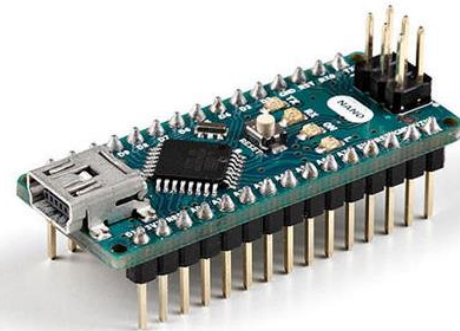


Komponentovo orientované a udalosťami riadené programovanie Arduino zariadení

AUTOR: PATRIK PEKARČÍK

VEDÚCI: RNDR. FRANTIŠEK GALČÍK, PHD.

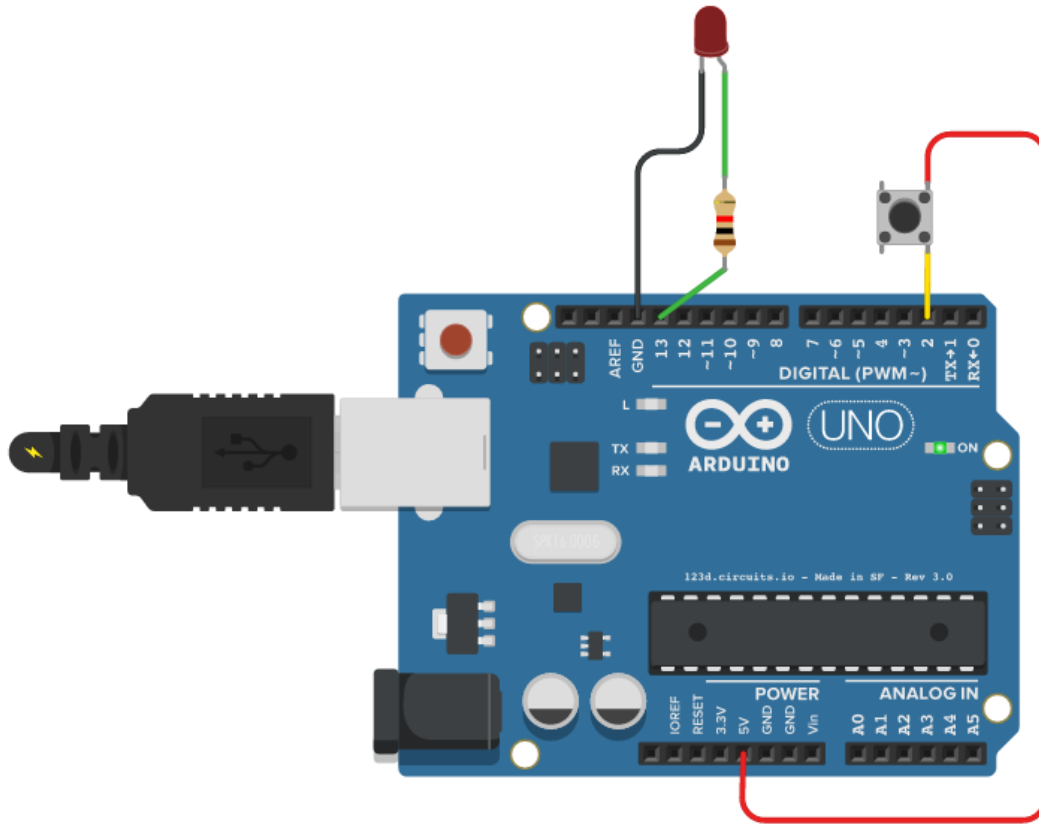
Motivácia



Parametre

	Arduino UNO	Arduino Nano
Microcontroller	ATmega328P	Atmel ATmega168 or ATmega328
Operating Voltage	5V	5 V
Input Voltage (limit)	6-20V	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)	14 (of which 6 provide PWM output)
Analog Input Pins	6	8
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	2 KB (ATmega328P)	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	1 KB (ATmega328P)	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz	16 MHz
Size	68.6 mm * 53.4 mm	45 mm * 18 mm
Weight	25 g	5 g

Demo projekt



Demo projekt

```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;
void setup () {
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT);
}
void loop () {
    if (buttonState == 0) {
        digitalWrite(ledPin, HIGH);
    }
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
    if (digitalRead(buttonPin) == HIGH) {
        buttonState = buttonState == 0 ? 1 : 0;
    }
}
```

Problém?

Po stlačení tlačidla sa nič nedeje!

```
if (buttonState == 0) {  
    digitalWrite(ledPin, HIGH);  
}  
delay(1000);  
digitalWrite(ledPin, LOW);  
delay(1000);  
if (digitalRead(buttonPin) == HIGH) {  
    buttonState = buttonState == 0 ? 1 : 0;  
}
```

Problém?

Po stlačení tlačidla sa nič nedeje!

```
if (buttonState == 0) {  
    digitalWrite(ledPin, HIGH);  
}  
delay(1000);  
digitalWrite(ledPin, LOW);  
delay(1000);  
if (digitalRead(buttonPin) == HIGH) {  
    buttonState = buttonState == 0 ? 1 : 0;  
}
```

delay(1000)

program je uspatý na 1 sekundu a neprijíma žiadne stlačenia.

StudioActivity > app > src > main > res > layout > main_activity.xml

main_activity.xml x

Palette

- Widgets
- TextView
- BUTTON
- OFF
- CheckBox
- RadioButton
- CheckedTextView
- Home

Component Tree

- android.support.constraint.CoordinatorLayout
- imageViewHeader
- android.support.design.widget.FloatingActionButton
- imageButtonCall
- imageButtonChat
- textViewPhone - @string/...
- textViewPhoneLabel - @string/...
- imageButtonHangout
- textViewPhone2 - @string/...
- textViewPhone2Label - @string/...
- LineDivider (View)
- imageButtonEmail
- textViewEmail - @string/e...
- textViewEmailLabel - @str...

Design | Text

1:1 n/a n/a Context: <no context>

Android Studio

Android Studio

(650) 555-1234
Google Voice

(650) 555-4321
Mobile

studio@android.com
Work

ImageView

phone_number
type_of_phone_number

phone_number
type_of_phone_number

studio@android.com
Work

Properties

ID: FloatingActionButton

layout_width: 56dp

layout_height: 56dp

FloatingActionButton

srcCompat: jade_black_24dp

backgroundTint: ttr/colorAccent

rippleColor: ontrolHighlight

tint: old_color/white

fabSize: normal

elevation: 7dp

[View all properties](#)


```
public class FooPanel extends JPanel implements
ActionListener {
    public FooPanel() {
        super();
        JButton btn = new JButton("Click Me!");
        btn.addActionListener(this);
        this.add(btn);
    }
    @Override
    public void actionPerformed(ActionEvent ae) {
        System.out.println("Button has been
clicked!");
    }
}
```

Komponent

Spracovanie udalosti

Naša vizioa - komponenty

Časovač

Názov: **blinkTimer**

Interval: **1000**

OnTick: **changeLed()**

Tlačidlo

Názov: **button**

Pin: **2**

OnClick: **buttonClick()**

Prepínač

Názov: **led**

Pin: **13**

```
boolean blika = true;
void buttonClick() {
    blika = !blika;
}
void changeLed() {
    if(blika) {
        led.revert();
    }
}
```

Podobné riešenia



- github.com/igormiktor/arduino-EventManager

Ciele práce

1. Preskúmať, analyzovať a porovnať existujúce prístupy, softvérové aplikácie a knižnice využívané pri programovaní Arduino zariadení
2. Preskúmať a analyzovať možnosti komponentového a udalosťami riadeného programovania s ohľadom na hardvérové obmedzenia Arduino zariadení

Ciele práce

3. Vychádzajúc z existujúcich open-source projektov a knižníc navrhnuť a implementovať užívateľsky prívetivé riešenie na jednoduché komponentovo-orientované a udalosťami riadené programovanie Arduino zariadení
4. Implementovať vzorové komponenty využiteľné pri návrhu a implementácii IoT riešení

Literatúra

1. Doukas, C. (2012) **Building Internet of Things with the Arduino**. CreateSpace Independent Publishing Platform, ISBN: 978-1470023430
2. Schwartz, M. (2016) **Internet of Things with Arduino Cookbook**. Packt Publishing, ISBN: 978-1785286582
3. Waher, P. (2015) **Learning Internet of Things**. Packt Publishing, ISBN 978-1783553532.



Ďakujem za pozornosť!