

# AquaGuardian System

**AquaGuardian System** este un sistem inteligent creat pentru monitorizarea si automatizarea unui acvariu. Ideea proiectului a pornit din dorinta de a realiza un sistem capabil sa mentina conditii stabile si sigure pentru pesti prin monitorizarea permanenta a parametrilor importanti si automatizarea unor procese esentiale.

## Scopul proiectului

Sistemul **monitorizeaza constant parametri importanti ai acvariului**, precum temperatura apei, nivelul apei si lumina ambientala, pentru a mentine un mediu sigur si stabil pentru pesti. Informatiile sunt afisate in **timp real** pe un **LCD conectat prin I2C**, astfel incat utilizatorul poate vedea rapid ce se intampla in acvariu.

Una dintre functiile principale ale proiectului este hranirea automata la ore fixe. Cu ajutorul modulului RTC DS3231, sistemul pastreaza timpul real si declanseaza automat **hranirea la ore prestabilite**, cum ar fi dimineata si seara. In plus, sistemul poate controla si iluminarea acvariului in functie de lumina detectata de senzorul LDR, contribuind la crearea unui mediu mai natural pentru pesti si plante.

Sistemul include si **conectivitate Bluetooth BLE**, permitand monitorizarea parametrilor principali direct de pe telefon. Utilizatorul poate vizualiza informatii precum temperatura apei sau starea iluminarii si poate controla anumite functii ale sistemului prin intermediul unei aplicatii mobile.

Mi s-a parut un proiect interesant deoarece combina mai multe concepte importante din domeniul sistemelor embedded: senzori, intreruperi, ADC, PWM, UART si comunicatie I2C. In plus, consider ca este un proiect util deoarece poate ajuta orice persoana care are un acvariu sa mentina un mediu mai sigur si mai stabil pentru pesti, fara sa verifice constant fiecare parametru manual.

## Descriere generală

Sistemul integreaza mai multe module hardware si software care comunica intre ele pentru a asigura functionarea automata a acvariului si monitorizarea parametrilor importanti.

- **Microcontrollerul ATmega328P** reprezinta unitatea centrala a sistemului si coordoneaza toate procesele. Acesta colecteaza date de la senzori, proceseaza informatiile si controleaza perifericele conectate.
- **Modulul RTC DS3231** este utilizat pentru pastrarea timpului real si pentru programarea actiunilor automate, precum hranirea pestilor sau controlul iluminarii la anumite ore.
- **Senzorul NTC waterproof** monitorizeaza temperatura apei, iar valorile citite sunt afisate pe display.
- Nivelul apei este monitorizat folosind **senzorul ultrasonic HC-SR04**, care masoara distanta dintre senzor si suprafata apei pentru a detecta eventuale scaderi ale nivelului.
- **Senzorul LDR** este utilizat pentru detectarea luminii ambientale, permitand controlul automat al iluminarii acvariului prin intermediul unui releu conectat la banda LED.
- Pentru hranirea automata este utilizat un **servomotor SG90** care actioneaza mecanismul de

distribuire a hranei la ore prestabilite.

- **Display-ul LCD** 16x2 conectat prin I2C afiseaza in timp real informatii importante despre sistem, precum temperatura apei, nivelul apei sau starea iluminarii.

Sistemul include si un **modul Bluetooth BLE HM-10/AT-09** care permite comunicatia cu telefonul mobil. Prin intermediul unei aplicatii mobile, utilizatorul poate monitoriza parametrii principali ai acvariului si poate controla anumite functii ale sistemului.

Comunicatia dintre module este realizata folosind mai multe interfete specifice sistemelor embedded:

- I2C pentru RTC si LCD
- UART pentru comunicatia Bluetooth
- PWM pentru controlul servomotorului



## Hardware Design

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

Componenta	Model	Interfata cu MCU	Functie
Microcontroller	ATmega328P Xplained Mini	-	Unitatea centrala a sistemului
RTC	DS3231	I2C	Pastreaza timpul real si programeaza actiunile automate
Senzor temperatura	NTC waterproof	ADC	Monitorizeaza temperatura apei
Senzor nivel apa	HC-SR04	GPIO	Masoara nivelul apei
Display LCD	LCD 16x2 I2C	I2C	Afiseaza informatii despre sistem
Servomotor	SG90 180°	PWM	Controleaza mecanismul de hranire
Modul releu	Releu 5V 1 canal	GPIO	Controleaza banda LED
Senzor lumina	LDR	ADC	Detecteaza lumina ambientala
Iluminare	Banda LED 12V	Alimentare externa prin releu	Iluminarea acvariului
Modul Bluetooth	HM-10 / AT-09 BLE	UART	Comunicatie cu telefonul
Buton	Push Button	GPIO	Control manual al sistemului

## Descrierea detaliata a modulelor hardware

### RTC DS3231

Modulul RTC DS3231 este utilizat pentru mentinerea timpului real chiar si atunci cand sistemul este oprit. Acesta permite programarea automata a hranirii si a iluminarii la anumite ore prestabilite.

RTC-ul comunica cu microcontrollerul prin interfata I2C:

- **SDA - PC4 (SDA)**
- **SCL - PC5 (SCL)**

Interfata I2C permite comunicatia seriala folosind doar doua fire de date, ceea ce reduce numarul de pini utilizati.

### NTC waterproof

Senzorul NTC waterproof este utilizat pentru monitorizarea temperaturii apei din acvariu. Acesta functioneaza ca o rezistenta variabila dependenta de temperatura si este conectat intr-un divizor rezistiv impreuna cu un rezistor de 10k $\Omega$ .

Conectare:

- **PC1 (A1)** - intrare analogica ADC

Valorile analogice citite de microcontroller sunt convertite in temperatura si afisate pe display-ul LCD.

### HC-SR04

Senzorul ultrasonic HC-SR04 este utilizat pentru monitorizarea nivelului apei.

Conexiuni:

- **TRIG - PD4 (D4)** - iesire digitala
- **ECHO - PD5 (D5)** - intrare digitala

Distanta este calculata pe baza duratei semnalului receptionat.

### LDR

Senzorul LDR detecteaza intensitatea luminii ambientale si permite controlul automat al iluminarii acvariului.

Conectare:

- **PC0 (A0)** - intrare analogica ADC

LDR-ul este utilizat impreuna cu un rezistor de 10k $\Omega$  intr-un divizor de tensiune.

## **Servomotor SG90**

Servomotorul SG90 controleaza mecanismul de hranire automata al pestilor.

Conectare:

- **PB1 (D9)** - pin PWM

PWM-ul permite controlul precis al unghiului de rotatie.

## **LCD 16x2 I2C**

Display-ul LCD afiseaza informatii importante despre sistem, precum temperatura apei, nivelul apei si starea iluminarii.

Conexiuni I2C:

- **SDA - PC4**
- **SCL - PC5**

Utilizarea interfetei I2C reduce numarul de pini necesari conectarii.

## **Modul releu**

Modulul releu controleaza banda LED de iluminare a acvariului.

Conectare:

- **PD7 (D7)** - iesire digitala

Releul este activat automat in functie de valoarea citita de LDR sau manual prin buton.

## **Modul Bluetooth BLE**

Modulul HM-10 / AT-09 permite comunicatia dintre sistem si telefon prin Bluetooth Low Energy.

Conexiuni UART:

- **TXD modul - PB2 (D10)**
- **RXD modul - PB3 (D11)**

Prin Bluetooth, utilizatorul poate monitoriza parametrii sistemului si controla anumite functii folosind o aplicatie mobila.

## Butoane de control

Sistemul utilizeaza doua butoane pentru control manual:

- **PD3 (D3)** - buton hranire manuala
- **PD6 (D6)** - buton control iluminare

Butonul de hranire utilizeaza intreruperi externe pentru detectarea rapida a apasarii.

Componenta	Pin ATmega328P	Tip Interfata
RTC DS3231	PC4 (SDA), PC5 (SCL)	I2C
LCD 16x2 I2C	PC4 (SDA), PC5 (SCL)	I2C
HC-SR04	PD4 (TRIG), PD5 (ECHO)	GPIO
Senzor NTC waterproof	PC1 (ADC1 / A1)	ADC
LDR	PC0 (ADC0 / A0)	ADC
Modul releu	PD7 (D7)	GPIO
Banda LED 12V	Alimentare externa prin releu	Alimentare externa
Servomotor SG90	PB1 (OC1A / D9)	PWM
Buton hranire	PD3 (INT1 / D3)	Intrerupere externa
Buton iluminare	PD6 (D6)	GPIO
Modul Bluetooth BLE HM-10 / AT-09	PB2 (RX / D10), PB3 (TX / D11)	UART

## Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Partea software a proiectului va fi dezvoltata in limbajul C/C++ si va fi completata pe masura realizarii si testarii hardware-ului.

Proiectul integreaza mai multe concepte si functionalitati studiate in cadrul laboratoarelor PM:

### Lab 1 (USART)

Sistemul va transmite in timp real datele de monitorizare, precum temperatura apei, nivelul apei si nivelul luminii ambientale, precum si diferite mesaje de stare sau alerta catre terminalul serial si catre conexiunea Bluetooth.

### **Lab 2 (Intreruperi)**

Va fi utilizata o intrerupere externa hardware pentru detectarea apasarii butonului de hranire manuala. In plus, sistemul va folosi semnalul de alarma generat de modulul RTC DS3231 pentru declansarea automata a hranirii la ore prestabilite.

### **Lab 3 (Timere si PWM)**

Servomotorul SG90 utilizat pentru mecanismul de hranire va fi controlat prin semnal PWM generat de microcontroller. De asemenea, timerele interne vor fi folosite pentru gestionarea anumitor evenimente si actiuni periodice din sistem.

### **Lab 4 (ADC)**

Convertorul Analog-Digital va fi utilizat pentru citirea senzorului LDR si monitorizarea luminii ambientale. Valorile citite vor fi folosite pentru controlul automat al iluminarii acvariului.

Se va realiza monitorizarea permanenta a parametrilor importanti ai acvariului si se va controla automat iluminarea si hranirea in functie de valorile senzorilor si de ora furnizata de RTC. Informatiile importante vor fi afisate pe display-ul LCD si transmise prin Bluetooth catre telefonul utilizatorului.

### **Lab 6 (I2C)**

Protocolul I2C este utilizat pentru comunicatia cu modulul RTC DS3231 si cu display-ul LCD 16x2. RTC-ul furnizeaza timpul real necesar pentru automatizarea hranirii si iluminarii, iar display-ul afiseaza informatii despre starea sistemului si valorile monitorizate.


## **Rezultate Obținute**

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## **Concluzii**

## **Download**

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC

crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

**Saptamana 1:** Am ales ideea proiectului și am stabilit funcționalitățile principale ale sistemului. În această etapă am studiat componentele necesare și modul în care acestea pot comunica între ele.

**Saptamana 2:** Am realizat planul proiectului, am ales modulele hardware necesare și am comandat componentele pentru realizarea sistemului.

**Saptamana 3:** Am realizat schema bloc a sistemului și am început redactarea documentației. De asemenea, am studiat conexiunile și protocoalele de comunicație utilizate de fiecare modul.

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/alexia.oprisan> 

Last update: **2026/05/16 08:18**