

Teodora CHIRIAC (78367) - Zaruri

Autorul poate fi contactat la adresa: **Login pentru adresa**

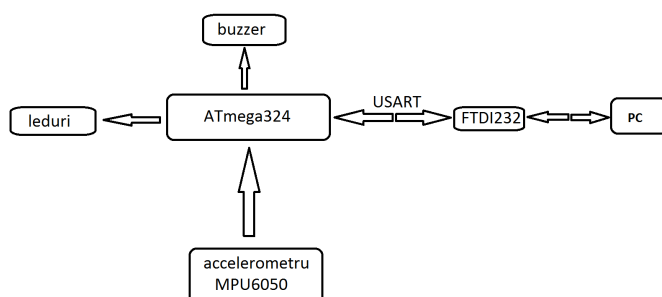
Acest proiect consta in simularea unui zar.

Introducere

Scopul proiectului este simularea unui zar.

La miscarea placutei se va genera un numer intre 1 si 6 care reprezinta zarul si va fi auzit un zgomot specific aruncarii zarurilor. Proiectul este creat pentru a ajuta utilizatorul sa joace jocuri care necesita aruncarea cu zarul oriunde si sa economiseasca timp.

Descriere generală

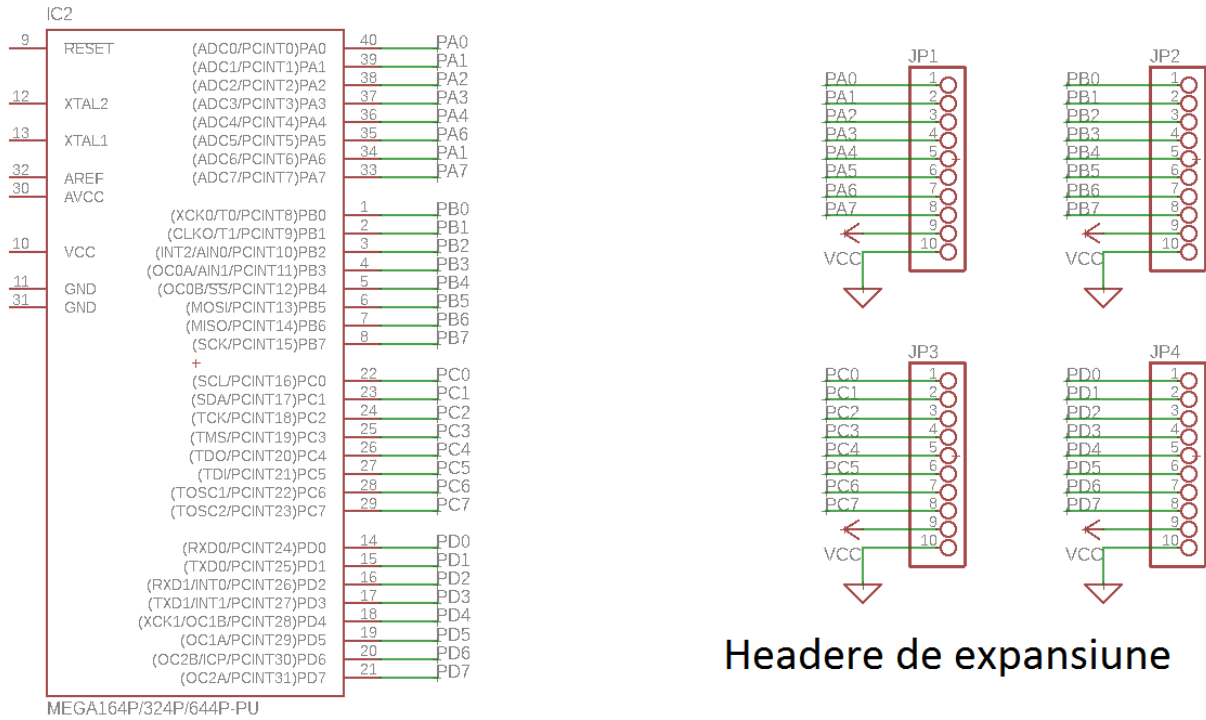


Hardware Design

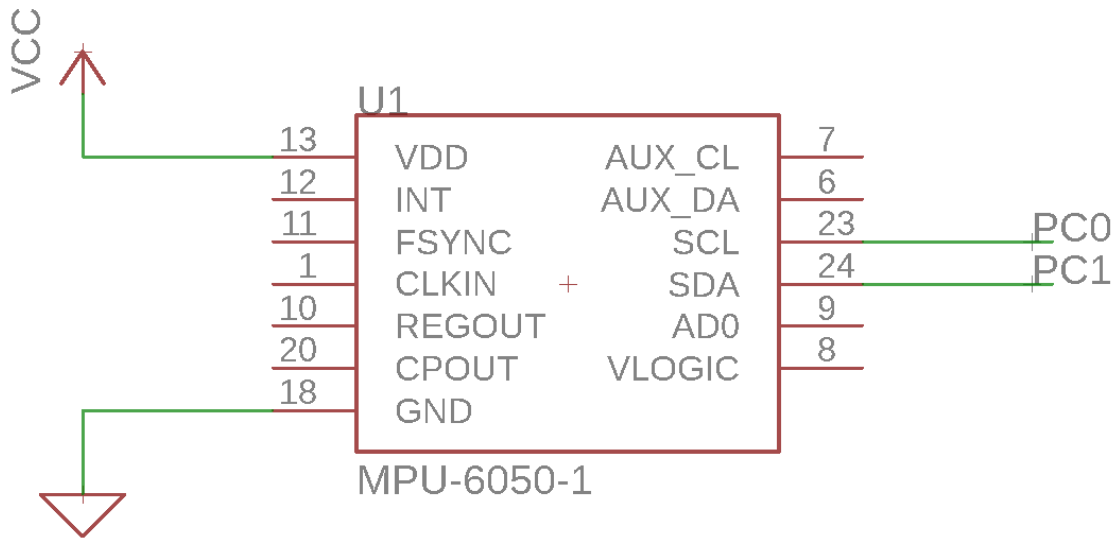
Lista piese:

- Modul accelerometru si giroscop MPU6050 GY
- Modul FTDI232
- leduri
- buzzer
- fire conectoare

Scheme electrice:

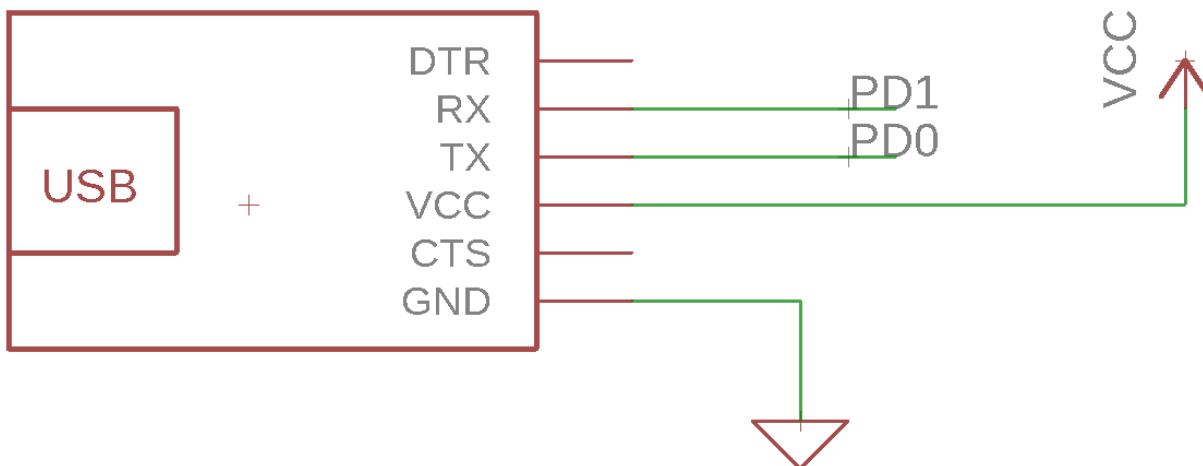


Headere de expansiune

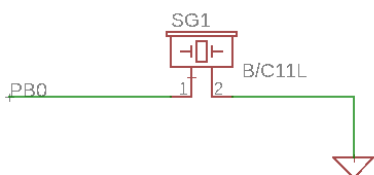


Schema accelerometru

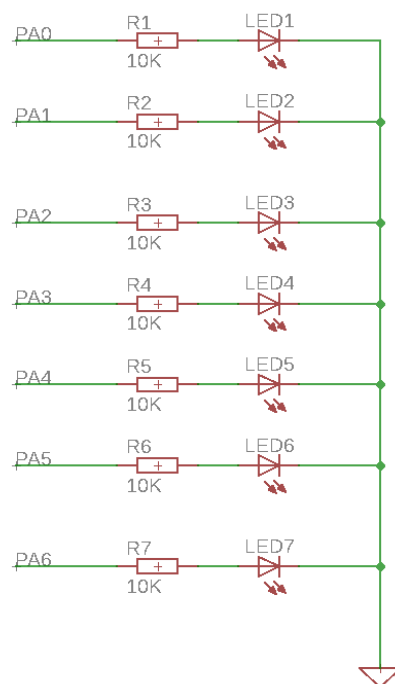
U\$1 USB-SERIAL-FT232R



Schema FTDI



Schema buzzer si leduri



Software Design

Mediul de dezvoltare folosit:

- AtmelStudio pentru dezvoltarea codului si compilare
- Eagle pentru realizarea schemelor electrice

- HIDBootFlash pentru incarcarea programului

Biblioteci folosite:

- usart.h din laboratorul 1: <http://cs.curs.pub.ro/wiki/pm/lab/lab1>
- i2cmaster.h: <https://github.com/mkschreder/avr-ultimate-driver-pack>
- mpu6050.h: <https://github.com/mkschreder/avr-ultimate-driver-pack>

Functii folosite:

- USART0_init() - pentru initializarea comunicatiei seriale
- mpu6050_init() - pentru initializarea accelerometrului
- void afisare(int zar) - afiseaza pe leduri valoarea zarului
- mpu6050_getConvData - citeste valorile acceleratiilor pe cele 3 axe
- USART0_print() - pentru afisarea valorii zarului si a acceleratiei

Rezultate Obținute



Concluzii

Proiectul a iesit cu succes si functioneaza bine dupa parerea mea, fiind primul proiect de acest gen la care am lucrat. A fost un proiect relativ usor de realizat, cu mici probleme pe care le-am intampinat, cum ar fi ca nu am reusit sa lucrez cu LCD-ul pe care l-am achizitionat initial si nu am mai gasit in stoc unul pe care sa stiu sa il folosesc si a trebuit sa schimb strategia si sa folosesc leduri si interfata seriala. La inceput nu eram hotarata intre buzzer si difuzor dar difuzorul nu a functionat cum trebuie cu accelerometrul asa ca am ramas la buzzer. Cu toate aceste schimbari proiectul a fost destul de facil de realizat, am gasit si o biblioteca pentru accelerometrul MPU6050 pe internet (linkul este specificat mai sus) impreuna cu una pentru interfata I2C. Pentru comunicatia seriala am folosit biblioteca din laboratorul 1. Codul l-am realizat pornind de la scheletul de laborator.

Download

Bootloader, GccApplication2, surse, README:

[chiriacteodora_331ca_proiectpm_surse.zip](#)

Poze:

[chiriacteodora_331ca_proiectpm_poze.zip](#)

Schematici si poze:

[chiriacteodora_scheme.zip](#)

Jurnal

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

Bibliografie/Resurse

Resurse software:

- Datasheet ATmega: http://cs.curs.pub.ro/wiki/pm/_media/doc8272.pdf
- Biblioteca MPU6050 si I2C: <https://github.com/mkschreder/avr-ultimate-driver-pack>
- Biblioteca USART si schelet de cod: <http://cs.curs.pub.ro/wiki/pm/lab/lab1> ,
<http://cs.curs.pub.ro/wiki/pm/lab/lab6>
- Bootloader: <http://cs.curs.pub.ro/wiki/pm/tutorial/bootloader>
- AtmelStudio: <https://www.microchip.com/>
- Eagle: <https://www.autodesk.com/products/eagle/free-download>

Resurse hardware:

- Optimus digital: <https://www.optimusdigital.ro/>
- Placa de baza: <http://cs.curs.pub.ro/wiki/pm/cablaj/proiect2018>

- Documentația în format [PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2018/amocanu/25>



Last update: **2021/04/14 15:07**