

Laser Turret

Introducere

- Proiectul consta dintr-o “tureta” care se misca, atat vertical, cat si orizontal, aceasta putand fi controlata prin intermediul mouse-ului de la laptopul/calculatorul la care este conectata placa. De asemenea, tureta are si un laser in varful acesteia, cu care utilizatorul poate tinti obiectele din jurul sau.
- Scopul acestui proiect este de pur divertisment, cu care utilizatorii pot juca diferite “minigame”-uri ce au legatura cu tintirea obiectelor (preferabil nu a oamenilor!!).
- Ideea a pornit din doua surse: turetele din jocurile de tip MOBA/Tower Defense, care trag cu un proiectil la un anumit interval de timp, iar a doua este un laser obisnuit, de buzunar, folosit pentru a te juca impreuna cu animalele de companie (lucru care as vrea sa-l faci si eu cu pisica mea :))).
- Utilitatea proiectului este, in principal, educationala. Faptul ca poti controla un “robot” fizic folosindu-te de controalele de la calculator, alegerea componentelor potrivite, calcularea si testarea consumului, cat si altele sunt lucruri cu care trebuie sa experimenteze orice inginer. :)

Descriere generală

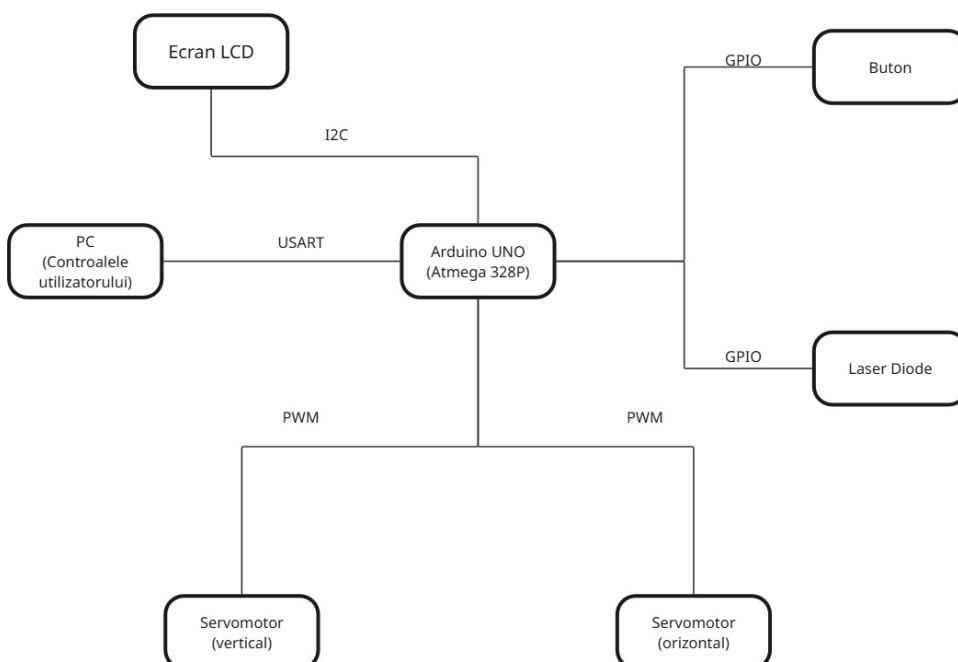


Diagrama Hardware

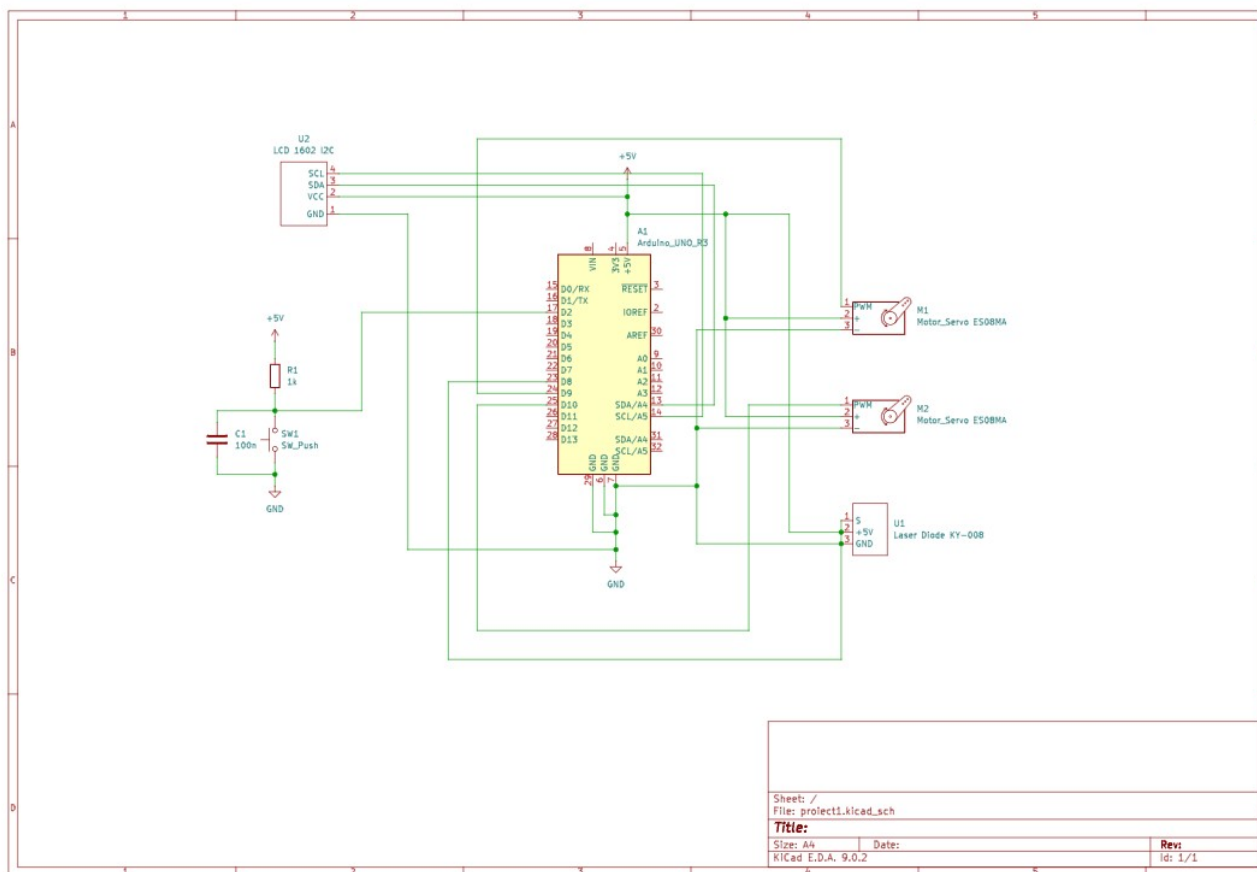
Hardware Design

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

Nume componentă	Link	Cantitate	Preț unitar (lei)	Preț final (lei)
Placuta compatibila Arduino Uno R3	Link	1	49.99	49.99
Breadboard HQ (400 Points)	Link	1	4.56	4.56
Ecran LCD I2C 1602	Link	1	16.34	16.34
Servo SG90	Link	1	11.99	11.99
Servo ES08MA II	Link	1	28.39	28.39
Baterii 1.5V	Link	4	3.20	12.80
Suport baterii (4xAA)	Link	1	4.86	4.86
Condensator 1000 μ F	Link	2	0.59	1.18
Modul Laser KY-008	Link	1	4.28	4.28
Fire Male to Male	Link	15	0.28	4.20
Fire Female to Male	Link	9	0.44	3.99
Preț total:			142.55 lei	

Schema electrica:



Utilizare PINI :

- **Arduino Uno R3:**
 - **D2** - pin digital conectat la dioda laser KY-008
 - **D9** - pin PWM conectat la servo motor SG90 (M1)
 - **D10** - pin PWM conectat la servo motor ES08MA II (M2)
 - **A4 (SDA)** - pin I2C conectat la LCD 1602
 - **A5 (SCL)** - pin I2C conectat la LCD 1602
 - **5V, GND** - pentru alimentarea componentelor
- **Servo SG90 (M1):**
 - **PWM** - conectat la **D9**
 - **VCC, GND** - conectate la 5V și GND
- **Servo ES08MA II (M2):**
 - **PWM** - conectat la **D10**
 - **VCC, GND** - conectate la 5V și GND
- **Laser KY-008:**
 - **S (semnal)** - conectat la **D2**
 - **VCC** - conectat la 5V
 - **GND** - conectat la GND
- **LCD 1602 I2C:**
 - **SDA** - conectat la **A4**
 - **SCL** - conectat la **A5**
 - **VCC** - conectat la 5V
 - **GND** - conectat la GND

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 plănuieți să le implementați
- (etapa 3) surse și funcții implementate

Din moment ce PC-ul trebuie să comunice în permanență cu placa Arduino pentru a putea trimite comenzile, trebuie să existe cod care rulează pe ambele dispozitive.

PC:

- O mică aplicație scrisă în Java care deschide o fereastră în care utilizatorul poate mișca mouse-ul pentru a controla tureta
- Partea de GUI realizată folosind Java Swing
- Trimiterea coordonatelor Mouse-ului și verificarea dacă utilizatorul a apăsat Click se realizează folosind biblioteca externă SerialComm.

Placuta:

- Scris în Arduino C/C++, astfel încât să poată să "handle-uisca" input-ul de la utilizator transmis prin Serial și să facă modificările necesare în Hardware, folosind bibliotecile LiquidCrystal_I2C, Wire și Servo.
- După ce primește input-ul de la utilizator (coordonate/acțiunea ca a dat click), se măpăază coordonatele la sistemul de referință al servo-urilor (unghiuri de la 0 la 180) și "mută" servo-urile.
- Dacă primește "click", verifică dacă a trecut cooldown-ul de la ultima dată când utilizatorul a activat laserul (folosind Timer-ul 2 al plăcii), setează PIN-ul pe HIGH, în cazul în care cooldown-ul a trecut
- Resetează Timer-ul în momentul în care utilizatorul nu mai ține click-ul apăsat
- Scrie pe ecranul LCD dacă utilizatorul poate activa laser-ul / cât timp mai are până când poate face asta.

Notiuni din laboratoare folosite:

- I2C (folosind biblioteca LiquidCrystal_I2C)
- Timer (implementat cu registrii)
- PWM (biblioteca Servo)
- GPIO (implementat cu registrii)
- USART (implementat cu registrii)
- Întreruperi

Funcții:

- `setup_timer()` - Configurează Timer2 pentru a genera o întrerupere la fiecare 1 ms.
- `ISR(TIMER2_COMPA_vect)` - Incrementează contorul timerMs la fiecare 1 ms pentru a măsura timpul.

- `setup()` – Initializeaza servo-urile, laserul, LCD-ul, comunicatia seriala si porneste Timer2.
- `parseDataX(data)` – Extrage si converteste valoarea X din sirul primit prin serial.
- `parseDataY(data)` – Extrage si converteste valoarea Y din sirul primit prin serial.
- `loop()` – Citeste datele seriale, controleaza servo-urile si laserul in functie de click si cooldown, si actualizeaza afisajul LCD.
- `usart_init(baud)` – Configureaza manual registrii USART pentru comunicare seriala la baud rate-ul specificat (RX/TX activate, format 8N1).
- `usart_send_char©` – Trimite un singur caracter prin portul serial folosind registrul UDR0, asteptand ca bufferul de transmisie sa fie gol.
- `usart_send_string(str)` – Trimite un sir de caractere prin portul serial caracter cu caracter, apeland `usart_send_char()`.
- `usart_receive_char()` – Primeste un caracter de pe portul serial, blocand executia pana cand unul este disponibil in registrul UDR0.
- `usart_read_line(buffer, maxLen)` – Citeste caractere de pe portul serial intr-un buffer pana la caracterul de newline (`\n` sau `\r`), construind astfel o linie completa de text.

IDE-uri folosite:

- Visual Studio Code pentru partea de Java
- Arduino IDE pentru codul de pe placuta

Rezultate Obținute


In urma implementarii, am reusit sa obtin o “tureta” care se poate misca pe ambele axe (verticala si orizontala) folosind mouse-ul de la laptop cu performante bune, cu un delay si o acuratete foarte bune. De asemenea, ecranul afiseaza precis cooldown-ul abilitatii de a trage, acesta fiind actualizat o data pe secunda, fiind mentionat atunci cand utilizatorul poate sa “traga”.

[Demo1](#)


[Demo2](#)

Concluzii

In urma realizarii acestui proiect, mi-am consolidat multe din cunostintele acumulate la laboratorul si cursul de PM, in special comunicarea seriala si controlarea motoarelor prin PWM, reusind sa obtin un proiectel interesant si distractiv.

Pe de partea cea mai grea parte din realizarea proiectului a fost cea pe care o consideram cea mai usoara inainte sa ma apuc de proiect, si anume comunicarea seriala. Am intampinat multe probleme la aceasta, cea mai mare fiind faptul ca trimitrea datelor prin serial de la laptop la arduino ducea la umplerea buffer-ului de trimitere a datelor (ceva ce nu stiam ca se poate intampla ) , manifestarea acestui fenomen nefiind intocmai evidenta). De asemenea, am avut probleme legate de setup-ul USART-ului (felul in care setasem formatul frame-ului era diferit de felul in care era configurat frame-ul by default in biblioteca `commSerial` din java).

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.

Resurse

[Repository](#)

Bibliografie

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

[Datasheet ATmega328P](#) [Tutorial Servo](#) [Tutorial LCD](#) [Biblioteca I2C](#) [Biblioteca Servo](#)

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2025/eradu/andrei.chirimeaca> 

Last update: **2025/05/30 10:22**

