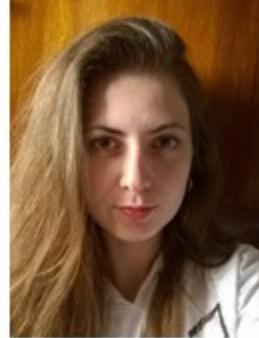
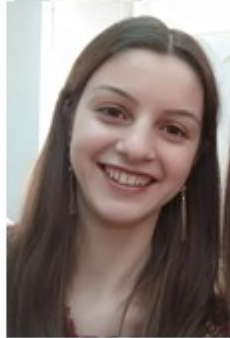


Studenti:



Arcana Iuliana  
Irimia Diana-  
Cătălina  
Proca Mihaela

[arcanaiuliana@gmail.com](mailto:arcanaiuliana@gmail.com) [diana.irimia.catalina@gmail.com](mailto:diana.irimia.catalina@gmail.com) [mihaela.proca94@yahoo.com](mailto:mihaela.proca94@yahoo.com)

Etape proiect:

**1. Rezumat, resurse materiale, timp/atribuții echipă**

Tema: Sistem de irigare comandat automat și prin browser.

Rezumat: XMC 1100 primește date de la senzori și trimite comenzi către Raspberry Pi. Aceasta afișează datele în browser și trimite comenzi, fie date de utilizator, fie automat către XMC 1100 care comandă motoarele și pompița.

Resurse materiale:

- Raspberry PI 3 Model B
- XMC 1100
- Driver motor bazat pe cipul L298N
- 2 motoare DC de 9V
- Pompiță 5V
- 2 baterii de 9V legate în serie
- tranzistor npn tip 122
- Fire de legătură
- Furtun pentru pompiță
- Senzori de umiditate a solului
- kit mașinuță(schelet, 4 roți, suport motoare)

Timp: 14 săptămâni, 3-4h/săpt

Atribuții:

Arcana Iuliana :

- Implementare scripturi python și realizarea comunicării seriale între XMC1100 și Raspberry PI
- Configurație hardware senzori umiditate
- Documentatie

Irimia Diana Cătălina:

- Configurație software (creare site)
- Contribuție hardware

Proca Mihaela:

- Configurație hardware
- Creare program de comandă a motoarelor și a pompei

## 2. Raport sintetic, alternative soluții și ciorne

Ideea de a realiza un sistem de irigare care poate fi controlat de la distanță este foarte utilă. Atunci când ai acces mereu la date care să îți spună starea solului, poți controla printr-o singură comandă sistemul obținând astfel rezultate mai rapide, fără a fi necesară monitorizarea directă.

Sistemul inițial este conceput pentru un număr mic de plante, dar poate fi extins în așa fel încât să poată fi folosit, de exemplu, într-o seră.

Am ales Raspberry Pi deoarece corespunde cerințelor noastre dar în locul acesteia se putea folosi și BananaPi sau Parallella.

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

<https://www.raspberrypi.org/learning/hardware-guide/>

[https://www.terraelectronica.ru/show\\_pdf.php?pdf=%2Fds%2Fpdf%2FTechicRP3.pdf](https://www.terraelectronica.ru/show_pdf.php?pdf=%2Fds%2Fpdf%2FTechicRP3.pdf)

<https://www.parallella.org/>

<http://www.banana-pi.org/>

Link-uri către alte proiecte asemănătoare:

<http://www.instructables.com/id/Raspberry-Pi-Controlled-Irrigation-System/>

<https://www.youtube.com/watch?v=nUHizmyt74>

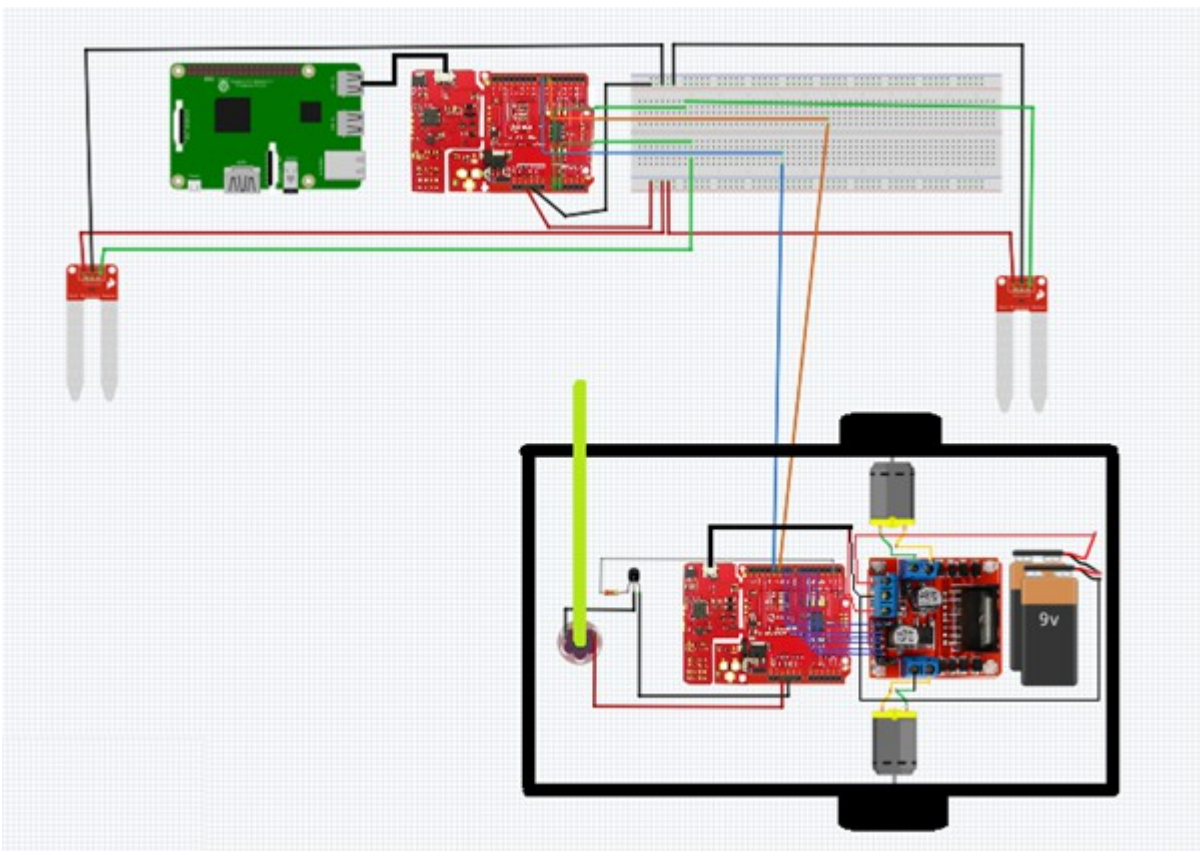


Fig1. Ciornă soluție

### 3. Implementare soluție, secvență de test

Plăcile de dezvoltare folosite în realizarea proiectului sunt **XMC1100** și **Raspberry Pi 3 Model B**. Am utilizat comunicația serială pentru recepția și transmiterea de date între cele două plăci. Pe Raspberry Pi am configurat serverul web Apache, iar cele două plăci XMC1100 au fost programate în Arduino IDE.

Principiu de funcționare:

Placa Infineon este alimentată la driver, driverul fiind alimentat de cele 2 baterii de 9V. Motoarele sunt conectate la drivere prin plusul și minusul de la fiecare motor.

De la driver avem conectați 3 pini pentru fiecare motor:

- 1 pin de enable-pornește/oprește motor
- 1 pin de direcție-înainte
- 1 pin de direcție-înapoi,

aceștia fiind conectați la XMC1100.

Pompa este conectată prin intermediul tranzistorului la XMC1100.

În funcție de datele primite pe cei doi pini de intrare (1 sau 2) ai plăcuței Infineon și starea în care se află mașinuța, aceasta se va deplasa (sau nu) și va uda planta, reținând noua stare. De exemplu, dacă de pe interfață se apasă butonul **udă planta 1**, atunci XMC 1100 va primi pe unul din cei 2 pini de intrare 1. Dacă, în momentul respectiv, mașina se află la planta 2, din program se va apela funcția **backward()** și mașina va merge cu spatele până la planta destinație, după care se va apela funcția **wet()**. Ca urmare a execuției acesteia, planta va fi udată și se va memora noua stare a mașinii, în cazul de față **planta 1**, și va rămâne în stand by până la o nouă schimbare a pinilor de intrare.

Alți 2 pini sunt folosiți pentru a primi comanda de udare a plantei 1, respectiv 2, comanda fiind transmisă de un alt XMC1100 care primește datele de la senzorii de umiditate de la cele două plante.

Valorile înregistrate de cei doi senzori de umiditate sunt transmiși pe cale serială de pe XMC1100 către Raspberry Pi. Aici sunt preluate și stocate într-un fișier printr-un script python și transmise către server, unde vor fi accesibile utilizatorului. La comanda de udare a plantei 1 sau 2, se trimit comenzi către XMC1100 care va acționa în modul prezentat anterior.

Fișiere folosite pentru realizarea comunicației seriale:

#### **writeSensorData.py**

```
serial_port = '/dev/ttyACM0';
baud_rate = 9600;

ser = serial.Serial(serial_port, baud_rate)

try:
    while 1:
        line = ser.readline().rstrip()
        hum1,hum2=line.split(",")
        f=open('humidity.dat','w+')
        f.write('{}'.format(hum1))
        f.write(',')
        f.write('{}'.format(hum2))
```

#### **scriptPlantă2.py**

```
def fnct():
```

```

print("entered\n")
port='/dev/ttyACM0'
try:

    ser=serial.Serial('/dev/ttyACM0',9600)
    ser.flushInput()
    time.sleep(2)
    print("Connection to "+port+" established succesfully!\n")
    file = open("/var/www/html/planta2.txt","r")
    ch=file.readline(1)
    if ch=='2':
        ser.write('2')
    else:
        print(ch)
    os.remove("/var/www/html/planta2.txt")
    ser.close()
except Exception as e:
    print(e)

def main():
    while 1:
        fnct()

main()

```

#### **senzor\_umiditate.ino**

```

#include <Wire.h>
#define pinsenzor1 17
#define pinsenzor2 18
#define LED1 7
#define LED2 6

char number;
int citiredate1, citiredate2;
int procentumiditate1,procentumiditate2;

void setup()
{
    Serial.begin(9600);
    pinMode(LED1,OUTPUT);
    pinMode(LED2,OUTPUT);
}

void loop()
{
    citiredate1 = analogRead(pinsenzor1); // citire valoare pe intrarea analogica
    citiredate2 = analogRead(pinsenzor2);
    procentumiditate1 = map(citiredate1, 1023,0,0,100); // transformare date primare in procente
    procentumiditate2 = map(citiredate2, 1023,0,0,100);

    Serial.print(procentumiditate1);
    Serial.print(",");
    Serial.print(procentumiditate2);
    Serial.print("\n");

    number=Serial.read();
    if((number-'0')==1)
    {
        digitalWrite(LED1,HIGH);
        delay(3000);
        digitalWrite(LED1,LOW);
    }
}

```

```

else if((number-'0')==2)
{
    digitalWrite(LED2,HIGH);
    delay(3000);
    digitalWrite(LED2,LOW);
}
else if (procentumiditate1 <50)
{
    digitalWrite(LED1,HIGH);
    delay(3000);
    digitalWrite(LED1,LOW);
}
else if(procentumiditate2 <50)
{
    digitalWrite(LED2,HIGH);
    delay(3000);
    digitalWrite(LED2,LOW);
}

delay(3000);
}

```

### Index.html



```

<html>
  <body>
    <head>
      <title>SI</title>
      <meta charset="utf-8">
      <link rel="stylesheet" type="text/css" href="css/register.css">
    </head>
    <form action="read_credentials.php" method="post">
      <h1>LOGIN</h1>
      <fieldset>
        <label for="name">Nume</label>
        <input type="text" id="name" name="user_name">
        <label for="password">Parola</label>
        <input type="password" id="password" name="user_password">
      </fieldset>
      <button type="submit" value="click" name="submit">SUBMIT</button>
    </form>

```





```
</body>  
</html>
```