**UNIVERSITATEA POLITEHNICA ,,TIMISOARA”**

**FACULTATEA DE AUTOMATICA SI CALCULATOARE**

**CALCULATORE SI TEHNOLOGIA INFORMATIEI**

**PROIECT**

**SISTEME INCORPORATE**

**APLICATIA ,,AUTOMOTIVE 1”**

*GitHub:* [*https://github.com/mirifi2k/si-project*](https://github.com/mirifi2k/si-project)

**LAPADAT RAZVAN**

**MIH BOGDAN**

**KOVACS ADAM**

**AN III**

**UNIVERSITATEA POLITEHNICA ,,TIMISOARA”**

**2018 – 2019**

**SISTEME INCORPORATE**

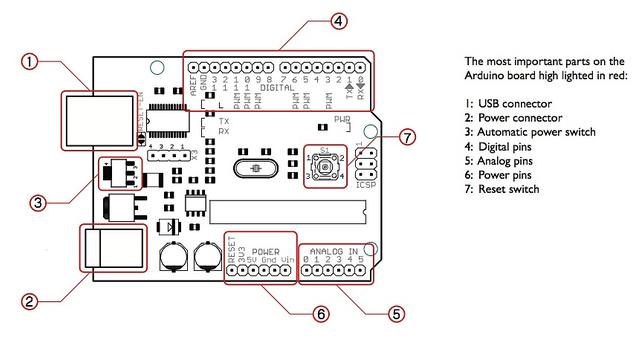
**APLICATIA ,,AUTOMOTIVE 1”**

* **Enunt**

Solutie automata pentru monitorziarea locurilor dintr-o parcare publica implementata cu ajutorul microcontroller-ului Arduino UNO si senzori fotoreflectivi de miscare TCRT5000. Monitorizarea locurilor se face prin intermediul celor 6 senzori si se afiseaza pe un LCD cu o interfata I2C. Bariera este controlata de un servomotor atasat.Comunicarea cu placa se face prin intermediul portului USB aflat pe placa Arduino, folosind modulul UART. De asemenea, codul este incarcat pe placa prin intermediul portului USB.

* **Detalii tehnice despre microcontroller-ul Arduino UNO**

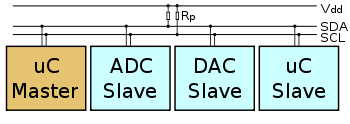
Arduino UNO este un microcontroller bazat pe microcontroller-ul detasabil AVR ATmega828. Are 20 de pini digitali de intrare/iesire (dintre care 6 pot fi utilizati ca intrari analog), un resonator de 16MHz si un port de incarcare. Programele pot fi incarcate foarte usor folosind IDE-ul Arduino. Versiunea R3 este a treia versiune si ultima de Arduino.



*Schema placii Arduino UNO*

* **Modulele utilizate in cadrul proiectului**
  + **Modulul I2C (Inter-Integrated Circuit)**

Este un protocol serial cu doua fire folosit pentru a conecta dispositive de viteza redusa, cum ar fi microcontrollere, EEPROM-uri, convertoare analog-digital/digital-analog, interfete I/O si alte perfierice similar aflate in sistemele incorporate. A fost inventat de compania Philips si acum este folosit de aproape toti producatorii importanti de circuite integrate. Fiecare dispozitiv de tip I2C slave are nevoie de o adresa, obtinuta de la NXP.



*Magistrala I2C*

I2C este popular deoarece este simplu de utilizat, poate fi mai mult de un master, doar viteza magistralei superioare este definita si sunt necesare doar doua fire cu rezistente pentru a conecta un numar nelimitat de dispositive I2C.

Interfata I2C foloseste doar doua fire: SCL (serial clock) si SDA (serial data). Ambele trebuie conectate impreuna cu un resistor la VCC.

In cadrul proiectului nostru, interfata I2C a fost folosita pentru conectarea LCD-ului la placa. Aceasta interfata reduce numarul de fire folosite la numai 4 (in loc de 16).

* + **Modulul UART (USB)**

Am folosit comunicarea serial (cunoscuta si ca UART) pentru a trimite date de la Arduino catre computer prin porturile seriale USB COM.

UART: receptor/transmitator asincron universal (abreviat UART), este o componenta hardware care translateaza datele intre tipurile serial si parallel. UART sunt utilizate de regula impreuna cu standardele de comunicare cum ar fi RS232, RS422. Am folosit UART la transferul datelor de la Arduino la computer prin porturile USB.

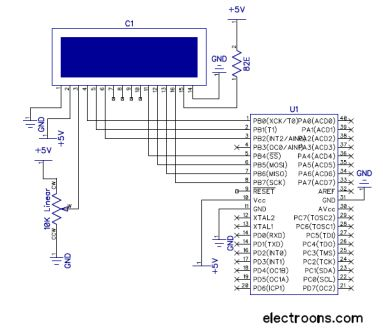
* **Sistemul de afisaj**

Pentru afisare am folosit un LCD (LCD 1602) bazat pe o interfata I2C. LCD1602 este un afisaj cu cristale lichid (character-type). Modelul 1602 afiseaza 2 linii a cate 16 caractere fiecare. Fiecare caracter este afisat intr-o matrice pixel de 5x7. LCD-ul are doua register, numite Command si Data.

In cadrul proiectului nostru, am folosit acest afisaj pentru a afisa numarul de locuri libere din parcare, cat si care dintre aceste locuri sunt libere.



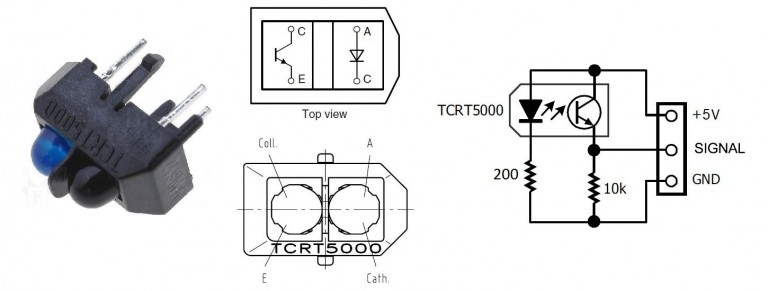
*Afisaj LCD 1602*



*Schema bloc afisaj LCD1602*

* **Senzori folositi**

Singurii senzori pe care i-am folosit in cadrul proiectului au fost senzorii fotoreflectivi de miscare TCRT5000 (in numar de 8). Senzorii TCRT5000 sunt senzori reflectivi, care include un emitator cu infrarosu si un fototranzistor intr-un ambalaj cu plumb care blocheaza lumina vizibila. Senzorul TCRT5000 functioneaza impreuna cu senzorul general de infrarosu, ca si cum senzorul ar avea un tub de emisie infrarosu, si un tub de receptie infrarosu.

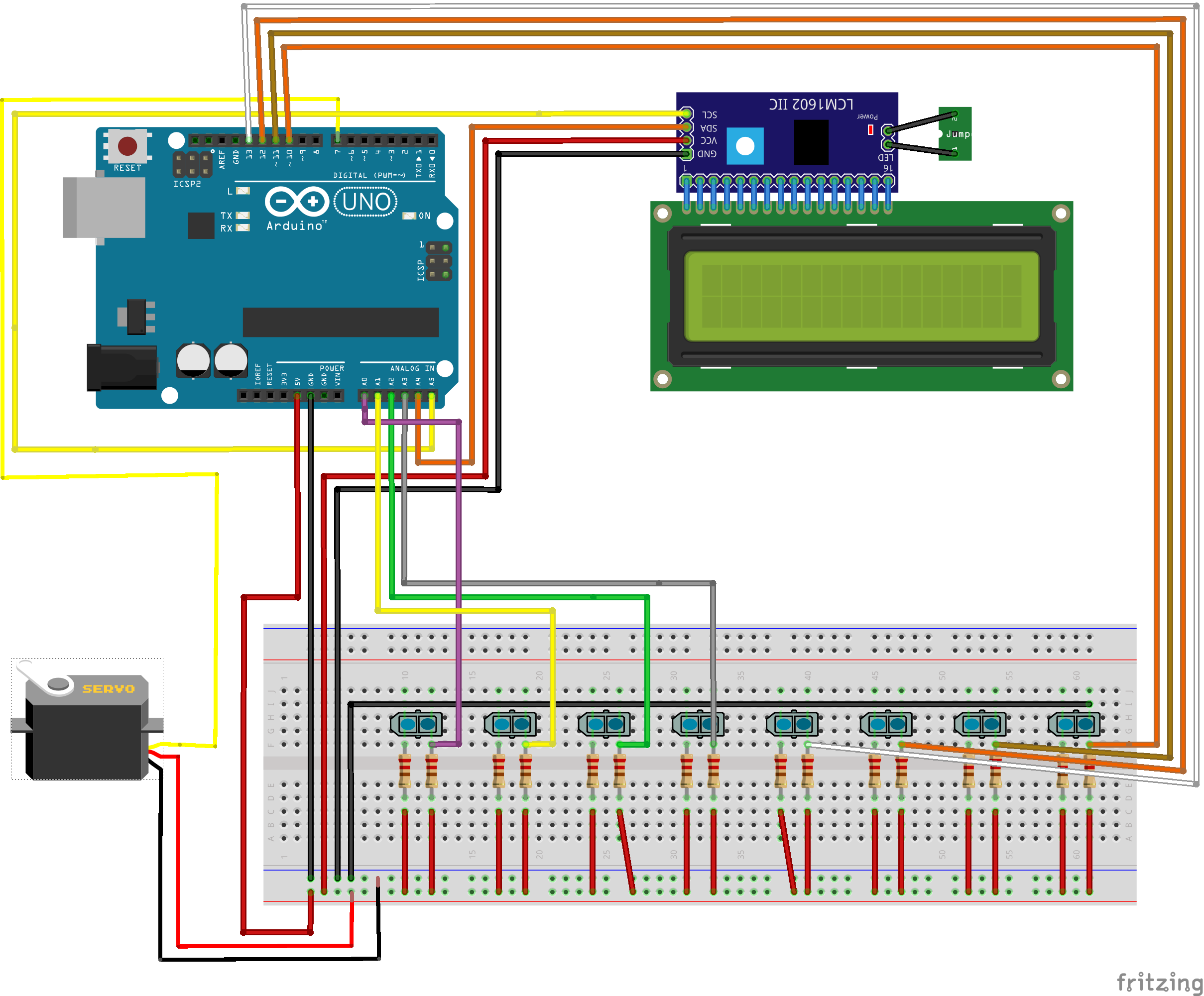


*Senzorul TCRT5000*

In cadrul proiectului nostru, 4 din acesti senzori au fost legati la 4 pini analog al placii Arduino, iar restul la pini digitali, pentru a obtine rezultate cat mai exacte (datorita limitarii placii Arduino UNO la numai 6 pini analogici).

Citirea valorilor intoarse de senzori a fost realizata folosind functiile *analogRead* si *digitalRead.*

* **Conectarea hardware**

 Schema conectarii hardware a fost realizata folosind tool-ul Fritzing.

* **Codul programului**

**#include <Wire.h>**

**#include <LiquidCrystal\_I2C.h>**

**#include <Servo.h>**

**// 16x2 I2C LCD**

**LiquidCrystal\_I2C lcd(0x27, 16, 2);**

**Servo servo;**

**const unsigned short parkingLotsPins[6] = {**

**10, 11, 12, 13, A3, A2**

**};**

**const unsigned short leavingSensorPin = A0;**

**const unsigned short enteringSensorPin = A1;**

**int enter, leave; // value readings for leaving/entering sensors**

**void setup() {**

**Serial.begin(9600);**

**/\***

**\* Pins Setup.**

**\*/**

**pinMode(leavingSensorPin, INPUT);**

**pinMode(enteringSensorPin, INPUT);**

**for (int i = 0; i < 6; i++) {**

**pinMode(parkingLotsPins[i], INPUT);**

**}**

**/\***

**\* Servo setup.**

**\*/**

**servo.attach(7);**

**servo.write(0);**

**/\***

**\* LCD Setup.**

**\*/**

**lcd.init();**

**lcd.backlight();**

**lcd.setCursor(0, 0);**

**lcd.print("Free park spots:");**

**}**

**void loop() {**

**unsigned short freeSpots;**

**lcd.setCursor(0, 1);**

**for (int i = 0; i < 6; i++) {**

**int val;**

**if (i < 4) {**

**if ((val = digitalRead(parkingLotsPins[i])) == HIGH) {**

**lcd.print(i + 1);**

**lcd.print(",");**

**}**

**}**

**else {**

**if ((val = analogRead(parkingLotsPins[i])) > 550) {**

**lcd.print(i + 1);**

**lcd.print(",");**

**}**

**}**

**}**

**lcd.print("\b ");**

**enter = analogRead(enteringSensorPin); // reading of the parking entering sensor**

**leave = analogRead(leavingSensorPin); // reading of the parking leaving sensor**

**if ((freeSpots = getFreeSpots()) < 1) { // in case there are no free spots => barrier closes and displays message**

**servo.write(0);**

**delay(1000);**

**lcd.setCursor(0, 1);**

**lcd.print("Parking is full.");**

**}**

**if (leave <= 550) { // if someone wants to leave**

**for (int i = 0; i <= 60; i++) { // open the barrier**

**servo.write(i);**

**delay(30);**

**}**

**while ((leave = analogRead(leavingSensorPin)) <= 550); // barrier stays up till the car leaves**

**delay(500); // wait 500ms**

**for (int i = 60; i >= 0; i--) { // close the barrier**

**servo.write(i);**

**delay(30);**

**}**

**}**

**if (enter <= 550) {**

**for (int i = 0; i <= 60; i++) { // open the barrier**

**servo.write(i);**

**delay(30);**

**}**

**while ((enter = analogRead(enteringSensorPin)) <= 550); // barrier stays up till the car enters the parking**

**delay(500); // wait 500ms**

**for (int i = 60; i >= 0; i--) { // close the barrier**

**servo.write(i);**

**delay(30);**

**}**

**}**

**if ((freeSpots = getFreeSpots()) > 0) {**

**lcd.setCursor(13, 1);**

**lcd.print("C:");**

**lcd.print(freeSpots);**

**}**

**}**

**unsigned short getFreeSpots() { // function to return the free spots count**

**unsigned short count = 0x06;**

**for (unsigned short i = 0x0; i < 0x6; i++) {**

**if (i < 4) {**

**if (digitalRead(parkingLotsPins[i]) == LOW) { // spots 1-4 are connected to digital pins**

**count --;**

**}**

**} else {**

**if (analogRead(parkingLotsPins[i]) <= 550) { // spots 5-6 are connected to analog pins**

**count --;**

**}**

**}**

**}**

**return count;**

**}**

* **Bibliografie**
  + http://wikipedia.org/
  + http://google.com/
  + https://www.elecrow.com/tcrt5000-reflective-infrared-sensor-photoelectric-switches-10pcs.html
  + <https://www.vishay.com/docs/83760/tcrt5000.pdf>