

به نام خدا

گزارش کار آزمایش هفتم ریزپردازنده

گروه 7

آریان بوکانی 9731012

وسایل استفاده شده در این آزمایش:

- Arduino Mega 2560
- LCD 16*2
- Keypad
- Servo Motor
- Button

در این آزمایش قصد داریم یعقوب برقی را شبیه سازی کنیم.

برای هندل کردن LCD و منوهای مربوط به محصولات دستگاه از کتابخانه MenuBackend.h استفاده می کنیم. برای استفاده از این کتابخانه باید ابتدا یک شی از این کلاس بسازیم و به آن دو تابع پاس دهیم. تابع اولی وقتی صدا زده می شود که یک آیتم در منو استفاده یا به اصطلاح use شود. همچنین تابع پاس شده دومی وقتی اجرا می شود که تغییری در آیتم حال حاضر داده شود.

```
MenuBackend menu = MenuBackend (menuUseEvent, menuChangeEvent);
```

هر قسمت متفاوت در این منو یک MenuItem است. برای ساخت چنین شی ای باید MenuItem() را صدا زده که سازندهی کلاس MenuItem است. و به عنوان ورودی به آن نام شی را پاس می دهیم.

```
MenuItem hotdog = MenuItem ("Hotdog");
```

در ادامه تابع هایی که از این کلاس مورد استفاده قرار گرفته اند آورده شده اند:

1.item.add(menuitem):

این تابع، شی item را به شی menuitem که یک MenuItem است اضافه می کند. در واقع منوی جدیدی را به menuitem اضافه می کند. در این حالت می توان از توابع menuitem.moveDown() و menuitem.moveUp() برای جابه جا شدن بین منوها استفاده کرد.

2.item.addRight(menuitem):

این تابع شی item را به menuitem در همان منو اضافه می کند. در این حالت می توان از تابع menuitem.moveRight() و menuitem.moveLeft() برای جابه جا شدن بین این آیتم ها استفاده کرد.

3.item.getCurrent():

در این قسمت menu یک شی از MenuBackend است و با صدا زدن تابع getCurrent، منویی که در حال حاضر در آن قرار دارد را برمی گرداند و بسته به نیاز می توان مثلا تابع getName را روی آن صدا زد.

پی‌نوشت:

ورودی دو تابع پاس داده شده به سازنده‌ی MenuBackend، یعنی menuChangeEvent و menuUseEvent، به ترتیب MenuChangeEvent و MenuUseEvent هستند. که می‌توان از آنها برای یافتن آیتم مربوط به هر کدام از event ها استفاده کرد.

همچنین در این آزمایش برای استفاده مناسب از کلیدها و گرفتن نویزهای اضافی از کتابخانه‌ی Bounce2.h استفاده کرده‌ایم. ابتدا چهار instance از کلاس Bounce گرفته‌ایم. سپس توسط تابع attach پایه‌ای که می‌خواهیم کلید با آن در ارتباط باشد را به این شی ارتباط می‌دهیم. پس از این مرحله توسط تابع interval زمان عملیات Debouncing را برای هر شی Bounce مشخص می‌کنیم. توسط تابع update می‌توان وضعیت دکمه را دریافت کرد. اگر وضعیت آن تغییر کرده بود مقدار true و در غیر این صورت false برمی‌گرداند. توسط تابع fell می‌توان دریافت که آیا کلید از حالت HIGH به LOW تغییر کرده است یا خیر. اگر جواب مثبت باشد دارای خروجی true و در غیر این صورت false است.

کد:

```
#include <Servo.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <MenuBackend.h>
#include <Bounce2.h>

int pressedButton = 0;
String numberOfProducts = "";
bool wantToSubmit = false, pay = false;
char key = ' ';

Bounce esc = Bounce ();
```

```
Bounce enter = Bounce ();
```

```
Bounce left = Bounce ();
```

```
Bounce right = Bounce ();
```

```
const byte ROWS = 4;
```

```
const byte COLS = 3;
```

```
const int rightKey = 23, leftKey = 22, enterKey = 24, escKey = 25;
```

```
char keys[ROWS][COLS] = {
```

```
{'1','2','3'},
```

```
{'4','5','6'},
```

```
{'7','8','9'},
```

```
{'*', '0', '#'}  
};
```

```
byte rowPins[ROWS] = { A11, A10, A9, A8 };
```

```
byte colPins[COLS] = { A12, A13, A14};
```

```
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

```
LiquidCrystal lcd(12, 13, 11, 10, 9, 8);
```

```
Servo servos[12];
```

```
int getPrice(String product){
```

```
    if(product == "Coffee")return 5;
```

```
    if(product == "Tea")return 2;
```

```
if(product == "Mocha")return 5;
if(product == "Orange Juice")return 4;
if(product == "Apple Juice")return 4;
if(product == "Cherry Juice")return 4;
if(product == "Hotdog")return 10;
if(product == "Burger")return 12;
if(product == "Cheeseburger")return 15;
if(product == "Olivier Salad")return 8;
if(product == "Chick Ham")return 10;
if(product == "Meat Ham")return 10;
return 0;
}
```

```
int getPin(String product){
    if(product == "Coffee")return 3;
    if(product == "Tea")return 2;
    if(product == "Mocha")return 1;
    if(product == "Orange Juice")return 0;
    if(product == "Apple Juice")return 7;
    if(product == "Cherry Juice")return 6;
    if(product == "Hotdog")return 5;
    if(product == "Burger")return 4;
    if(product == "Cheeseburger")return 11;
    if(product == "Olivier Salad")return 10;
    if(product == "Chick Ham")return 9;
```

```

    if(product == "Meat Ham")return 8;
}

void menuUseEvent (MenuUseEvent used){
    lcd.clear();
    lcd.home();
    lcd.print(used.item.getName());
    if (getPrice(used.item.getName())) {
        lcd.setCursor(14, 0);
        if(!pay)
            lcd.print(getPrice(used.item.getName()));
        if(wantToSubmit && !pay){
            numberOfProducts = "";
            lcd.setCursor(0, 1);
            lcd.print("Amount:");
        }else if(pay){
            lcd.setCursor(0, 1);
            lcd.print("Total:"+String(numberOfProducts.toInt()
getPrice(used.item.getName())));
            HandleServos();
            pay = false;
            numberOfProducts = "";
        }
    }
}
}

```

*

```

void menuChangeEvent (MenuChangeEvent changed){
    lcd.clear();
    lcd.home();
    MenuItem newMenuItem = changed.to;
    lcd.print(newMenuItem.getName());
    if (getPrice(newMenuItem.getName())) {
        lcd.setCursor(14, 0);
        lcd.print(getPrice(newMenuItem.getName()));
    }
}

MenuBackend menu = MenuBackend (menuUseEvent, menuChangeEvent);
MenuItem drink = MenuItem ("Drink");
    MenuItem hotDrink = MenuItem ("Hot");
        MenuItem coffee = MenuItem ("Coffee");
        MenuItem tea = MenuItem ("Tea");
        MenuItem mocha = MenuItem ("Mocha");
MenuItem coldDrink = MenuItem ("Cold");
    MenuItem orange = MenuItem ("Orange Juice");
    MenuItem apple = MenuItem ("Apple Juice");
    MenuItem cherry = MenuItem ("Cherry Juice");
MenuItem sandwich = MenuItem ("Sandwich");
    MenuItem hotSandwich = MenuItem ("Hot");
        MenuItem hotdog = MenuItem ("Hotdog");

```

```
MenuItem burger = MenuItem ("Burger");
```

```
MenuItem cheeseburger = MenuItem ("Cheeseburger");
```

```
MenuItem coldSandwich = MenuItem ("Cold");
```

```
MenuItem olivier = MenuItem ("Olivier Salad");
```

```
MenuItem chick = MenuItem ("Chick Ham");
```

```
MenuItem meat = MenuItem ("Meat Ham");
```

```
void setup() {
```

```
  lcd.begin(16, 2);
```

```
  pinMode(leftKey, INPUT);
```

```
  pinMode(rightKey, INPUT);
```

```
  pinMode(enterKey, INPUT);
```

```
  pinMode(escKey, INPUT);
```

```
  esc.attach (escKey);
```

```
  enter.attach (enterKey);
```

```
  right.attach (rightKey);
```

```
  left.attach (leftKey);
```

```
  esc.interval (50);
```

```
  enter.interval (50);
```

```
  right.interval (50);
```

```
  left.interval (50);
```



```

menu.getRoot().add (drink);
drink.addRight (sandwich);
drink.add (hotDrink).addRight (coldDrink);
hotDrink.add(coffee).addRight(tea).addRight(mocha);
coldDrink.add(orange).addRight(apple).addRight(cherry);
sandwich.add (coldSandwich).addRight (hotSandwich);
hotSandwich.add(hotdog).addRight(burger).addRight(cheeseburger);
coldSandwich.add(olivier).addRight(chick).addRight(meat);
menu.toRoot ();

```

```

for(int i = 0; i < 12; i++){
    if(i <= 3)
        servos[i].attach(3-i, 1000, 2000);
    else
        servos[i].attach(10+i, 1000, 2000);
    servos[i].write(0);
}

```

```

}

```

```

void loop() {

```

```

    key = kpd.getKey();
    if(wantToSubmit && key != NO_KEY && key != '*' && key != '#'){
        numberOfProducts += key;
    }
}

```

```

    lcd.clear();
    lcd.home();
    lcd.print(menu.getCurrent().getName());
    lcd.setCursor(14, 0);
    lcd.print(getPrice(menu.getCurrent().getName()));
    lcd.setCursor(0, 1);
    lcd.print("Amount:" + String(numberOfProducts.toInt()));
}
buttonEvent();
changeMenu();
}

```

```

void buttonEvent (){
    enter.update ();
    right.update ();
    left.update ();
    esc.update ();
    pressedButton = 0;
    if (left.fell ())
        pressedButton = leftKey;
    else if (right.fell ())
        pressedButton = rightKey;
    else if (enter.fell ())
        pressedButton = enterKey;
    else if (esc.fell ())

```

```

        pressedButton = escKey;
    }

void changeMenu () {
    MenuItem currentMenu = menu.getCurrent();
    if (pressedButton == leftKey){
        menu.moveLeft();
    }else if (pressedButton == rightKey) {
        menu.moveRight();
    }else if (pressedButton == escKey) {
        wantToSubmit = false;
        pay = false;
        menu.toRoot();
    }else if (pressedButton == enterKey) {
        if(!(currentMenu.moveDown ())) {
            if(wantToSubmit){
                pay = true;
            }
            wantToSubmit = !pay;
            menu.use();
        }
        else {
            menu.moveDown();
        }
    }
}

```

```

pressedButton = 0;

}

void HandleServos(){
  for(int i = 0; i < numberOfProducts.toInt(); i++){
    int openTime = millis();
    servos[getPin(menu.getCurrent().getName())].write(180);
    while(millis() - openTime <= 1000);
    servos[getPin(menu.getCurrent().getName())].write(0);
    if( i != numberOfProducts.toInt() - 1)
      while(millis() - openTime <= 2000);
  }
}

```

چند نمونه از اجرا:





