

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



## **Sada demo aplikací nad FreeRTOS**

MIKROPROCESOROVÉ A VESTAVĚNÉ SYSTÉMY

2021/2022

AUTOR:

TOMÁŠ ČECHVALA (XCECHV03)

# Obsah

<b>1</b>	<b>Zadanie</b>	<b>2</b>
<b>2</b>	<b>Návod</b>	<b>2</b>
<b>3</b>	<b>Popis riešenia zadania</b>	<b>2</b>
3.1	Program a jeho implementácia . . . . .	2
3.2	TaskBlikaj . . . . .	2
3.3	TaskTouch . . . . .	2
3.4	Priebeh programu . . . . .	2
<b>4</b>	<b>Splnenie podmienok zadania</b>	<b>3</b>
4.1	Riadenie vstupných / výstupných periférií a využitie samostatných taskov . . . . .	3
4.2	Použitie komunikačnej alebo synchronizačnej služby . . . . .	3
4.3	Použitie aspoň jednej služby FreeRTOS pre prácu s časom . . . . .	3
4.4	Interakcia s užívateľom pomocou termináloveho okna . . . . .	3
<b>5</b>	<b>Prílohy</b>	<b>4</b>
<b>6</b>	<b>Zdroje</b>	<b>6</b>

# 1 Zadanie

Zadaním projektu bolo vytvoriť jednoduchý (bare-metal) projekt umožňujúci a demonštrujúci riadenie základných periférií mikrokontrolerom na technickom vybavení ESP32 z úrovne úloh FreeRTOS. Pre obsluhu každej periférie bolo potrebné využiť samostatnú úlohu (task), použiť alespoň jednu komunikačnú alebo synchronizačnú službu FreeRTOS (napríklad fronta, semafor) a alespoň jednu službu FreeRTOS pre prácu s časom, napríklad `vTaskDelay()`, pre interakciu s užívateľom je taktiež možné využiť terminálové okno.

## 2 Návod

Na spustenie projektu na zariadení ESP32 je potrebný micro usb kábel a následujúce softwarové produkty: operačný systém Windows 10/Linux a aplikácia Arduino IDE. Pomocou aplikácie Arduino IDE dokážeme preložiť a nahráť kód do zariadenia esp32. Pre správne fungovanie Arduino IDE odporúčam pozrieť tento návod [1, Installing using Boards Manager]. Po vykonaní inštalácie je potrebné nainštalovať ESP32 platformu a v board menu vybrať **WEMOS D1 MINI ESP32** a reštartovať Arduino IDE. Pomocou Arduino IDE preložíme kód a nahráme ho na dosku ESP32.

## 3 Popis riešenia zadania

V nasledujúcich sekciách je popísané riešenie projektu a implementácia daných funkcií. Celá implementácia projektu sa nachádza v jednom súbore `xcechv03.ino`. K vypracovaniu bola použitá schéma ESP32 a taktiež prezentácia venovaná ESP32, ktoré sa nachádzajú v školských materiáloch [2][3]. Na nájdenie potrebných informácií ako narábať s jednotlivými funkciami som využil oficiálnu dokumentáciu pre Arduino<sup>1</sup>.

### 3.1 Program a jeho implementácia

Funkčnosť programu spočíva v dvoch taskoch, ktoré som implementoval - `TaskBlikaj` a `TaskTouch`.

### 3.2 TaskBlikaj

`TaskBlikaj` demonštruje funkčnosť výstupnej periférie pomocou LED žiarovky.

### 3.3 TaskTouch

`TaskTouch` demonštruje funkčnosť vstupnej periférie pomocou senzoru na dotyk prstom na zariadení ESP32 a taktiež aj funkčnosť výstupnej periférie pomocou LED žiarovky.

### 3.4 Priebeh programu

Po spustení programu začne svietiť LED žiarovka na 3 sekundy následne zhasne na 2.5 sekundy a na terminálovom okne, ktoré máme otvorené môžeme vidieť počítadlo, ktoré ráta koľkokrát sa LED žiarovka rozsvieti pomocou `taskBlikaj`. Následne sa celý proces zapnutia a vypnutia LED zopakuje.

---

<sup>1</sup><https://www.arduino.cc/reference/en/>

Počas toho ako beží TaskBlikaj prebieha aj TaskTouch ten zaznamenáva každú sekundu elektrické zmeny na vybranom GPIO 4 pine (TOUCH0) a vypisuje ich v terminálovom okne. Akonáhle príde k dotyku senzoru prstom, príde k zmene hodnoty, keďže náš prst obsahuje statickú elektrinu. Preto sa tieto senzory nazývajú aj kapacitné snímače. Hodnoty funkcie TaskTouch sú zapisované do premennej touch\_sensor\_value. Ak program zistí, že sa dotýkame senzoru, pozastaví sa TaskBlikaj a každú sekundu pokiaľ sa dotýkame senzoru LED žiarovka dvakrát zabliká. Akonáhle odstránime prst zo snímača. Obnoví sa TaskBlikaj a program sa vráti do pôvodného stavu, počas ktorého bežia oba tasky.

## **4 Splnenie podmienok zadania**

### **4.1 Riadenie vstupných / výstupných periférií a využitie samostatných taskov**

Riadenie periférií je implementované v samostatných taskoch TaskBlikaj a TaskTouch pomocou LED a touch senzoru.

### **4.2 Použitie komunikačnej alebo synchronizačnej služby**

Na implementáciu tohto bodu zadania nám slúžia funkcie vTaskSuspend() a vTaskResume(). Funkcia vTaskSuspend() má za úlohu pozastaviť TaskBlikaj akonáhle zaznamená touch senzor dotyk prstu pomocou tasku TaskTouch. Na prístup k TaskBlikaj, slúži task1\_handle, ktorý môžeme použiť ako parameter funkcie vTaskSuspend() na pozastavenie TaskBlikaj. Funkcia vTaskResume() taktiež využíva parameter task\_handle na prístup k tasku TaskBlikaj. Akonáhle sa prestaneme dotýkať prstom touch senzoru, funkcia vTaskResume() obnoví fungovanie TaskBlikaj.

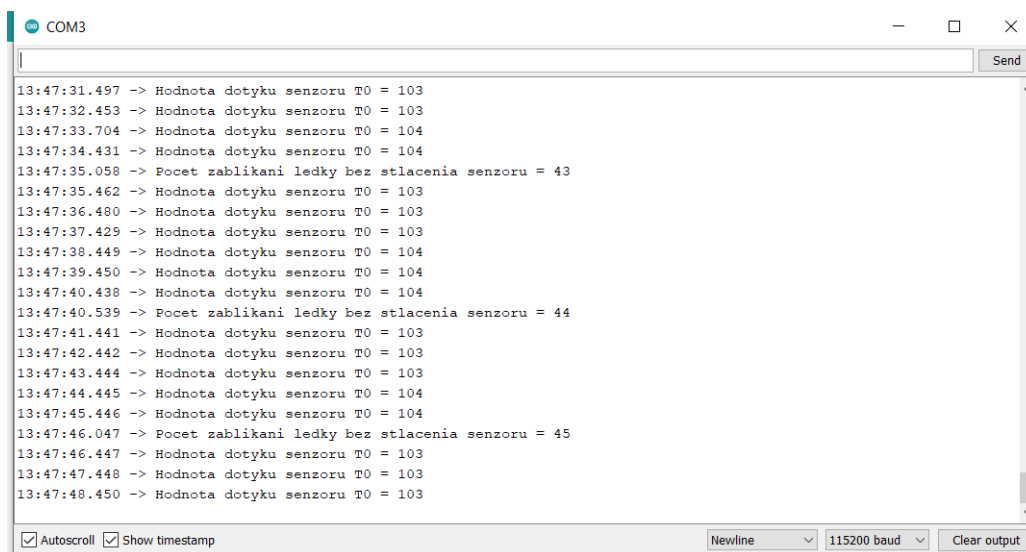
### **4.3 Použitie aspoň jednej služby FreeRTOS pre prácu s časom**

Na splnenie tejto požiadavky využívam funkciu vTaskDelay(), ktorá slúži na vytvorenie časových intervalov medzi jednotlivými taskami, bliknutiami LED žiarovky a výpisu textu v terminálovom okne.

### **4.4 Interakcia s užívateľom pomocou termináloveho okna**

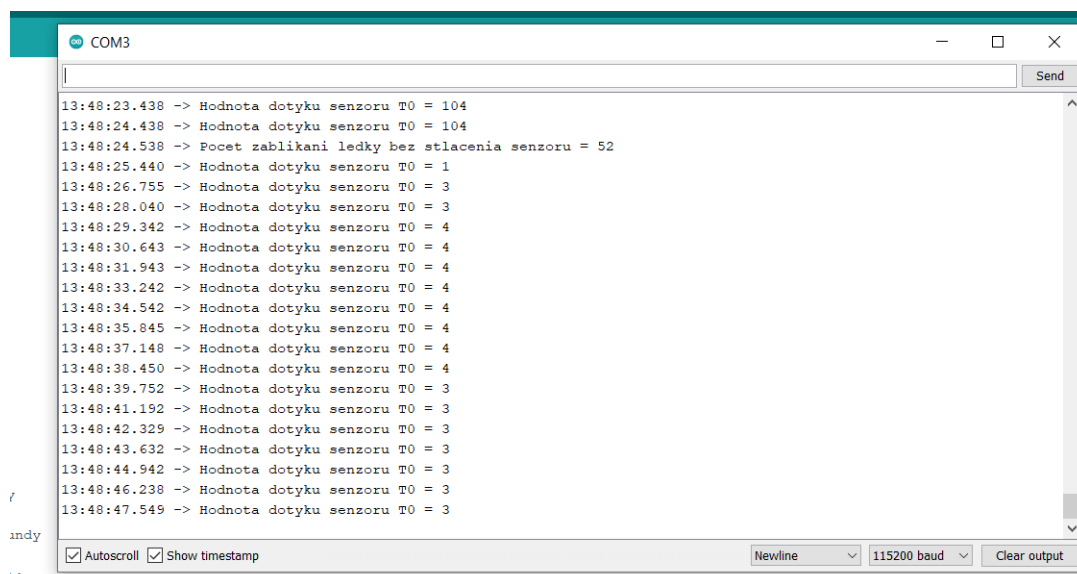
Projekt spĺňa aj túto funkcionality, v terminálovom okne vypisuje TaskBlikaj počet bliknutí, ktoré vykoná LED žiarovka pomocou tohto tasku. TaskTouch vypisuje v terminálovom okne každú sekundu hodnotu, ktorú nameral touch senzor.

## 5 Prílohy



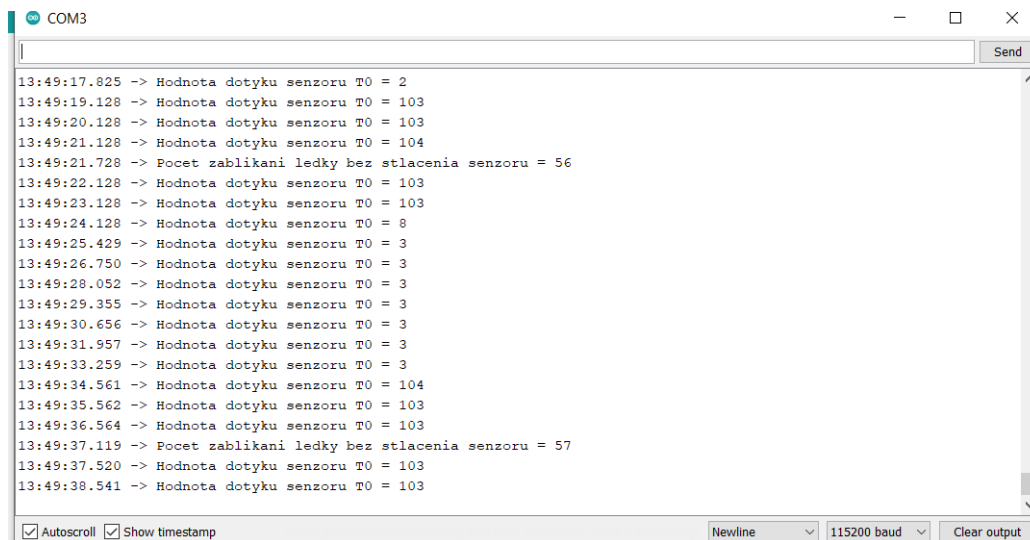
```
13:47:31.497 -> Hodnota dotyku senzoru T0 = 103
13:47:32.453 -> Hodnota dotyku senzoru T0 = 103
13:47:33.704 -> Hodnota dotyku senzoru T0 = 104
13:47:34.431 -> Hodnota dotyku senzoru T0 = 104
13:47:35.058 -> Pocet zablikani ledky bez stlacenia senzoru = 43
13:47:35.462 -> Hodnota dotyku senzoru T0 = 103
13:47:36.480 -> Hodnota dotyku senzoru T0 = 103
13:47:37.429 -> Hodnota dotyku senzoru T0 = 103
13:47:38.449 -> Hodnota dotyku senzoru T0 = 104
13:47:39.450 -> Hodnota dotyku senzoru T0 = 104
13:47:40.438 -> Hodnota dotyku senzoru T0 = 104
13:47:40.539 -> Pocet zablikani ledky bez stlacenia senzoru = 44
13:47:41.441 -> Hodnota dotyku senzoru T0 = 103
13:47:42.442 -> Hodnota dotyku senzoru T0 = 103
13:47:43.444 -> Hodnota dotyku senzoru T0 = 103
13:47:44.445 -> Hodnota dotyku senzoru T0 = 104
13:47:45.446 -> Hodnota dotyku senzoru T0 = 104
13:47:46.047 -> Pocet zablikani ledky bez stlacenia senzoru = 45
13:47:46.447 -> Hodnota dotyku senzoru T0 = 103
13:47:47.448 -> Hodnota dotyku senzoru T0 = 103
13:47:48.450 -> Hodnota dotyku senzoru T0 = 103
```

Obrázek 1: Výpis v terminálovom okne v stave kedy sa nedotýkam senzoru a bežia oba tasky



```
13:48:23.438 -> Hodnota dotyku senzoru T0 = 104
13:48:24.438 -> Hodnota dotyku senzoru T0 = 104
13:48:24.538 -> Pocet zablikani ledky bez stlacenia senzoru = 52
13:48:25.440 -> Hodnota dotyku senzoru T0 = 1
13:48:26.755 -> Hodnota dotyku senzoru T0 = 3
13:48:28.040 -> Hodnota dotyku senzoru T0 = 3
13:48:29.342 -> Hodnota dotyku senzoru T0 = 4
13:48:30.643 -> Hodnota dotyku senzoru T0 = 4
13:48:31.943 -> Hodnota dotyku senzoru T0 = 4
13:48:33.242 -> Hodnota dotyku senzoru T0 = 4
13:48:34.542 -> Hodnota dotyku senzoru T0 = 4
13:48:35.845 -> Hodnota dotyku senzoru T0 = 4
13:48:37.148 -> Hodnota dotyku senzoru T0 = 4
13:48:38.450 -> Hodnota dotyku senzoru T0 = 4
13:48:39.752 -> Hodnota dotyku senzoru T0 = 3
13:48:41.192 -> Hodnota dotyku senzoru T0 = 3
13:48:42.329 -> Hodnota dotyku senzoru T0 = 3
13:48:43.632 -> Hodnota dotyku senzoru T0 = 3
13:48:44.942 -> Hodnota dotyku senzoru T0 = 3
13:48:46.238 -> Hodnota dotyku senzoru T0 = 3
13:48:47.549 -> Hodnota dotyku senzoru T0 = 3
```

Obrázek 2: Výpis v terminálovom okne v stave kedy sa dotýkam nepretržite touch senzoru



```
13:49:17.825 -> Hodnota dotyku senzoru T0 = 2
13:49:19.128 -> Hodnota dotyku senzoru T0 = 103
13:49:20.128 -> Hodnota dotyku senzoru T0 = 103
13:49:21.128 -> Hodnota dotyku senzoru T0 = 104
13:49:21.728 -> Pocet zablikani ledky bez stlacenia senzoru = 56
13:49:22.128 -> Hodnota dotyku senzoru T0 = 103
13:49:23.128 -> Hodnota dotyku senzoru T0 = 103
13:49:24.128 -> Hodnota dotyku senzoru T0 = 8
13:49:25.429 -> Hodnota dotyku senzoru T0 = 3
13:49:26.750 -> Hodnota dotyku senzoru T0 = 3
13:49:28.052 -> Hodnota dotyku senzoru T0 = 3
13:49:29.355 -> Hodnota dotyku senzoru T0 = 3
13:49:30.656 -> Hodnota dotyku senzoru T0 = 3
13:49:31.957 -> Hodnota dotyku senzoru T0 = 3
13:49:33.259 -> Hodnota dotyku senzoru T0 = 3
13:49:34.561 -> Hodnota dotyku senzoru T0 = 104
13:49:35.562 -> Hodnota dotyku senzoru T0 = 103
13:49:36.564 -> Hodnota dotyku senzoru T0 = 103
13:49:37.119 -> Pocet zablikani ledky bez stlacenia senzoru = 57
13:49:37.520 -> Hodnota dotyku senzoru T0 = 103
13:49:38.541 -> Hodnota dotyku senzoru T0 = 103
```

Obrázek 3: Výpis v terminálovom okne v stave kedy sa dotýkam touch senzoru a následne ho pustím

## 6 Zdroje

### Reference

- [1] How to install Arduino. [online]. Dostupné z: <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>
- [2] Bidlo, M.: Obsluha a programování platformy Wemos D1 R32 s MCU ESP32. [online]. Dostupné z: [https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Flectures%2F08-Wemos\\_D1\\_R32\\_scheme.pdf](https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Flectures%2F08-Wemos_D1_R32_scheme.pdf)
- [3] Bidlo, M.: Obsluha a programování platformy Wemos D1 R32 s MCU ESP32. [online]. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Flectures%2F08-ESP32-Wemos-D1-R32.pdf>