Shooting range

Introducere

Proiectul propus constă într-un sistem embedded autonom capabil să detecteze poziția unui punct laser proiectat pe un fototranzistor și să orienteze automat un pistol Nerf pe axa orizontală către zona respectivă, declanșând ulterior trăgaciul.

Sistemul utilizează o matrice de senzori optici (fotorezistori cu comparatoare LM393) pentru detectarea precisă a luminii laser, un servomotor de putere (MG996R) pentru orientarea stânga-dreapta a platformei și un servomotor secundar (SG90) dedicat activarii trăgaciului. Sistemul este alimentat printr-un sistem de baterii litiu-ion.

Scopul proiectului este realizarea unei platforme embedded interactive care îmbină detecția optică, controlul mecanic și procesarea în timp real pe microcontrolerul ATmega328P.

Descriere generală

Sistemul este compus din mai multe module hardware interconectate, fiecare avand un rol in functionarea pistolului automat. Detectia tintei este realizata prin intermediul unui vector liniar 1×3 de senzori optici montati pe un plan orizontal. Fiecare senzor reprezinta o zona tinta distincta, iar detectia se bazeaza pe identificarea unei surse de lumina directionate catre unul dintre senzori. In momentul in care un senzor detecteaza lumina, microcontrollerul ATmega328P identifica pozitia acesteia si determina directia corespunzatoare pentru orientarea platformei.

Orientarea pistolului Nerf este realizata utilizand un mecanism format din un servomotor, ce realizeaza miscarea orizontala (stanga/dreapta) Tragaciul pistolului este actionat separat de un servomotor SG90 datorita cuplului ridicat necesar pentru apasarea mecanica a mecanismului de tragere.

Sistemul este controlat de placa de dezvoltare bazata pe microcontrollerul ATmega328P Xplained Mini, care gestioneaza citirea senzorilor, logica de decizie si controlul actuatoarelor. Alimentarea este realizata independent prin baterii Li-Ion 18650 si un regulator LM2596 pentru stabilizarea tensiunii necesare componentelor electronice si servomotoarelor.

Hardware Design

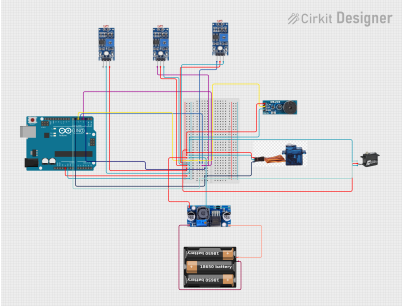
Componentele principale utilizate in proiect sunt:

- ATmega328P

- 3x senzori optici fototranzistori LM393
- 1x servomotor SG90
- 1x servomotor MG996R
- LM2596 Voltage Regulator
- baterii 18650 Litiu-Ion
- breadboard si componente pasive

Schema bloc:

Schema electrică:



Software Design

Procesul de scriere și compilare a firmware-ului utilizează mediul de dezvoltare Arduino IDE integrat cu toolchain-ul AVR-GCC, o alegere care permite combinarea flexibilă a funcțiilor de nivel înalt cu programarea bare-metal direct pe regiștri. Legarea componentelor software este realizată prin intermediul compilatorului avr-g++, asigurându-se astfel o optimizare riguroasă a codului pentru microcontrolerul ATmega328P și un management eficient al memoriei Flash în timpul execuției.

Pentru generarea semnalelor de control destinate servomotoarelor fără a perturba celelalte funcții software, s-a implementat un algoritm de Software PWM generat în fundal prin intermediul unei Rutine de Serviciu a Întreruperilor (ISR). Folosind Timer1 configurat în modul CTC (Clear Timer on Compare Match) cu declanșare la fiecare 0.1 milisecunde, un contor global numără tick-urile până la atingerea perioadei standard de 20 de milisecunde (50 Hz), moment în care pinii asociați servomotoarelor sunt trecuți în starea HIGH. Oprirea impulsului (starea LOW) se face independent pentru motorul de orientare a platformei și cel al trăgaciului prin compararea în timp real a contorului cu valorile țintă modificate dinamic de program.

În interiorul buclei principale, sistemul execută un algoritm de detectare a intensității luminoase prin interogarea succesivă a canalelor 0, 1 și 2 ale convertorului analog-digital (ADC). Filtrarea semnalelor primite de la fototranzistorii se bazează pe compararea valorilor brute cu un prag static predefinit THRESHOLD, interpretând orice valoare mai mică de 190 ca fiind o detectare validă a punctului laser (datorită conexiunii hardware în care senzorul trage pinul spre masă la iluminare). Arbitrarea deciziei în cazul în care mai mulți senzori sunt activați simultan se face printr-o structură condițională în cască, oferind prioritate fizică senzorilor de la stânga la dreapta și prevenind astfel comportamentele mecanice divergente sau oscilațiile nedorite ale platformei.

Structura funcțională a codului este divizată în subprograme specializate, începând cu timer1_init_interrupt care configurează ceasul de fundal în modul CTC și execute_target_sequence care rulează mașina de stări blocantă responsabilă de emiterea avertismentelor sonore, mișcarea precisă a platformei pe trei direcții prestabilite, acționarea trăgaciului și revenirea automată în starea neutră. Funcția standard setup se ocupă de inițializarea shell-ului interactiv și configurarea direcției regiștrilor DDRD (ieșiri) și DDRC (intrări), în timp ce loop menține active sarcinile din background ale interfeței seriale. Punctul central de execuție este funcția main, care după activarea întreruperilor globale prin instrucțiunea sei și pornirea modulului ADC hardware prin adc_init, guvernează bucla infinită în care se procesează în timp real datele senzorilor și se trimit periodic mesaje de diagnostic (debug) prin Serial Monitor la fiecare 3 secunde folosind funcții non-blocante.

Laboratoare utilizate în implementare

- Laboratorul 1 - I/O Digital
- Laboratorul 2 - Întreruperi externe & Timere
- Laboratorul 3 - Întreruperi
- Laboratorul 4 - USART
- Laboratorul 5 - ADC

Rezultate Obţinute

* Tutorial Shooting Range

Concluzii

In urma realizarii proiectului, am aprofundat conceptele din laboratoarele folosite, precum am si realizat un sistem functional de targetting folosind fototranzistori.

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună .

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Bibliografie/Resurse

Github: https://github.com/DavidStancu/PM-ShootingRange

* ATmega328P Xplained Mini User Guide

Export to PDF

Tutoriale youtube:

* Pan-Tilt bracket assembly tutorial - YouTube