|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Střední průmyslová škola elektrotechnická Havířov** | | | **HAW** | | | **Třída: 3.A** | | |
|
| **Skupina:**  1 | | |
|
| **Pokročilá práce s LED a tlačítkem** | | | | | | **Datum: 14. 3. 2025** | | |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | **Jméno učitele:** Tereza Hermanová | |  |
|  |  |  |  |  |  |  | | |
|  |  |  |  |  |  | **Jméno:** Samuel Černý |  |  |
|  |  |  |  |  |  |  | |  |
|  |  |  |  |  |  | **Známka:** | | |
|  |  |  |  |  |  |

**Zadání:**

**Zadání:** Rozblikejte postupně všechny LED, právě blikající LED se mění stisknutím tlačítka.

**Diagram:**

**Obsah obrázku text, snímek obrazovky, Písmo, design

Obsah vygenerovaný umělou inteligencí může být nesprávný.**

**Teoretický rozbor:**

**Přípravek**

Pro realizaci zadání, které jsme dostali, jsem využil programovatelný logický obvod (dále PLD).  
PLD jsou číslicové obvody, které lze konfigurací naprogramovat. PLD mají oproti neprogramovatelným logickým obvodům (dále NPLD) mnoho výhod jako např. zpřehlednění a snížení počtu pinů, jsou mnohem spolehlivější než NPLD, jsou jednoduše modifikovatelné a velmi výkonné. Avšak PLD mají i pár nevýhod jako např. to, že musíme znát programovací jazyk pro dané PLD, nebo i to že při poruše je nutno vyměnit celé PLD. V mém případě jsem využíval přípravek z řady ARM, konkrétně tedy STM32F407VG.

**Něco o prostředí**

K vyvíjení programu jsem využíval prostředí Visual Studio Code (dále VSC). V tomto prostředí pracuji už minimálně 4 roky a absolutně mi vyhovuje. Do VSC můžeme přidávat různá rozšíření např. na práci s jazyky C a CPP, tohle nám umožňuje v prostředí vyvíjet mnoho projektů. Já si do VSC přidal rozšíření, které se nazývá PlatformI. Tohle rozšíření přidává do VSC menu ve kterém máme možnost si založit projekt na naše PLD, pokud používáme Rasberry Pi, tak si ho můžeme založit na něj. V mém konkrétním případě jsem si vybral STM32F407VG. Během kódování máme několik různých možností, které nám PlatformIO přidává. Nejdůležitější z nich jsou: „build’’ která nám připraví bitový soubor a „upload“ která následně nahraje bitový soubor na náš přípravek. Celý zdrojový kód byl psán v vývojovém jazyce C.

**Samotný projekt**

Když nahrajeme projekt na ARM, bliká LED na portu 12. Každé stisknutí tlačítka posune blikání o port výš. Po dosažení portu 15 se vrátíme zpět na 12.Aby to fungovalo, aktivujeme hodiny pro porty A a D, nastavíme LED jako výstupy a ve smyčce hlídáme stisk tlačítka. Pokud ano, zvýšíme proměnnou x, což určuje, která LED svítí. Blikání je řešeno krátkým rozsvícením a zhasnutím s prodlevou. Celé to běží pořád dokola.

**Zdrojový kód**

**#include <stm32f4xx.h>**

**#include <chrono.h>**

***//start je 12, protože to je první port, kde se nachází LED***

***//end je 15, protože to je poslední port, kde se nachází LED***

**#define start 12**

**#define end 15**

**int main()**

**{**

***//inicializace proměnných***

**int i;**

**int x = 0;**

***//lines, které umožní, aby mi fungoval SysTick*, *potřebuju to ať mi funguje delay***

**SystemCoreClockUpdate();**

**SysTick\_Config(SystemCoreClock / 1000);**

***//aktivace hodin pro A a D***

**RCC->AHB1ENR |= (1UL << 0);**

**RCC->AHB1ENR |= (1UL << 3);**

***//loop na inicializaci všech 4 LED***

**for (i = start; i <= end; i++)**

**{**

**GPIOD->MODER &= ~(3 << (i \* 2));**

**GPIOD->MODER |= (1 << (i \* 2));**

**}**

***//nekonečný loop ve kterém se kontroluje stlačení tlačítka a blikají LED***

**while (1)**

**{**

***//pokud bude stlačeno tlačítko, tak se přídá 1 k x***

**if (GPIOA->IDR & (1UL << 0))**

**{**

**x++;**

**delay(100);**

**}**

***//všechny LED se vypnou***

**GPIOD->ODR &= ~(1UL << 12);**

**GPIOD->ODR &= ~(1UL << 13);**

**GPIOD->ODR &= ~(1UL << 14);**

**GPIOD->ODR &= ~(1UL << 15);**

***//zapne se LED***

**GPIOD->ODR |= (1 << (start + (x % 4)));**

**delay(50);**

***//vypne se LED***

**GPIOD->ODR &= ~(1 << (start + (x % 4)));**

**delay(50);**

**}**

**return 0;**

**}**

**Zhodnocení:**

Když jsem viděl zadání poprvé upřímně jsem nevěděl, co s tím. Později, ale mi začalo docházet, jak tuto úlohu splnit. Příště bych v tomto kódu rozhodně nevyužil modulo, protože to je zbytečně složité.