



UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Dispozitiv de măsurare a forței cu afișaj

Profesor coordonator:

Prof. Univ. Dr. Ing. Burileanu Corneliu

Studenți:

Petre Alexandru-Ionuț

Călătoae Diana-Dumitra

Grupa: 432B

An universitar: 2018-2019

Cuprins

1. Introducere	4
2. Placa de dezvoltare Intel Galileo	4
2.1. Prezentare generală.....	4
2.2. Specificații tehnice	5
2.3. Schema logică	6
2.4. Porturile de intrare / ieșire	6
3. Etapa de realizare hardware	8
3.1. Schema electrică a circuitului	8
3.2. Implementarea hardware	9
4. Etapa de realizare software	11
4.1. Schema logică	11
4.2. Implementarea software	12
5. Concluzii	14
6. Bibliografie	15

1. Introducere

Descrierea temei

Acest proiect constă în dezvoltarea unui dispozitiv de măsurare a forței și afișarea acesteia, utilizând placa Intel Galileo. Dispozitivul semnalează prin aprinderea ledului verde că forța măsurată cu ajutorul senzorului de forță se află în intervalul specificat, în timp ce aprinderea ledului roșu semnalează faptul că forța măsurată nu se află în intervalul specificat. Valoarea forței măsurate este afișată pe o bară de leduri și pe afișajele cu șapte segmente.

A fost o provocare de a implementa acest proiect, deoarece am avut ocazia de a lucra cu microcontrolerul Intel Galileo și cu Arduino IDE. Datorită acestui proiect am avut oportunitatea de a utiliza cunoștințele teoretice și de a le implementa într-o aplicație practică.

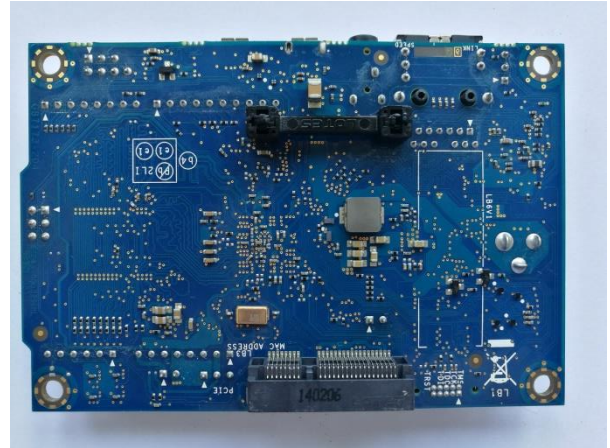
2. Placa de dezvoltare Intel Galileo

2.1. Prezentare generală

Intel Galileo Gen 2 se bazează pe Intel Quark SoC X1000, un sistem de clasă Intel Pentium pe un chip (SoC). Aceasta este prima placă bazată pe arhitectura Intel proiectată pentru a fi compatibilă din punct de vedere hardware și software cu pinii de pe shield-urile pentru Arduino Uno R3.

Intel Galileo include un slot complet mini-PCI Express, port Ethernet 100Mb, header USB TTL UART, port USB Host, port USB Client și 8 MByte NOR flash și un suport de card microSD de până la 32GB. Procesorul este capabil să ruleze cod scris într-un limbaj de programare care este foarte similar cu limbajul C++.

Aceasta suportă o distribuție de Linux, Yocto 1.4 Poky Linux. Placa Intel Galileo Gen 2 se conectează la portul USB al calculatorului folosind un cablu de tip USB Micro-B. Aceasta poate fi programată cu ajutorul mediului de dezvoltare Arduino.



Placa de dezvoltare Intel Galileo

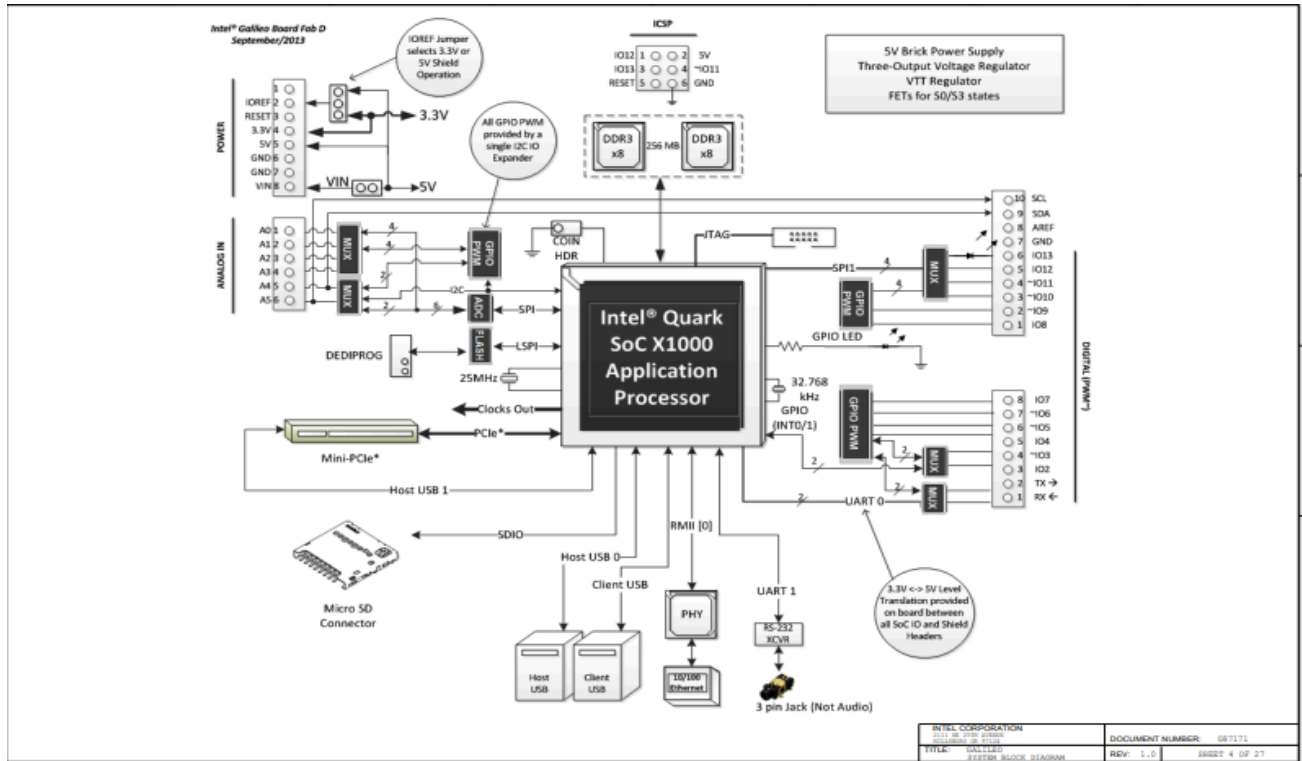
2.2. Specificații tehnice

- Procesor: Intel Quark X1000
- Memorie flash: 8 Mbyte – bootloader; 256-512Kbyte – pentru programe
- Frecvență procesor: 400 MHz
- Memorie SRAM: 512KB
- Memorie EEPROM: 11KB
- Pini digitali I/O: 14(dintre care 6 pot oferi output PWM)
- Pini analogi input: 6

Limbaje suportate:

- Arduino
- JavaScript și Node.js
- C++
- Java

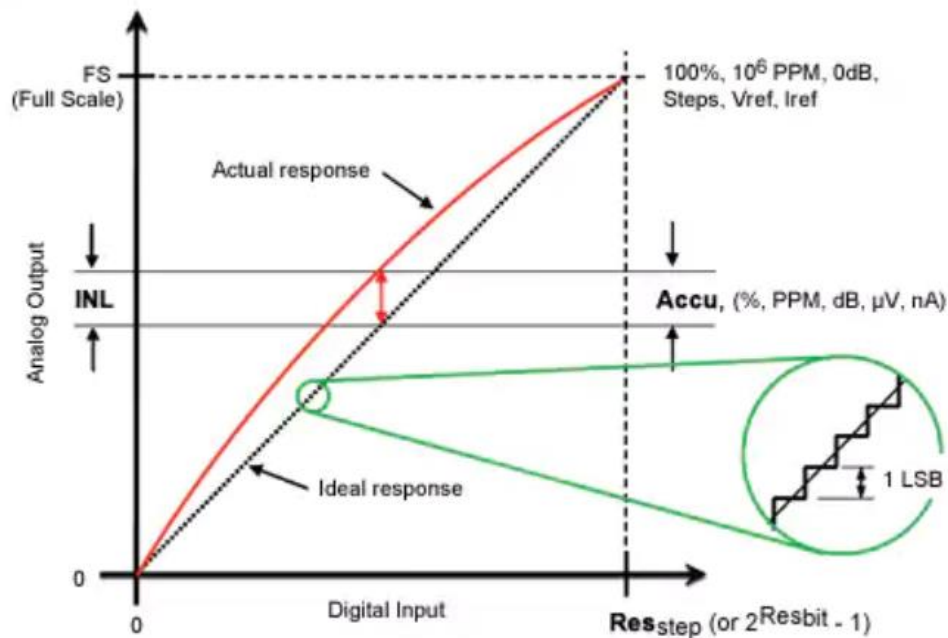
2.3. Schema logică



Porturile de intrare / ieșire - GPIO:

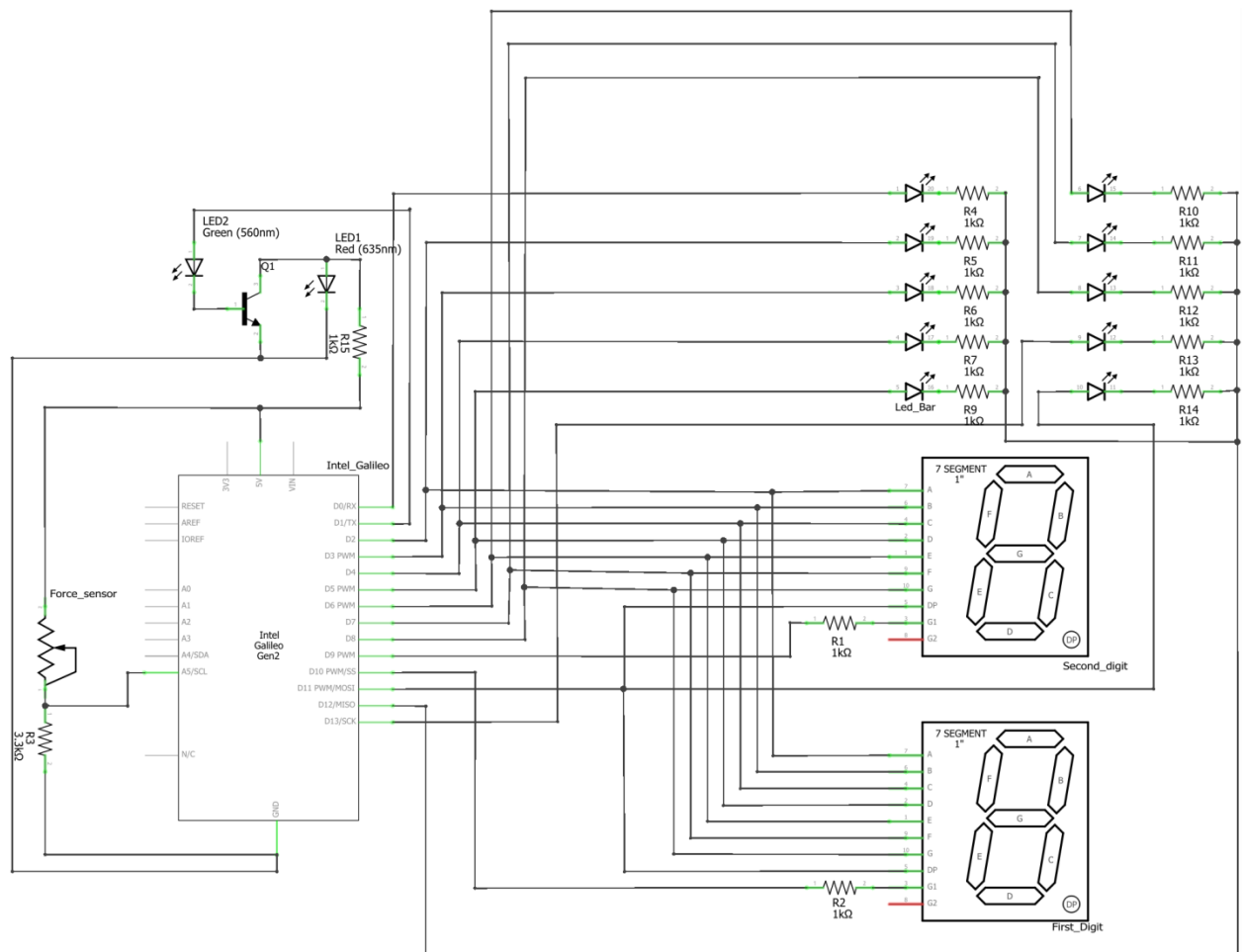
Porturile de intrare / ieșire de uz general prezintă 14 pini digitali (Tx, Rx, IO2 – IO12); toți acești pini pot fi folosiți ca pini de intrare, cât și ca pini de ieșire. Acești pini pot fi utilizați cu ajutorul funcțiilor: `pinMode()`, `digitalWrite()`, `digitalRead()`. În momentul în care cei 14 pini sunt folosiți ca ieșire, curentul maxim este de 10mA; dar cand sunt folosiți ca intrare, acceptă un curent maxim de 25mA. Rezistența internă de pull-up este de 10K Ω . Dintre cei 14 pini, 6 dintre ei pot fi folosiți și pentru PWM (Pulse Width Modulation).

Pinii analogici de intrare sunt A0-A5 (6 pini), implicit măsoară tensiuni între 0-5V; aceștia pot fi accesați cu ajutorul funcției `analogRead()`, cei 6 pini folosesc un convertor ADC de tip SAR AD7298. Rezoluția este pe 12b, rezultând 2^{12} (4096) valori discrete diferite.



3. Etapa de realizare hardware

3.1. Schema electrică a circuitului



3.2. Implementarea hardware

Din punct de vedere hardware, pentru afișare am optat pentru o bară de leduri și două afișaje cu 7 segmente; atât bara de leduri, cât și afișajele cu 7 segmente au pini conectați la pini digitali.

Pentru afișajele cu 7 segmente avem:

- Segmentul A conectat la pinul D2
- Segmentul B conectat la pinul D3
- Segmentul C conectat la pinul D4
- Segmentul D conectat la pinul D5
- Segmentul E conectat la pinul D6
- Segmentul F conectat la pinul D7
- Segmentul G conectat la pinul D8
- Punctul este conectat la pinul D11

Pentru bara de leduri avem:

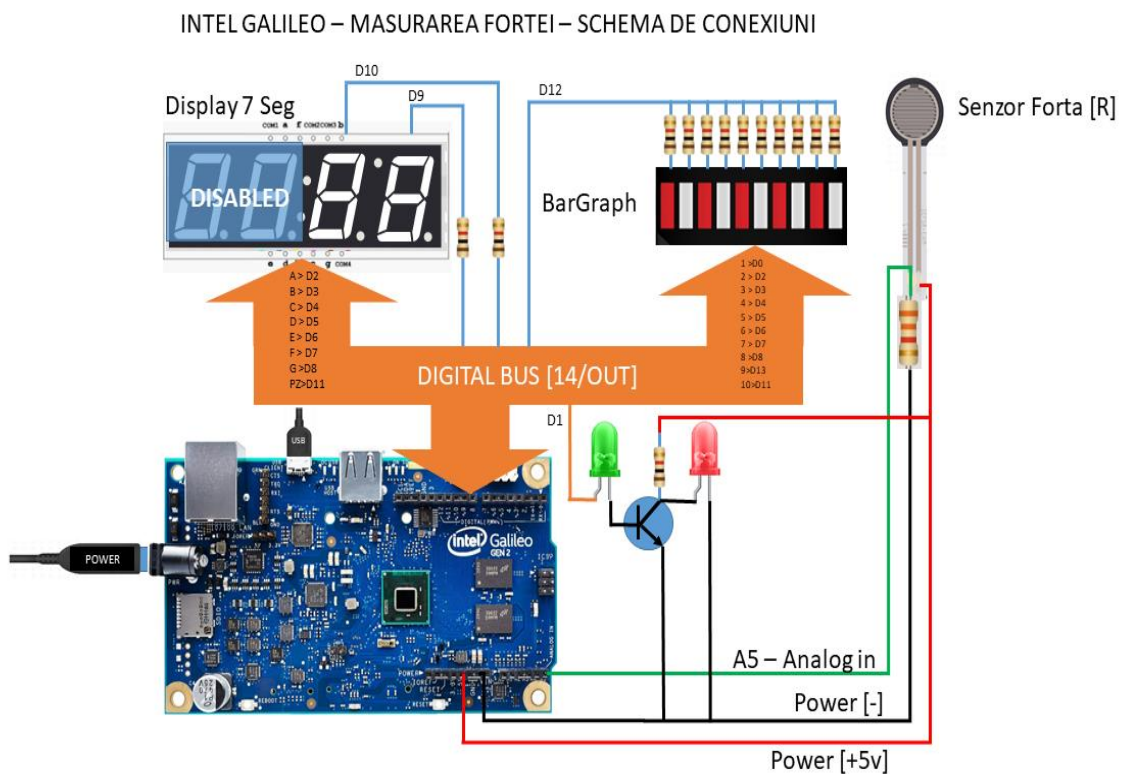
- Ledul 1 conectat la pinul D0
- Ledul 2 conectat la pinul D2
- Ledul 3 conectat la pinul D3
- Ledul 4 conectat la pinul D4
- Ledul 5 conectat la pinul D5
- Ledul 6 conectat la pinul D6
- Ledul 7 conectat la pinul D7
- Ledul 8 conectat la pinul D8
- Ledul 9 conectat la pinul D13
- Ledul 10 conectat la pinul D11

Bara de leduri este conectată la zece rezistențe care sunt conectate între catodul fiecărui led și pinul D12, toate rezistențele au aceeași valoare ($R_4=R_5=R_6=R_7=R_9=R_{10}=R_{11}=R_{12}=R_{13}=R_{14}=1K\Omega$).

Senzorul de forță are un pin conectat la 5V și celălalt pin conectat la A5, între acest punct și masă se află o rezistență $R_3=3.3K\Omega$.

Ledul verde este conectat la baza tranzistorului bipolar npn și pinul D1, acesta se aprinde când tranzistorul se află în conducție; ledul roșu este conectat între colectorul tranzistorului npn și masă (emitorul tranzistorului este conectat la masă), acesta este aprins în momentul în care tranzistorul se află în blocare. Tranzistorul de tip npn are rol de inversor. Rezistența $R_{15}=1K\Omega$ se află între colectorul tranzistorului și 5V.

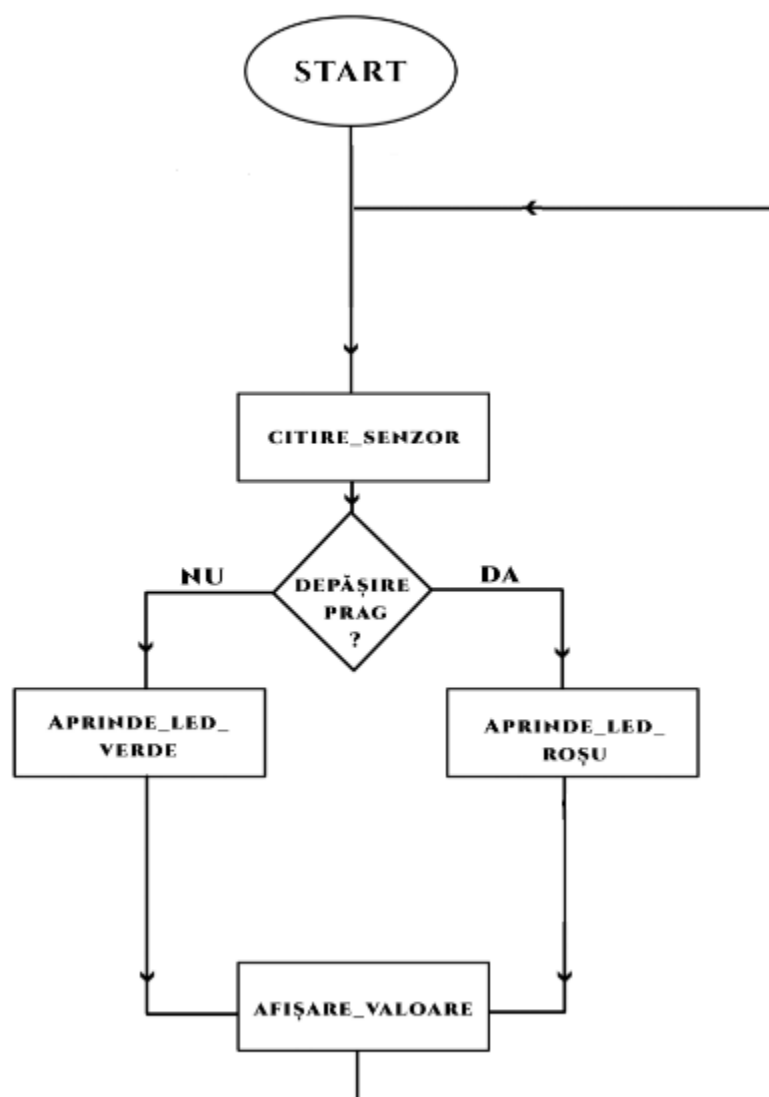
Pentru o mai bună ilustrare a schemei, am realizat următorul model:



C2 General

4. Etapa de realizare software

4.1. Schema logică



4.2. Implementarea software

Din punct de vedere software a fost dorită o implementare cât mai simplă din punct de vedere al logicii, acest lucru se poate observa și în codul ce urmează a fi comentat.

- În cadrul funcției ce rulează o singură dată –setup() – se pornește comunicarea serială și se setează pini 0-13 de tip ieșire:

```
void setup()
{
  Serial.begin(9600);
  while (!Serial) ; // wait for serial port to connect. Needed for native USB
  Serial.println("start");
  for (int i=0;i<14;i++)
  {
    pinMode(i, OUTPUT); // taking all pins from 0-13 as output
  }
}
```

- funcția loop rulează până la oprirea alimentării și citește forța senzorului, după care o scalează pentru a fi cuprinsă în intervalul [0.0 , 9.9], în cazul în care forța nu se află în acest interval nu vom avea o valoare afișată pe afișajele cu 7 segmente și nici pe bara de leduri, pentru acest caz avem ledul roșu aprins. În cazul în care valoarea este cuprinsă în interval se aprind segmentele corespunzătoare valorii forței citite:

```
void loop()
{
  float fsrForce = fsr.getForce(); //getting force from sensor
  int Force= fsrForce /45 ; //scaling force in [0.0 , 9.9]
  Serial.print(F("Force: ")); Serial.print(Force); Serial.println(F(" [g]"));

  int FirstDigit = Force/10;
  int SecondDigit = Force-FirstDigit*10;

  if (Force > 99){
    digitalWrite(1, LOW); // turning on red led
  }
  else {
    digitalWrite(1, HIGH); // turning on green led
  }
  digitalWrite(11, HIGH); // turning off decimal point
```

```

int Digit = FirstDigit;
digitalWrite(10, HIGH); // turning on first digit
digitalWrite(12, HIGH); // turning off led bar
digitalWrite(9, LOW); // turning off second digit

        //Setarea segmentelor corespunzătoare
        //primului digit

delay(9);

digitalWrite(11, HIGH); // turning off decimal point
digitalWrite(10, LOW); // turning off first digit
digitalWrite(9, HIGH); // turning on second digit
digitalWrite(12, HIGH); // turning off led bar
Digit = SecondDigit;

        //Setarea segmentelor corespunzătoare
        //pentru digitul doi

delay(9);

digitalWrite(10, LOW); // turning off first point
digitalWrite(9, LOW); // turning off second point
digitalWrite(12, LOW); // turning on led bar
digitalWrite(11, HIGH); // turning off decimal point
digitalWrite(8, HIGH);
Digit = FirstDigit;

        //Setarea ledurilor din bara de leduri
        //corespunzătoare

}

```

5. Concluzii

În cadrul mediului de dezvoltare Arduino se poate folosi orice limbaj de programare (C/C++/Java etc.) pentru a scrie programele. Compilatorul inclus în cadrul aplicației poate înțelege toate aceste limbaje, pe care le traduce în limbaj mașină și le trimite către placa de dezvoltare.

După cum am precizat încă de la începutul documentației prezente, scopul proiectului nostru a fost implementarea unui dispozitiv de măsurare a forței și afișarea acesteia pentru o anumită gamă de valori, în funcție de valoarea forței se aprinde ledul verde sau ledul roșu. Acest dispozitiv a fost realizat cu ajutorul lui Intel Galileo. Dispozitivul realizându-se cu ajutorul mediului de dezvoltare Arduino.

Noi considerăm că mediul de dezvoltare Arduino este prietenos și o provocare pentru noi să lucrăm cu acest mediu de dezvoltare, cât și realizarea unui proiect care implică parte software și hardware

6. Bibliografie

[1] - Specificații tehnice pentru placa Intel Galileo

<http://dlmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Boards/Galileo%20Schematic.pdf>

[2] - Ghid de începători cu Intel Galileo

<https://learn.sparkfun.com/tutorials/galileo-getting-started-guide>

[3] – Ghid componente și Arduino

www.wiki.eprolabs.com

[4] - <https://www.robofun.ro/>