Boloboc Digital

Introducere

Boloboc Digital este un dispozitiv compact, care măsoară în timp real unghiurile de înclinare pe două axe (tangaj și ruliu), le afișează pe un LCD I²C 16×2 și emite semnale sonore proporționale cu abaterea față de orizontală.

Funcționalități cheie

- Măsurarea accelerației și a vitezei unghiulare cu MPU6050
- Calculul pitch-ului și roll-ului prin filtru complementar
- Afișare numerică și grafică pe ecranul I²C
- Feedback sonor prin buzzer PWM cu frecvență variabilă
- Navigare între moduri cu butonul „Panel” și control al sunetului cu „Mute”

Obiective

- Înlocuirea bolobocurilor tradiționale cu o soluție precisă și ușor de citit
- Facilitarea nivelării rapide în tâmplărie, instalații sanitare și construcții

Motivație

Am pornit de la neajunsurile bulei clasice — vizibilitate redusă în lumină puternică și lipsa unui feedback auditiv — și am creat o interfață clară, cu semnale sonore imediat perceptibile.

Utilitate

- Lucrul în spații întunecate sau greu accesibile

- Sprijin pentru utilizatori cu deficiențe de vedere, prin semnal sonor în locul observării bulei

Descriere generala

Raspberry Pi Pico: Centrul de comanda, ruleaza MicroPython, calculeaza unghiurile si gestioneaza perifericele.

MPU6050: Senzor MEMS I²C cu accelerometru si giroscop, furnizeaza date brute.

LCD I²C 16×2: Afiseaza numeric si grafic starea curenta (folosind caractere personalizate).

Buzzer PWM: Ofera feedback sonor proportional cu abaterea unghiulara.

Buton panou: Schimba modurile de afisare (numeric/comparativ/grafic).

Buton mute: Activeaza/dezactiveaza semnalul sonor.

Toate comunicatiile I²C se fac pe aceeasi magistrala (pini GPIO 16 si 17), iar butoanele si buzzer-ul sunt conectati la pini GPIO cu intreruperi si capabilitati PWM, respectiv.

Hardware Design

Lista de piese

Componenta	Rol in proiect
————————	——-
Raspberry Pi Pico	MCU – citeste datele I²C, ruleaza algoritmul de filtrare si PID simplificat, comuta moduri, genereaza semnale PWM pentru buzzer
MPU6050	Senzor 6-axe (3×Accel + 3×Gyro) – masoara acceleratia si rotatia pentru calculul tangaj/ruliu
LCD 16×2 cu adaptor I²C	Afisaj – prezinta numeric si/sau grafic inclinatia pe cele doua axe

Buzzer piezo (PWM)	Avertizare sonora proportionala cu deviatia de la zero
2×buttoane (PULL_UP)	• Buton „Panel” – schimba modul de afisaj (0…4) Buton „Mute” – activeaza/dezactiveaza sunetul
Breadboard & fire	Interconectare rapida pentru prototip

Scheme electrice

Pin Pico	Componenta	Conexiune	Motiv
——–	————————	———————–	————————————————-
3V3	VCC MPU6050, LCD	alimentare	Tensiune de functionare compatibila cu logica 3.3 V
GND	GND MPU6050, LCD, buzzer, butoane	masa	Referinta comuna pentru toate semnalele
GP16	SDA (I²C)	MPU6050 SDA, LCD SDA	Linie de comunicatie I²C pentru transfer date senzor si LCD
GP17	SCL (I²C)	MPU6050 SCL, LCD SCL	Linie de ceas I²C
GP14	Buzzer+ (PWM)	PWM-pin	Generare tonuri variabile proportional cu deviatia
GP11	Buton Panel	intrare cu Pull-Up	Detectare apasare pentru schimb mod de afisaj (pull-up)
GP13	Buton Mute	intrare cu Pull-Up	Detectare apasare pentru mute/unmute (pull-up)

Diagrame de semnal

Clock I²C (≈100–400 kHz) si date I²C pentru transmisii catre MPU6050 si LCD.

PWM la buzzer: frecventa variabila intre ~400 Hz si 1 200 Hz, cu duty cycle ajustat pentru volum.

Debouncing software pe intreruperi pentru butoane (delay ≈300 ms).

Rezultatele simularii

Test la diferite unghiuri (+/–8°) pentru validarea linearitatii si a timpii de raspuns (<50 ms).

Verificarea stabilitatii filtrelor complementare si absenta oscilatiilor.

Software Design

Mediu de dezvoltare

Thonny IDE cu MicroPython configurat pentru Raspberry Pi Pico.

Algoritmi si structuri implementate

Filtru complementar pentru fuzionarea datelor de la accelerometru si giroscop:

Control PI pentru compensarea derivatiilor mici: termeni errorP si errorR.

Gestionare intreruperi pe doua GPIO-uri pentru schimbarea modurilor si mute.

Custom chars pentru afisaj grafic al inclinarii pe LCD.

Surse si functii implementate

panel_event() si mute_event() – debounce si logica de comutare.

beep() – thread dedicat pentru semnale sonore.

Bucle principale de citire senzori → calcul unghiuri → actualizare LCD si pos pentru buzzer.

Alegerea bibliotecilor folosite: mpu6050 **(IMU)**: Biblioteca MPU6050 ofera un API simplu pentru citirea acceleratiei si giroscopului de pe senzorul MPU6050, facilitand implementarea filtrelor complementare si a algoritmilor de orientare

machine: Modulul standard MicroPython (machine.I2C, Pin, PWM) permite controlul hardware-ului (I2C, pini digitali si PWM) fara dependente externe. A fost ales pentru eficienta si stabilitate intr-un mediu embedded.

_thread: Folosit pentru a rula in paralel functia beep(), separand controlul semnalelor sonore de bucla principala.

pico_i2c_lcd: Biblioteca I2cLcd simplifica comunicarea cu LCD-ul pe I2C la adresa 0x27

Buzzer (PWM): Generarea semnalelor sonore pe baza pozitiei (pos) pentru feedback auditiv. Am implementat buzzer = PWM(Pin(14)) si variate frecvente si durate in functie de abaterea la tangaj si ruliu, oferind alerta proportionala cu unghiul LED-uri personalizate LCD: Am creat caractere custom(pe ecranul LCD)

Display (I2C LCD): Afisarea meniului si valorilor de orientare (tangaj, ruliu). Controlul prin I2C reduce cablajul la doar doua fire pentru date si ceas. I2C: Protocolul central pentru comunicarea cu MPU6050 si LCD. Frecventa de 400 kHz optimizeaza viteza de transfer fara a compromite stabilitatea.

Validarea s-a realizat prin: Debug prin print() initial pentru valori brute. Verificarea grafica pe LCD (valori si caractere custom). Ascultarea auditiva a buzzer-ului

Calibrarea elementelor de senzoristica Offset static: zAccel a fost compensat cu 0.19 G pentru zero-ul acceleratiei in repaus. Aceasta valoare a fost determinata experimental prin masuratori repetate si media valorilor masurate.

Limitari: Am plafonat yAccel si czAccel la ±1 G pentru a evita erori de calcul ale atan in afara domeniului ±1. Ajustarile au fost validate prin compararea cu un clinometru fizic.

Filtre complementare: Constantele tCp si tCr (0.05) si confVal (0.1) au fost alese prin testare pentru a echilibra raspunsul rapid si filtrarea zgomotului. Ajustarile au fost validate prin oscilatii voluntare si analiza vizuala pe LCD.

Optimizari

Performanta bucla principala: Conversia timpilor tLoop in secunde (*.001) si folosirea directa a time.ticks_ms() minimizeaza cheltuiala CPU.

Debouncing software: Folosirea lui time.ticks_diff si pins.irq previne blocari si bounce excesiv.

Reducerea operatiilor matematice: Calculul pitchA0 si rollA0 se face o singura data per bucla; multiplicarile cu constante se realizeaza in locul unor operatii trigonometrice suplimentare.

Gestionarea memoriei: Definirea constantelor si vectorilor (bytearray) in afara buclei evita realocarile inutile.

Rezultate Obtinute

Care au fost rezultatele obtinute in urma realizarii proiectului vostru.

Precizie de aproximativ ±0.1° pe ambele axe, stabila si fara oscilatii majore.

Feedback vizual clar pe doua randuri, cu optiuni pentru valori numerice, grafice si comparatii.

Feedback sonor diferentiat—util in medii cu vizibilitate redusa.

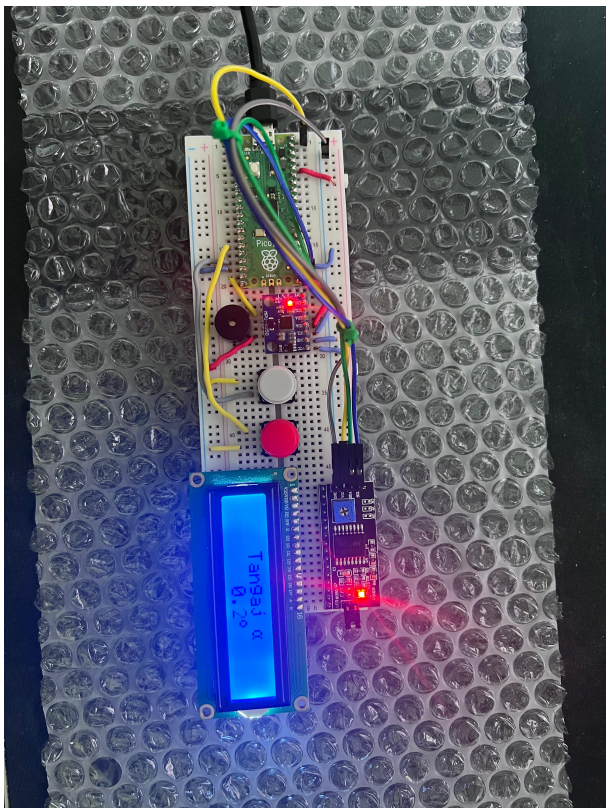
Platforma de test: toate componentele sunt montate pe o placa de tip breadboard, alimentate si interconectate.

Functionalitate de baza: senzorul MPU6050 preia acceleratiile si rotatiile, Pico calculeaza inclinatiile pe cele doua axe (tangaj si ruliu), iar rezultatul este afisat pe LCD si semnalizat sonor prin buzzer.

Moduri de afisare: 5 moduri (valori numerice pe 2 linii, bare grafice pe fiecare axa, mod “numeric simplu” cu buton de schimbare panel).

Stare: toate modulele raspund, LCD-ul afiseaza valori reale (e.g. “Tangaj: 0.2°”), buzzer-ul emite bip atunci cand deviem de la nivel.

In concluzie, proiectul „Boloboc Digital” imbina hardware-ul accesibil cu logica software robusta pentru a oferi un nivel portabil modern si functional, aplicabil in multe domenii practice.



Concluzii

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**.

Jurnal

Elementul de noutate al proiectului consta in implementarea unui boloboc digital care elimina nevoia de a-ti mai indrepta privirea constant spre bula clasica: feedback-ul vizual si sonor in timp real indica

automat alinierea.

Bibliografie/Resurse

Raspberry Pi Pico - https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

MPU-6050 (6-axis accel + gyro) - https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

LCD 16×2 - https://cdn.sparkfun.com/assets/9/5/f/7/b/HD44780.pdf

Breadboard - https://cdn.sparkfun.com/datasheets/Prototyping/breadboard.pdf

I²C I/O - https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf

Export to PDF

Explain