

Roxana FRUNZĂ (78533) - Peter The Talking Rabbit

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

Introducere

Proiectul presupune realizarea unei jucării care poate interacționa cu copii prin apăsarea unor butoane. În funcție de butoanele apăstate, ursulețul reacționează în diferite moduri.

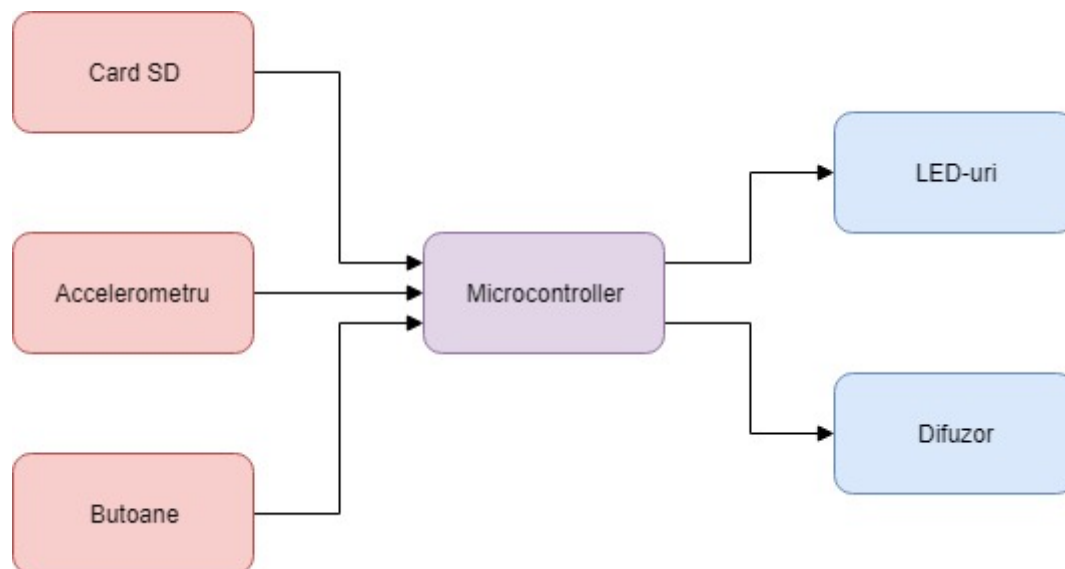
Proiectul se adresează în special copiilor, pentru care un singur animal de pluș nu mai este suficient. Astfel, acest proiect își propune să ofere copiilor un prieten de jucărie cu care poate interacționa, care îi va stârni curiozitatea și de a le dezvolta simțurile, de la stimularea simțului vizual, auditiv sau tactil până la a învăța cum să interacționeze cu lumea din jur.

Descriere generală

Jucăria va primi input de la utilizator prin apăsarea unuia dintre cele 5 butoane plasate în lăbuțe (2), urechi (2), gât (1). La rândul său, jucăria va interacționa pentru cu utilizatorul prin transmiterea de fișiere audio sau aprinderea de LED-uri. Se va utiliza un accelerometru pentru determinarea mișcărilor și în funcție de acestea, jucăria va reacționa în diferite moduri.

În funcție de butoanele apăstate și de mișcările realizate, ursulețul va avea diferite replici prin care se va construi o conversație cu utilizatorul și îl va invita să realizeze diferite acțiuni (să cânte împreună un cântecel, să spună o poveste, să joace un joculeț, etc).

[Schema bloc](#)



Peter este pornit prin apăsarea butonului de la gât. Acesta se va prezenta și pentru a trece în starea următoare se apasă pe butonul de pe lăbuța stângă. Acum utilizatorul are 3 posibilități: să asculte un cântecel (prin apăsarea butonului de pe urechea stângă), să asculte o replică amuzantă (prin apăsarea butonului de pe urechea dreaptă) sau să joace un joc în care trebuie înveselit iepurașul prin diferite acțiuni (prin apăsarea butonului de pe labuța stângă).

Dacă se alege jocul, utilizatorul are datoria de a-l înveseli pe Peter prin diferite acțiuni: să se joace cu urechiușele lui sau să îl legene. După fiecare acțiune, Peter îi va oferi feedback utilizatorului printr-o replică și nivelul de bucurie a acestuia va crește cu 1 pentru fiecare acțiune efectuată. Când nivelul de bucurie ajunge la 7, led-urile galbene se sting și se aprind cele verzi, arătându-i acestuia că a reușit să îl înveselească.

Chiar dacă îi place să se distreze, Peter obosește foarte ușor. De aceea, după 4 acțiuni, jucăria îl va avertiza pe utilizator că a obosit și așteaptă să fie culcat. Până când Peter nu doarme, utilizatorul nu va mai putea interacționa cu el. Odată ce s-a odihnit, Peter este pregătit să se joace din nou.

Hardware Design

Listă piese

Componentă	Cantitate
Placa de bază cu toate <i>componentele</i> acesteia	1
Led verde	2
Led rosu	2
Butoane	5
Difuzor	1
Modul amplificator LM386	1
Modul soclu microSD	1
Accelerometru ADXL345	1
Rezistor 220Ω	2
Placa de test universala 70×90 mm	1
Fire mamă-mamă	30

Header de pini (40p)	1
Cablu UTP	5m
Cablu USB	1

Scheme electrice



Software Design

1. Mediul de dezvoltare

Aplicația a fost dezvoltată în Programmer's Notepad (WinAVR Edition) și încărcată pe microcontroller folosind HID Boot Flash.

2. Biblioteci incluse

```
#include <avr/io.h>
#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdbool.h>
#include <stdio.h>
#include <string.h>

#include "pff.h"
#include "adxl345.h"
```

3. Descrierea codului aplicației

- **pff.c, pff.h** - implementarea PetitFAT pentru a deschide sistemul de fișiere al aplicației de pe un microcard SD formatat FAT32 și a citi informațiile byte cu byte din fișierele, cu posibilitatea parcurgerii recursive a fișierului respectiv.

```
/* Petit FatFs module application interface */

FRESULT pf_mount (FATFS*); /*
Mount/Unmount a logical drive */
FRESULT pf_open (const char*); /* Open a
file */
FRESULT pf_read (void*, WORD, WORD*); /* Read data from
the open file */
FRESULT pf_write (const void*, WORD, WORD*); /* Write data to the open
file */
FRESULT pf_lseek (DWORD); /*
Move file pointer of the open file */
FRESULT pf_opendir (DIR*, const char*); /* Open a directory
```

```
*/
FRESULT pf_readdir (DIR*, FILINFO*);           /* Read a directory
item from the open directory */
```

- **sd.c, sd.h** - implementarea funcțiilor folosite pentru comunicarea efectivă cu cardul SD folosind SPI

```
/******\
 * Functii pentru utilizarea cardului SD      *
 \*****/

DSTATUS disk_initialize();
DRESULT disk_readp(BYTE *buf, DWORD lba, WORD offset, WORD cnt);
DRESULT disk_writep(const BYTE *buf, DWORD sa);
```

- **spi.c, spi.h** - implementarea funcțiilor pentru inițializare și exchange pentru SPI

```
/* SPI config */
#define SPI_PORT      PORTB
#define SPI_DDR       DDRB
#define SPI_MISO      PB6
#define SPI_MOSI      PB5
#define SPI_SCK       PB7

void SPI_init();
uint8_t SPI_exchange(uint8_t data);
```

- **i2cmaster.h, twimaster.c** - implementarea funcțiilor pentru interfața I2C

```
/** defines the data direction (reading from I2C device) in
i2c_start(),i2c_rep_start() */
#define I2C_READ    1

/** defines the data direction (writing to I2C device) in
i2c_start(),i2c_rep_start() */
#define I2C_WRITE    0

extern void i2c_init(void);
void i2c_stop(void);
extern unsigned char i2c_start(unsigned char addr);
extern unsigned char i2c_rep_start(unsigned char addr);
extern void i2c_start_wait(unsigned char addr);
extern unsigned char i2c_write(unsigned char data);
extern unsigned char i2c_readAck(void);
extern unsigned char i2c_readNak(void);
extern unsigned char i2c_read(unsigned char ack);
#define i2c_read(ack)  (ack) ? i2c_readAck() : i2c_readNak();
```

- **adxl345.c, adxl345.h** - implementarea funcțiilor pentru citirea datelor de la accerelerometru prin interfața I2C

```
//definitions
#define ADXL345_ADDR (0x53<<1) //device address
```

```
#define ADXL345_RANGE2G 0x00 //sensitivity to 2g and measurement mode
#define ADXL345_RANGE4G 0x01 //sensitivity to 4g and measurement mode
#define ADXL345_RANGE8G 0x02 //sensitivity to 8g and measurement mode
#define ADXL345_RANGE16G 0x03 //sensitivity to 16g and measurement mode
#define ADXL345_FULLRANGE 0 //1 to enable 0 to disable

//functions declarations
extern void adxl345_init();
extern void adxl345_calibrateoffset();
extern void adxl345_getdata(double *ax, double *ay, double *az);
#if ADXL345_GETATTITUDE == 1
extern void adxl345_getpitchroll(double ax, double ay, double az, double
*pitch, double *roll);
#endif
```

- **main.c** - implementarea funcțiilor de inițializare pini intrare/ieșire, funcție de redare melodie și automatul cu stări care modelează comportamentul jucăriei

Jucăria se afla inițial în starea de sleep (starea 0) și pentru ca Peter să se trezească trebuie apăsat pe butonul de la gât. Se trece în starea 1, iar prin apăsarea butonului de pe lăbuța dreaptă, utilizatorul are acum 3 moduri de a interacționa cu Peter.

- Dacă apasă pe butonul de pe urechea dreaptă, Peter spune o replică în funcție de starea sa de spirit: o replică răutacioasă dacă este trist și una amuzantă dacă este fericit. După ce s-a spus replica, se revine în starea 1.
- Dacă se apasă pe butonul de pe urechea stângă, Peter va cânta un cântecel. Și de această dată se ține cont de starea lui de spirit: dacă este fericit va cânta o melodie veselă, altfel una tristă. După ce s-a spus replica, se revine în starea 1.
- Dacă se apasă pe butonul de pe lăbuța stângă, utilizatorul începe un joc prin care va încerca să îl înveselească pe Peter. Automatul cu stări este în starea 3

Jocul are ca scop creșterea nivelului de bucurie al lui Peter. Utilizatorul are 3 acțiuni care îl bucură pe Peter: dacă îi mângâie urechiușele (prin apăsarea uneia din cele două butoane) sau dacă îl leagănă. Fiecare acțiune îi crește nivelul de bucurie al lui Peter cu 1.

Dupa 4 interacțiuni, Peter obosește și ajunge în starea 5 unde va rămâne până când va fi culcat. LED-urile galbene se aprind intermitent pentru a semnaliza că Peter este obosit și orice interacțiune este ignorată. Odată ce Peter s-a culcat (a fost delectată mișcare), va reveni în starea care era înainte.

Utilizatorul poate întrerupe jocul prin apăsarea butonului de pe lăbuța dreaptă, nivelul de fericire fiind păstrat. Cât timp Peter este trist, sunt aprinse LED-urile galbene, iar când Peter este fericit se aprind LED-urile verzi.

Rezultate Obținute

Din păcate, nu am reușit să redau melodiile de pe cardul SD, în ciuda multor încercări, cu fișiere cu bitrate și sampling rate de diferite valori. Jucăria ține cont de interacțiunile pe care le are cu utilizatorul și reacționează așa cum am descris mai sus, însă, din păcate, a fost necesar un compromis.

Peter nu spune replicile pe care am petrecut (cam) mult timp să le aleg cu grijă, ci redă niște sunete specifice în funcție de starea în care se află.



Concluzii

A fost un proiect interesant, din care am învățat multe despre programarea unui microcontroller și pot spune că am dobândit cunoștințe în diferite domenii, în special pe partea de hardware. Am folosit pentru prima dată în viața mea un letcon, m-am ars și tăiat la degete de atât de multe ori încât poate ar trebui să fac secțiune specială cu rănilor suferite 🤔😅. Am învățat să am grijă cum conectez amplificatorul pentru că dintr-o mică "neatenție" am ars unul și a trebuit să cumpăr altul 😞

Îmi pare rău că nu am reușit și partea de redare fișiere audio. Proiectul ar fi ieșit mult mai amuzant dacă jucăria ar fi reacționat cu replicile pe care le-am pregătit în loc de niște sunete random. Pentru generațiile viitoare aș recomanda să aleagă un proiect care nu folosește card SD pentru că atât mie cât și multor colegi le-a oferit doar bătăi de cap.

În final, aș vrea să le mulțumesc prietenilor că m-au ajutat și suportat pe parcursul acestor săptămâni cât am lucrat la proiect, și părinților pentru susținerea financiară fără de care realizarea proiectului nu ar fi fost posibilă, și pentru sprijinul moral oferit când Peter a decis să rămână mut.

Download

[Arhiva proiectului](#)

Bibliografie/Resurse

Resurse Software

- [Laboratorul 0 PM](#)
- [Laboratorul 4 PM](#)
- [Biblioteca Accelerometru ADXL345](#)

Resurse Hardware

- [Atmega 324 Datasheet](#)
- [LM386 Amplifier Datasheet](#)
- [ADXL345 Accelerometer Datasheet](#)

Componente

[Modul Amplificator LM386](#) [Modul Slot Card MicroSD](#) [Difuzor 8 ohmi](#) [Modul Accelerometru ADXL345](#)

- Documentația în format [PDF](#)
- Schema electrica in EAGLE: [Jucarie Interactiva](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/adraghici/roxana.frunza>



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