice gen de suprafata.



Rotile sunt puse în miscare de catre 2 motoare pe care le-am legat la un shield de motoare L 298.

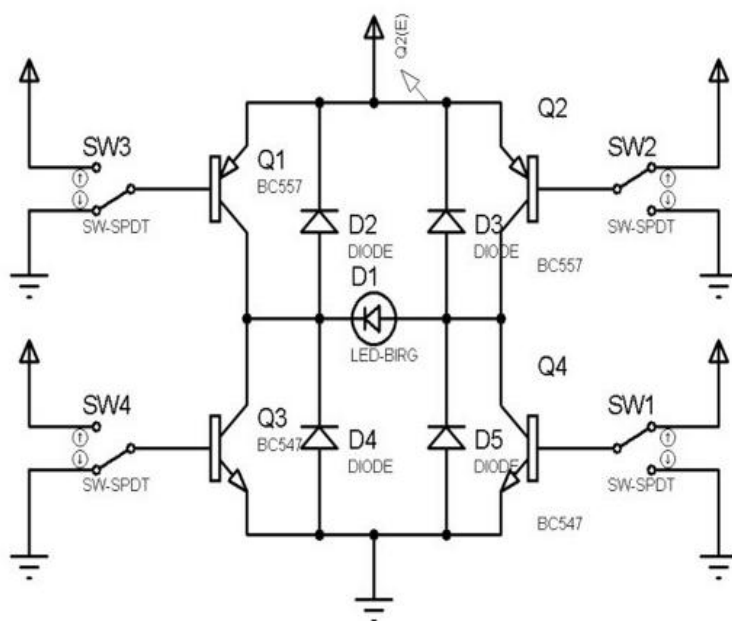
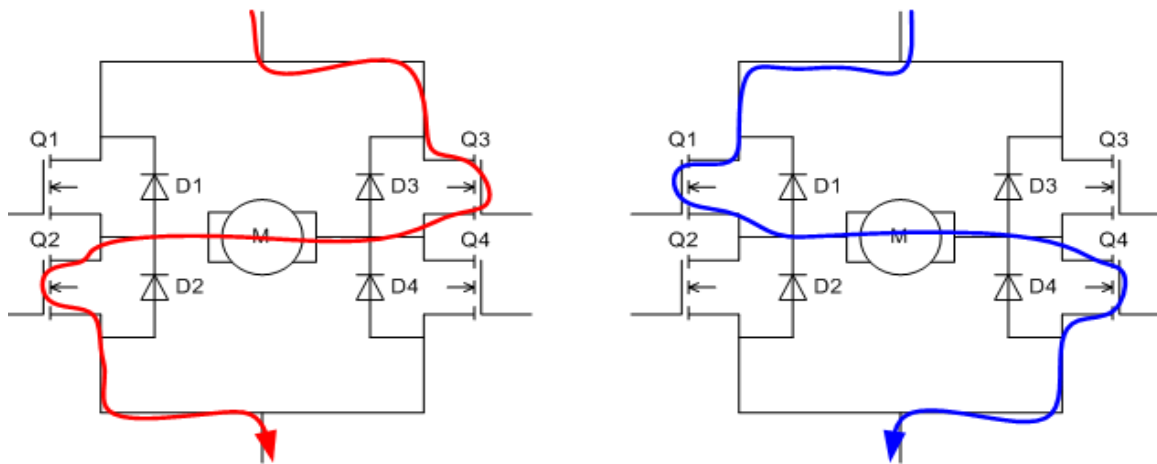
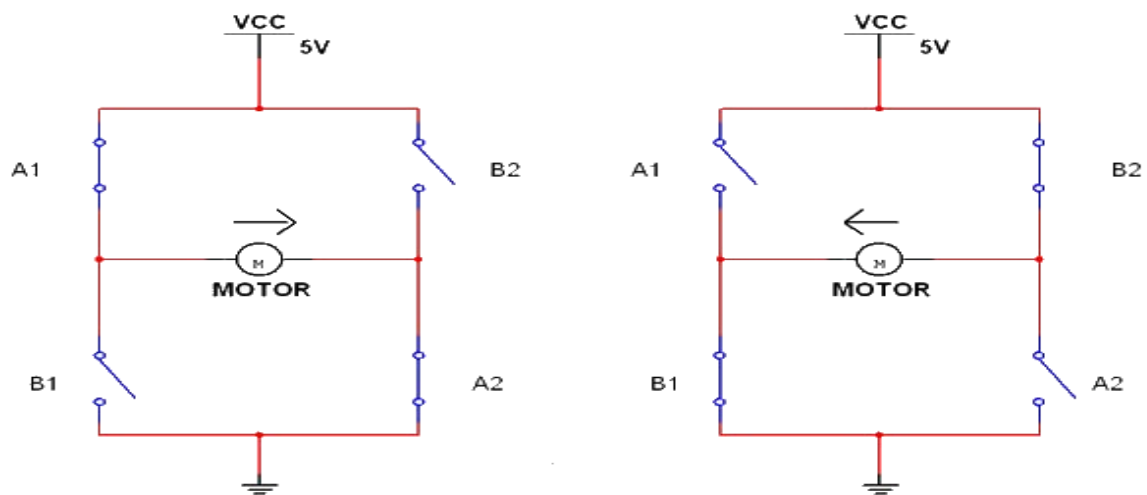


Shield-ul de motoare L298 este un shield care poate fi utilizat foarte usor pentru a controla motoarele de curent continuu. Exista o multitudine de drivere de motoare, diferenta majora intre ele fiind cat de multa putere pot conduce, cat de puternice pot fi motoarele pe care le pot controla. Driver-ul din aceasta sectiune este bazat pe integratul L298, fiind un driver de nivel mediu din punct de vedere al puterii conduse. Poate controla motoare care necesita cel mult 2 Amperi. Noi am ales acest shield deoarece este relativ ieftin si usor de programat.

Alimentarea shield-ului se face prin pinii cu surub VIN +si GND -. si se alimenteaza in functie de necesitatile motoarelor. Noi l-am alimentat cu ajutorul unor acumulatori de tip Li-ion de 3,7 V, 3500mAh, 2 buc. Acest shield are o punte H (eng. H Bridge) care este un circuit electronic ce permite aplicarea unei tensiuni pe o sarcină în orice sens. Aceste circuite sunt adesea folosite în robotică și alte aplicații pentru a permite motoarelor de curent continuu să ruleze înainte și înapoi. Punțile H sunt disponibile ca circuite integrate sau pot fi construite din componente discrete, tranzistoare bipolare sau MOS.

Puntea H are numele derivat de la modul obișnuit de desenare a circuitului. Aceasta este singura cale de tip solid state de a comanda motorul în ambele direcții. Atunci când întrerupătoarele A1 și A2 (în conformitate cu figura) sunt închise și B1 și B2 sunt deschise o tensiune pozitivă va fi aplicate la nivelul motorului. Prin deschiderea întrerupătoarelor A1 și A2 și închiderea întrerupătoarelor B1 și B2, această tensiune este inversata, astfel să permită funcționarea inversă a motorului. În practică întrerupatoarele A1,A2,B1,B2 sunt tranzistoare bipolare

Mod de funcționare:



SSursa

Driver-ul se conecteaza la platforma Arduino folosind 4 pini digitali (3, 5, 6 si 9) prin infigere directa in pinii placii Arduino.

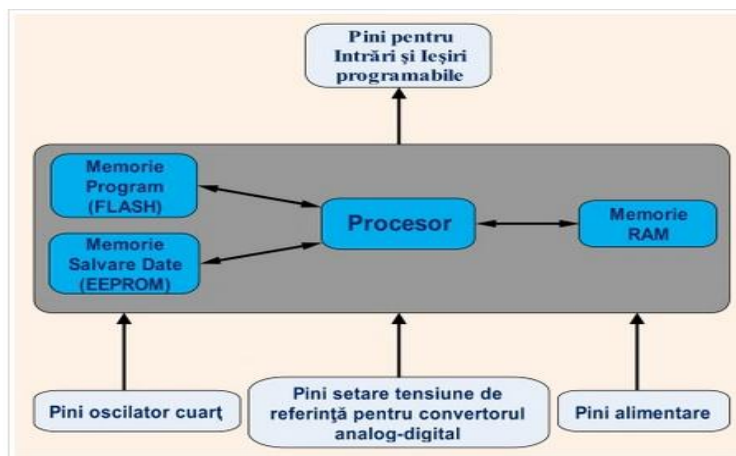
Placa Arduino

Arduino este un mic calculator programabil, care poate comanda leduri, becuri, motoare, ecrane digitale. El poate interpreta informațiile primite de la butoane, senzori precum giroscopul, accelerometrul, o telecomandă, senzori de lumină, de temperatură, de distanță, de mișcare. În proiectele complexe poate fi comandat prin recunoaștere vocală, bluetooth, se poate conecta la internet și poate fi creierul roboților industriali.



Noi am folosit o placa Arduino Uno care are urmatoarele caracteristici tehnice:

- Microcontroller: ATmega328
- Tensiune de lucru: 5V
- Tensiune de intrare (recomandată): 7-12V
- Tensiune de intrare (limite): 6-20V
- Pini digitali de intrare/ieșire: 14 (dintre care 6 suportă ieșire PWM, , toți pinii marcați cu semnul ~ pot fi folosiți și pentru PWM (pulse with modulation)
- Pini de intrare analogică: 6
- Intensitate per pin de intrare/ieșire: 40 mA
- Intensitate pentru pinul de 3,3V: 50 mA
- Memorie flash: 32 KB (ATmega328), dintre care 0,5 KB sunt folosiți de bootloader
- SRAM: 2 KB (ATmega328);EEPROM: 1 KB (ATmega328)
- Tact microcontroller: 16 MHz

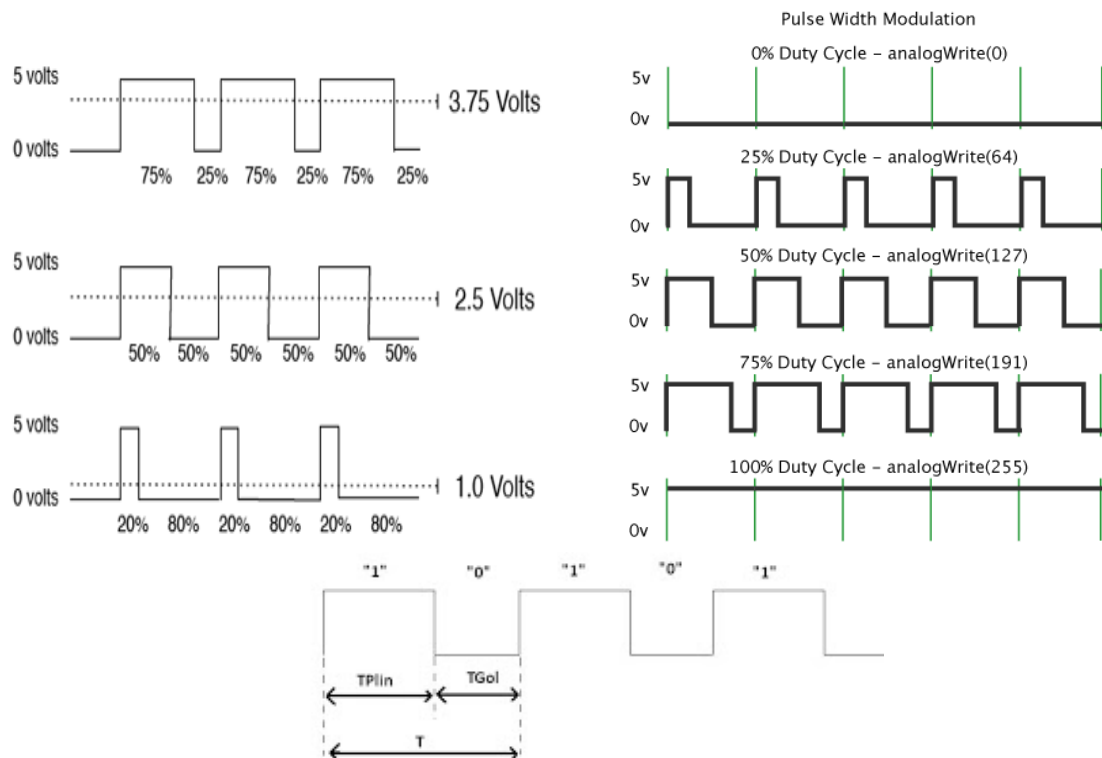


Surse: <http://www.hobbytronica.ro/ce-este-un-microcontroller/>

<https://www.arduino.cc/>

Ce este PWM ?

Termenul de PWM provine din limba engleza de la Pulse Width Modulation ceea ce inseamna Modulatia Pulsurilor in Lungime si este o tehnica de a simula un semnal analogic folosind un semnal digital. Semnalul digital poate lua doar 2 stari: ON sau OFF ; „1“ sau „0“ ; 5V sau 0V. Un semnal PWM te ajuta sa obtii o multime de valori cuprinse intre 5 si 0V. Semnalele PWM sunt folosite în general pentru a controla viteza unui motor de curent continuu. Aceste motoare permit controlul vitezei prin modificarea tensiunii de alimentare însă viteza se modifică neliniar. Pentru a controla această viteză liniar s-a recurs la metoda de control prin semnale PWM. Semnalele PWM au următoarea formă:



În schema de mai sus observăm că semnalul este de perioadă T . Această perioadă este compusă din timpul în care semnalul este 1 logic (T_{Plin} care se mai numește și factor de umplere) și timpul în care semnalul este 0 logic (T_{Gol}). De obicei intervalul T este dat de 256 de tacte de ceas astfel T_{Plin} sau T_{Gol} putând modifica viteza motorului cu pasul de aproximativ 0.4% la fiecare modificare cu o unitate a registrului de control al factorului de umplere.

Semnalele PWM sunt foarte utile in diverse situatii iar placa Arduino nu duce lipsa de aceasta functie. O placa Arduino UNO poate genera pana la 6 semnale PWM prin pinii 3, 5, 6, 9, 10 si 11 deoarece microcontroller-ul este echipat cu un modul hardware special care se ocupa exclusiv de aceste semnale.

Cum generez un semnal PWM cu placa Arduino ?

Exemplu: Cum pot sa comand un motor brushless cu controller ESC (Electronic Speed

Controller):

```

#include <Servo.h>

Servo esc;

int throttle = 0;

void setup()
{
  Serial.begin(9600);
  esc.attach(6);
  delay(15);
  esc.write(30);
  delay(2000);
}

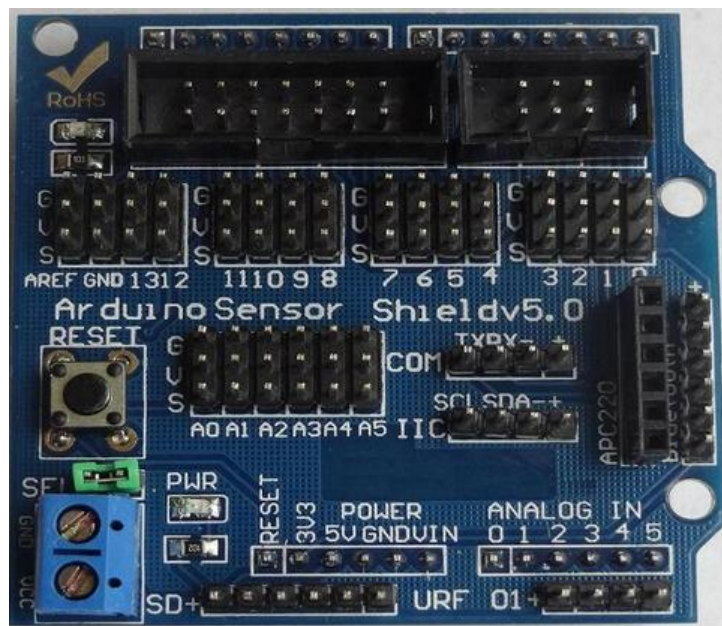
void loop()
{
  for (throttle = 0; throttle <=179; throttle++ ) {
    esc.write(throttle);
    Serial.println(throttle);
    delay(400);
  }

  for (throttle = 179; throttle >=0; throttle-- ) {
    esc.write(throttle);
    Serial.println(throttle);
    delay(400);
  }
}

```

Surse: https://www.arduino.cc/en/Tutorial/PWM#UxJAW_1_uSo,
http://www.robofun.ro/docs/curs/64_600f0173-a7ca-47d0-a7e8-7244f6f95991/Arduino-SoftwarePWM.pdf

Sensor shield Arduino Sensor Shieldv5.0



Caracteristici tehnice:

- PIN13 LED Pilot
- Digital IO Ports D0-D13
- Analog IO Ports A0-A5
- Reset Button
- Power In
- Power LED
- SD Card Interface
- Ultrasonic Interface
- RS232 Interface(TTL)
- Bluetooth Interface
- APC220 Interface
- IIC Interface
- 12864 LCD Parallel Interface
- 12864 LCD Serial Interface

Compatibilitate:

- Arduino NG (Compatible)
- Arduino Diecimila (Compatible)
- Arduino Duemilanove (Compatible)
- Arduino UNO (Compatible)

Surse: 1. <https://arduino-info.wikispaces.com/SensorShield>

2. http://robotbase.en.alibaba.com/product/1157780742-211868459/Sensor_Shield_V5_0_for_Robotic_controller.html

Senzori

Senzorii sunt componente hardware care furnizează computerului informații despre locația și împrejurimile computerului și multe altele. Programele din computer pot accesa informații de la senzori, apoi pot stoca sau utiliza acele informații pentru a ne ajuta cu în activitățile zilnice sau pentru a îmbunătăți experiența computerului. Senzori folosiți de noi sunt:

1.Senzor de proximitate

Un senzor de proximitate este un senzor capabil de a detecta prezența de obiectele în apropiere, fără nici un contact fizic. Un senzor de proximitate emite adesea unde electromagnetice sau un fascicul de radiații electromagnetice și caută schimbări în mediul înconjurător, asemenea unui radar. El se activează automat în timpul unui apel iar distanța la care poate depista un obiect este, în medie, 50mm.

Senzorul folosit de noi este: Ultrasonic Ranging Module HC - SR04



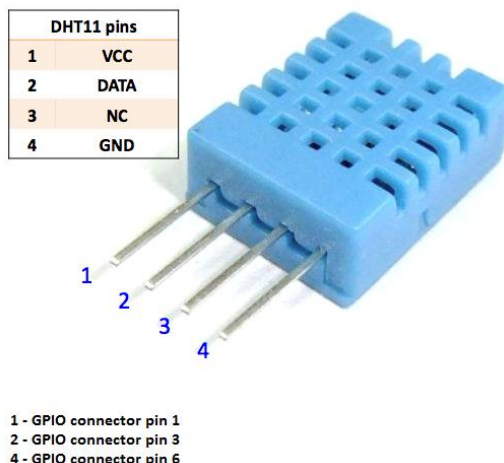
Si are urmatoarele caracteristici:

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Sursa: <http://www.micropik.com/PDF/HCSR04.pdf>

2. Senzorul de temperatura si umiditate

Noi am folosit un senzor de temperatura si umiditate DHT11 cu ajutorul caruia se pot face citiri la fiecare 2 secunde, deoarece ca necesar de pini este foarte modest, avand nevoie doar de un singur pin digital pentru conectarea cu Arduino, este usor de programat si ieftin.



Caracteristicile tehnice ale acestui senzor sunt:

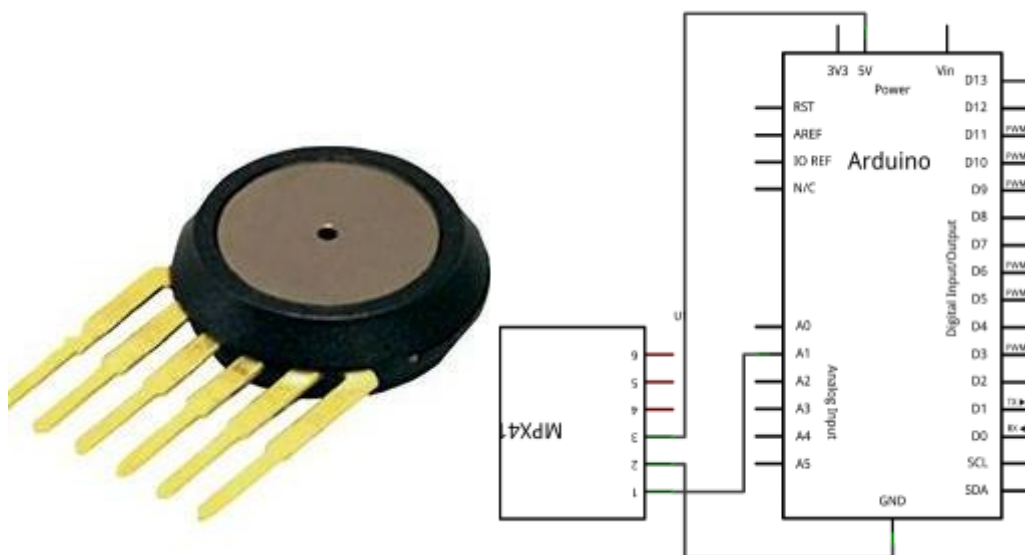
Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			± 1%RH	
Accuracy	25 °C		± 4%RH	
	0-50 °C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0 °C	30%RH		90%RH
	25 °C	20%RH		90%RH
	50 °C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25 °C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
Temperature				
Resolution		1 °C	1 °C	1 °C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1 °C	
Accuracy		± 1 °C		± 2 °C
Measurement Range		0 °C		50 °C
Response Time (Seconds)	1/e(63%)	6 S		30 S

Sursa: <http://www.micropik.com/PDF/dht11.pdf>

3.Senzor de presiune

Pentru a masura presiunea atmosferica am folosit un senzor de presiune MPX4115A. Acesta are 6 pini, din care sunt folositi 3: primul reprezinta Vout, al doilea este GND-ul, iar al treilea este Vin. Noi am ales acest senzor, desi este mai scump, deoarece are o precizie mare si un timp de raspuns redus.

Formula pentru calcularea presiunii este: $V_{out} = V_{in} * (0.009 * P - 0.095)$, unde P reprezinta valoarea presiunii atmosferice masurate in kilopascali.



Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P _{OP}	15	—	115	kPa
Supply Voltage ⁽²⁾	V _S	4.85	5.1	5.35	Vdc
Supply Current	I _o	—	7.0	10	mAdc
Minimum Pressure Offset @ V _S = 5.1 Volts ⁽³⁾	V _{off}	0.135	0.204	0.273	Vdc
Full Scale Output @ V _S = 5.1 Volts ⁽⁴⁾	V _{FSO}	4.725	4.794	4.863	Vdc
Full Scale Span @ V _S = 5.1 Volts ⁽⁵⁾	V _{FSS}	4.521	4.59	4.659	Vdc
Accuracy ⁽⁶⁾	—	—	—	±1.5	%V _{FSS}
Sensitivity	V/P	—	46	—	mV/kPa
Response Time ⁽⁷⁾	t _R	—	1.0	—	ms
Output Source Current at Full Scale Output	I _{o+}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	mSec
Offset Stability ⁽⁹⁾	—	—	±0.5	—	%V _{FSS}

Surse: <http://media.digikey.com/pdf/Data%20Sheets/Freescale%20Semi/MPX4115A>

3.Senzor de gaz pentru detectarea monoxidului de carbon

Am folosit: MQ-7 Semiconductor Sensor for Carbon Monoxide.



Caracteristicile tehnice ale acestui senzor sunt:

Model			MQ-7
Sensor Type			Semiconductor
Standard Encapsulation			Plastic cap
Target Gas			carbon monoxide
Detection range			10~500ppm CO
Standard Circuit Conditions	Loop Voltage	V_c	$\leq 10V$ DC
	Heater Voltage	V_H	5.0V \pm 0.1V AC or DC (High tem.) 1.5V \pm 0.1V AC or DC (Low tem.)
	Heater Time	T_L	60 S \pm 1S (High tem.), 90 S \pm 1S (Low tem.)
	Load Resistance	R_L	Adjustable
Sensor character under standard test conditions	Heater Resistance	R_H	29 Ω \pm 3 Ω (room tem.)
	Heater consumption	P_H	$\leq 900mW$
	Sensitivity	S	$R_s(\text{in air})/R_s(\text{in } 150\text{ppm CO}) \geq 5$
	Output Voltage	V_s	2.5V~4.3V (in 150ppm CO)
	Concentration Slope	α	$\leq 0.6(R_{300\text{ppm}}/R_{50\text{ppm CO}})$
Standard test conditions	Tem. Humidity		20 $^{\circ}C$ \pm 2 $^{\circ}C$; 55% \pm 5%RH
	Standard test circuit		V_c : 5.0V \pm 0.1V; V_H (High tem.): 5.0V \pm 0.1V; V_H (Low tem.): 1.5V \pm 0.1V
	Preheat time		Over 48 hours

Sursa: <https://www.sparkfun.com/products/9403>

Bluetooth

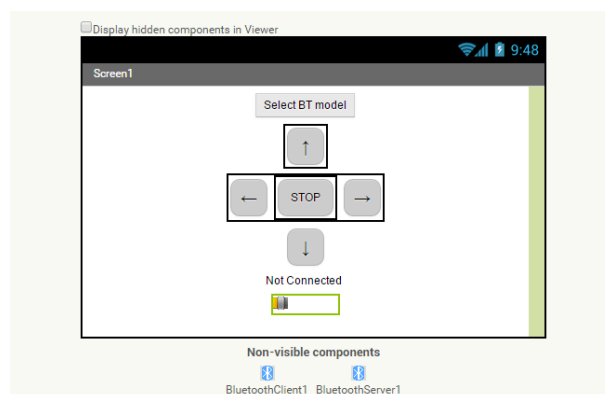
Pentru comunicare noi am folosit modulul HC-06 Bluetooth. Acesta comunică printr-un port serial, la o viteză de 9600 bit/s. Funcționează pe o rază de până la 50 m. Am ales acest model, deoarece este relativ ieftin și corespunde necesitatilor noastre.



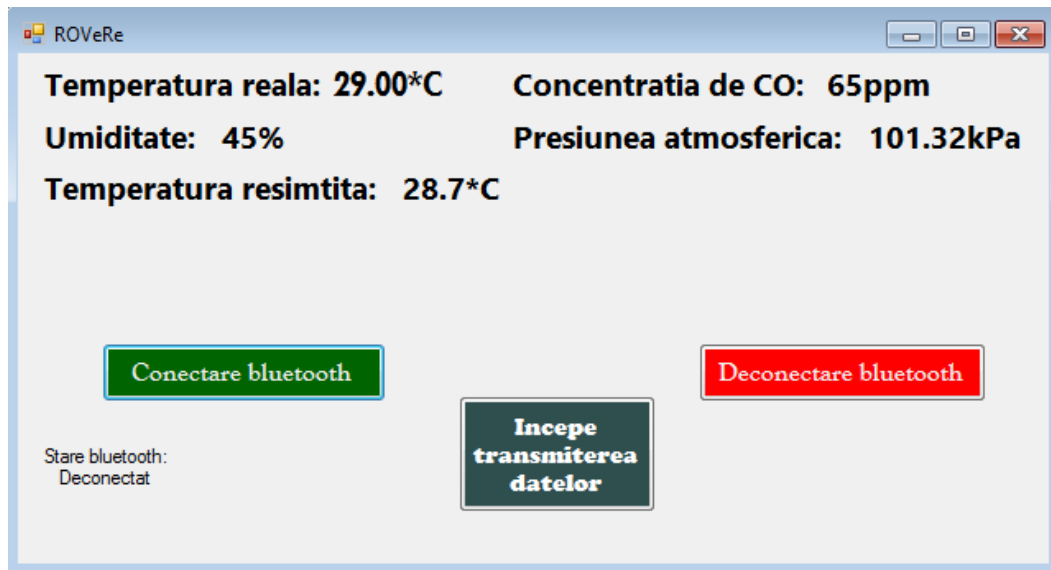
Software

Partea de software a robotului a fost puțin mai greu de realizat, deoarece trebuiau luate în considerare mai multe aspecte. De aceea, noi am ales să facem trei coduri pentru ca ROVeRe să facă tot ce ne-am propus.

În primul rând, am rezolvat mișcarea lui. Pentru a-l controla, am creat o aplicație folosind MIT App Inventor. Aceasta este una simplă, având șase butoane și un slider. Un buton este folosit pentru a începe conexiunea cu placa, slider-ul transmite valori între 0 și 255, reprezentând viteza de deplasare, în timp ce restul butoanelor sunt pentru a controla direcția de mișcare.



În al doilea rând, pentru a putea citi și prelucra datele primite de la plăcuța Arduino, am făcut o aplicație (WindowsFormApplication) în C#. Aceasta afișează pe ecran datele primite prin Bluetooth și le memorează într-un tabel Excel, de unde urmează să fie prelucrate.



Ultima, dar și cea mai importantă parte, a fost scrierea codului pentru Arduino. Pentru asta am folosit mediul de programare dat de cei care crează și distribuie aceste plăcuțe. Limbajul folosit este asemănător cu C++.

Orice cod scris în acest mediu de programare conține două părți: setup și loop. Liniile de cod scrise în prima parte se execută o singură dată la începutul rulării, în timp ce cele din a doua se execută repetat (după cum îi spune și numele) până la încheierea rulării.

Există câteva biblioteci incluse în pachet, dar sunt și foarte multe care se găsesc de downloadat de pe internet și care pot fi extrem de utile. Noi am folosit o singura bibliotecă, dht.h, pentru a ne ușura citirea datelor de la senzorul de temperatură și umiditate. Exemplul de cod de mai jos, exemplifică funcționarea acestuia:

```
#include <DHT.h>
#define DHTPIN 2;// pinul la care este conectat senzorul
#define DHTTYPE DHT11;//tipul senzorului
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
}

void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\n");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print("Heat index: ");
  Serial.print(hic);
  Serial.print(" *C ");
}
```

Buget

1. Arduino UNO, pret 109 lei
2. Driver motoare L298, pret 79 lei
3. Arduinosensor Shieldv5.0, pret 45 lei
4. 3 pcs. HC-06 Wireless Bluetooth Transceiver RF Main Module Serial For Arduino, pret 45 lei
5. Senzor de temperatura si presiune, pret 10 lei
6. Senzor de presiune, pret 25 lei
7. Senzor de monozid de carbon, pret 30 lei
8. Acumulatori, pret 50 lei