

# PaperFolds

## Introducere

**NUME:** Roșu Armand-Alexandru

**GRUPA:** 331CD

Proiectul va consta într-o aplicație cu instrucțiuni spre realizarea a diverse origami-uri. Instrucțiunile vor fi redactate pe un display sub forma unor imagini, însoțite de un mesaj text și de un semnal sonor (diferit în funcție de ce buton este apăsat). Utilizatorul va apăsa pe diverse butoane pentru a alege ce origami vrea să realizeze, pentru a trece de la o imagine la alta, pentru a reveni la ecranul principal sau pentru a primi un mesaj text spre a citi ce are de făcut la pasul respectiv.

## Descriere generală

Robotul conține în primul rând o placuță arduino, ce este alimentată de la un port USB al laptop-ului meu. Din pinii de power, pleacă un cablu cu 5V către o linie + a breadboard-ului și una cu GND către cea - minus alăturată. Din analog 0-3, pleacă câte unul către fiecare buton (mai exact către capătul ce are un rezistor spre borna plus a breadboard-ului), care apoi merg tot spre 5V.

Display-ul are următoarele porturi astfel conectate: GND la GND, VCC la 5V, SCL la digital 13, SDA la 11, RES la 8, DC la 9, CS la 10 și BL în aer. Cardul SD este similar, acesta având următorii pini: GND la GND, 3.3V neconectat, 5V la 5V, CS la 2, MOSI la 12, SCK tot la 13, MISO la 12 și GND la GND.

Amplificatorul este alimentată de la un port USB separat (pinii power având în paralel și un condensator), are două boxe (fiecare cu câte un condensator înseriat) și pinii L, G, R, legați la pinul 3, ground, respectiv tot pinul 3.

Schema bloc:



## Hardware Design

Acestea sunt piesele pe care va trebui să le utilizez:

- arduino

- ecran LCD
- breadboard
- cabluri
- butoane
- card SD
- boxe
- amplificator

Schemă electrică:



## Software Design

Biblioteci folosite:

- SPI.h (folosită pentru comunicarea cu dispozitivele de tip SPI)
- SD.h (pentru lucrul cu cardul SD)
- TFT.h (pentru lucrul cu display-ul)

Cod Arduino IDE:

```
#include <SPI.h>
#include <SD.h>
#include <TFT.h>

#define sd_cs 4
#define lcd_cs 10
#define dc 9
#define rst 8

#define NOTE_C0 16

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
```

```
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
```

```
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978

int sensorPin = A0;
int sensorValue = 200;

int sensorPin2 = A1;
int sensorValue2 = 200;

int sensorPin3 = A2;
int sensorValue3 = 200;

int sensorPin4 = A3;
int sensorValue4 = 200;

int current_screen = 0;
int instr = 0;
int step = 0;

TFT TFTscreen = TFT(lcd_cs, dc, rst);

PImage logo;

int x = 10;
int y = -300;

void setup() {
  TFTscreen.begin();
```

```
TFTscreen.background(255, 255, 255);

TFTscreen.stroke(0, 0, 255);
TFTscreen.println();
TFTscreen.println(F("Arduino TFT Bitmap Example"));
TFTscreen.stroke(0, 0, 0);
TFTscreen.println(F("Open serial monitor"));
TFTscreen.println(F("to run the sketch"));

Serial.begin(9600);
while (!Serial) {
}

TFTscreen.background(255, 255, 255);

Serial.print(F("Initializing SD card..."));
if (!SD.begin(sd_cs)) {
  Serial.println(F("failed!"));
  return;
}
Serial.println(F("OK!"));

TFTscreen.begin();
TFTscreen.background(255, 255, 255);

logo = TFTscreen.loadImage("origami.bmp");
if (!logo.isValid()) {
  Serial.println(F("error while loading arduino.bmp"));
}
tone(3, NOTE_C0, 300);
delay(300);
noTone(3);
}

void loop() {
  sensorValue = analogRead(sensorPin);
  sensorValue2 = analogRead(sensorPin2);
  sensorValue3 = analogRead(sensorPin3);
  sensorValue4 = analogRead(sensorPin4);

  if (current_screen == 0) {
    if (sensorValue < 200) {
      tone(3, NOTE_A3, 300);
      TFTscreen.background(255, 255, 255);
      x = 10;
      y = -10;
      current_screen = 1;
      step = 1;
      delay(300);
      noTone(3);
    }
  }
}
```

```
if (sensorValue2 < 200) {
  tone(3, NOTE_D4, 300);
  TFTscreen.background(255, 255, 255);
  x = 10;
  y = -160;
  current_screen = 2;
  delay(300);
  noTone(3);
  step = 1;
}
} else {
if (sensorValue < 200) {
  if (step > 1) {
    tone(3, NOTE_A3, 300);
    TFTscreen.background(255, 255, 255);
    x += 195;
    step -= 1;
    if (instr == 1) {
      y += 400;
      instr = 0;
    }
    delay(300);
    noTone(3);
  }
}
if (sensorValue2 < 200) {
  if (step < 7) {
    tone(3, NOTE_D4, 300);
    TFTscreen.background(255, 255, 255);
    x -= 195;
    step += 1;
    if (instr == 1) {
      y += 400;
      instr = 0;
    }
    delay(300);
    noTone(3);
  }
}
if (sensorValue3 < 200) {
  tone(3, NOTE_D2, 300);
  if (instr == 0) {
    instr = 1;
    TFTscreen.background(255, 255, 255);
    y -= 400;
    delay(300);
    noTone(3);
  } else {
    instr = 0;
    TFTscreen.background(255, 255, 255);
    y += 400;
```

```
        delay(300);
        noTone(3);
    }
}
if (sensorValue4 < 200) {
    tone(3, NOTE_E2, 300);
    TFTscreen.background(255, 255, 255);
    x = 10;
    y = -300;
    current_screen = 0;
    delay(300);
    noTone(3);
}
}

if (logo.isValid() == false) {
    return;
}

Serial.println(F("drawing image"));

TFTscreen.image(logo, x, y);
}
```

## Laboratoare Utilizate

### Laboratorul 0: GPIO

GPIO mă ajută pentru a utiliza plăcuța Arduino UNO.

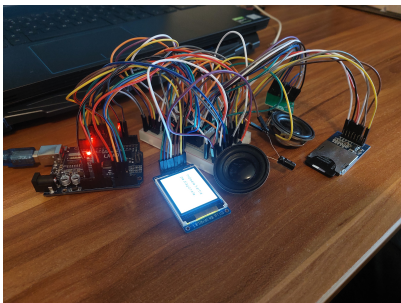
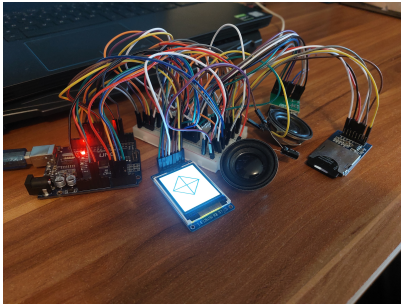
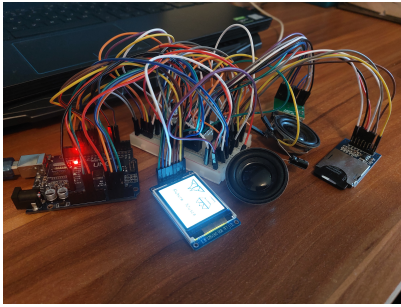
### Laboratorul 4: ADC

Acesta este folosit în detectarea faptului dacă un buton este apăsat sau nu.

### Laboratorul 5: SPI

Serial Peripheral Interface îmi trebuie în lucrul cu cardul SD și cu display-ul.

## Rezultate Obținute



## Download

<https://github.com/ArmandRosu/PaperFolds.git>

## Bibliografie/Resurse

Materiale folosite în scrierea codului:

- <http://www.arduino.cc/en/Tutorial/TFTBitmapLogo>
- Arduino UNO - File - Examples - Digital - toneMelody

[Export to PDF](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2024/cpatru/armand.rosu>

Last update: 2024/05/26 16:01





