

ELEKTRONICKÉ POČÍTAČE

(MIKROPROCESOROVÁ TECHNIKA)

Název úlohy

Ovládání LED diody pomocí UART

Číslo úlohy

2

Zadání:

Vytvořte pomocí UART sériové komunikace zapojení a program, které umožní uživateli zapnout a vypnout LED diodu pomocí příkazů ON a OFF (psáno velkými i malými písmeny).

Součásti:

- 1× LED dioda
- 1× rezistor 500 Ω
- program PuTTY
- UART

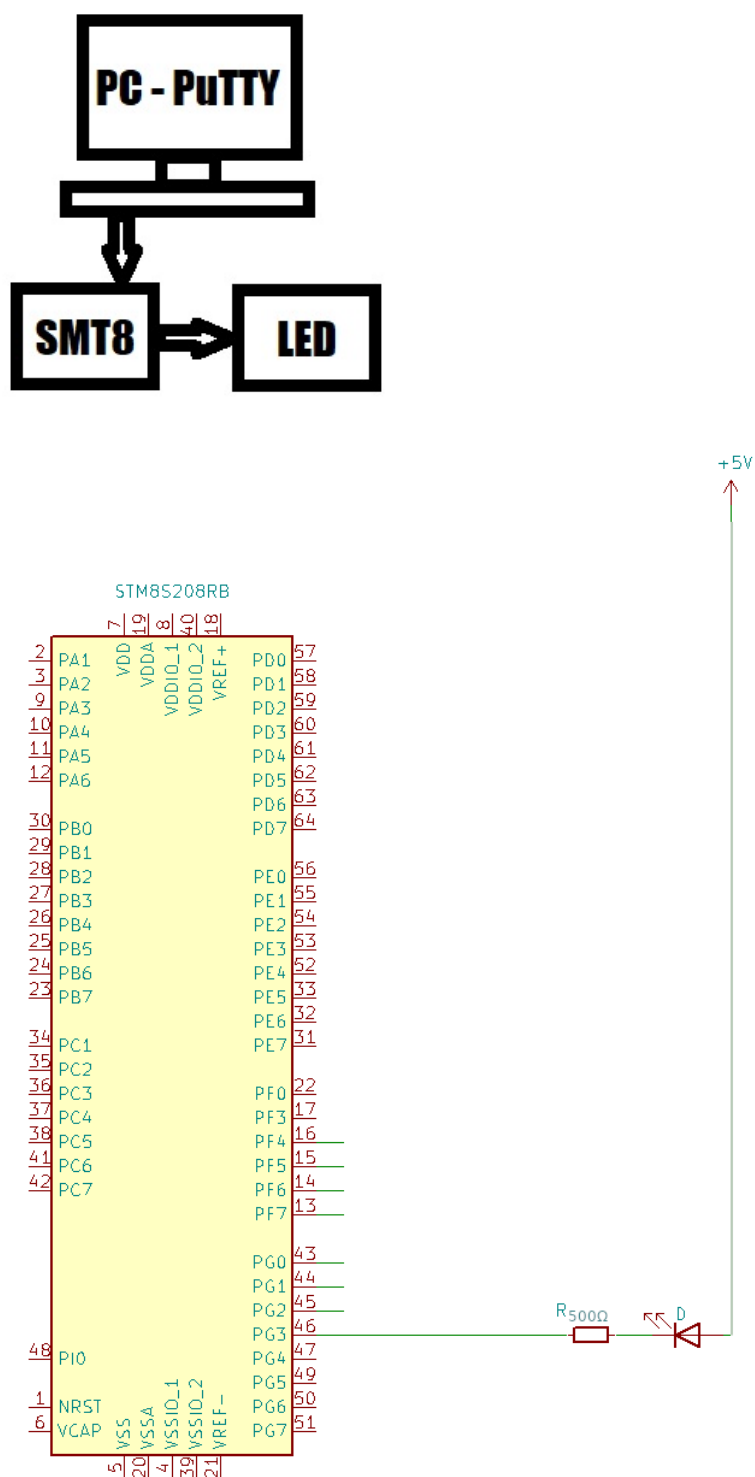
Poř. č. 13	Příjmení a jméno PÁLENÍK Matouš		Třída 4B	Skupina 2	Školní rok 2021/22		
Datum zadání 16. 3. 2022		Datum odevzdání 19. 4. 2022	Počet listů 6	příprava	Klasifikace měření	protokol	obhajoba
Protokol obsahuje:			popis zapojení schéma zapojení popis funkce programu	vývojový diagram zhodnocení, co jsem se naučil zdrojový kód main.c			

SLOVNÍ POPIS ZAPOJENÍ:

STM8 je připojen k PC, ze kterého je napájen. LED dioda je umístěna na nepájivém poli a je připojena katodou přes rezistor $R=500\Omega$ na pin STM8 s označením PG3, anodou pak na napájení + 5V.

Komunikace po sériové sběrnici UART je v počítači zaopatřena programem PuTTY.

SCHÉMA ZAPOJENÍ:



SLOVNÍ POPIS FUNKCE PROGRAMU:

Program pomocí cyklů hledá v uživateli zadaných znacích příkazy ON a OFF, na základě kterých pak zapíná nebo vypíná LED diodu. Velikost písmen nehraje žádnou roli, program se zaměřuje pouze na posloupnost písmen, ať už velkých či malých.

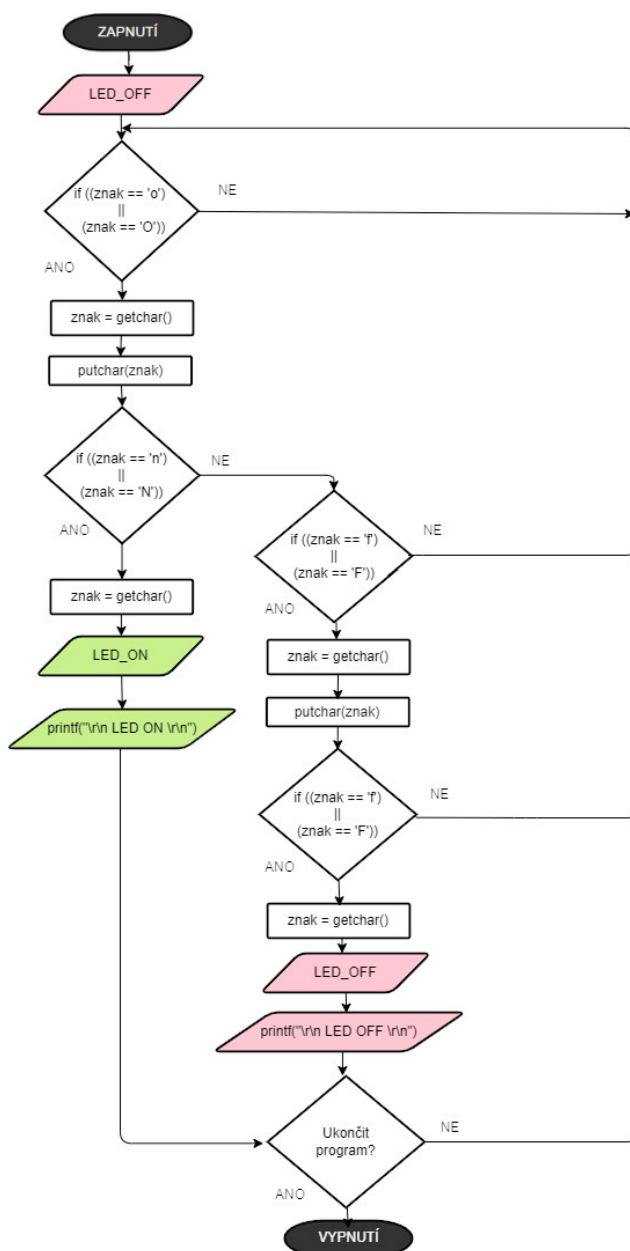
Ve výchozím stavu je LED dioda vypnutá. Následně program čte každý zadaný znak z klávesnice a zkoumá, zda se nejedná o písmeno „o“. Pokud toto písmeno zaznamená, zjišťuje, zda po něm uživatel zadal písmeno „n“ nebo „f“.

V případě, že po písmenu „o“ program zaznamená „n“, znamená to, že uživatel zadal „on“ a tak program zapne LED diodu a vrátí se na začátek, tedy opět hledá písmeno „o“.

V případě, že po písmenu „o“ následuje „f“, program nadále zkoumá, zda dalším znakem bude také „f“. Pokud ano, uživatel zadal „off“ a tudíž program vypíná LED diodu a opět se vrací na začátek.

V případě že uživatel zadá „o“ a po něm pak jiný znak než „n“ a „f“, program se vrací na začátek.

VÝVOJOVÝ DIAGRAM:



ZHODNOCENÍ A CO JSEM SE NAUČIL

Zapojení funguje a splňuje zadání ve všech jeho bodech. Ošetřil jsem možnost zadání příkazu ON a OFF jak velkými, tak malými písmeny. Během práce na tomto projektu jsem lépe pochopil některé pojmy a zápis v jazyce C. Vzhledem k jednoduchosti a malému rozsahu se jednalo o můj nejrychleji zpracovaný projekt do předmětu EPO (MIT). Jsem rád, že jsem jej zdárně dokončil a dovršil tak snad potřebný počet bodů ke klasifikaci.

FOTODOKUMENTACE



VIDEO:

<https://youtube.com/playlist?list=PLDPwFwnBWuhMsP5TsFjbr4T41LkfAcIbV>



ZDROJOVÝ KÓD main.c:

```
1  /*
2  2022-04-20
3  Projekt JARNÍ: Ovládání LED diody pomocí UART
4  Matouš Páleník, 4B
5  Obsahuje: 1 LED dioda
6           : UART komunikace přes PuTTY
7  */
8
9  #include "stm8s.h"
10 #include "assert.h"
11 #include <stdio.h> //standartní knihovna pro vstup a výstup, obsahuje getchar i putchar
12
13 #define LED_PORT GPIOG
14 #define LED_PIN GPIO_PIN_3
15 #define LED_OFF GPIO_WriteHigh(LED_PORT, LED_PIN)
16 #define LED_ON GPIO_WriteLow(LED_PORT, LED_PIN)
17 #define LED_REVERSE GPIO_WriteReverse(LED_PORT, LED_PIN)
18
19 char znak; //char je stejné jako int8_t
20
21 void setup(void){
22     CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovat MCU na 16MHz
23
24     UART1_DeInit();
25     UART1_Init((uint32_t)115200, //rychlost
26                UART1_WORDLENGTH_8D, //počet bitů ve slově
27                UART1_STOPBITS_1, //počet stopbutů
28                UART1_PARITY_NO, //bez kontroly parity
29                UART1_SYNCMODE_CLOCK_DISABLE, //asynchronní režim
30                UART1_MODE_TXRX_ENABLE //vysílání i příjem
31            );
32     UART1_Cmd(DISABLE); //celkově zakáže UART1
33     UART1_Cmd(ENABLE); //celkově povolí UART1
34
35     //LED dioda
36     GPIO_Init(LED_PORT, LED_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);
37 }
38
39 // hlavní smyčka programu
40 void main(void){
41     setup();
42     LED_OFF;
43     while (1){
44         znak = getchar(); //zaznamená znak
45         putchar(znak); //vypíše znak
46
47         if ((znak == 'o')||(znak == 'O')) {
48             //jakmile program zaznamená písmeno "o", ptá se,
49             //zda za ním následuje "n" nebo "f" (pro slova ON a OFF)
50             znak = getchar();
51             putchar(znak);
52
53             if ((znak == 'n')||(znak == 'N')) {
54                 znak = getchar();
55                 LED_ON;
56                 printf("\r\nLED ON\r\n"); //informace, že LED svítí
57             }
58         }
```

```

59         if ((znak == 'f')||(znak == 'F')) {
60             znak = getchar();
61             putchar(znak);
62
63             if ((znak == 'f')||(znak == 'F')) {
64                 znak = getchar();
65                 LED_OFF;
66                 printf("\r\nLED OFF\r\n"); //informace, že LED nesvítí
67             }
68         }
69     }
70 }
71 }
72
73
74 char putchar (char c)
75 {
76     /* Write a character to the UART1 */
77     UART1_SendData8(c);
78     /* Loop until the end of transmission */
79     while (UART1_GetFlagStatus(UART1_FLAG_TXE) == RESET);
80
81     return (c);
82 }
83
84 char getchar (void)
85 {
86
87     int c = 0;
88     /* Loop until the Read data register flag is SET */
89     while (UART1_GetFlagStatus(UART1_FLAG_RXNE) == RESET);
90     c = UART1_ReceiveData8();
91     return (c);
92 }

```