

```
// Coffee dispenser code
// Karol Merta
// Príloha C

// Libraries
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <Adafruit_GFX.h>
#include <MCUFRIEND_kbv.h>

MCUFRIEND_kbv tft;
Adafruit_PWMServoDriver driverPCA =
Adafruit_PWMServoDriver(0x40);

// LCD pins definition
#define YP A1
#define XM A2
#define YM 7
#define XP 6
#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4

// Colours definition
#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 ORANGE 0xFD20
#define DARKORANGE 0xFB60
#define MAROON 0x7800
#define BLACKM 0x18E3
#define SILVER 0xC618
#define GOLD 0xFEAA
#define BROWN 0xA145
#define LIME 0x07E0
#define GREENYELLOW 0xAFE5
#define PURPLE 0x780F
#define OLIVE 0x7BE0
#define LIGHTGREY 0xC618
#define DARKGREY 0x7BEF
#define NAVY 0x000F
#define DARKGREEN 0x03E0
#define DARKCYAN 0x03EF
```

```
// RGB LED pins
```

```
int red_pin = 29;
```

```
int green_pin = 31;
```

```
int blue_pin = 33;
```

```
// Initialization of pin and variable that reads  
the presents of the bag
```

```
int clip_pin = 27;
```

```
int clip_val;
```

```
// Initialization of pins and variables that
```

```
read joystick's x and y axes and a button state
int Xpin=A6;
int Ypin=A7;
int SWpin=10;

int Xval;
int Yval;
int SWval;

// Minimal and maximal values of joystick
const int XYmin = 0;
const int XYmax = 1023;

// Whitch button we Hoover over
short menu_sel = 0; // 0 - Arabica, 1 - Robusta,
2 - BrazilMild, 3 - A+R_Mix
short page_sel = 0; // 0 - Volume, 1 - Cancel, 3
- Confirm

// Whitch page are we currently on
short curr_page = 4; // 4 - Menu_page, 0 -
Arabica, 1 - Robusta, 2 - BrazilMild, 3 - A+R_Mix

// True if joystick was pressed false otherwise
bool clicked = false;
bool old = true;

// Selected amount of coffee
short volume = 1;
const int max_vol = 10;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    tft.reset();
```

```
    tft.begin(tft.readID());
```

```
    driverPCA.begin();
```

```
    driverPCA.setPWMPfreq(60);
```

```
    pinMode(Xpin, INPUT);
```

```
    pinMode(Ypin, INPUT);
```

```
    pinMode(SWpin, INPUT);
```

```
    digitalWrite(SWpin, HIGH);
```

```
    pinMode(red_pin, OUTPUT);
```

```
    pinMode(green_pin, OUTPUT);
```

```
    pinMode(blue_pin, OUTPUT);
```

```
    white_light();
```

```
    menu();
```

```
}
```

```
// Main functionality based on current page,  
selected button and joystick click
```

```
void loop() {
```

```
    Xval = analogRead(Xpin);
```

```
    Yval = analogRead(Ypin);
```

```
    SWval = digitalRead(SWpin);
```

```
delay(5);
```

```
if(SWval == 1 && !old) {  
    clicked = true;  
}
```

```
if(!clicked) {  
    if(curr_page == 4) {  
        up_down_menu();  
    }else{  
        up_down_page();  
        if(page_sel == 0) {  
            change_volume();  
        }  
    }  
}else{  
    if(curr_page == 4) {  
        page_sel = 0;  
        curr_page = menu_sel;  
        page_change();  
    }else{  
        if(page_sel == 1) {  
            menu_sel = 0;  
            volume = 1;  
            curr_page = 4;  
            page_change();  
        }  
        if(page_sel == 2) {  
            confirm_function();  
        }  
    }  
}
```

```

        }
    }
    clicked = false;
    old = SWval;
}

//Runs servo motors in case clip_val == 0
otherwise display insert_bag_page
void confirm_function(){
    clip_val = digitalRead(clip_pin);
    if(clip_val == 1){
        red_light();
        insert_bag_page();
        while(clip_val == 1){
            clip_val = digitalRead(clip_pin);
        }
    }
    if(clip_val == 0){
        green_light();
        confirmation_page();
        delay(1000);
        run_servo();
        delay(1500);
        menu();
        curr_page = 4;
    }
    white_light();
}

```

// Base function for running servo motors. In case of the page 3 we run two motors

simultaneously

```
void run_servo() {
```

```
    if(curr_page == 3) {
```

```
        for(int i = 0; i < volume; i++) {
```

```
            delay(1500);
```

```
            driverPCA.setPWM(0, 0, 633);
```

```
            driverPCA.setPWM(1, 0, 633);
```

```
            delay(1500);
```

```
            driverPCA.setPWM(0, 0, 84);
```

```
            driverPCA.setPWM(1, 0, 84);
```

```
        }
```

```
    }else{
```

```
        for(int i = 0; i < volume; i++) {
```

```
            delay(1500);
```

```
            driverPCA.setPWM(curr_page, 0, 633);
```

```
            delay(1500);
```

```
            driverPCA.setPWM(curr_page, 0, 84);
```

```
        }
```

```
    }
```

```
}
```

```
// Change display based on current page variable
```

```
void page_change() {
```

```
    switch (curr_page) {
```

```
        case 0:
```

```
            ArabicaPage();
```

```
            break;
```

```
        case 1:
```

```

    RobustaPage ();
    break;

case 2:
    BrazilMildPage ();
    break;

case 3:
    AandR_MixPage ();
    break;

case 4:
    menu ();
    break;
}
}

// Divides range of possible values of joysticks
// y axis and increase or decrease volume
// range = 0 right
// range = 1 and 2 neutral
// range = 3 left
void change_volume () {
    int Yrange = map (Yval, XYmin, XYmax, 0, 3);
    switch (Yrange) {
        case 0:
            if (curr_page != 3) {
                if (volume < max_vol) {
                    volume += 1;
                }
            }
            else {

```



```

        if(volume < max_vol/2) {
            volume += 1;
        }
    }
    page_buttons();
    break;

case 3:
    if(volume > 1) {
        volume -= 1;
    }

    page_buttons();
    break;
}
}

// Divides range of possible values of joysticks
// x axis and increase or decrease page selection
// range = 0 up
// range = 1 and 2 neutral
// range = 3 down
void up_down_page() {
    int Xrange = map(Xval, XYmin, XYmax, 0, 3);
    switch(Xrange) {

        case 0:
            if(page_sel == 0) {
                page_sel = 2;
            }else{
                page_sel -= 1;
            }
        }
    }
}

```

```

        }
    page_buttons ();
    break;

case 3:
    if (page_sel == 2) {
        page_sel = 0;
    } else {
        page_sel += 1;
    }
    page_buttons ();
    break;
}
}

```

// Displays page buttons according to variable page_sel

```

void page_buttons () {
    switch (page_sel) {
        case 0:
            VolumeON ();
            CancelOFF ();
            ConfirmOFF ();
            break;

        case 1:
            VolumeOFF ();
            CancelON ();
            ConfirmOFF ();
            break;
    }
}

```

```
case 2:
```

```
    VolumeOFF();
```

```
    CancelOFF();
```

```
    ConfirmON();
```

```
    break;
```

```
}
```

```
}
```

```
// Divides range of possible values of joysticks  
x axis and increase or decrease menu_sel variable
```

```
// range = 0 up
```

```
// range = 1 and 2 neutral
```

```
// range = 3 down
```

```
void up_down_menu() {
```

```
    int Xrange = map(Xval, XYmin, XYmax, 0, 3);
```

```
    switch(Xrange) {
```

```
        case 0:
```

```
            if(menu_sel == 0) {
```

```
                menu_sel = 3;
```

```
            }else{
```

```
                menu_sel -= 1;
```

```
            }
```

```
            menu_buttons();
```

```
            break;
```

```
        case 3:
```

```
            if(menu_sel == 3) {
```

```
                menu_sel = 0;
```

```
            }else{
```

```
        menu_sel += 1;
    }
    menu_buttons();
    break;
}
}
```

// Displays menu buttons according to menu selection variable

```
void menu_buttons() {
    switch(menu_sel) {
        case 0:
            ArabicaButtonON();
            RobustaButtonOFF();
            BrazilMildButtonOFF();
            AandR_MixButtonOFF();
            break;

        case 1:
            ArabicaButtonOFF();
            RobustaButtonON();
            BrazilMildButtonOFF();
            AandR_MixButtonOFF();
            break;

        case 2:
            ArabicaButtonOFF();
            RobustaButtonOFF();
            BrazilMildButtonON();
            AandR_MixButtonOFF();
            break;
```

```
case 3:
    ArabicaButtonOFF();
    RobustaButtonOFF();
    BrazilMildButtonOFF();
    AandR_MixButtonON();
    break;
}
```

```
// Displays confirmation page
```

```
void confirmation_page() {
    tft.fillScreen(MAROON);
    tft.setRotation(0);
    tft.setTextSize(4);
    tft.setTextColor(WHITE);
    tft.setCursor(0, 40);
    tft.print("ENJOY YOUR");
    tft.setCursor(45, 80);
    tft.print("COFFEE");
    tft.setCursor(45, 160);
    tft.print("HAVE A");
    tft.setCursor(25, 200);
    tft.print("NICE DAY");
}
```

```
// Displays insert bag page
```

```
void insert_bag_page() {
    tft.fillScreen(MAROON);
    tft.setRotation(0);
    tft.setTextSize(3);
```

```
tft.setTextColor(WHITE);  
tft.setCursor(20, 40);  
tft.print("THE OUTFLOW");  
tft.setCursor(35, 70);  
tft.print("IS EMPTY");  
tft.setCursor(4,100);  
tft.print("PLEASE INSERT");  
tft.setCursor(20,130);  
tft.print("A PAPER BAG");  
tft.setTextSize(4);  
tft.setCursor(90,160);  
tft.print("!!!");  
}
```

```
// Displays menu page
```

```
void menu() {  
  tft.fillScreen(MAROON);  
  tft.setRotation(0);  
  Header();  
  ArabicaButtonON();  
  RobustaButtonOFF();  
  BrazilMildButtonOFF();  
  AandR_MixButtonOFF();  
  UpDown();  
}
```

```
// Displays Arabica page
```

```
void ArabicaPage() {  
  tft.fillScreen(DARKORANGE);  
  tft.setTextSize(4);  
  tft.setTextColor(BLACK);
```

```
tft.setCursor(35, 10);  
tft.print("ARABICA");  
tft.fillRect(0, 50, 240, 4, BLACK);
```

```
Comment_Arabica();  
VolumeON();  
ConfirmOFF();  
CancelOFF();  
}
```

```
// Displays Robusta page
```

```
void RobustaPage() {  
  tft.fillScreen(SILVER);  
  tft.setTextSize(4);  
  tft.setTextColor(BLACK);  
  tft.setCursor(35, 10);  
  tft.print("ROBUSTA");  
  tft.fillRect(0, 50, 240, 4, BLACK);
```

```
Comment_Robusta();  
VolumeON();  
ConfirmOFF();  
CancelOFF();  
}
```

```
// Displays BrazilMild page
```

```
void BrazilMildPage() {  
  tft.fillScreen(GREEN);  
  tft.setTextSize(4);  
  tft.setTextColor(BLACK);  
  tft.setCursor(0, 10);
```

```
tft.print("BrazilMild");
tft.fillRect(0,50,240,4,BLACK);

Comment_BrazilMild();
VolumeON();
ConfirmOFF();
CancelOFF();
}

// Displays AandR_MixPage page
void AandR_MixPage() {
  tft.fillScreen(BLUE);
  tft.setTextSize(4);
  tft.setTextColor(BLACK);
  tft.setCursor(15, 10);
  tft.print("AandR_Mix");
  tft.fillRect(0,50,240,4,BLACK);

  Comment_AandR_Mix();
  VolumeON();
  ConfirmOFF();
  CancelOFF();
}

// Displays Header page
void Header() {
  tft.setTextSize(4);
  tft.setTextColor(WHITE);
  tft.setCursor(40, 20);
  tft.print("COFFEE");
```



```
tft.setCursor(15, 65);
tft.print("DISPENSER");
tft.fillRect(0,100,240,4,WHITE);
}

// Displays Arabica button as not selected
void ArabicaButtonOFF() {
    tft.fillRoundRect(10, 110, 195, 40, 195,
DARKORANGE);
    tft.setTextSize(3);
    tft.setTextColor(BLACK);
    tft.setCursor(45, 120);
    tft.print("ARABICA");
}

// Displays Arabica button as selected
void ArabicaButtonON() {
    tft.fillRoundRect(10, 110, 195, 40, 195,
WHITE);
    tft.setTextSize(3);
    tft.setTextColor(BLACK);
    tft.setCursor(45, 120);
    tft.print("ARABICA");
}

// Displays Robusta button as not selected
void RobustaButtonOFF() {
    tft.fillRoundRect(10, 160, 195, 40, 195,
SILVER);
    tft.setTextSize(3);
    tft.setTextColor(BLACK);
```

```
tft.setCursor(45, 170);
tft.print("ROBUSTA");
}

// Displays Robusta button as selected
void RobustaButtonON() {
  tft.fillRoundRect(10, 160, 195, 40, 195,
WHITE);
  tft.setTextSize(3);
  tft.setTextColor(BLACK);
  tft.setCursor(45, 170);
  tft.print("ROBUSTA");
}

// Displays BrazilMild button as not selected
void BrazilMildButtonOFF() {
  tft.fillRoundRect(10, 210, 195, 40, 195, LIME);
  tft.setTextSize(3);
  tft.setTextColor(BLACK);
  tft.setCursor(20, 220);
  tft.print("BrazilMild");
}

// Displays BrazilMild button as selected
void BrazilMildButtonON() {
  tft.fillRoundRect(10, 210, 195, 40, 195,
WHITE);
  tft.setTextSize(3);
  tft.setTextColor(BLACK);
  tft.setCursor(20, 220);
  tft.print("BrazilMild");
}
```

```
}

// Displays AandR_Mix button as not selected
void AandR_MixButtonOFF() {
    tft.fillRoundRect(10, 260, 195, 40, 195, BLUE);
    tft.setTextSize(3);
    tft.setTextColor(BLACK);
    tft.setCursor(45, 270);
    tft.print("A+R_Mix");
}

// Displays AandR_Mix button as selected
void AandR_MixButtonON() {
    tft.fillRoundRect(10, 260, 195, 40, 195,
WHITE);
    tft.setTextSize(3);
    tft.setTextColor(BLACK);
    tft.setCursor(45, 270);
    tft.print("A+R_Mix");
}

// Displays comment for arabica page
void Comment_Arabica() {
    tft.setTextSize(2);
    tft.setTextColor(BLACK);
    tft.setCursor(10, 60);
    tft.print("Arabica is known      for its
smooth,      complex flavor and distinct lack
of      bitterness with      earthy notes.");
}
```

```
// Displays comment for Robusta page
void Comment_Robusta() {
    tft.setTextSize(2);
    tft.setTextColor(BLACK);
    tft.setCursor(10, 60);
    tft.print("Robusta coffee is notoriously
bitter. Mostly used for an espresso, and as a
filler in coffee blends.");
}

// Displays comment for BrazilMild page
void Comment_BrazilMild() {
    tft.setTextSize(2);
    tft.setTextColor(BLACK);
    tft.setCursor(5, 60);
    tft.print("Brazilian coffee is characterized by
itsextremely delicate taste. Enjoy this
velvety coffee with a hint of hazelnut.");
}

// Displays comment for AandR_Mix page
void Comment_AandR_Mix() {
    tft.setTextSize(2);
    tft.setTextColor(BLACK);
    tft.setCursor(10, 60);
    tft.print("A mix of Arabica and Robusta is
the most popular blend that combines the
best of both. Mixed in a ratio of 1: 1. ");
}
```

```
// Displays Confirm button as not selected
void ConfirmOFF() {
    tft.fillRoundRect(30, 275, 180, 40, 180,
BLACK);
    tft.setTextSize(3);
    tft.setTextColor(WHITE);
    tft.setCursor(55, 285);
    tft.print("Confirm");
}
```

```
// Displays Confirm button as selected
void ConfirmON() {
    tft.fillRoundRect(30, 275, 180, 40, 180,
WHITE);
    tft.setTextSize(3);
    tft.setTextColor(BLACK);
    tft.setCursor(55, 285);
    tft.print("Confirm");
}
```

```
// Displays Confirm button as not selected
void CancelOFF() {
    tft.fillRoundRect(30, 225, 180, 40, 180, RED);
    tft.setTextSize(3);
    tft.setTextColor(WHITE);
    tft.setCursor(70, 235);
    tft.print("Cancel");
}
```

```
// Displays Confirm button as selected
void CancelON() {
```

```
tft.fillRoundRect(30, 225, 180, 40, 180,
WHITE);
tft.setTextSize(3);
tft.setTextColor(RED);
tft.setCursor(70, 235);
tft.print("Cancel");
}

// Displays Volume selection as not selected
according to current page
void VolumeOFF() {
    tft.setTextSize(4);
    tft.setTextColor(BLACK);
    tft.setCursor(90, 180);

    switch(curr_page) {
        case 0:
            tft.fillRoundRect(50, 180, 195, 40, 195,
DARKORANGE);
            tft.print(volume * 25);
            break;

        case 1:
            tft.fillRoundRect(50, 180, 195, 40, 195,
SILVER);
            tft.print(volume * 25);
            break;

        case 2:
            tft.fillRoundRect(50, 180, 195, 40, 195,
LIME);
```

```

        tft.print(volume * 25);
        break;

    case 3:
        tft.fillRoundRect(50, 180, 195, 40, 195,
BLUE);
        tft.print(volume * 50);
        break;
    }
    tft.print("g");
    tft.fillTriangle(40, 205, 40, 175, 15, 190,
RED);
    tft.fillTriangle(200, 205, 200, 175, 225, 190,
BLACK);
    }

// Displays Volume selection as selected
according to current page
void VolumeON() {
    tft.setTextSize(4);
    tft.setTextColor(WHITE);
    tft.setCursor(90, 180);

    switch(curr_page) {
        case 0:
            tft.fillRoundRect(50, 180, 195, 40, 195,
DARKORANGE);
            tft.print(volume * 25);
            break;

        case 1:

```

```

        tft.fillRect(50, 180, 195, 40, 195,
SILVER);
        tft.print(volume * 25);
        break;

    case 2:
        tft.fillRect(50, 180, 195, 40, 195,
LIME);
        tft.print(volume * 25);
        break;

    case 3:
        tft.fillRect(50, 180, 195, 40, 195,
BLUE);
        tft.print(volume * 50);
        break;
}

tft.print("g");
tft.fillTriangle(40, 205, 40, 175, 15, 190,
RED);
tft.fillTriangle(200, 205, 200, 175, 225, 190,
BLACK);
}

// Displays arrows that show possibility to move
up and down
void UpDown() {
    tft.fillTriangle(210, 170, 230, 170, 220, 140,
BLACK);
    tft.fillTriangle(210, 240, 230, 240, 220, 270,

```



```
BLACK);
```

```
}
```

```
// Ligts up green light
```

```
void green_light() {
```

```
    digitalWrite(red_pin, LOW);
```

```
    digitalWrite(green_pin, HIGH);
```

```
    digitalWrite(blue_pin, LOW);
```

```
}
```

```
// Ligts up red light
```

```
void red_light() {
```

```
    digitalWrite(red_pin, HIGH);
```

```
    digitalWrite(green_pin, LOW);
```

```
    digitalWrite(blue_pin, LOW);
```

```
}
```

```
// Ligts up white light
```

```
void white_light() {
```

```
    digitalWrite(red_pin, HIGH);
```

```
    digitalWrite(green_pin, HIGH);
```

```
    digitalWrite(blue_pin, HIGH);
```

```
}
```