

Komponentovo orientované a udalosťami riadené programovanie Arduino zariadení

AUTOR: PATRIK PEKARČÍK

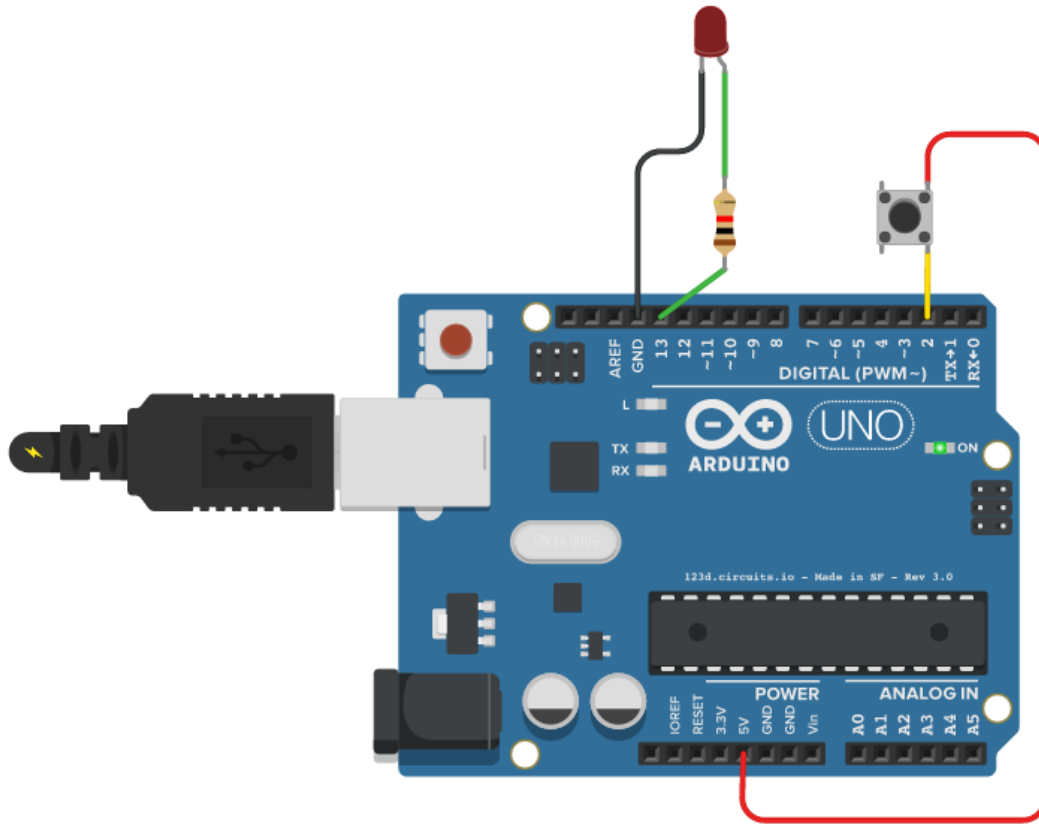
VEDÚCI: RNDR. FRANTIŠEK GALČÍK, PHD.

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
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();
        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

Náš spôsob - komponenty

Časovač

Názov: **blinkTimer**

Interval: **1000**

OnTick: **changeLed()**

Tlačidlo

Názov: **button**

Pin: **2**

OnClick: **buttonClick()**

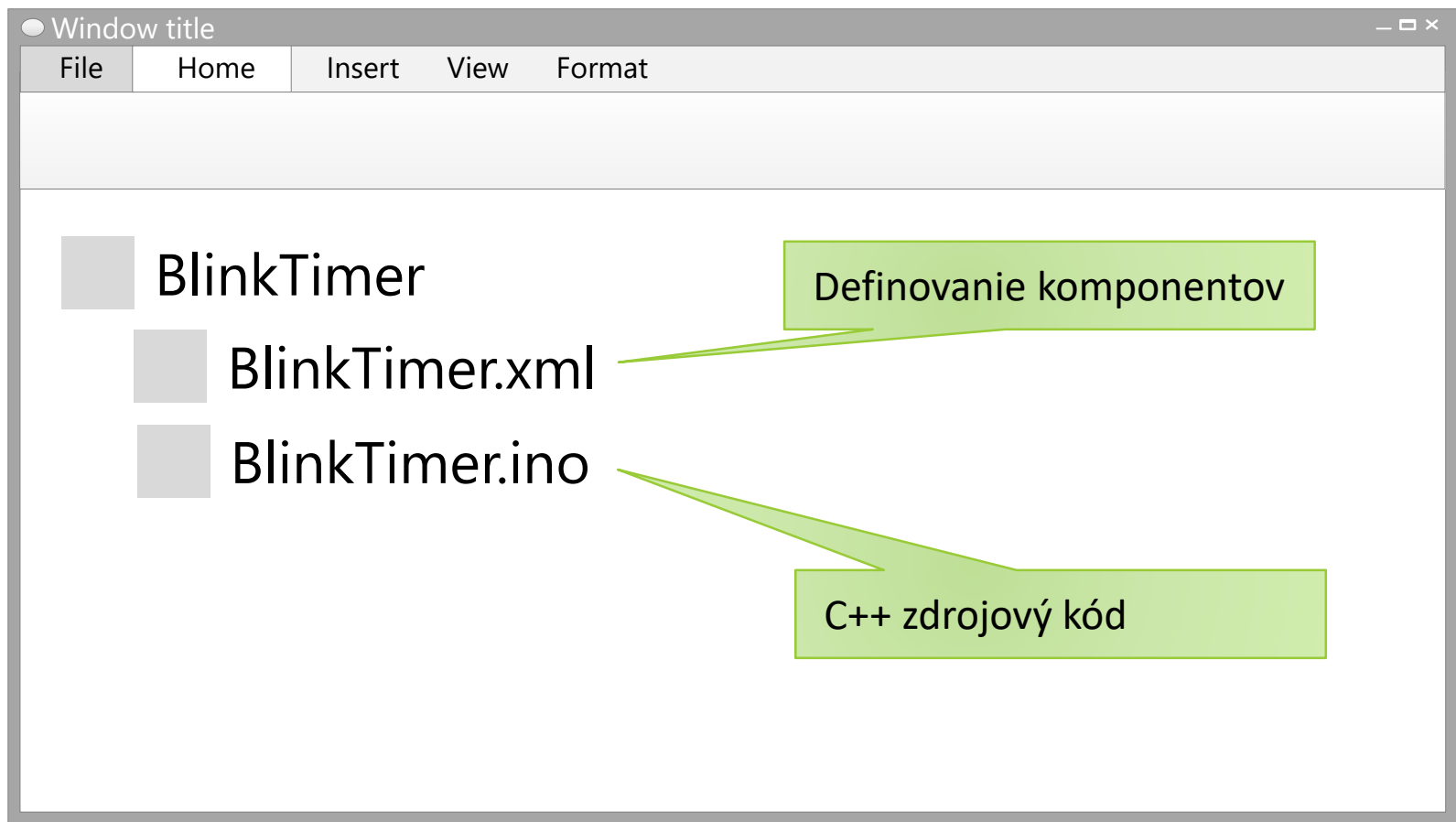
Prepínač

Názov: **led**

Pin: **13**

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

Štruktúra projektu



BlinkTimer.xml

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


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();  
}
```

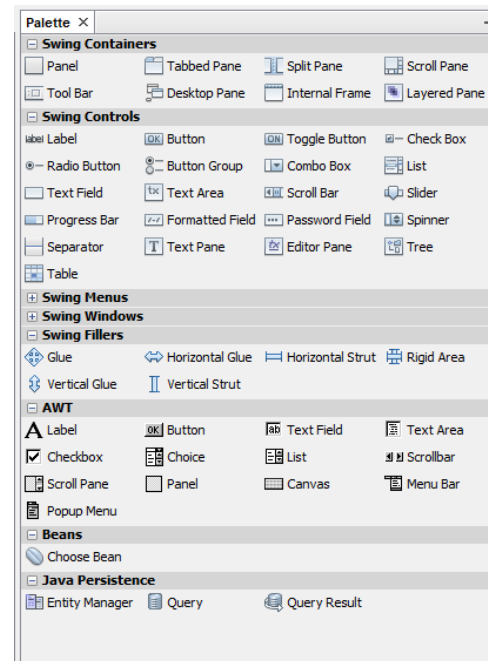
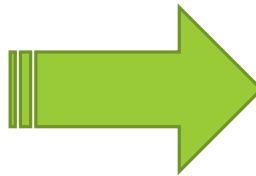
IDE Editor

The screenshot displays an IDE interface with several panels:

- Toolbox:** A tree view of components under 'Nástroje' and 'acp'. The 'acp' folder contains sub-folders like 'common', 'debug', 'display', 'libs', 'messenger', 'network', 'rfd', and 'serial'. Each folder contains specific component files.
- Visual editor:** A workspace containing three components: 'acp.commo...', 'LED1', and 'TIMER1'.
- Editor:** A code editor showing C++ code for a sketch. The code includes headers and a summary of available objects. Line 9 is highlighted in yellow.
- Property editor:** A table showing properties for the 'TIMER1' component.

Property	Value
Názov komponentu	TIMER1
Typ komponentu	acp.common.timer
InitialDelay	
Enabled	<input type="checkbox"/>
Interval	
OnTick	<input type="button" value="+"/>
- Console:** A window showing the output of the compilation process, including error messages: 'Invalid or undefined value of property 'Pin' of component 'acp.common.measured_switch 1'. Looking for library 'user32'' and 'Adding paths from jna.library.path: null'.
- ACP Builder:** A dialog box with 'Settings' and 'Project' sections. It includes fields for 'ACP modules path', 'Arduino library path', 'Project xml', and 'Library name'. A 'Build' button is visible.
- Message:** A small dialog box with an information icon and the text 'Build completed.' and an 'OK' button.

IDE Editor - zlepšenie



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!