Tracker Turret

Introducere

Tracker Turret este un sistem embedded capabil să detecteze și să urmărească automat o persoană folosind procesare video în timp real și control hardware al unei turete motorizate. Proiectul combină algoritmi de computer vision, comunicație serială și control hardware low-level într-o platformă integrată.

Ideea proiectului a pornit de la dorința de a crea un sistem care îmbină procesarea video cu controlul mecanic în timp real. Sistemul utilizează camere video conectate la o placă Avnet MaaXBoard pentru detectarea și urmărirea unei persoane folosind OpenCV. Datele rezultate sunt transmise către un microcontroler Arduino Nano ESP32 care controlează mișcarea servo motoarelor și afișarea informațiilor pe un display OLED.

Scopul principal al proiectului este realizarea unei platforme autonome capabile să:

* detecteze fețe în timp real * urmărească poziția unei persoane * controleze mecanic orientarea turetei * afișeze starea sistemului pe display * comunice între module hardware diferite

Proiectul poate avea aplicații în:

* robotică * sisteme autonome * supraveghere inteligentă * automatizare * cercetare în computer vision și embedded systems

Elementul de noutate al proiectului constă în integrarea unui sistem Linux embedded capabil de procesare video avansată cu un microcontroler dedicat controlului hardware în timp real, utilizând mai multe protocoale hardware și tehnici low-level studiate în laborator.

Descriere generală

Sistemul este împărțit în două componente principale:

* modulul de procesare video * modulul de control hardware

==== Modulul de procesare video

Procesarea video este realizată pe placa Avnet MaaXBoard i.MX8M care rulează Linux embedded. Pe aceasta rulează un program Python bazat pe OpenCV responsabil pentru:

* capturarea imaginilor de la cameră * detectarea fețelor folosind clasificatori Haar Cascade * estimarea poziției țintei în imagine * generarea comenzilor de control

În funcție de poziția feței detectate în cadru:

* dacă fața este în stânga → se transmite comanda de rotire stânga * dacă fața este în dreapta → se transmite comanda de rotire dreapta * dacă fața este centrată → tureta se oprește * dacă nu există față detectată → sistemul intră în modul SEARCHING

Comenzile sunt transmise prin UART către microcontroler.

==== Modulul de control hardware

Controlul hardware este realizat folosind Arduino Nano ESP32 programat folosind ESP-IDF și drivere hardware low-level.

Microcontrollerul este responsabil pentru:

* controlul servo motoarelor folosind PWM hardware * controlul display-ului OLED prin I2C * recepționarea comenzilor prin UART * actualizarea stării sistemului

Comenzile primite de la MaaXBoard sunt:

Comandă	Funcție
C0	Oprire / țintă centrată
C1	Rotire stânga
C2	Rotire dreapta
C3	Mod SEARCHING

În funcție de comanda primită:

* servo motorul este mișcat * display-ul OLED este actualizat * starea sistemului este modificată

Display-ul OLED afișează:

* DETECTED → atunci când există o față detectată * SEARCHING → atunci când nu există țintă detectată

Schema bloc

* Cameră USB → MaaXBoard i.MX8M * OpenCV → Detectare față și tracking * Algoritm decizie → Determinare direcție țintă * UART → Transmitere comenzi * Arduino Nano ESP32 → Control hardware * PWM → Control servo motor * I2C → Control display OLED * OLED SSD1306 → Afișare stare sistem

Arhitectura sistemului

==== Fluxul complet de funcționare

1. Camera transmite imagini către MaaXBoard. 2. OpenCV procesează imaginea și detectează fața. 3. Sistemul determină poziția feței în cadru. 4. Se generează o comandă UART. 5. Comanda este transmisă către Arduino Nano ESP32. 6. ESP32 actualizează poziția servo motorului. 7. ESP32

actualizează mesajul de pe display-ul OLED.

Acest proces rulează continuu în timp real.

Implementare hardware

==== Stadiul actual al implementării hardware

În momentul actual au fost implementate și validate:

* comunicația UART între MaaXBoard și ESP32 * controlul servo motorului folosind PWM * comunicația I2C cu display-ul OLED SSD1306 * detectarea fețelor folosind OpenCV * transmiterea comenzilor de tracking în timp real

Sistemul funcționează complet end-to-end:

* detectare față * transmitere comandă * mișcare servo * actualizare display

Componente hardware

Componentă	Rol în proiect
Cameră USB	Captură imagine pentru tracking
Avnet MaaXBoard i.MX8M	Procesare video și rulare OpenCV
Arduino Nano ESP32	Control hardware și comunicare
Servo motor	Mișcarea turetei
Display OLED SSD1306 I2C	Afișarea stării sistemului
Breadboard	Realizarea conexiunilor
Fire Dupont	Conexiuni electrice
Alimentare USB	Alimentarea sistemului
Convertor UART	Debugging și testare comunicație

Conexiuni hardware

==== UART MaaXBoard → ESP32

MaaXBoard	ESP32	Funcție
UART2_TX	RX0	Transmitere comenzi
GND	GND	Referință comună

Protocolul UART utilizează:

* baudrate: 9600 * comenzi text de forma C0/C1/C2/C3

==== Servo motor

Servo	ESP32
Signal	GPIO9
VCC	Alimentare externă
GND	GND comun

Servo motorul este controlat folosind PWM hardware la 50Hz.

==== Display OLED SSD1306

OLED	ESP32
SDA	A4
SCK/SCL	A5
VCC	3.3V
GND	GND

Display-ul utilizează protocolul I2C și a fost detectat la adresa 0x3C.

Functionalități utilizate din laborator

==== PWM

PWM-ul hardware este utilizat pentru controlul servo motorului.

Avantaje:

* control precis al poziției * funcționare fluidă * utilizare eficientă a hardware-ului

==== UART

UART este utilizat pentru comunicația dintre MaaXBoard și ESP32.

Avantaje:

* implementare simplă * comunicație robustă * latență redusă

==== I2C

I2C este utilizat pentru controlul display-ului OLED SSD1306.

Avantaje:

* număr redus de fire * integrare simplă * compatibilitate ridicată

==== Timere hardware

Timerele hardware sunt utilizate pentru:

* generarea semnalului PWM * actualizarea periodică a servo motorului * sincronizarea task-urilor FreeRTOS

Implementare software

==== Stadiul actual al implementării software

Au fost implementate:

* captură video OpenCV * detectare fețe Haar Cascade * tracking în timp real * generare comenzi UART * parser UART pe ESP32 * control servo PWM * control OLED SSD1306 * task-uri FreeRTOS pentru paralelizare

Biblioteci utilizate

Bibliotecă	Rol
OpenCV	Detectare și procesare imagine
pyserial	Comunicație UART pe Linux
ESP-IDF	Programare low-level ESP32
driver/uart.h	Control UART hardware
driver/ledc.h	PWM hardware
driver/i2c.h	Comunicație I2C
FreeRTOS	Task scheduling

Motivarea alegerii tehnologiilor

==== OpenCV

OpenCV oferă:

* algoritmi optimizați * detectare rapidă * compatibilitate Linux embedded * integrare simplă cu Python

==== ESP-IDF

ESP-IDF permite:

* acces direct la periferice hardware * control low-level * performanță ridicată * utilizarea driverelor hardware oficiale

==== UART

UART a fost ales deoarece:

* este simplu de implementat * este robust * necesită puține resurse hardware * oferă latență mică

Optimizări realizate

Pentru creșterea performanței au fost realizate următoarele optimizări:

* reducerea rezoluției imaginii pentru creșterea FPS-ului * transmiterea comenzilor sub formă compactă * utilizarea task-urilor separate pentru UART și servo * utilizarea PWM hardware în loc de software PWM * utilizarea comunicației I2C hardware * reducerea latenței UART prin scăderea baudrate-ului și stabilizarea comunicației

Validare și testare

Sistemul a fost validat prin:

* testarea comunicației UART * testarea PWM pentru servo * scanarea magistralei I2C * detectarea display-ului la adresa 0x3C * validarea tracking-ului facial * testarea mișcării turetei în timp real

Au fost validate:

* funcționarea comunicației dintre module * afișarea mesajelor pe OLED * reacția servo motorului la poziția feței * funcționarea în timp real a sistemului

Calibrare

Calibrarea sistemului a fost realizată prin:

* ajustarea dead-zone-ului pentru tracking * reglarea vitezei servo motorului * ajustarea parametrilor Haar Cascade * configurarea rezoluției optime pentru cameră * ajustarea frecvenței comenzilor UART

Scopul calibrării a fost obținerea unei mișcări stabile și reducerea oscilațiilor sistemului.

Structura software

Software-ul este împărțit în:

==== MaaXBoard Linux

* captură video * detectare fețe * logică tracking * generare comenzi UART

==== ESP32

* task UART * task servo * control OLED * PWM hardware * I2C hardware

Componentele comunică folosind un protocol UART simplu bazat pe comenzi text.

Demonstrarea funcționalității

Pentru demonstrarea proiectului vor fi prezentate:

* detectarea unei persoane în timp real * rotirea turetei după poziția țintei * afișarea stării pe display-ul OLED * transmiterea comenzilor UART * funcționarea sistemului embedded complet

Vor fi atașate:

* imagini cu conexiunile hardware * capturi din timpul funcționării * demonstrație video a proiectului

Concluzii

Tracker Turret demonstrează integrarea cu succes a:

* computer vision * embedded Linux * control hardware low-level * comunicații seriale * protocoale hardware * sisteme real-time

Proiectul reprezintă o combinație practică între software și hardware și evidențiază utilizarea conceptelor studiate în laborator într-un sistem complet funcțional.