**MIT**

**Závěrečný projekt – Zobrazení souřadnic**

**Funkce:**

1. Zobrazení souřadnic na digitálním displayi
2. Zobrazení aktuálního datumu

**Potřeby:**

* Mikrokontrolér: Budu potřebovat mikrokontrolér jako základ pro moje zařízení.
* Displej: LCD displej, nebo LED.
* Snímač polohy: Snímač GPS
* Software: (programování v C)
* Napájení: Napájení bude zajištěno baterií
* Konektory a kabely: Budu potřebovat kabely a konektory pro propojení všech součástí.

**Stručný popis činnosti programu**

* Program je navržen pro mikrokontrolér STM8 a zajišťuje zpracování GPS dat, které přicházejí přes UART. Zpracovaná data jsou následně zobrazována na LCD displeji připojeném přes I2C. Program střídavě zobrazuje zeměpisnou šířku a délku spolu s datem. Rovněž program každých 10 sekund přepíná stav LED diody.

**Popis kodu**

* **Záhlaví a Definice Konstant**

**#include "main.h"**

#include "milis.h"

#include <stdbool.h>

#include <stdio.h>

#include <stm8s.h>

#include "uart1.h"

#include <string.h>

#include "LCD\_I2C.h"

#include "delay.h"

**#define BUFFER\_SIZE 128**

* **#include "main.h"** a další #include příkazy: Tyto hlavičkové soubory zahrnují knihovny a funkce potřebné pro práci s mikrokontrolérem STM8, UART komunikací, I2C komunikací a LCD displejem.
* **#define BUFFER\_SIZE 128**: Definuje velikost bufferu pro ukládání přijatých dat z GPS modulu.
* **Inicializační Funkce**

void init(void) {

**CLK\_HSIPrescalerConfig(CLK\_PRESCALER\_HSIDIV1);**

**GPIO\_Init**(LED\_PORT, LED\_PIN, GPIO\_MODE\_OUT\_PP\_LOW\_SLOW);

#if defined(BTN\_PORT) || defined(BTN\_PIN)

#endif

**init\_milis();**

**init\_uart1();**

**UART3\_DeInit**();

**UART3\_Init**(9600, UART3\_WORDLENGTH\_8D, UART3\_STOPBITS\_1, UART3\_PARITY\_NO, UART3\_MODE\_TXRX\_ENABLE);

}

**CLK\_HSIPrescalerConfig(CLK\_PRESCALER\_HSIDIV1)**: Nastavení hodinového prescaleru, který určuje frekvenci hodin MCU na 16 MHz.

**GPIO\_Init**(LED\_PORT, LED\_PIN, GPIO\_MODE\_OUT\_PP\_LOW\_SLOW): Inicializace GPIO pro LED jako výstupní pin.

**init\_milis**(): Inicializace časovače pro měření uplynulého času.

**init\_uart1()**: Inicializace UART1 pro debugování.

**UART3\_DeInit() a UART3\_Init(...)**: Inicializace UART3 pro komunikaci s GPS modulem s rychlostí 9600 baudů.

* **Globální Proměnné**

char latitude[16] = {0};

char longitude[16] = {0};

char date[16] = {0};

Deklarace globálních proměnných pro uložení zeměpisné šířky, délky a datumu získaných z GPS dat.

* **Funkce pro Zpracování NMEA Zpráv**

void **processNMEA**(char \*buffer) {

if (strstr(buffer, "$GPGGA") != NULL) {

char \*token = strtok(buffer, ",");

int fieldIndex = 0;

while (token != NULL) {

fieldIndex++;

if (fieldIndex == 2) {

***// Pole pro čas v GGA větě***

}

else if (fieldIndex == 3) {

strncpy(latitude, token, sizeof(latitude) - 1);

} ***// Uložení zeměpisné šířky***

else if (fieldIndex == 4) {

} ***// Pole pro směr zeměpisné šířky***

else if (fieldIndex == 5) {

strncpy(longitude, token, sizeof(longitude) - 1); ***// Uložení zeměpisné délky***

}

token = strtok(NULL, ",");

}

} else if (strstr(buffer, "$GPRMC") != NULL) {

***// Kontrola, zda buffer obsahuje $GPRMC zprávu***

char \*token = strtok(buffer, ",");

int fieldIndex = 0;

while (token != NULL) {

fieldIndex++;

if (fieldIndex == 9) {

***// Pole pro datum v RMC větě***

strncpy(date, token, sizeof(date) - 1);

***// Uložení datumu***

}

token = strtok(NULL, ",");

}

}

}

**processNMEA**(char \*buffer): Zpracovává přijatá NMEA data z GPS modulu.

Kontroluje, zda buffer obsahuje $GPGGA nebo $GPRMC zprávu.

Rozděluje zprávu na jednotlivá pole a ukládá zeměpisnou šířku, délku a datum do odpovídajících globálních proměnných.

* **Funkce pro Zobrazení na LCD**

void **displayLatitude**() {

LCD\_I2C\_SetCursor(0, 0); ***// Nastavení kurzoru na začátek prvního řádku***

LCD\_I2C\_Print("Dnes je:");

LCD\_I2C\_Print(date); ***// Zobrazení datumu na prvním řádku***

LCD\_I2C\_SetCursor(0, 1); ***// Nastavení kurzoru na začátek druhého řádku***

LCD\_I2C\_Print("Sirka:");

LCD\_I2C\_Print(latitude); ***// Zobrazení zeměpisné šířky na druhém*** ***řádku***

}

void **displayLongitude**() {

LCD\_I2C\_SetCursor(0, 0); ***// Nastavení kurzoru na začátek prvního*** ***řádku***

LCD\_I2C\_Print("Dnes je:");

LCD\_I2C\_Print(date); ***// Zobrazení datumu na prvním řádku***

LCD\_I2C\_SetCursor(0, 1); ***// Nastavení kurzoru na začátek druhého*** ***řádku***

LCD\_I2C\_Print("Delka:");

LCD\_I2C\_Print(longitude); ***// Zobrazení zeměpisné délky na druhém*** ***řádku***

}

* **displayLatitude()**: Nastaví kurzor na LCD displeji a zobrazí datum a zeměpisnou šířku.
* **displayLongitude()**: Nastaví kurzor na LCD displeji a zobrazí datum a zeměpisnou délku.
* **Hlavní Funkce Programu**

**int main(void)** {

uint32\_t time = 0;

char buffer[BUFFER\_SIZE] = {0};

uint8\_t bufferIndex = 0;

bool displayLat = true;

***// Inicializace I2C LCD displeje***

LCD\_I2C\_Init(0x27, 16, 2);

**init();**

***// Počáteční zobrazení zeměpisné šířky***

**displayLatitude();**

**while (1)** {

***// Kontrola, zda byla přijata data přes UART3***

if (UART3\_GetFlagStatus(UART3\_FLAG\_RXNE) != RESET) {

char receivedChar = UART3\_ReceiveData8();

if (receivedChar == '\n' || receivedChar == '\r') {

buffer[bufferIndex] = '\0';

processNMEA(buffer);

***// Zpracování přijatých NMEA dat***

bufferIndex = 0;

memset(buffer, 0, sizeof(buffer));

} else {

if (bufferIndex < BUFFER\_SIZE - 1) {

buffer[bufferIndex++] = receivedChar;

}

}

}

***// Kontrola, zda uplynulo 10 sekund pro změnu zobrazení***

if (milis() - time > 10000) {

time = milis();

displayLat = !displayLat;

***// Přepnutí mezi zobrazením zeměpisné šířky a délky***

if (displayLat) {

displayLatitude();

} else {

displayLongitude();

}

***// Přepnutí stavu LED diody***

GPIO\_WriteReverse(LED\_PORT, LED\_PIN);

}

}

}

**int main(void)**: Hlavní funkce programu, která provádí inicializace a hlavní smyčku.

LCD\_I2C\_Init(0x27, 16, 2): Inicializace I2C LCD displeje s adresou 0x27 a rozměrem 16x2 znaky.

**init()**: Volání inicializační funkce.

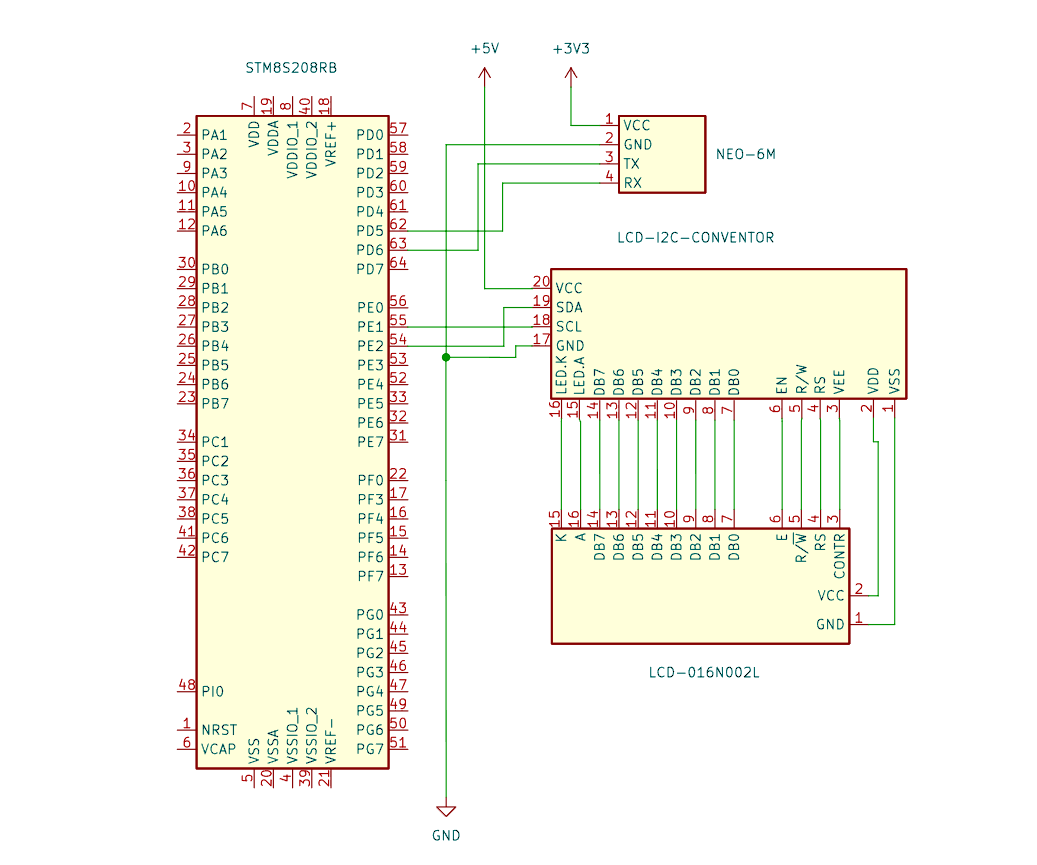
**displayLatitude()**: Počáteční zobrazení zeměpisné šířky.

Hlavní smyčka **(while (1)):**

Kontroluje, zda byla přijata data přes UART3.

Pokud je přijata celá zpráva (označená \n nebo \r), zpracuje ji a uloží do bufferu.

Každých 10 sekund přepíná mezi zobrazením zeměpisné šířky a délky a přepíná stav LED diody.

**Schéma KiCad**

**Diagram:**

