

# Flood Security System - Chiper Alexandra-Diana

Studentă: *Alexandra-Diana CHIPER (117182)*

Grupă: **335CB**

An universitar: **2022-2023**

## Introducere

Sistem de alarmă și protecție împotriva inundațiilor și prevenirea accidentelor mortale produse prin electrocutare. Nivelul apei dintr-o încăpere tehnică este permanent monitorizat cu ajutorul unor senzori de nivel al apei. Dacă se detectează prezența apei în încăpere, atunci cu ajutorul unor pompe de apă controlate prin intermediul microcontrollerului, aceasta va fi evacuată în exterior.

### Pentru cine poate fi util acest proiect?

Acest proiect poate fi util în casele construite în zone ușor inundabile sau pentru cazurile în care nivelul pânzei de apă freatică este fluctuant și apare o defecțiune la flotorul de acționare al pompei de evacuare din bașă.

Sursă de inspirație (#16):

- <https://all3dp.com/2/most-useful-arduino-projects/>

## Descriere generală

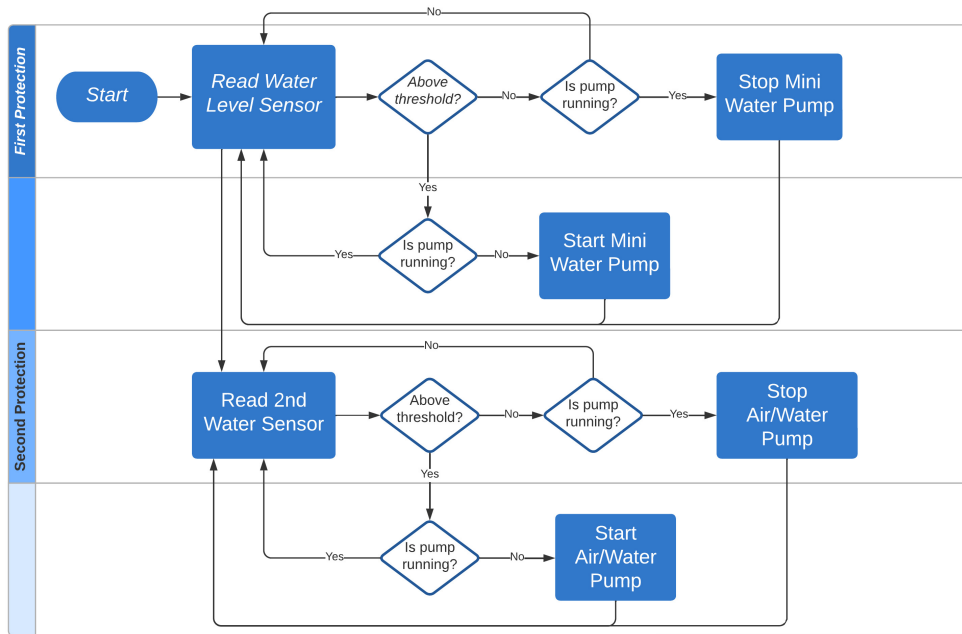
Când apa atinge nivelul de alertă este pornită pompa de serviciu din bașă. Dacă nivelul apei continuă să crească și ajunge la nivelul de pericol este pornită și a doua pompă de siguranță, cu debit mai mare. Cele două praguri de alertă sunt citite cu ajutorul a doi senzori de nivel al apei, plasați la înălțimi diferite.

Starea curentă a nivelului apei va fi afișată pe un ecran LCD și abstractizată vizual printr-un LED RGB astfel:

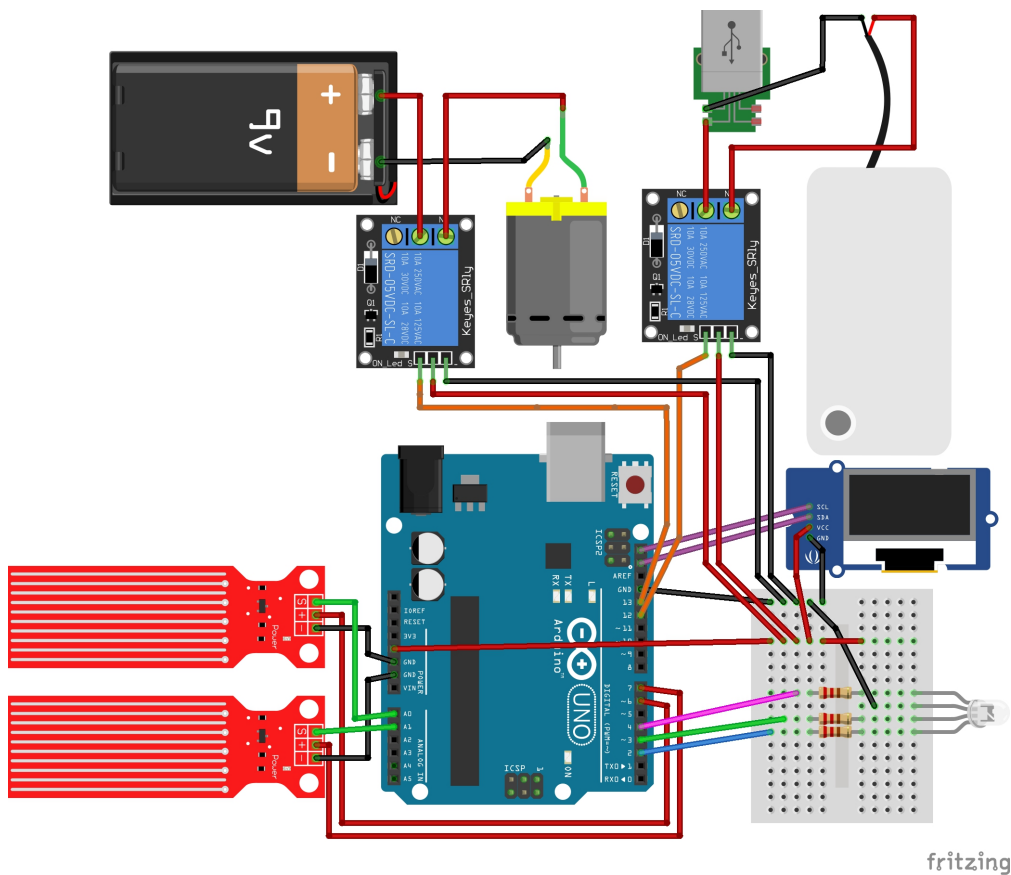
- Lipsa apei la nivelul sensorului - **EMPTY** și ledul **VERDE**
- 1 cm deasupra nivelului pardoselii - **LOW** și ledul este **ALBASTRU**
- 2.5 cm deasupra nivelului pardoselii - **MEDIUM** și ledul este **GALBEN**
- 4 cm deasupra nivelului pardoselii - **HIGH** și ledul este **ROȘU**
- peste 30 de cm - se afișează intermitent **DANGER** și ledul este **ROȘU**

Dacă imaginile apar prea mici, faceți click pe ele.

### Organigrama



### Schema electrică



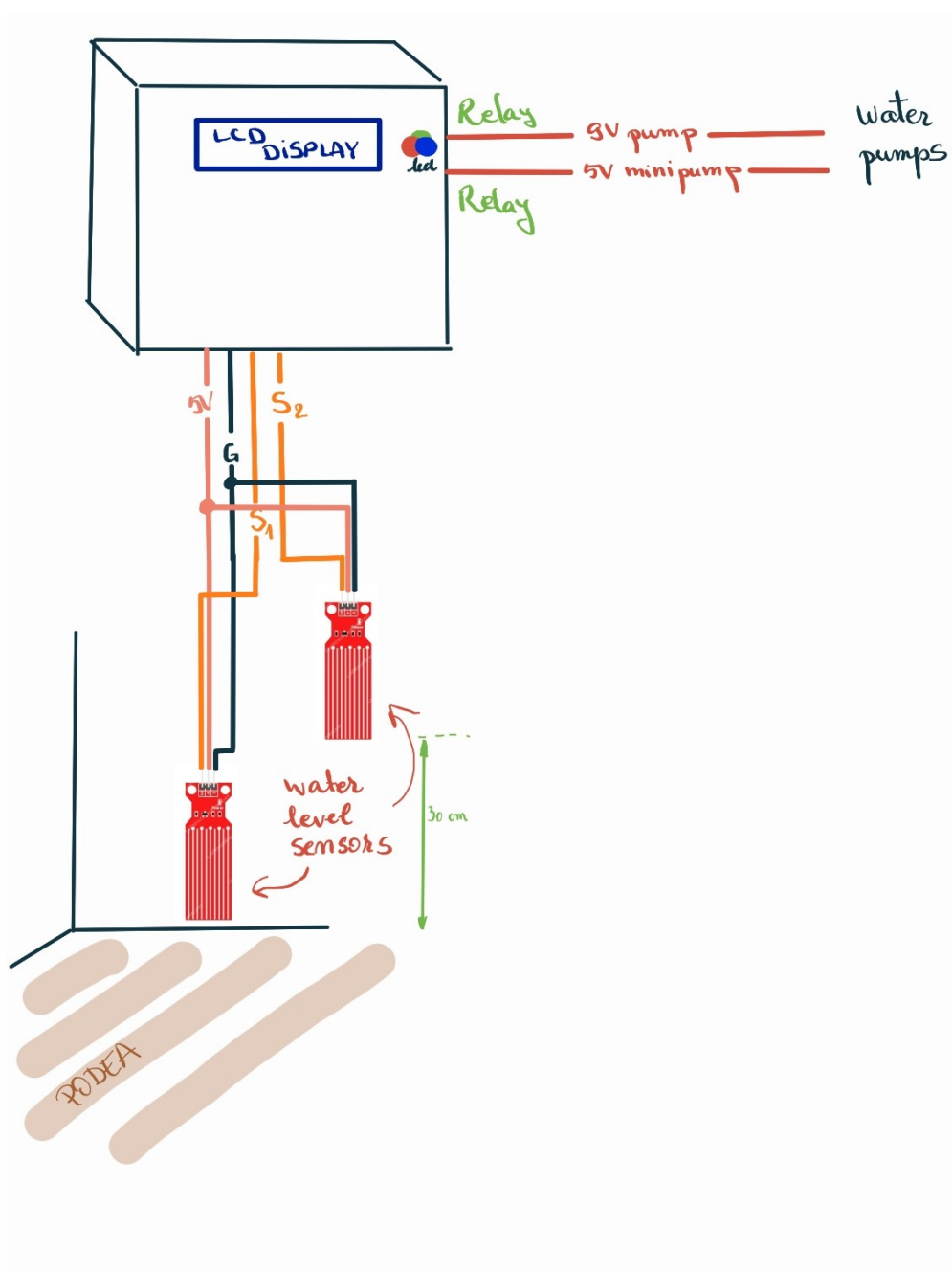
### Schemă



## Hardware Design

Componenta principală este plăcuța arduino, la care se leagă

- Input
  - 2 senzori analogici de nivel al apei
- Output
  - ecran LCD 16x2 cu interfață I2C
  - 2 relee cu care sunt comandate pompele de apă
  - led RGB pentru semnalizare luminoasă

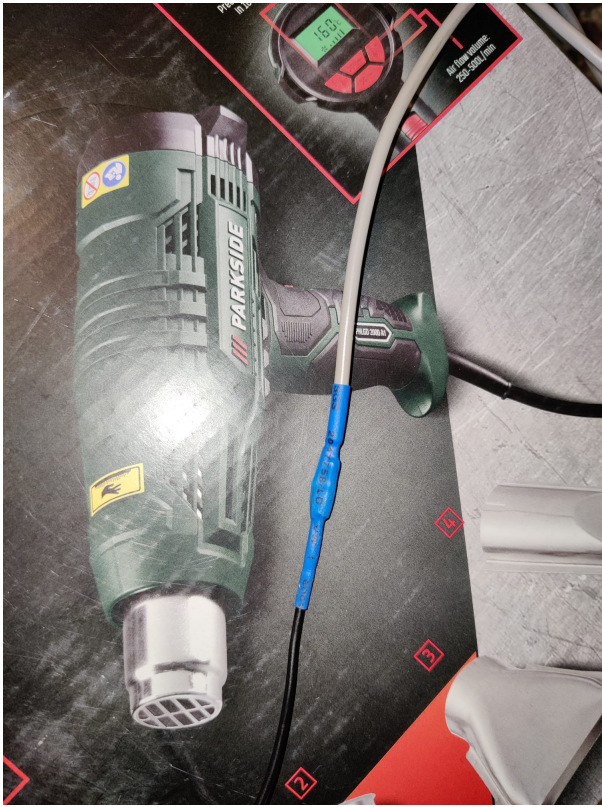


## Lista de componente

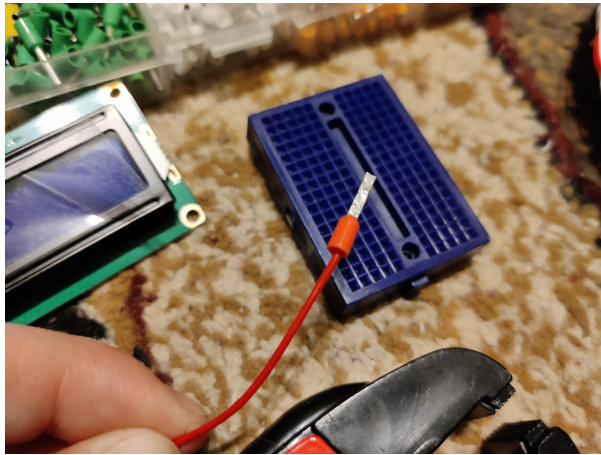
Componentă	Cantitate
Placă de dezvoltare compatibilă Arduino UNO R3	1
Senzor de nivel al apei	2
Ecran LCD cu interfață I2C	1
Releu cu comandă high-level 5V	2
Baterie 9V	1
Power bank 5V	2
Cablu USB	1
Conector rapid 3×1 și 5×1 pini	2
Mufă alimentare (mamă + tată)	2
Doză derivație	1
LED RGB	1

## Lipituri

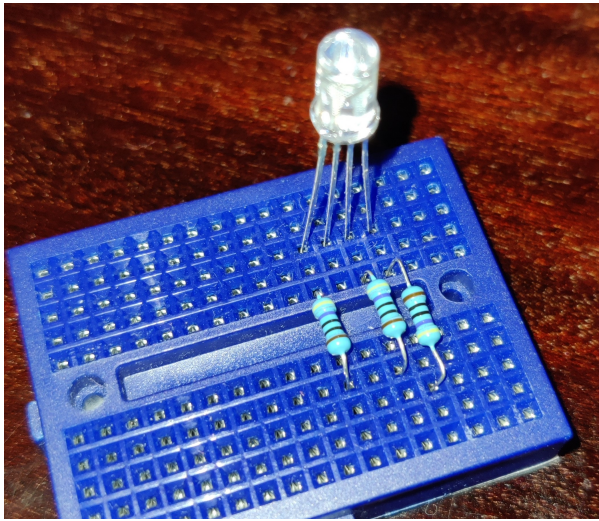




### Conectare



### Finisaje Proiect



## Software Design

- **mediu de dezvoltare:** Arduino IDE și VS Code
- biblioteci folosite: LiquidCrystal\_I2C.h

Notiuni folosite: Timere, Întreruperi, ADC, I2C, UART

## Mod de funcționare

Cu ajutorul Timer-ului 1 de pe plăcuța Arduino se generează o întrerupere la fiecare 0.5s (500 ms) care declanșează startul conversiei AD pentru primul senzor legat la pinul A1 analogic. La rândul său, modulul ADC generează o întrerupere când conversia s-a terminat, iar în rutina de tratare a acestei întreruperi se citește rezultatul într-una din două variabile, în funcție de un flag setat în funcția loop().

## Funcția loop()

La fiecare 0.5s se verifică valoarea primului senzor, care se încadrează apoi între niste praguri astfel:

1. **valoare\_senzor** < noWater\_threshold
  - totul este ok
2. noWater\_threshold < **valoare\_senzor** < lowerThreshold

- se aprinde LED-ul verde pentru a atentiona persoanele de existenta unei pelicule de apă la nivelul podelei
  - ambele pompe de apă sunt oprite
3. `lowerThreshold < valoare_senzor < upperThreshold`
- se aprinde LED-ul albastru pentru a atentiona persoanele de existenta unui strat mai consistent de apă la nivelul podelei
  - prima pompă de apă (cea de capacitate mai mică) este deschisă
  - pompa de capacitate mai mare este închisă
4. `valoare_senzor > upperThreshold`
- se aprinde LED-ul roșu pentru a atentiona persoanele de prezenta unui strat de apă care depășește 30 de cm, adică există posibilitatea de curentare sau de pericole mortale în cazul în care se intră în încăperea afectată fără echipament corespunzător
  - ambele pompe sunt pornite
  - se citește și valoarea celui de-al doilea senzor.
    - dacă acesta detectează prezența apei, atunci cea de-a doua pompă este pornită
    - dacă nu, cea de-a doua pompă este oprită

Starea curentă a sistemului este afișată pe ecranul LCD și în paralel este printată și pe interfața serială.

## Funcția `setup()`

Se inițializează pinii pentru senzori, relee pentru pompe și led-uri. Se configurează Timer1 și modulul ADC astfel:

[timer\\_setup.ino](#)

```
void config_timer1() {
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;

    // tc = f_cpu / (prescaler * f_needed) - 1 = f_cpu / prescaler * time
    // - 1
    OCR1A = (F_CPU / PRESCALER) * SAMPLING_INTERVAL_S - 1; // Calculate
    // the compare value

    TCCR1B |= (1 << WGM12); // CTC mode
    TCCR1B |= (1 << CS12); // 256 prescaler
}

void init_timer1() {
    TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt
}
```

[adc\\_setup.ino](#)

```
void config_adc() {
  ADMUX = 0;

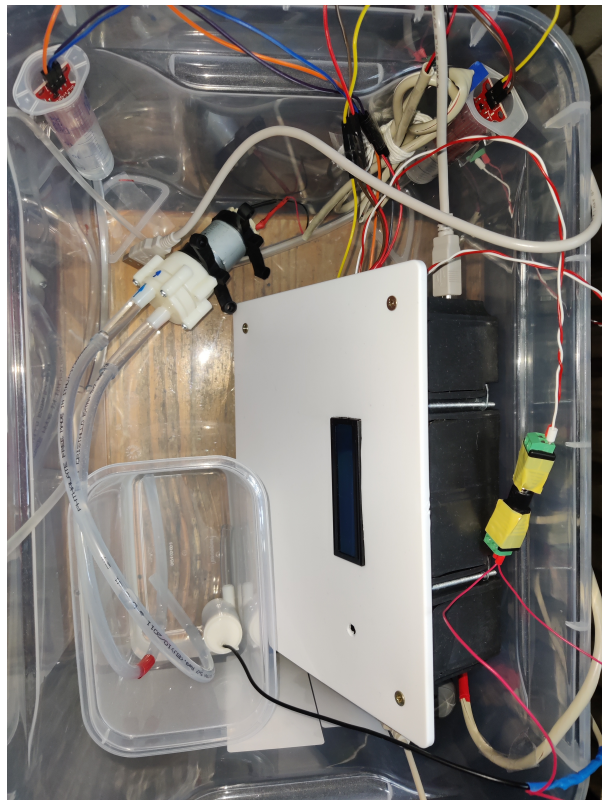
  /* AVCC with external capacitor at AREF pin */
  ADMUX |= (1 << REFS0);

  ADCSRA = 0;
  // Set the ADC prescaler to 128 for a 16MHz clock, giving an ADC
  // clock of 125kHz
  ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
}

void init_adc() {
  ADCSRA |= (1 << ADEN); // Enable ADC
  ADCSRA |= (1 << ADIE); // Enable ADC interrupts
}
```

## Rezultate Obținute

Proiectul funcționează așa cum a fost proiectat. Proiectul poate funcționa atât independent, prin alimentarea plăcuței arduino la o baterie de 9V sau în modul de management, cu plăcuța arduino conectată prin usb la un calculator care are un emulator pentru consola serială, pentru a putea urmări mesajele printate în cazul în care ecranul LCD sau LED-ul de informare se defectează.




## Demo proiect

- [Demo proiect](#)

## Concluzii

Proiectul a fost interesant de realizat, fiind primul proiect de acest fel, care să implice și partea de design și realizare hardware.

- A fost mai dificilă parte de calibrare a senzorilor (stabilirea pragurilor) deoarece pare că sunt mai sensibili spre vârf și mai puțin sensibili spre partea superioară și a durat ceva până să mă prind.
- O alta problemă întâlnită a fost comportamentul diferit al proiectului atunci când alimentam arduino de la o baterie de 9V față de ce făcea când era conectat prin usb la calculator. Problema era de la ADC, și mai exact de la faptul că AVcc este legat la supply voltage și nu la 5V de pe arduino ⇒ rezultatele senzorilor erau raportate fie la 5V fie la 9V și pragurile stabilite nu mai corespundeau .
- **Soluție:** Dacă vreau ca arduinoul să nu mai fie conectat la laptop îl alimentez de la un power bank la 5V.
- Tot legat de ADC, inițial intenționam ca ADC să aibă auto-trigger pe timer1 match A, dar am descoperit în datasheet că pentru timer1 nu are event decât la overflow sau la match B.- **Soluție:** În rutina de tratarea a întreruperii timer-ului 1 de match A pornesc "manual" conversia ADC.
- Față de schema electrică, senzorii de nivel al apei sunt acum alimentați direct la 5V și nu la pinii 7 și 8 (erau inițial conectați așa pentru că senzorii se corodează mai repede dacă stau mult alimentați și în contact cu apa, și, inițial, le dădeam drumul doar când voiam să le citesc valoarea și așteptam 10 ms înainte de analogRead ca să nu apară rezultate eronate → problema a fost că în momentul când am trecut la varianta cu timer și ADC cu regiștrii trebuia să aștept acele 10 ms în ISR și cum la laborator am învățat că nu e bine să faci delay în ISR am renunțat la acel delay prin faptul că acum senzorii sunt permanent powered on).

A fost overall o experiență nouă (lipit fire, sertizat cbluri, înghesuit și organizat cabluri în cutie, etc) care ar fi putut fi foarte plăcută dacă nu erau atâtea deadline-uri în perioada aceasta :(

## Jurnal

- 7 mai 2023
  1. creare pagină proiect
  2. completare descriere, introducere, schemă bloc și lista de componente
- 7-14 mai 2023
  1. testare componente
  2. adăugare schema electrică
  3. asamblare versiune inițială a proiectului
- 27-28 mai 2023
  1. îmbunătățire software de testare

2. adăugare folosire timer
  3. ADC cu regiștrii
  4. analogRead cu regiștrii
- 28-30 mai 2023
    1. adăugat și completat documentația la partea de rezultate și concluzii
    2. adăugat mai multe poze

## Bibliografie/Resurse

### Resurse

- [Relay](#)
- <https://www.instructables.com/Arduino-Timer-Interrupts/>
- [https://www.tablesgenerator.com/markdown\\_tables#](https://www.tablesgenerator.com/markdown_tables#)
- <https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/>
- <https://circuitdigest.com/microcontroller-projects/interfacing-water-level-sensor-with-arduino>
- [https://electronoobs.com/eng\\_arduino\\_tut156.php](https://electronoobs.com/eng_arduino_tut156.php)
- <https://www.arnabkumardas.com/arduino-tutorial/adc-register-description/>
- [Datasheet](#)
- [arduino pinout](#)
- <https://www.viralsciencecreativity.com/post/arduino-water-level-indicator-automatic-water-pump-system>
- <https://www.robotique.tech/robotics/control-a-water-pump-by-arduino/>

### Download

old: [flood\\_alarm.zip](#)

most recent: [flood\\_alarm\\_2.zip](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/fstancu/flood-alarm>



Last update: **2023/05/30 06:07**