Co-funded by the
Erasmus+ Programme
of the European Union

ROBOSTEM IO2 - Module on microcontroller programming

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Title

Building a simple Tic-Tac-Toe game with an Arduino

Required elements

- An Arduino Uno
- A 2.8” Touch Screen

Other:

- Computer + cable to program the Arduino
- Basic understanding of electronics
- Arduino programming fundamentals

Assembly steps

Step 1

Prepare Arduino UNO [1] and a 2.8” Touch Screen [2]. The breadboard should be positioned in a way that the positive voltage rail is closer to you

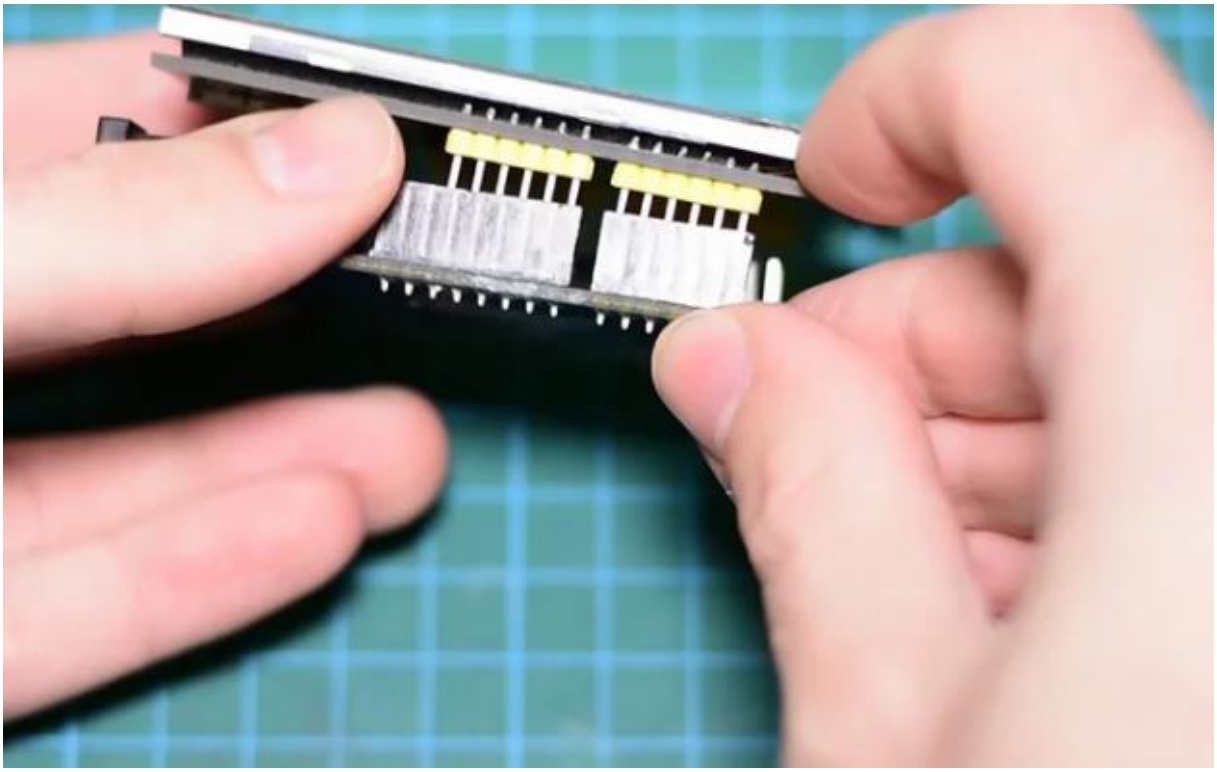
[1] Arduino UNO

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

[2] 2.8” Touch Screen

Step 2

Connect Arduino UNO and a 2.8” Touch Screen. [3]

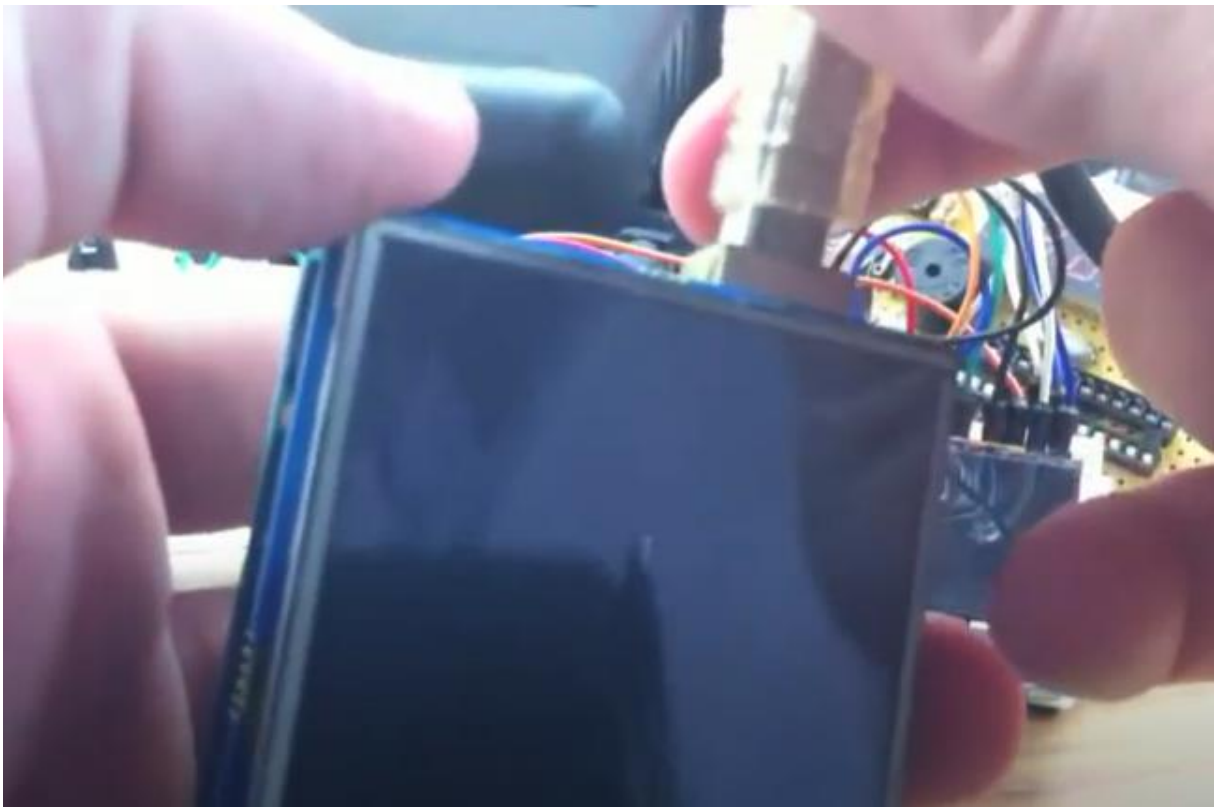


[3] Assembly

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 3

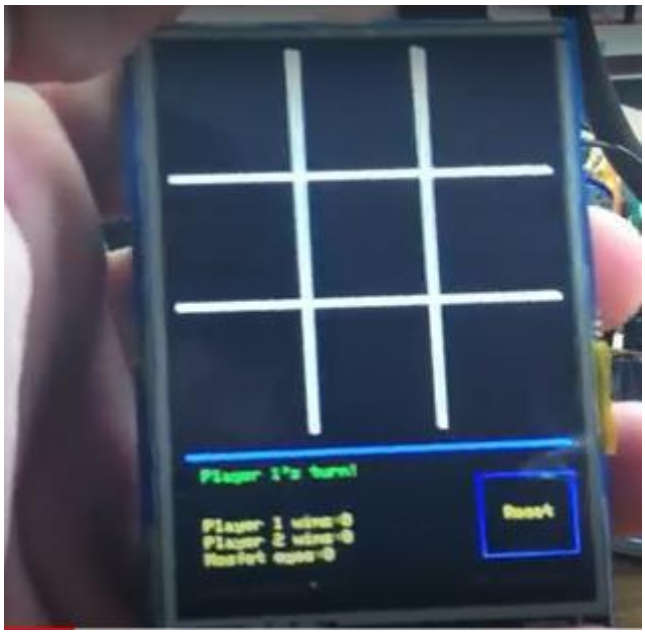
Plug the cable to connect it with a computer.



[4] Connecting assembled parts with computer

Step 4

After connecting, we can load the code and we are ready to play. *[5]*



[5] Ready to play

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Coding steps

Step 1

Include necessary libraries

```
#include "TFTLCD.h"
#include "TouchScreen.h"
#include <EEPROM.h>
#if not defined USE_ADAFRUIT_SHIELD_PINOUT
#error "For use with the shield, make sure to #define USE_ADAFRUIT_SHIELD_PINOUT in the
TFTLCD.h library file"
#endif
```

Step 2

These are the pins for the shield!

```
#define YP A1  // must be an analog pin, use "An" notation!
#define XM A2  // must be an analog pin, use "An" notation!
#define YM 7   // can be a digital pin
#define XP 6   // can be a digital pin

#define TS_MINX 150
#define TS_MINY 120
#define TS_MAXX 920
#define TS_MAXY 940
```

Step 3

For better pressure precision, we need to know the resistance between X+ and X- Use any multimeter to read it. For the one we're using, its 300 ohms across the X plate.

```
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
```

Step 4

Color definitions - in 5:6:5.

```
#define  BLACK           0x0000
#define  BLUE            0x001F
#define  RED             0xF800
#define  GREEN           0x07E0
#define CYAN            0x07FF
#define MAGENTA         0xF81F
#define YELLOW          0xFFE0
#define WHITE           0xFFFF
#define TEST            0x1BF5
#define JJCOLOR         0x1CB6
#define JJORNG          0xFD03
```

```
TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, 0);
int i = 0;
int backlight = 3;

int upperleft = 0;
int uppermid = 0;
int upperright = 0;
int midleft = 0;
int center = 0;
int midright = 0;
int lowerleft = 0;
int lowermid = 0;
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
int lowerright = 0;
int ul = 1;
int um = 1;
int ur = 1;
int ml = 1;
int cent = 1;
int mr = 1;
int ll = 1;
int lm = 1;
int lr = 1;
int turn = 1;
int gameover = 0;
int ponewins = 0;
int ptwowins = 0;
int mosfets = 0;
char playerone [10];
char playertwo [10];
char eyes [10];
void setup(void) {
  tft.reset();
  pinMode(backlight, OUTPUT);
  Serial.begin(9600);
  for(i = 0 ; i <= 255; i+=1) {
    analogWrite(backlight, i);
    delay(2);
  }
  tft.reset();
  tft.initDisplay();
  tft.fillScreen(BLACK);
  tft.drawString(40, 150, "Tic Tac Touch", WHITE, 2);
  delay(1500);
  drawboard();
 // tft.drawChar(28, 20, 'X', RED, 5);
 // tft.drawChar(108, 20, 'X', RED, 5);
 // tft.drawChar(188, 20, 'X', RED, 5);
 // tft.drawChar(28, 100, 'X', RED, 5);
 // tft.drawChar(108, 100, 'X', RED, 5);
 // tft.drawChar(188, 100, 'X', RED, 5);
 // tft.drawChar(28, 180, 'X', RED, 5);
 // tft.drawChar(108, 180, 'X', RED, 5);
 // tft.drawChar(188, 180, 'X', RED, 5);
  pinMode(13, OUTPUT);
}
#define MINPRESSURE 10
#define MAXPRESSURE 1000
void loop()
{

  digitalWrite(13, HIGH);
  Point p = ts.getPoint();
  digitalWrite(13, LOW);
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 5

If you're sharing pins, you'll need to fix the directions of the touchscreen pins!

```
//pinMode(XP, OUTPUT);
pinMode(XM, OUTPUT);
pinMode(YP, OUTPUT);
//pinMode(YM, OUTPUT);
// we have some minimum pressure we consider 'valid'
// pressure of 0 means no pressing!
if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {

  /*
  Serial.print("X = ");
   Serial.print(p.x);
   Serial.print("\tY = ");
   Serial.print(p.y);
   Serial.print("\tPressure = ");
   Serial.println(p.z);
   */
  // turn from 0->1023 to tft.width
  p.x = map(p.x, TS_MINX, TS_MAXX, 240, 0);
  p.y = map(p.y, TS_MINY, TS_MAXY, 320, 0);

  Serial.print("p.y:");
```

Step 6

This code will help you get the y and x numbers for the touchscreen.

```
   Serial.print(p.y);
   Serial.print("   p.x:");
   Serial.println(p.x);


  // Upper Left
  if ((p.y > -4 && p.y < 74 && p.x > 3 && p.x < 82) && (ul == 1) && (gameover == 0)) {
    if (turn == 1) {
    tft.drawChar(28, 20, 'O', GREEN, 5);
    upperleft = 1;
    }
    if (turn == 2) {
    tft.drawChar(28, 20, 'X', RED, 5);
    upperleft = 2;
    }
    ul = 0;
    turntoggle();
    showturn();
  }




  // Upper Mid
  if ((p.y > -4 && p.y < 74 && p.x > 91 && p.x < 164) && (um == 1) && (gameover == 0)) {
    if (turn == 1) {
    tft.drawChar(108, 20, 'O', GREEN, 5);
    uppermid = 1;
    }
    if (turn == 2) {
    tft.drawChar(108, 20, 'X', RED, 5);
    uppermid = 2;
    }
    um = 0;
    turntoggle();
    showturn();
  }
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    // Upper Right
    if ((p.y > -4 && p.y < 74 && p.x > 166 && p.x < 243) && (ur == 1) && (gameover == 0)) {
      if (turn == 1) {
      tft.drawChar(188, 20, 'O', GREEN, 5);
      upperright = 1;
      }
      if (turn == 2) {
      tft.drawChar(188, 20, 'X', RED, 5);
      upperright = 2;
      }
      ur = 0;
      turntoggle();
      showturn();
    }


    // Mid Left
    if ((p.y > 80 && p.y < 153 && p.x > 3 && p.x < 82) && (ml == 1) && (gameover == 0)) {
      if (turn == 1) {
      tft.drawChar(28, 100, 'O', GREEN, 5);
      midleft = 1;
      }
      if (turn == 2) {
      tft.drawChar(28, 100, 'X', RED, 5);
      midleft = 2;
      }
      ml = 0;
      turntoggle();
      showturn();
    }


    // Center
    if ((p.y > 80 && p.y < 153 && p.x > 91 && p.x < 164) && (cent == 1) && (gameover == 0))
{
      if (turn == 1) {
      tft.drawChar(108, 100, 'O', GREEN, 5);
      center = 1;
      }
      if (turn == 2) {
      tft.drawChar(108, 100, 'X', RED, 5);
      center = 2;
      }
      cent = 0;
      turntoggle();
      showturn();
    }


    // Mid Right
    if ((p.y > 80 && p.y < 153 && p.x > 166 && p.x < 243) && (mr == 1) && (gameover == 0))
{
      if (turn == 1) {
      tft.drawChar(188, 100, 'O', GREEN, 5);
      midright = 1;
      }
      if (turn == 2) {
      tft.drawChar(188, 100, 'X', RED, 5);
      midright = 2;
      }
      mr = 0;
      turntoggle();
      showturn();
    }
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    // Lower Left
    if ((p.y > 162 && p.y < 240 && p.x > 3 && p.x < 82) && (ll == 1) && (gameover == 0)) {
      if (turn == 1) {
      tft.drawChar(28, 180, 'O', GREEN, 5);
      lowerleft = 1;
      }
      if (turn == 2) {
      tft.drawChar(28, 180, 'X', RED, 5);
      lowerleft = 2;
      }
      ll = 0;
      turntoggle();
      showturn();
    }


    // Lower Mid
    if ((p.y > 162 && p.y < 240 && p.x > 91 && p.x < 164) && (lm == 1) && (gameover == 0))
{
      if (turn == 1) {
      tft.drawChar(108, 180, 'O', GREEN, 5);
      lowermid = 1;
      }
      if (turn == 2) {
      tft.drawChar(108, 180, 'X', RED, 5);
      lowermid = 2;
      }
      lm = 0;
      turntoggle();
      showturn();
    }


    // Lower Right
    if ((p.y > 162 && p.y < 240 && p.x > 166 && p.x < 243) && (lr == 1) && (gameover == 0))
{
      if (turn == 1) {
      tft.drawChar(188, 180, 'O', GREEN, 5);
      lowerright = 1;
      }
      if (turn == 2) {
      tft.drawChar(188, 180, 'X', RED, 5);
      lowerright = 2;
      }
      lr = 0;
      turntoggle();
      showturn();
    }


    // Reset Area
    if (p.y > 270 && p.y < 318 && p.x > 189 && p.x < 246) {
      turn = 1;
      ul = 1;
      um = 1;
      ur = 1;
      ml = 1;
      cent = 1;
      mr = 1;
      ll = 1;
      lm = 1;
      lr = 1;
      upperleft = 0;
      uppermid = 0;
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
      upperright = 0;
      midleft = 0;
      center = 0;
      midright = 0;
      lowerleft = 0;
      lowermid = 0;
      lowerright = 0;
      gameover = 0;
      drawboard();
    }



    if ((upperleft == 1) && (uppermid == 1) && (upperright == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((upperleft == 2) && (uppermid == 2) && (upperright == 2) && (gameover == 0)) {
      playertwowin();
    }
    if ((midleft == 1) && (center == 1) && (midright == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((midleft == 2) && (center == 2) && (midright == 2) && (gameover == 0)) {
      playertwowin();
    }
    if ((lowerleft == 1) && (lowermid == 1) && (lowerright == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((lowerleft == 2) && (lowermid == 2) && (lowerright == 2) && (gameover == 0)) {
      playertwowin();
    }
    if ((upperleft == 1) && (midleft == 1) && (lowerleft == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((upperleft == 2) && (midleft == 2) && (lowerleft == 2) && (gameover == 0)) {
      playertwowin();
    }
    if ((uppermid == 1) && (center == 1) && (lowermid == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((uppermid == 2) && (center == 2) && (lowermid == 2) && (gameover == 0)) {
      playertwowin();
    }
    if ((upperright == 1) && (midright == 1) && (lowerright == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((upperright == 2) && (midright == 2) && (lowerright == 2) && (gameover == 0)) {
      playertwowin();
    }

    if ((upperleft == 1) && (center == 1) && (lowerright == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((upperleft == 2) && (center == 2) && (lowerright == 2) && (gameover == 0)) {
      playertwowin();
    }

    if ((upperright == 1) && (center == 1) && (lowerleft == 1) && (gameover == 0)) {
      playeronewin();
    }
    if ((upperright == 2) && (center == 2) && (lowerleft == 2) && (gameover == 0)) {
      playertwowin();
    }
    if ((upperleft != 0) && (uppermid != 0) && (upperright != 0) && (midleft != 0) &&
(center != 0) && (midright != 0) && (lowerleft != 0) && (lowermid != 0) && (lowerright !=
0) && (gameover == 0)) {
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    catseye();
  }

}
}

void catseye() {
  tft.fillRect(10, 260, 96, 8, BLACK);
  tft.drawString(10, 275, "Mosfet Eye!", WHITE, 2);
  mosfets++;
  updatewins();
  gameover = 1;
}

void playeronewin() {
  tft.fillRect(10, 260, 96, 8, BLACK);
  tft.drawString(10, 275, "Player 1 wins!", WHITE, 2);
  ponewins++;
  updatewins();
  gameover = 1;
}
void playertwowin() {
  tft.fillRect(10, 260, 96, 8, BLACK);
  tft.drawString(10, 275, "Player 2 wins!", WHITE, 2);
  ptwowins++;
  updatewins();
  gameover = 1;
}
void turntoggle() {
  if (turn == 1) {
    turn = 2;
    return;
  }
  if (turn == 2) {
    turn = 1;
  }
}
void updatewins() {
  tft.fillRect(94, 290, 24, 8, BLACK);
  itoa (ponewins, playerone, 10);
  tft.drawString(94, 290, playerone, YELLOW);

  tft.fillRect(94, 300, 24, 8, BLACK);
  itoa (ptwowins, playertwo, 10);
  tft.drawString(94, 300, playertwo, YELLOW);

  tft.fillRect(82, 310, 24, 8, BLACK);
  itoa (mosfets, eyes, 10);
  tft.drawString(82, 310, eyes, YELLOW);
}



void showturn() {
  if (turn == 1) {
    tft.fillRect(10, 260, 96, 8, BLACK);
    tft.drawString(10, 260, "Player 1's turn!", GREEN);
  }
  if (turn == 2) {
    tft.fillRect(10, 260, 96, 9, BLACK);
    tft.drawString(10, 260, "Player 2's turn!", RED);
  }
}
void drawboard() {
  tft.fillScreen(BLACK);
tft.fillRect(78, 0, 4, 240, WHITE);
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
tft.fillRect(158, 0, 4, 240, WHITE);
tft.fillRect(0, 78, 240, 4, WHITE);
tft.fillRect(0, 158, 240, 4, WHITE);
tft.fillRect(0, 250, 240, 4, BLUE);
tft.drawRect(180, 270, 60, 50, BLUE);
tft.drawString(196, 290, "Reset", YELLOW);
showturn();
tft.drawString(10, 290, "Player 1 wins:", YELLOW);
tft.drawString(10, 300, "Player 2 wins:", YELLOW);
tft.drawString(10, 310, "Mosfet eyes:", YELLOW);
updatewins();
}
```

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.