

# Komponentovo orientované a udalosťami riadené programovanie Arduino zariadení

## Úvod

*IoT* alebo „internet vecí“ je pojem resp. téma, o ktorej dnes počujeme zo všetkých médií. Vývoju *IoT* zariadení sa venujú známe spoločnosti, ako napríklad firma *Philips*, ktorá medzi inými ponúka aj inteligentné osvetlenie. Popularizácii *IoT* pomohol aj príchod zariadení *Arduino*. *Arduino* je open-source platforma s mikrokontrolerom *ATmega*, ktorá vývojárom neponúkla iba hardvér s procesorom, ale aj pomerne jednoduché vývojové prostredie *Arduino IDE*. Vývoj na platforme *Arduino* je založený na programovacom jazyku *C++*.

Program pre platformu *Arduino* sa skladá z dvoch základných funkcií:

- **setup ()** – funkcia spustená iba raz po zapnutí zariadenia
- **loop ()** – periodicky spúšťaná funkcia, pokiaľ je zariadenie zapnuté

Tento prístup nepodporuje efektívny multitasking, na aký sú programátori zvyknutí z programovania pre operačný systém. K rozšíreniu týchto zariadení prispela aj ich nízka cena, ktorá sa pohybuje od niekoľkých dolárov. Cena zariadenia *Arduino Nano* sa pohybuje okolo 2 dolárov, za čo dostaneme úložný priestor *FLASH* 32 kB a operačnú pamäť *SRAM* 2048 B. Obmedzená veľkosť operačnej pamäte nám zatvára dvere pred použitím komplexnejšieho operačného systému, čo však nevyklučuje vytvorenie minimalistického plánovača úloh pre toto zariadenie. Pri návrhu možného fungovania sa chceme inšpirovať komponentovo orientovaným prístupom, aký poznáme z vývoja používateľských aplikácií v operačných systémoch (napr. *Swing* v jazyku *Java*).

## Postup práce

Naším prvým krokom je vývoj plánovača úloh pre komponenty, ktorý bude vedieť efektívne a v reálnom čase vykonávať udalosti vytvorené komponentami. Znova si musíme uvedomiť, že máme k dispozícii iba 2048 B operačnej pamäte. Ďalším faktom je, že čím viac pamäte spotrebuje plánovač úloh, tým menej komponentov budeme môcť použiť. To znamená, že v prípade, ak by sme využili celú pamäť pre plánovač, tak vieme iba plánovať a nemôžeme urobiť nič viac. Preto sa pri návrhu zameriame na využitie čo najmenej časti tejto operačnej pamäte.

Spolu s vývojom plánovača vytvoríme aj základné komponenty:

- **Časovač** – komponent, ktorý bude v zadanom intervale spúšťať preddefinované udalosti
- **Prepínač** – komponent, ktorý bude zapínať alebo vypínať port zo zariadenia, napríklad zapne/vypne *Led* diódu
- **Senzor** – komponent, ktorý bude sledovať port zariadenia, v ktorom je pripojený senzor: ak sa na senzore zmení hodnota, tak spustí definovanú udalosť (napríklad po stlačení tlačidla sa vykoná zapnutie Prepínača)

Pri tvorbe musíme - rovnako ako pri tvorbe plánovača úloh - dbať na veľkosť operačnej pamäte (2048 B).

To, čo otvorilo dvere zariadeniam *Arduino*, bolo práve vývojové prostredie *Arduino IDE*. Preto sa v ďalšej kapitole budeme venovať tvorbe vývojového prostredia podobného tomu, ktoré sme použili pri vývoji *Swing* aplikácií. V našom prostredí bude používateľ kliknutím vytvárať jednotlivé komponenty, ktorým nastaví jednotlivé parametre. Z týchto dát sa automaticky vygeneruje zdrojový kód pre plánovač úloh. Pri návrhu vývojového prostredia a generátora kódu musíme dbať na budúcu rozšíriteľnosť dostupných typov komponentov. V tejto kapitole sa prevažne budeme venovať

- analýze zdrojového kódu z pohľadu párovania parametrov z komponentov ku zdrojovým kódom
- analýze kódu z pohľadu hľadania kolízií, kde bude viac komponentov súčasne používať tie isté prostriedky
- problematike kompilátora pre tieto zariadenia

Posledným krokom bude implementácia ďalších bežne používaných komponentov, kde pri vývoji sa budeme znova riadiť všetkými už spomenutými podmienkami.

## 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

- (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.