## Komponentovo orientované a udalosťami riadené programovanie Arduino zariadení

AUTOR: PATRIK PEKARČÍK

VEDÚCI: RNDR. FRANTIŠEK GALČÍK, PHD.

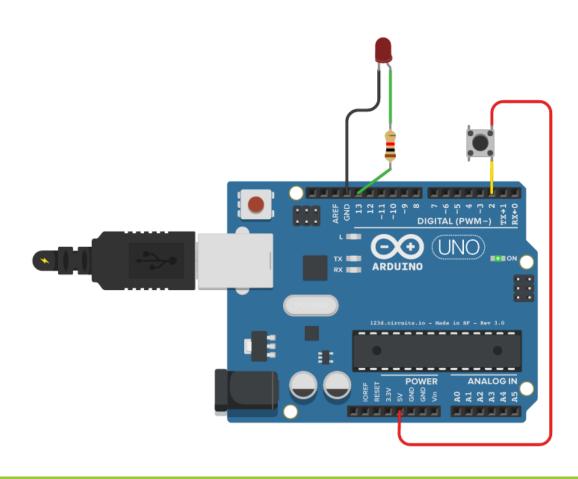


# Motivácia



Parametre	Arduino UNO	Arduino Nano	
Microcontroller	ATmega328P	Atmel ATmega168 or ATmega328	
<b>Operating Voltage</b>	5V	5 V	
Input Voltage (limit)	6-20V	6-20 V	
Digital I/O Pins	14 (of which 6 provide PWM	14 (of which 6 provide PWM	
	output)	output)	
Analog Input Pins	6	8	
Flash Memory	<b>32 KB</b> (ATmega328P) of which 0.5 KB used by bootloader	<b>16 KB</b> (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	
Price	\$ 2.00	\$ 2.00	

# 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;
```

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

# Náš spôsob - komponenty

#### Časovač

Názov: **blinkTimer** 

Interval: 1000

OnTick: changeLed()

### Prepínač

Názov: led

Pin: **13** 

#### Tlačidlo

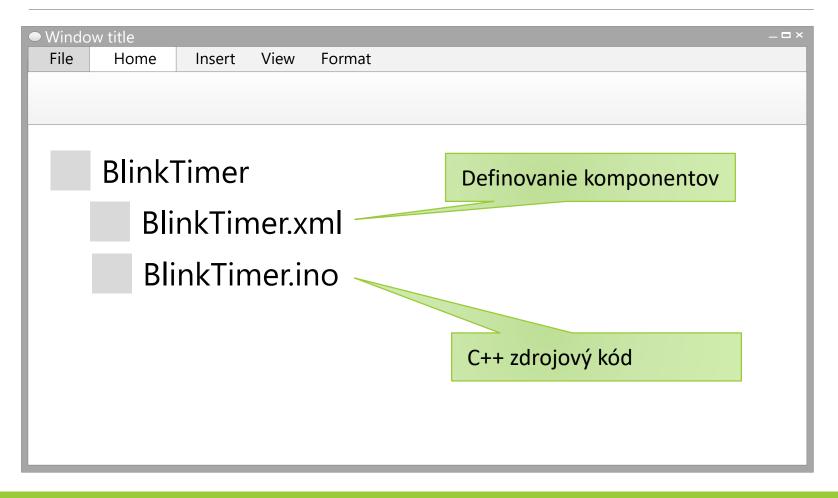
Názov: **button** 

Pin: **2** 

OnClick: buttonClick()

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

# Štruktúra projektu



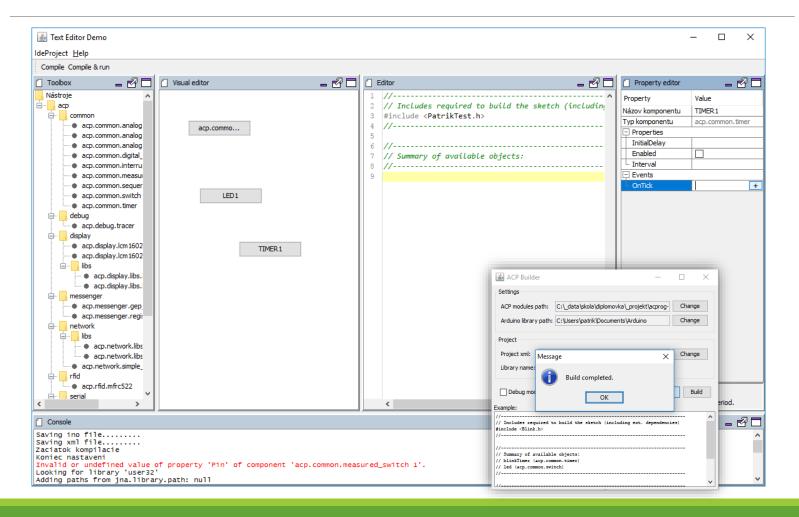
### BlinkTimer.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
project platform="ArduinoUno">
 <components>
   <component>
     <name>blinkTimer
     <type>acp.common.timer</type>
     cproperties>
       cproperty name="Enabled">true</property>
       cproperty name="Interval">1000</property>
     </properties>
     <events>
       <event name="OnTick">onBlink</event>
     </events>
   </component>
 </components>
 <ide>...</ide>
```

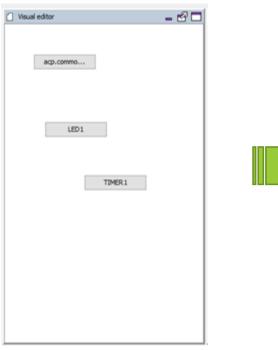
### BlinkTimer.ino

```
// Includes required to build the sketch (including ext. dependencies)
#include <Blink.h>
//----
//----
// Summary of available objects:
// blinkTimer (acp.common.timer)
// led (acp.common.switch)
//----
//----
// Event callback for blinkTimer.OnTick
void onBlink() {
 led.revert();
```

### **IDE** Editor



# IDE Editor - zlepšenie





Palette ×			
Swing Contain	ers		
Panel	Tabbed Pane	Split Pane	Scroll Pane
Tool Bar	E Desktop Pane	Internal Frame	Layered Pane
─ Swing Controls	i		
label Label	OK Button	ON Toggle Button	
●— Radio Button	🔭 Button Group	Combo Box	List
Text Field	tx Text Area	Scroll Bar	
Progress Bar	7-7 Formatted Field	··· Password Field	Spinner
Separator	T Text Pane	Editor Pane	Tree
Table			
Swing Menus			
Swing Window	s		
Swing Fillers ∧			_
Glue Glue		Horizontal Strut	🛱 Rigid Area
Vertical Glue			
■ AWT			
A Label	OK Button	ab Text Field	Text Area
Checkbox	- Choice	E List	<b>₫₫</b> Scrollbar
Scroll Pane	Panel	Canvas	Menu Bar
Popup Menu			
■ Beans			
Choose Bean			
<ul> <li>Java Persisten</li> </ul>	ice		
Entity Manager	Query	Query Result	

# Ciele práce

- Preskúmať, analyzovať a porovnať existujúce prístupy, softvérové aplikácie a knižnice využívané pri programovaní Arduino zariadení
- 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 navrhnúť a implementovať uživateľsky prívetivé riešenie na jednoduché komponentovo-orientované a udalosťami riadené programovanie Arduino zariadení
- 4. Implementovať vzorové komponenty využiteľné pri návrhu a implementácii IoT riešení

### Literatúra

- 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!