

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ**

# **BAKALÁŘSKÁ PRÁCE**

**Časomíra pro závody robotů**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2018/2019

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Alona HEPENKO**

Osobní číslo: **E15B0230P**

Studijní program: **B2612 Elektrotechnika a informatika**

Studijní obor: **Elektronika a telekomunikace**

Název tématu: **Časomíra pro závody robotů**

Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

### Z á s a d y p r o v y p r a c o v á n í :

Navrhněte a realizujte jednotku určenou jako časomíra pro závody robotů:

1. Uvažujte vhodné mechanické rozměry pro roboty používané ve výuce i soutěžích spolupřátaných FEL KAE.
2. Časoměrné zařízení bude vybaveno displejem se zobrazením času závodu/jízdy.
3. Umožněte ruční ovládání i připojení v PC.
4. Uvažujte variantu "okruh", tj. start a cíl jsou stejným bodem.
5. Alternativně uvažujte nezávislý start a cíl s nutností komunikace mezi bloky.

Rozsah grafických prací: podle doporučení vedoucího

Rozsah kvalifikační práce: 30 - 40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

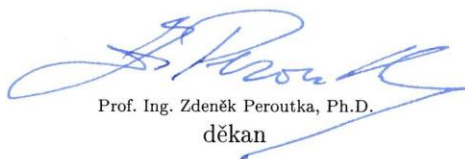
**Pinker, J.: "Mikroprocesory a mikropočítače", Praha: BEN - technická literatura, 2004, ISBN 80-7300-110-1**

Vedoucí bakalářské práce: Ing. Petr Weissar, Ph.D.

Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: 5. října 2018

Termín odevzdání bakalářské práce: 13. června 2019



Prof. Ing. Zdeněk Peroutka, Ph.D.  
děkan



Doc. Dr. Ing. Vjačeslav Georgiev  
vedoucí katedry

V Plzni dne 5. října 2018

## **Abstrakt**

Daná bakalářská práce se zabývá návrhem časomíry pro závody robotů, které se používají ve výuce a soutěžích spolupořádaných FEL KAE. V první části se popisuje princip realizace časomíry a jednotlivých bloků se kterých se skládá. Druhá část je věnována mechanické konstrukci časomíry a návrhu 3D modelu pro zachycení mikrokontrolérů. Třetí část se zabývá nastavením jednotlivých bloků a komunikace mezi nimi. Ve čtvrté části se řeší softwarová realizace časomíry v mikrokontrolerách a návrh uživatelského rozhraní pro PC sloužícího pro přehledné zobrazení výsledků závodů a pro jejich následující zpracování.

## **Klíčová slova**

Časomíra, Mikrokontrolér, Senzor, Displej, USART, I2C, SysTick, C#, Závod.

## **Abstract**

This bachelor thesis deals with the design of timekeeper for robot races, which are used in teaching and competitions co-organized by FEL KAE. The first part describes the principle of realization of timekeeper and individual blocks it is composed. The second part is devoted to mechanical construction of timekeeper and design of 3D model for microcontrollers capturing. The third part deals with setting of individual blocks and communication between them. The fourth part deals with software realization of timekeeper in microcontrollers and design of user interface for PC used for clear display of race results and for their subsequent processing.

## **Key words**

Timekeeper, Microcontroller, Sensor, Display, USART, I2C, SysTick, C#, Race.

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....

podpis

V Plzni dne 10.6.2019

Alona Hepenko

## **Poděkování**

Tímto bych ráda poděkovala vedoucímu bakalářské práce Ing. Petrovi Weissarovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

# Obsah

<b>OBSAH</b> .....	<b>8</b>
<b>ÚVOD</b> .....	<b>9</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>11</b>
<b>1 OBECNÝ POPIS</b> .....	<b>13</b>
1.1 MIKROKONTROLÉR .....	14
1.2 ROZŠÍŘUJÍCÍ DESKA .....	17
1.3 SENZORY VZDÁLENOSTI.....	18
1.3.1 VL6180X.....	19
1.3.2 VL53L0X.....	20
1.4 DISPLEJ .....	21
<b>2 MECHANICKÁ KONSTRUKCE</b> .....	<b>22</b>
<b>3 NASTAVENÍ MODULŮ</b> .....	<b>24</b>
3.1 POUŽITÉ NÁSTROJE.....	24
3.1.1 Výběr IDE.....	24
3.1.2 PuTTY.....	25
3.1.3 Uživatelské rozhraní.....	26
3.2 USART KOMUNIKACE MEZI MIKROKONTROLÉRY A PC .....	26
3.3 I2C KOMUNIKACE MEZI SENZORY, DISPLEJEM A MCU .....	28
3.4 NASTAVENÍ VL6180X .....	31
3.5 NASTAVENÍ VL53L0X.....	33
3.6 NASTAVENÍ DISPLEJE .....	33
3.7 NASTAVENÍ ČASOVAČE .....	34
<b>4 NÁVRH SW</b> .....	<b>35</b>
4.1 POPIS PRINCIPU SW REALIZACE ČASOMÍRY .....	35
4.2 UŽIVATELSKÉ ROZHŘANÍ .....	37
<b>ZÁVĚR</b> .....	<b>39</b>
<b>SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ</b> .....	<b>1</b>



## Úvod

Tato bakalářská práce je zaměřená na realizaci časomíry, která by byla schopná detekovat průjezd robota a zaznamenávat čas závodu. Časomíra musí být použitelná pro dva druhy závodu: start a cíl jsou ve stejném bodě (okruh) a v různých bodech (start-cíl). Zařízení musí mít takové mechanické rozměry, aby bylo schopné detekovat průjezd různých robotů, které se používají ve výuce i v soutěžích spolupořádaných FEL KAE. Časomíra by měla mít pevnou, jednoduše přemístitelnou konstrukci. Časomíra musí být schopná zaznamenávat průchod robotů i při vysoké rychlosti, záznam času musí být dostatečně přesný. Čas závodu musí být zobrazen na displeji. Zařízení musí komunikovat s PC. Přesnost měření nesmí být závislá na okolních podmínkách.

Existuje několik variant realizace časomíry. Jednou z nich je světelná závora. Tento druh zařízení tvoří dvě jednotky: vysílač a přijímač. Vysílačem je zdroj světla, což přijímač musí rozpoznat. Přijímač při přerušení paprsku musí odeslat signál do další jednotky pro následující zpracování. Nevýhodou tohoto druhu zařízení je závislost na okolním osvětlení. Pro odstranění tohoto nedostatku musí být schéma zapojení doplněno o další bloky. Avšak návrh desky plošného spoje pro tuto variantu je jednoduchý. Princip světelné závory se často používá v časoměrných zařízeních.

Další variantou je použití čipů v robotech a antény na místě startu a cíle. Při přiblížení čipu k anténě, se zachytí signál z čipu a pošle se do dalších obvodů pro následující zpracování. Výhodou této varianty je, že se najednou může zúčastnit závodů několik robotů. Nevýhoda této varianty spočívá v nepřesnosti, která bude mít význam pro závody na krátký čas.

V dané bakalářské práci pro realizaci časomíry byla zvolená následující varianta: použít senzor vzdálenosti připojený k MCU. Při průchodu robota přes paprsek, který vysílá senzor, se detekuje změna měřené vzdálenosti, spouští nebo zastavuje hodiny v MCU. Zařízení může být použité pro variantu s nezávislým startem a cílem, bloků MCU a senzory musí být dva. Oba bloky MCU by měli komunikovat mezi sebou, a jeden z nich musí být připojen k PC pro ovládání zařízení a zobrazení výsledků závodu. Senzor vzdálenosti musí provádět měření dostatečně rychle, aby byl schopný zachytit průjezd i

krátkých robotů. Výhodou použitých senzorů je, že jsou založené na technologii „Time of flight“, kde se měří čas, za který se paprsek odrazí zpět k senzoru, a z toho se pak vypočítává vzdálenost. To znamená, že výsledek měření není závislý na okolním světle a na odrazivosti předmětu. Výhodou použití MCU pro časomíru, je to, že hodiny v mikrokontroléru jsou přesnější než hodiny v PC. Navržené řešení odpovídá základním požadavkům časomíry pro závody robotů.

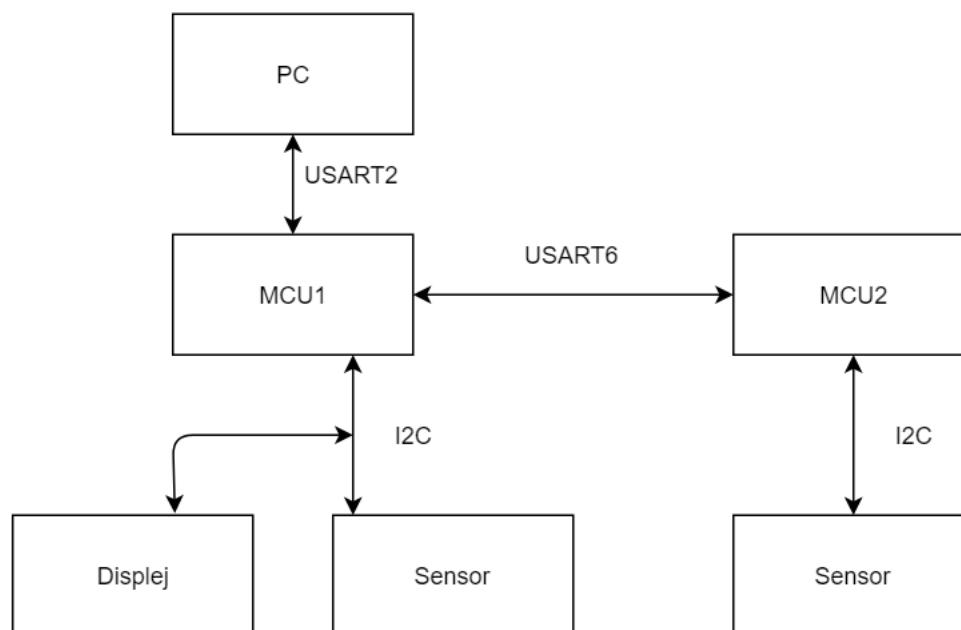
## Seznam symbolů a zkratk

3D .....	Three-dimensional space
A/D .....	Analog-to-digital converter
AHB .....	Advanced High-performance Bus
APB .....	Advanced Peripheral Bus
CMSIS .....	Cortex Microcontroller Software Interface Standard
COM .....	Communication port
CSV .....	Comma-separated values
D/A .....	Digital-to-analog converter
EEPROM .....	Electrically erasable programmable read-only memory
FPU .....	Floating-point unit
GND .....	Ground
GPIO .....	General-purpose input/output
I2C .....	Inter-Integrated Circuit
I2S .....	Inter-IC Sound
IDE .....	Integrated development environment
I/O .....	Input-output
IoT .....	Internet of things
LED .....	Light-emitting diode
MCU .....	Microcontroller unit
MPU .....	Memory protection unit
NVIC .....	Nested vectored interrupt controller
PC .....	Personal computer
PMOS .....	P-channel metal-oxide semiconductor
RISC .....	Reduced instruction set computer
RS-232 .....	Recommended Standard 232
SCB .....	System control block
SCL .....	Serial Clock
SDA .....	Serial Data Line
SDIO .....	Secure Digital Input Output
SPAD .....	Single-photon avalanche diode
SRAM .....	Static random-access memory,

SSH.....	Secure Shell
UART .....	Universal asynchronous receiver-transmitter
USART .....	Universal synchronous and asynchronous receiver-transmitter
USB .....	Universal Serial Bus, univerzální sériová sběrnice
VCSEL .....	Vertical Cavity Surface-Emitting Laser
VDD .....	Voltage Drain Drain
VS .....	Visual Studio
Atd. ....	A tak dále
Tj. ....	To je

## 1 Obecný popis

Pro realizaci časomíry byly použity následující bloky: PC, MCU1 s připojenou rozšiřující deskou vybavenou displejem a senzorem vzdálenosti, MCU2 s připojeným senzorem vzdálenosti. Tento model umožňuje 2 druhy závodu: start a cíl ve stejném místě – varianta “okruh”, start a cíl jsou v různých místech – varianta “start-cíl”.



Obr. 1 Blokové schéma časomíry

Pro vývoj časomíry byl zvolen mikrokontrolér firmy STMicroelectronics, model STM32F411RE, protože tento mikrokontrolér se používá ve výuce v rámci předmětu Mikroprocesory a počítače. Mám také s ním zkušenost a práci s ním v připravené knihovně. Do prvního MCU je připojená deska X-NUCLEO-6180XA1 se senzorem VL6180X a displejem ATA2453BG1. Do druhého MCU je připojen senzor VL53L0X. Zvolila jsem si dva různé snímače, abych si vyzkoušela obě varianty a mohla je porovnat mezi sebou.

Senzory vzdálenosti jsou připojené k mikrokontrolérům přes I2C rozhraní. Senzor stále měří vzdálenost a při zmenšení vzdálenosti pod 10 cm, zaznamená průchod nějakého předmětu bránou, který zapne nebo vypne časovač v mikrokontroléru. Čas jízdy je

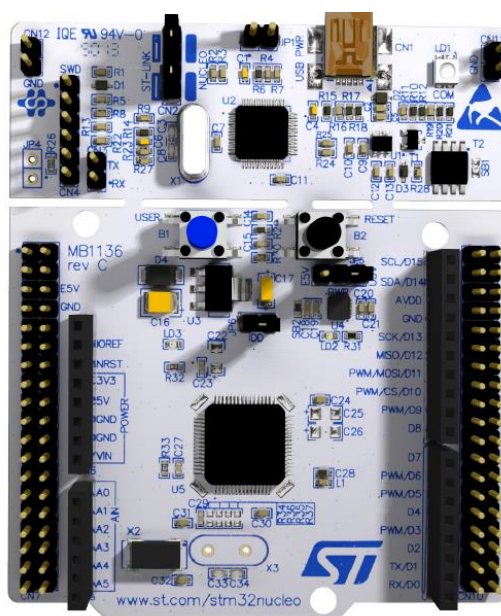
zobrazován na displeji, kterým je vybavená rozšiřující deska připojená k prvnímu mikrokontroléru přes stejné I2C rozhraní kde je i senzor.

Pro variantu “okruh” se používá pouze první mikrokontrolér. Při prvním průchodu bránou se začne odpočítávat čas, a při druhém průchodu časovač se vypne. Pro variantu “start-cíl” jsou zapotřebí oba MCU. Při průchodu přes první bránu se zapne časovač, a při průchodu přes druhou MCU2 pošle signál v MCU1, kterým zastaví časovač v MCU1.

MCU1 komunikuje s PC přes USART. Do počítače jsou posílána data o průběhu jízdy: signál začátku a konce jízdy. V programovacím jazyce C# je udělaná okenní aplikace, která té informace zobrazuje v pohodlné uživatelské podobě. Také aplikace umožňuje zadání jména účastníka jízdy a uložení těchto informací do CSV souboru.

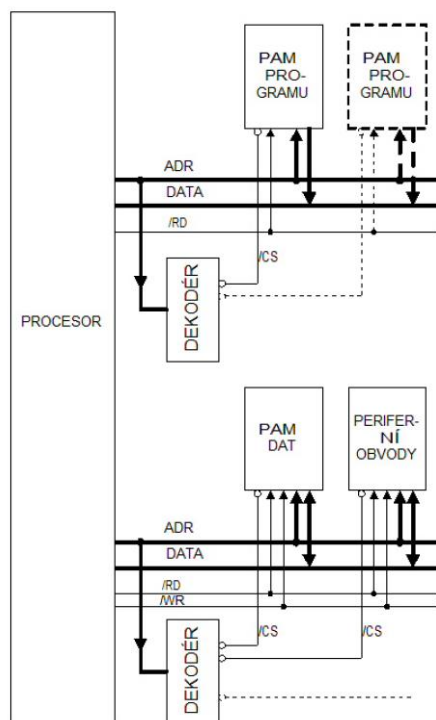
## 1.1 Mikrokontrolér

Pro vývoj časomíry byl zvolen mikrokontrolér firmy STMicroelectronics, model STM32F411RE. Výhodou mikrokontrolérů série STM32 je nízká cena a široká funkcionality. Tento mikrokontrolér je založen na vysoce výkonném 32bitovém RISC procesoru Cortex-M4 pracujícím na frekvenci do 100 MHz. STM32F411RE patří do řady produktů STM32 Dynamic Efficiency, které kombinují energetickou účinnost, výkon a integraci.



Obr. 2 Mikrokontrolér STM32F411RE [1]

Mikrokontrolér je založen na Harvardské architektuře, což znamená, že jedna soustava sběrnic pracuje s datovou pamětí a periférií a druhá soustava s programovou pamětí. Tato architektura zrychluje činnost procesoru, protože je možné současně provádět několik operací, například zapisovat data a číst další instrukci. Spousty současných mikropočítačů jsou založeny právě na Harvardské architektuře.



Obr. 3 Blokové schéma Harvardské architektury [2]

Mikrokontrolér obsahuje vysokorychlostní zabudované paměti (512 kB Flash paměť, 128 kB SRAM) a velký rozsah vstupu/výstupu a periférií připojených ke dvěma sběrnicím APB, dvěma sběrnicím AHB a 32bitové sběrnice multi-AHB. Zařízení nabízí jeden 12bitový analogově digitální převodník. Nízkonapěťové hodiny reálného času, šest univerzálních 16bitových časovačů včetně jednoho časovače s pulzně šířkovou modulací pro řízení motoru a dvou univerzálních 32bitových časovačů. Podporuje též funkce standardních a pokročilých komunikačních rozhraní:

- 3 rozhraní I2
- 5 rozhraní SPI
- 5 rozhraní I2S, z nichž 2 jsou plně duplexní.

- 3 USARTy
- Rozhraní SDIO
- Rozhraní USB 2.0

Mikrokontrolér je vybaven dvěma tlačítky: USER, které se dají použít ve vlastní aplikaci, a RESET. Na desce jsou umístěné tři LED: první LD1 označuje probíhající komunikaci přes USB, druhá LD2 je pro vlastní použití, třetí LD3 označuje připojené napájení. Po připojení napájení červené LD3 a LD1 musí svítit, LD1 a zelená LD2 musí blikat.

STM32F411RE pracuje v rozsahu teplot - 40 až + 125 °C s napájením od 1,7 do 3,6 V. Napájení je zajištěno buď hostitelským PC přes USB kabel 'Type-A to Mini-B' nebo externím zdrojem: VIN (7V-12V), E5V (5V) nebo +3,3 V na napájecích pinech CN6 nebo CN7. V dané bakalářské práci jsou mikrokontroléry napájené z PC. Deska STM32 Nucleo je rozdělena na dvě části: část ST-LINK a cílovou část STM32. ST-LINK část může být odříznuta pro zmenšení velikosti desky. V tomto případě může být zbývající cílová část STM32 napájena pouze VIN, E5V a 3,3 V na konektorech CN7 nebo VIN a 3.3V na konektorech Arduino CN6.

Procesor Cortex-M4 je vysoce výkonný 32.bitový procesor určený pro mikrokontroléry. Cortex-M4 a obsahuje následující periférie:

NVIC - Nested vectored interrupt controller, je řadič přerušení s nízkou latencí, to znamená, že doba od vyskytnutí požadavku na přerušení do zpracování první instrukce podprogramu je velice krátká.

SCB - System control block, je systémový řídicí systém, rozhraní programátora k procesoru. Poskytuje informace o implementaci a řízení systému, včetně konfigurace, řízení a reportování systémových výjimek.

SysTick – 24bitový časovač s počítáním dolů. Dá se použít jako časovač reálného času nebo jako obyčejný časovač.



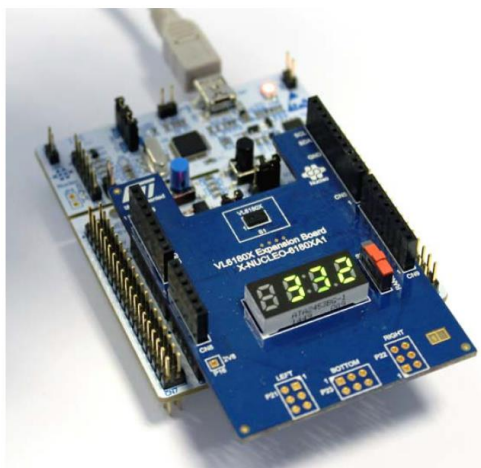
MPU - Memory protection unit, jednotka ochrany paměti která zlepšuje spolehlivost systému pomocí určení atributů pro různé adresové prostory. Poskytuje tak místo do 8 různých adresových prostorů.

FPU - Floating-point unit, jednotka která umožňuje provedení operací s hodnotami a s plovoucí desetinnou čárkou.

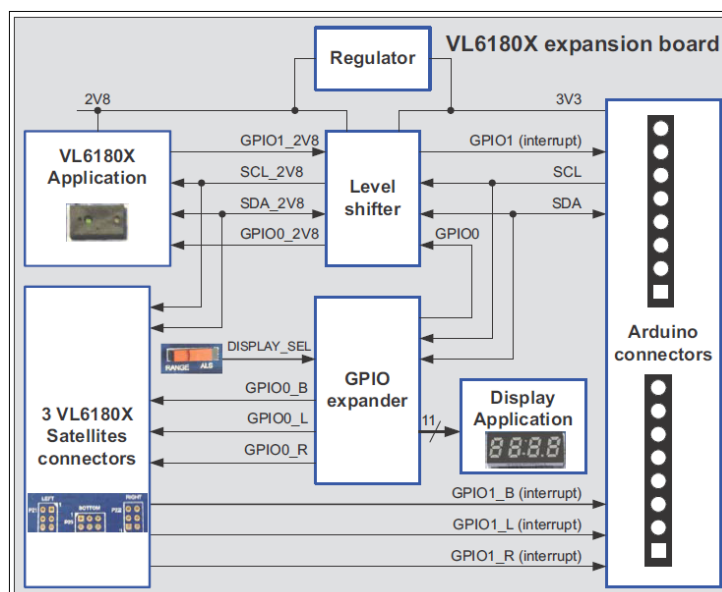
Díky výše uvedeným vlastnostem jsou mikrokontroléry STM32F411RE vhodné pro široký rozsah aplikací: řízení motoru a řízení aplikací, lékařské vybavení, průmyslové aplikace, PLC, střídače, jističe, tiskárny a skenery, poplachové systémy, domácí audio spotřebiče.

## 1.2 Rozšiřující deska

X-NUCLEO-6180XA1 je rozšiřující deska, vyhodnocovací a vývojový systém snímače přiblížení a okolního světla, navržený pro VL6180X, senzor založený na technologii FlightSense společnosti ST. VL6180X je senzor, který umí měřit vzdálenost do 10 cm a intenzitu osvětlení do 100 kLuxů, rozpoznávat gesta. Je velice přesný, změřené hodnoty nezávisí na odrazivosti cíle. K rozšiřující desce se dají připojit ještě 3 senzory, což umožňuje vývoj kreativních aplikací, například, rozpoznávání různých druhů gest. Také deska je vybavená displejem, který zobrazuje hodnotu vzdálenosti v mm nebo intenzitu osvětlení v Luxech, nebo data, které se odešlou přes I2C rozhraní. Senzor musí být napájen 2,8 V. Základní deska poskytuje konektory s napájením 3,3 V nebo 5 V, proto je rozšiřující deska vybavená regulátorem, který převádí 3,3 V na 2,8 V.



Obr. 4 Rozšiřující deska připojená k desce STM32 Nucleo [3]



Obr. 5 Blokové schéma rozšiřující desky [3]

Rozšiřující deska VL6180X je kompatibilní s rodinou elektronických desek STM32 Nucleo a Arduino. VL6180X komunikuje s mikroprocesorem vývojové desky STM32 Nucleo prostřednictvím I2C rozhraní dostupného na konektorech Arduino. Několik rozšiřujících desek ST mohou být umístěny jedna přes druhou, což umožňuje například vývoj aplikací s rozhraním Bluetooth nebo WiFi. Na rozdíl od první verze desky, druhá verze je vybavená GPIO expandérem, který umožňuje ovládání displeje. Díky této změně se uvolnilo několik vývodů GPIO, které mohou být použité s dalšími rozšiřujícími deskami.

### 1.3 Senzory vzdálenosti

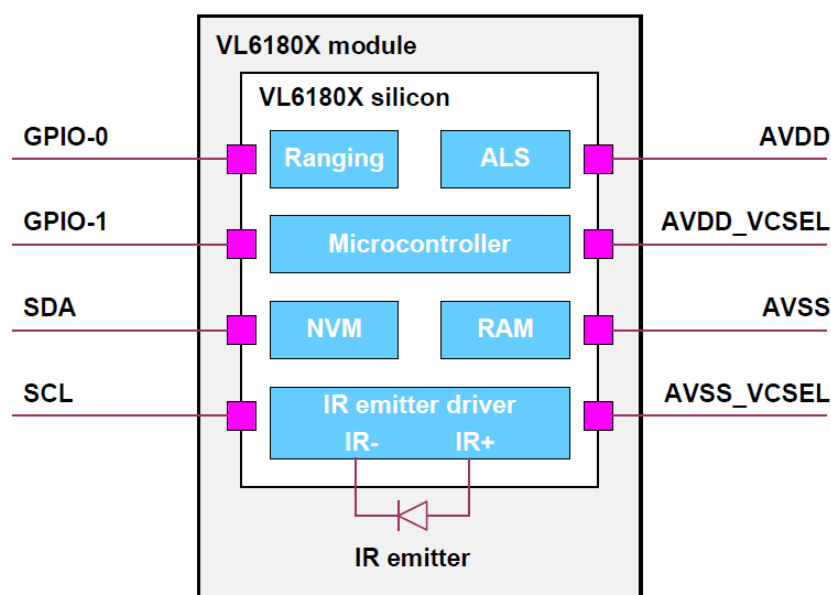
Oba senzory, VL6180X i VL53L0X, jsou založené na patentované technologii FlightSense společnosti ST. Jedná se o technologii, která umožňuje měřit absolutní vzdálenost nezávisle na odrazivosti povrchu a na okolním světle. Namísto odhadu vzdálenosti měřením množství světla odraženého zpět od objektu (což je výrazně ovlivněno barvou a povrchem), senzory přesně měří dobu, za kterou světlo se dostane cíle a odrazí se zpět k senzoru.

Senzory se skládají z vysílače v podobě VCSEL, a z přijímače který slouží SPAD, každý senzor má vlastní mikrokontrolér. Důležitou vlastností jsou malé rozměry senzorů. Oba senzory jsou schopné pracovat v teplotním rozmezí  $-20$  až  $+70^{\circ}\text{C}$ . a VL6180X, i

VL53L0X jsou vybavené I2C rozhraním, které umožňuje komunikaci senzorů s hlavním mikrokontrolérem. I2C pracuje na frekvenci 400 kHz.

### 1.3.1 VL6180X

Tento optický modul má v sobě tři funkce: snímač přiblížení, snímač okolního světla, světelný zdroj VCSEL, je snadno integrovatelný a má optimalizovanou optickou a mechanickou konstrukci. VCSEL laser vyzařuje vlnu délkou 850nm. Senzor umí měřit vzdálenost do 10 cm, což je pro účel dané bakalářské práce postačující. Je schopný změřit intenzitu osvětlení v rozmezí od 1 Luxu do 100 kLuxů. Na základě změřené vzdálenosti a intenzity osvětlení je senzor schopný rozpoznávat gesta. Provozní napětí pro daný senzor je od 2,6 V do 3 V, ale doporučené napětí je 2,7 až 2,9 V.



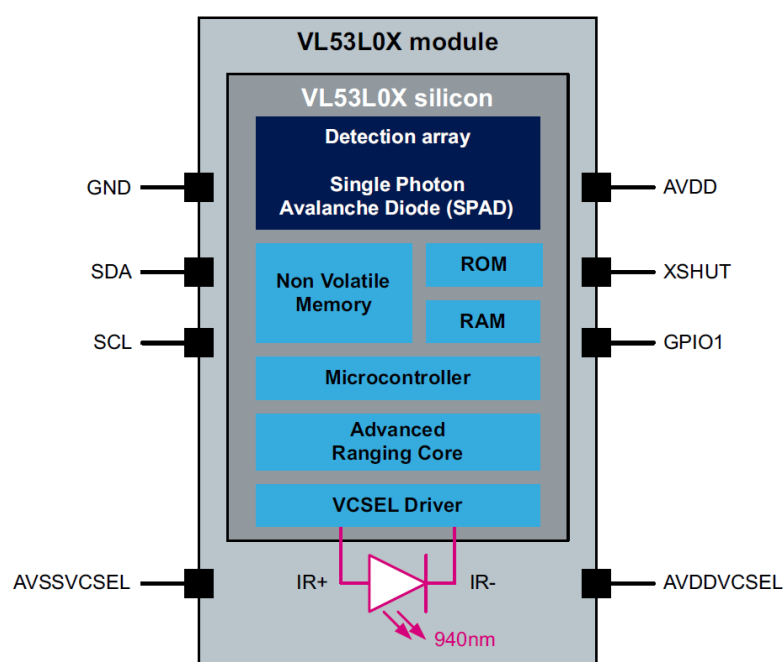
Obr. 6 Blokové schéma senzoru VL6180X [4]

Existuje knihovna poskytnuta výrobcem, která umožňuje práci v režimu měření vzdálenosti nebo intenzity osvětlení, režimu přerušení při měření vzdálenosti, zobrazení změřených hodnot na displeji rozšiřující desky, detekce různých druhů gest. Tato knihovna funguje na bázi STM32CubeHAL. V dané bakalářské práci není knihovna použita. Postačujícím je čtení hodnot vzdálenosti z příslušného registru senzoru a jejich následující zpracování v programu mikrokontroléru. Měření vzdálenosti lze provádět automaticky v intervalech definovaných uživatelem.

VL6180X se používá ve smartphonech, tabletech, notebookách, herních zařízeních v domácích spotřebičích a v průmyslových zařízeních.

### 1.3.2 VL53L0X

VL53L0X je senzor pro měření vzdálenosti a detekci gest. Je složitější a pokročilejší než VL6180X. Vnitřní mikrokontrolér a interní infračervené filtry VCSEL umožňují měření větších vzdáleností a zaručují větší imunitu proti okolnímu osvětlení. Senzor umí měřit vzdálenost do 2 m a funguje i v infračerveném světle. Laser vyzařuje vlnu délkou 940nm. Provozní napětí je od 2,6 V do 3,5 V.



Obr. 7 Blokové schéma senzoru VL53L0X [5]

Výrobce neposkytuje registrovou mapu pro senzor kvůli komplikovanosti. Jediná možnost práce se senzorem je použití API bázi STM32Cube. V dané bakalářské práci jsou použité zdrojové soubory API a komunikace se senzorem je uskutečněná pomocí public metod.

Senzor se používá pro detekci uživatele v PC, notebookách, tabletech, v IoT, v robotice, pro detekci rukou v automatických kohoutcích a v dávkovačích mýdla.

## 1.4 Displej

Existují dvě koncepce zobrazovacích panelů. Jedna je založena na velkých segmentových LED displejích (na Obr. 8) a druhá na vysoce svítivých LED diodách uspořádaných do segmentů (na Obr. 9). Každá koncepce má svoje výhody a nevýhody. Výhodou sedmi-segmentových LED zobrazovačů je pro jednoduchost návrhu desky plošných spojů. Tyto displeje mají většinou nízkou svítivost, která má za přímého slunce přes den, špatnou viditelnost. Tento problém je vyřešen u zobrazovačů s vysoce svítivými LED diodami. Ty jsou ale náročnější na návrh DPS. Jejich výhoda spočívá v tom, že jsou velice dobře čitelné i za přímého slunce.



Obr. 8 Displej ze sedmi-segmentových LED [6]



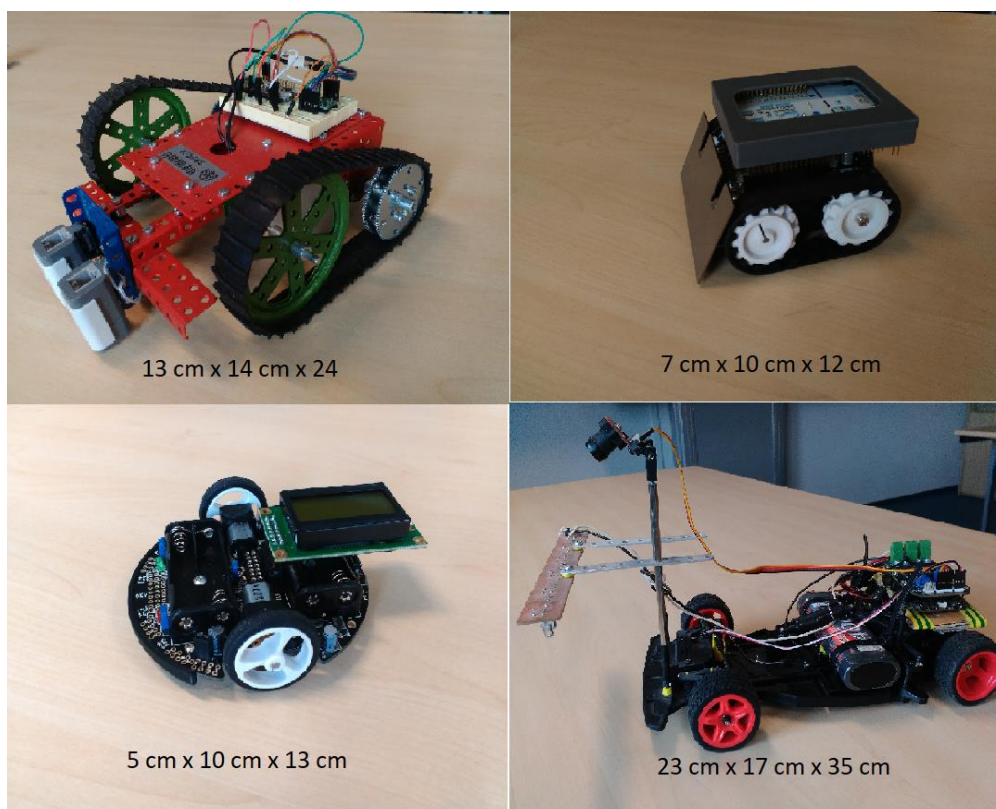
Obr. 9 Displej z vysoce svítivých LED [6]

Jako zobrazovací panel byl zvolen sedmi-segmentový displej, který je součástí rozšiřující desky. Nevýhodou tohoto displeje je malá svítivost a malá velikost číslic, také displej se nachází ze stejné strany jako senzor, což znamená, že úhel pohledu není dostatečně čitelný. To se kompenzuje tím, že čas jízdy bude také zobrazen na PC. Displej je čtyřmístný, což postačí pro daný typ závodu, protože jedná jízda trvá maximálně několik málo minut.

Na rozšiřující desce displej je připojen na expandér, který komunikuje s MCU1 přes stejné I2C jako senzor. To znamená, že se nemusí nastavovat další I2C rozhraní, stačí zapisovat data do displeje přes I2C adresu expandéru.

## 2 Mechanická konstrukce

Je několik variant umístění senzorů vůči robotům: jedná varianta je umístit senzor ze strany, druhá varianta je umístit senzor tak, aby paprsek směřoval dolů. Na *Obr. 10* jsou zobrazené roboti, které se používají ve výuce, s uvedenými rozměry. Vzhledem k tomu, že výška některých robotů dosahuje 23 cm a mohou být i vyšší, je lepší umístit senzor ze strany. Navíc pro umístění senzorů shora by měl být udělán složitější model brány. Mohla by také být navržena brána pro odraz paprsku, ale jelikož je v kódu nadefinována vzdálenost, při které se detekuje průchod robotů, tato brána nemusí být použita.



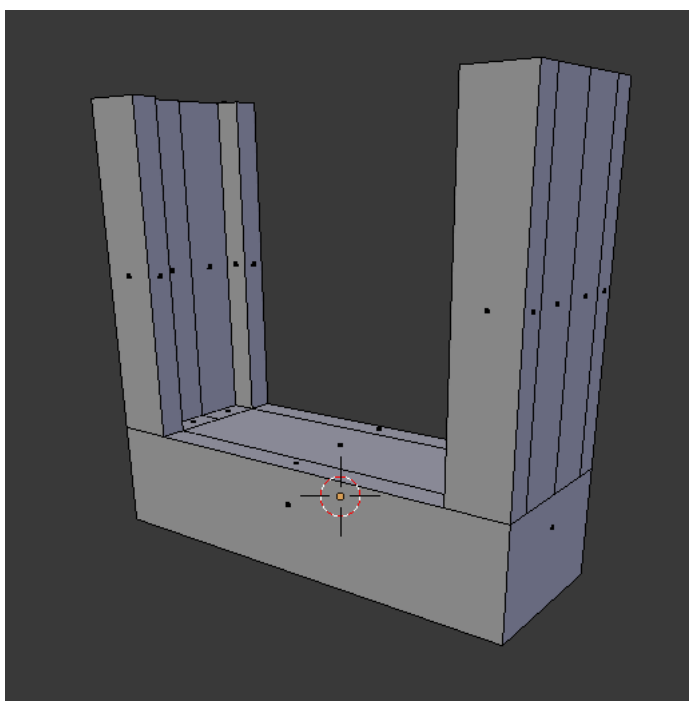
**Obr. 10** Roboti s uvedenými rozměry

Pro uchycení mikrokontroléru v takové poloze, aby paprsek senzoru směřoval paralelně k podlaze, je naprojektován 3D model stojánku. Pro vývoj modelu byl použit program “Blender”. Tento program je bezplatný, existuje instalační a portable verze.

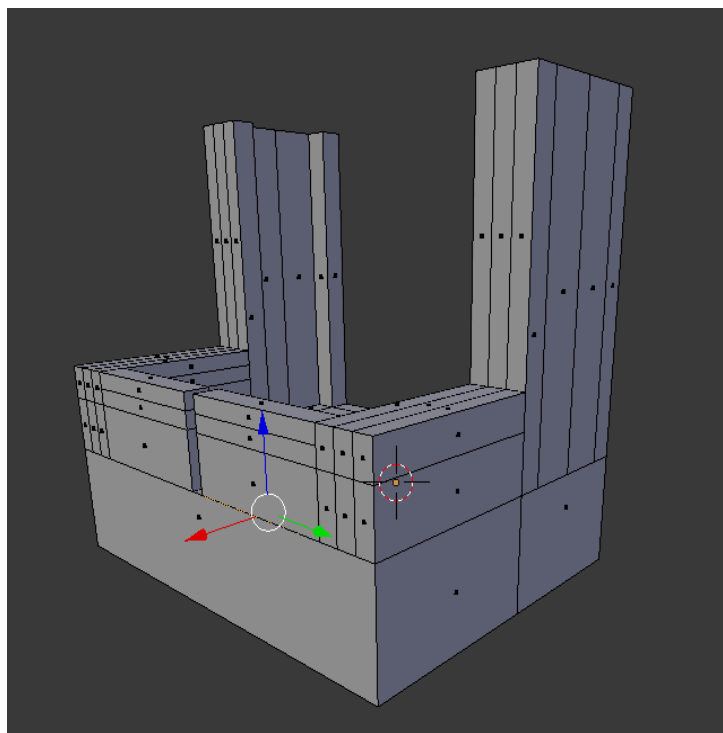
Při otevření programu je defaultně již přidán objekt: krychle. Aby model byl vytisknut správně a materiál byl homogenně rozdělen, konstrukce musí být monolitická. To znamená, že model stojánku musí být udělán s jedné krychle pomocí vytahování a oříznutí, aby se nenarušila geometrie v programu.

Pro MCU1 a MCU2 musí být udělané různé modely. MCU1 má na sobě umístěnou desku se senzorem, MCU2 má senzor připojený na drátech, proto pro MCU2 musí být udělaná dodatečná konstrukce pro zachycení senzoru. Mikro-kontroléry jsou již zabudované do pouzdra vytisknutého na 3D tiskárně, proto se při modelování počítá s rozměrem pouzdra. Celá konstrukce musí být umístěna na podstavci s nožičkami, aby MCU stál stabilně v kolmici k podlaze v této poloze. Jelikož nejvyšší robot je 5 cm vysoký, maximální výška na které může být umístěn senzor je 4,5 cm.

Na *Obr. 11* a *Obr. 12* jsou zobrazené 3D modely pro zachycení mikrokontrolérů.



**Obr. 11** 3D model stojánku pro MCU1



Obr. 12 3D model stojánku pro MCU2

## 3 Nastavení modulů

### 3.1 Použité nástroje

#### 3.1.1 Výběr IDE

Existuje spousta vývojových prostředí pro programování zabudovaných počítačů. Jelikož pro vývoj softwaru v práci a doma používám Microsoft Visual Studio, hledala jsem dostupné řešení pro vývoj pro zabudované systémy v tomto prostředí. Zjistila jsem, že existuje Visual GDB debugger, který umožňuje práci se zvoleným mikrokontrolérem a spoustou jiných modelů ve VS. Na technických fórech jsem přečetla pozitivní zpětnou vazbu o tomto debuggeru. Lidi, které mají zkušenost s Visual GDB ho považují za stabilní, výkonný a mnohofunkční. Výše popsaná kombinace IDE a debuggeru není zadarmo. VS stojí 45 dolarů/měsíc, ale pro studenty je zadarmo, a Visual GDB stojí 99 dolarů/rok s možností bezplatného použití na jeden měsíc.

Alternativou VS je vývojové prostředí založené na Eclipse - Atollic TrueStudio. Dané IDE je určeno speciálně pro STM32, je zadarmo i pro komerční použití. V rámci předmětu Mikroprocesory a počítače jsme vyvíjeli aplikace v tomto prostředí, proto pro vývoj programu bylo zvoleno dané IDE. Atollic má pohodlnou a logickou navigaci, přehledné strukturování kódu, které dělá vývoj pohodlnějším. Atollic umožňuje podívat se do RAM a

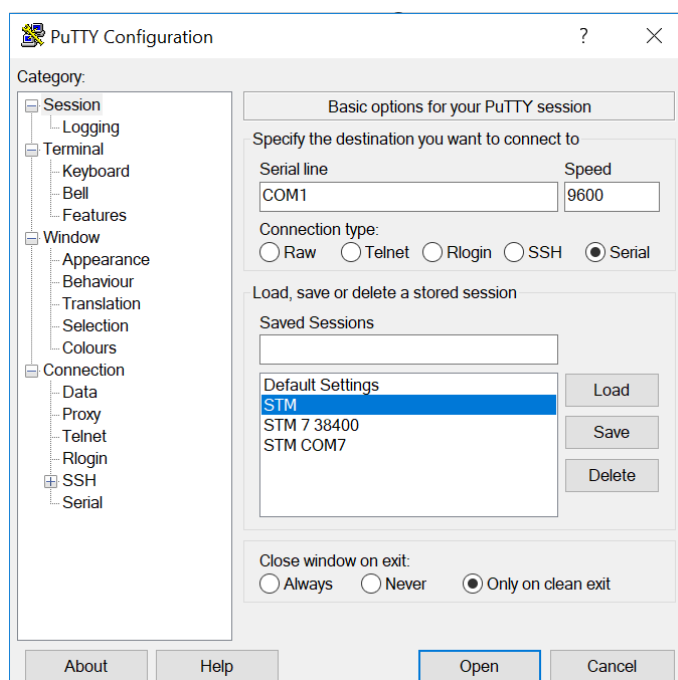


FLASH paměti, zjistit její velikost, kolik z toho je použité a kolik je volné. Také umožňuje výpočet použití stacku na základě zkompilevaného a nalinkovaného programu, což pomáhá v odhalování chyb.

### 3.1.2 PuTTY

Po připojení MCU do PC v počítači by měla proběhnout instalace ovladače pro sériovou komunikaci a automatické přidělení COM portu pro připojenou jednotku. S MCU se dá komunikovat pomocí libovolného terminálového programu. Pro práci byl zvolen program PuTTY. Tento program podporuje komunikaci na základě několika protokolů, například SSH, podporuje sériovou komunikaci. Je dostupný jak pro Windows, tak i pro Linux. Je jednoduchý v použití a přehledný.

Pro nastavení komunikaci v PuTTY se nejdřív musí zjistit který z COM portu je přidělen MCU a na jaké rychlosti probíhá komunikace. COM port se zjistí v okně “Zařízení a tiskárny” ve vlastnostech zvolené jednotky. Také ve vlastnostech se dají zjistit následující parametry: baud rate, počet datových bitů, parita, počet stop bitů. Baud rate je nastaven programově na hodnotu 115200 Bd při inicializaci USARTu.



Obr. 13 PuTTY

V dané bakalářské práci se PuTTY použil jenom pro vývoj: testování hodnot, které se posílaly ze senzoru, MCU1 a MCU2, displeje. Pro zobrazení výsledků závodů byla udělaná okenní aplikace.

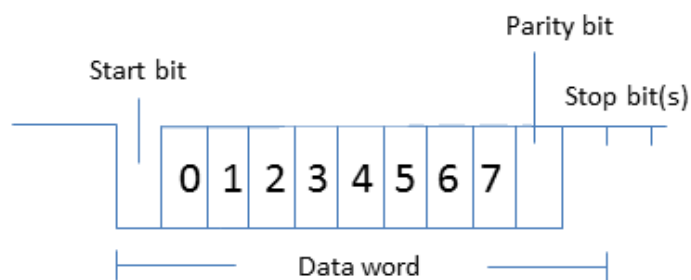
### 3.1.3 Uživatelské rozhraní

Pro vývoj uživatelského rozhraní bylo použito prostředí Microsoft Visual Studio 2017. Jak je již zmíněno, VS je bezplatné pro studenty. Aplikace je napsána v jazyce C# na frameworku Windows Forms. Tento framework umožňuje vytvoření formulářů, přidání do nich takových ovládacích prvků jako tlačítka, textová pole, checkboxy, tabulky atd. Je možné provázání několika formulářů na základě nějaké funkcionality, zobrazení přijatých přes USART dat a posílání dat do MCU. C# je objektově orientovaným jazykem, to znamená, že pracuje s třídami a jejich vlastnostmi, což perfektně popisuje ovládací prvky. K vytvoření formulářových aplikací jsem se naučila v rámci předmětu Programování pro embedded systémy.

## 3.2 USART komunikace mezi mikrokontroléry a PC

Mikrokontroléry série STM32 jsou vybavené interfejsem USART – Universal synchronous and asynchronous receiver-transmitter. Pro tento druh rozhraní je schopen pracovat ve dvou režimech: synchronním a asynchronním. V synchronním režimu je to polovičně duplexní systém, který může komunikovat s A/D a D/A převodníky, sériovou EEPROM atd. Využité jsou dva piny, jeden pro vysílání a příjem dat, a druhý pro časování, tudíž pro synchronizaci.

V asynchronním režimu je to plně duplexní systém. Používá se pro komunikaci s terminály, PC atd. Nejčastěji se používá pro komunikaci s PC na základě protokolu RS-232. Pro komunikaci se využívají dva piny: vysílání a příjem dat. Průběh signálu je zobrazen na *Obr. 14*.

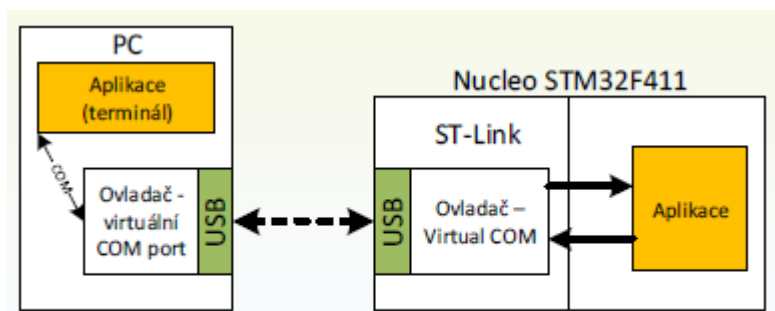


**Obr. 14 Průběh signálu USART [7]**

Synchronizace je zajištěna start a stop bity, pro tento druh se synchronizace nazývá znaková. Délka zprávy může být 5-8 bitů. Počet stop bitů je volitelný, 1 nebo 2. Před stop bitem může být nepovinný paritní bit. Je určený pro kontrolu správnosti vyslaných dat. Ten se generuje ve vysílací a přijímací části a výsledek se porovnává. Když výsledek není shodný, došlo k chybě při přenosu dat. Ale tento způsob není dostatečně spolehlivý, protože detekuje jenom lichý počet chyb.

V mikrokontroléru jsou k dispozici 3 USARTy: USART1, USART2, USART6. USART má široké možnosti nastavení: používá paritu, detekuje několik druhů chyb, generuje přerušení při chybách a při přenosu, má nastavitelnou přenosovou rychlost. USART2 se použije pro komunikaci mezi MCU a PC, USART6 pro komunikaci mezi dvěma MCU. Oba USARTy budou nakonfigurované v asynchronním režimu. USART2 je připojený na APB1 sběrnici, USART6 na APB2 sběrnici.

Pro připojení mikrokontroléru STM32F411RE k PC se nepoužívá protokol RS232. Na straně PC signál je převeden z virtuálního COM portu na USB. Na straně mikrokontroléru ST-Link převádí USB pakety na sériová data. Blokové schéma procesu je uvedeno na *Obr. 15*.



Obr. 15 Proces převezení dat mezi PC a MCU [8]

Výstup a vstup ST-Linku jsou připojené k hlavnímu procesoru na vstup a výstup USARTu. Podle schématu mikrokontroléru dané signály jsou připojené na piny PA3 a PA2. V datasheetu se pro uvedené piny dá použít jako alternativní funkce USART2. Podle datasheetu pin GPIOA3 je určený pro příjem dat, a GPIOA2 pro vysílání.

Pro komunikaci mezi dvěma MCU podle tabulky alternativních funkcí se dá použít USART6. Signál pro vysílání je umístěn na pinu PA11, pro příjem dat na pinu PA12. Nejdřív se musí propojit země obou MCU, a potom se pin pro vysílání dat na MCU1, připojí na pin pro příjem dat na MCU2 a naopak.

Pro inicializaci USARTů se musí provést povolení hodin příslušné periférie, povolení GPIO, a nastavení kontrolních registrů, zapsáním baud rate (kmitočet synchronizačních impulsů) do registru BRR, povolení komunikací v kontrolním registru.

### 3.3 I2C komunikace mezi senzory, displejem a MCU

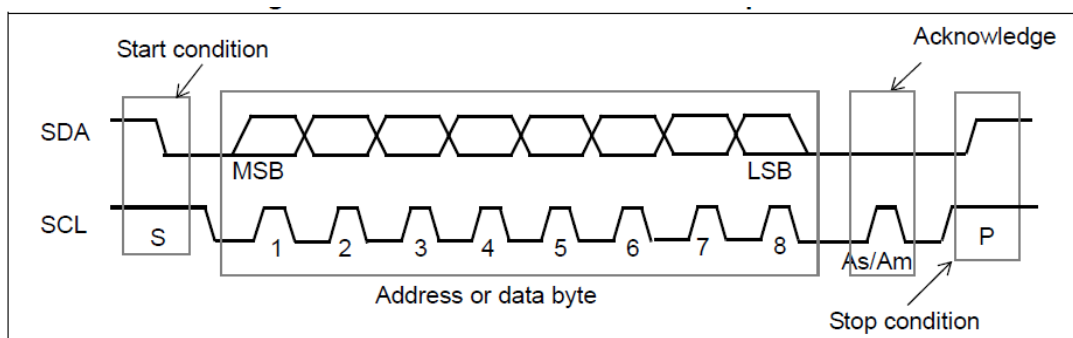
Senzory VL6180X a VL53L0X a také displej jsou vybaveny I2C rozhraním, prostřednictvím kterého je uskutečněna jejich komunikace s mikrokontrolerem.

I2C nebo IIC znamená Inter-Integrated Circuit, multi-masterová počítačová sériová sběrnice. Komunikace po této sběrnici je založená na arbitráži master-slave. Masterem, teda řídicím zařízením je to, které vstoupilo na sběrnici pro vysílání dat. Hodinové impulsy SCL generuje master, datové impulsy SDA generuje master při přenosu dat do slave, nebo slave při přenosu dat do masteru. Před vstupem na sběrnici zařízení kontroluje, jestli sběrnice je v klidu. Master vnucuje na vodič SCL log 1. Při vysílání dat zařízení současně

čte a kontroluje stav na sběrnici. Když slave chce vstoupit na sběrnici posílá log 0. Master vysílá 1, ale čte 0, to znamená, že došlo ke konfliktu. Zařízení, které zjistilo chybu, odstupuje od sběrnici, ale nepřestává přijímat. Při současném vstupu dvou nebo více zařízení na sběrnici se provádí arbitráž podle priority těchto zařízení.

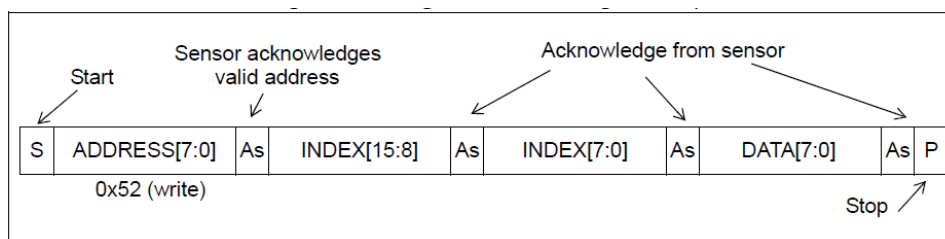
Informace jsou zabaleny v 8bitových paketech (bajtech), potom následuje potvrzovací bit,  $A_s$  pro zprávu od senzoru,  $A_m$  pro zprávu od masteru. Interní data jsou vytvářena vzorkováním datových impulsů SDA na náběžné hraně hodinových impulsů SCL. Externí data musí být stabilní během doby, když SCL je v logické 1. Výjimky jsou start (S) a stop (P) bity, když SDA klesá nebo stoupá, SCL je v logické 1.

První bajt zprávy obsahuje adresu zařízení, například 0x52 pro senzor VL6180X, a také určuje směr dat. Pokud je nejméně významný bit nenastavený (0x52), to znamená, že master zapisuje do senzoru. Je-li tento bit nastavený (0x53), pak je zpráva čtena masterem ze senzoru.



Obr. 16 Průběh signálu I2C [4]

Veškerá komunikace sériového rozhraní se senzorem musí začínat s podmínkou startu. Snímač potvrzuje, že obdržel platnou adresu tím, že nastaví SDA do logické 0. Nejméně významný bit datové zprávy určuje, jestli se data čtou nebo zapisují. Během sekvence zápisu druhý a třetí přijatý bajty poskytují 16bitový index, který ukazuje na jeden z interních 8bitových registrů.



Obr. 17 Struktura I2C zprávy [4]

Jak jsou data přijímána podřízeným zařízením, tak jsou i zapisována bit po bitu do sériového/paralelního registru. Poté, co byl každý datový bajt přijat podřízeným zařízením, je generováno potvrzení, data jsou pak uložena ve vnitřním registru adresovaném aktuálním indexem.

Během čtení zprávy obsah aktuálního registru je čten za bajtem adresy zařízení. Obsah tohoto registru je paralelně načítán do sériového/paralelního registru a vyřazen ze zařízení doběžnou hranou SCL.

Na konci každého bajtu, v sekvencích čtení i zápisu zpráv, přijímající zařízení vydá potvrzení (tj. senzor při zápisu a master při čtení). Zpráva může být ukončena pouze masterem sběrnice, a to buď vydáním podmínky zastavení, nebo záporným potvrzením po přečtení celé zprávy.

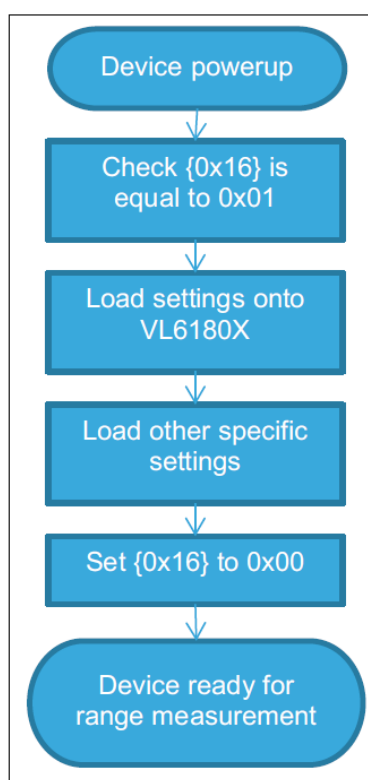
Rozhraní také podporuje automatickou inkrementaci indexování. Poté, co byl první datový bajt přenesen, je index automaticky zvýšen o 1. Master proto může vysílat datové bajty nepřetržitě do podřízeného zařízení, dokud podřízené zařízení nepřestane poskytovat potvrzení nebo master ukončí zápis dat s podmínkou zastavení. Pokud je použita funkce automatické inkrementace, master nemusí posílat indexy adres s datovými bajty.

Pro nastavení komunikace přes I2C do projektu byla implementovaná knihovna vytvořena panem Ing. Weissarem Ph.D, která se používá ve výuce v rámci předmětu Mikroprocesory a počítače. Knihovna obsahuje funkce pro inicializaci, reset, zahájení a ukončení komunikace, pro zápis a čtení dat přes I2C. Displej používá jednobajtovou adresu pro komunikaci, ale senzory používají dvoubajtové adresy, proto knihovna byla doplněná o funkce pro práci s dvoubajtovými adresy.

### 3.4 Nastavení VL6180X

První věc, která musí být udělaná pro uvedení do provozu VL6180X je připojení napájení. Vývody AVDD a AVDD\_VCSEL musí být napájené 2,8 volty, AVSS a AVSS\_VCSEL musí být připojené k zemi. Jelikož senzor je součástí rozšiřující desky, všechny vývody již jsou správně napojené a senzor je napájen z mikrokontroléru.

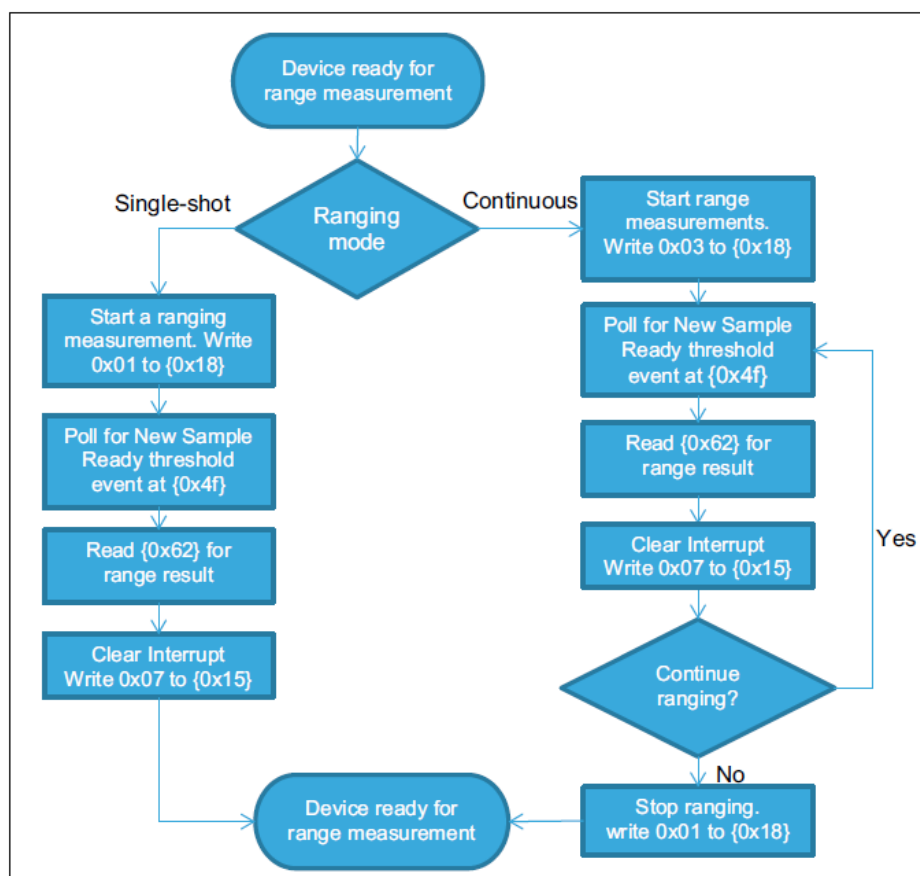
Dále musí proběhnout inicializace. Nejdřív musí být ověřené, jestli registr SYSTEM\_FRESH\_OUT\_OF\_RESET je nastaven na defaultní hodnotu 0x01. Tento registr se používá pro ověření podmínky resetu. Potom musí být nastavené další registry, které určují konfiguraci senzoru. Pro nastavení těchto registrů byly použité hodnoty doporučené v Application note. Potom se musí do registru SYSTEM\_FRESH\_OUT\_OF\_RESET zapsat 0x00, což pomůže hostiteli určit zda konfigurace byla správně načtená. VL6180X je připravený k provedení měření.



Obr. 18 Inicializační sekvence senzoru VL6180X [9]

Měření může být provedeno ve dvou režimech: Single-shot a Continuous. V režimu Single-shot senzor provádí jedno měření a jde do pohotovostního režimu softwaru. V režimu Continuous senzor provádí jedno měření za druhým s intervalem určeným

uživatelé až do vyskytnutí příkazu k zastavení. Pro realizaci časomíry, senzor musí pracovat v režimu Continuous.



Obr. 19 Algoritmus provedení měření vzdálenosti senzorem VL6180X [9]

Nejdříve musí být ověřeno, jestli je zařízení připraveno k provedení měření. Nulový bit registru `RESULT__RANGE_STATUS` musí být nastaven, to znamená, že senzor je připraven k příkazu začít měření. Měření se spustí zapsáním hodnoty `0x03` do registru `SYSRANGE__START`. Dále se musí počkat až senzor bude připraven k provedení dalšího měření. V Application note to je realizováno ve while smyčce, která blokuje provedení jiných částí programu, proto tato část kódu byla přepsaná na if podmínku. Hodnota vzdálenosti v mm se vyčítá z registru `RESULT__RANGE_VAL`. Zapsáním této hodnoty `0x07` do registru `SYSTEM__INTERRUPT_CLEAR` vyčistí status přerušení. Popsané příkazy probíhají v hlavní while smyčce. Měření lze zastavit zapsáním hodnoty `0x01` do registru `SYSRANGE__START`.



### 3.5 Nastavení VL53L0X

Daný senzor má 6 vývodů: VDD, VIN, GND, SDA, SCL, XSHUT, GPIO1. Senzor může být napájen napětím od 2,6 V do 3,5 V, proto vývod VIN se připojí na pin mikrokontroléru s napětím 3,3, a GND se připojí na zem. SCL a SDA jsou vývody určené pro I2C komunikaci. Té vývody se připojí na piny PB8 a PB9. Podle tabulky alternativních funkcí vývodu na těchto pinech se dá použít I2C1.

Pro práci s tímto senzorem bylo použito API poskytnuté výrobcem. Zdrojové soubory byly přesunuté do projektu v Atollic. Původně výrobcem bylo zamyšlené, že některé procesy budou probíhat v PC, a pak vypracovaná data budou poslána zpátky do MCU2. Některé funkce proto byly změněné tak, aby všechny procesy probíhaly v MCU2, je to logičtější a rychlejší.

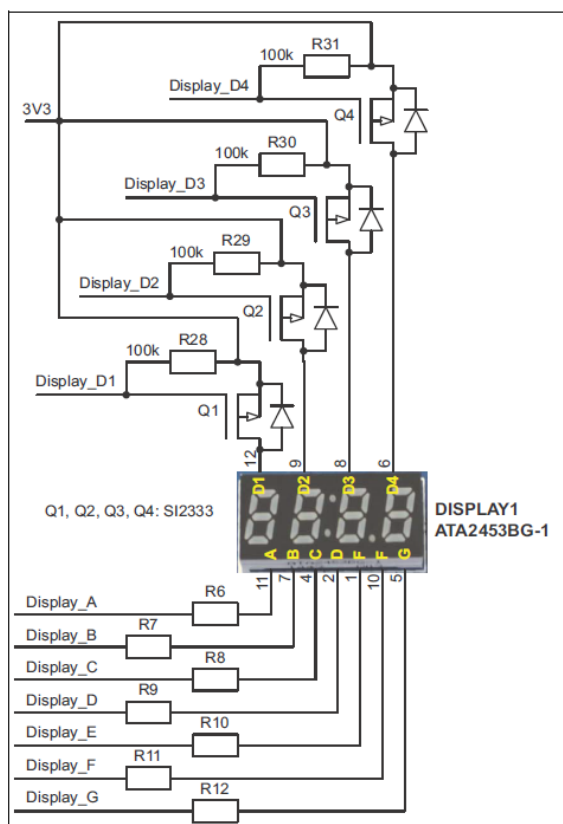
Nejdřív musí proběhnout inicializace senzoru. Stačí jenom zavolat funkce pro inicializaci a kalibraci dostupné v API. Pro měření vzdálenosti stačí zkontrolovat, jestli se nestala chyba a zavolat funkci pro provedení měření. Tato funkce se volá v hlavní nekonečné while smyčce s nastavitelným intervalem. Interval je odpočítán časovačem SysTick.

### 3.6 Nastavení displeje

Displej ATA2453BG1 umístěný na rozšiřující desce je připojen na expandér, který komunikuje s MCU1 přes stejné I2C jako senzor. To znamená, že se nemusí nastavovat další I2C rozhraní, stačí zapisovat data do displeje přes I2C adresu expandéru.

Nejdřív se musí nastavit všechny bity v registrech 0x14 a 0x15 do logické 1, dané registry znamenají směr přenosu informace. V datových registrech 0x12 a 0x13 se nastaví které segmenty a které cifry musí být rozsvícené. Bity D1-4 určují číslo cifry a bity A-G určují, které segmenty v zvolené cifře budou rozsvícené. Takový způsob uspořádání pomáhá minimalizovat počet vývodu a zjednodušit návrh desky plošného spoje. Jelikož pro připojení displeje se používá technologie PMOS, pro rozsvícení segmentů a cifer se do příslušných bitů musí zapsat logická 0.

Během vývoje jsem se setkala s problémem nastavení displeje, proto používám ho pouze pro zobrazení stavu závodu, když závod probíhá jsou rozsvícené pomlčky.



Obr. 20 Ovládaní displeje [3]

### 3.7 Nastavení časovače

Pro zaznamenání času byl zvolen časovač SysTick, protože je jednoduchý v nastavení, přesný, a dá se ho případně použít pro více účely než jenom záznam času jízdy.

Všechny procesory Cortex M4 jsou vybavené časovačem SysTick, který je součástí jádra. Jelikož SysTick běží v hodinách procesoru, pokud je signál hodin nastaven pro režim nízké spotřeby, čítač SysTick se zastaví.

SysTick je 24bitovým časovačem, který počítá dolů od hodnoty zapsané v registru STK\_LOAD do nuly, při přechodu z 0 načte hodnotu z registru STK\_LOAD a poté opět počítá dolů. Největší hodnota, která se může nastavit do registru STK\_LOAD je 16 777 215.

Pro práci s tímto časovačem se nejlíp využije přerušení, které se zavolá pokaždé co čítač dopočítá do 0. Univerzálním řešením je nastavit interval přerušení na každou milisekundu.

Pro nastavení časovače se nejdřív musí určit skutečná hodinová frekvence procesoru. Výrobce poskytuje knihovny CMSIS (Cortex Microcontroller Software Interface Standard) pro usnadnění práci s jádrem. Součástí těchto knihoven je globální proměnná `SystemCoreClock` do které je zapsána skutečná frekvence. Pro použitý procesor to je 16 MHz. Po změně nastavení hodin se hodnota této proměnné se musí zaktualizovat pomocí funkce `SystemCoreClockUpdate()`. Je také doporučeno vždy aktualizovat danou proměnnou na začátku programu.

Ovladače CMSIS poskytují funkci `SysTick_Config()`, která se používá k nastavení SysTicku. Do registru `STK_LOAD` se zapíše hodnota od které časovač bude počítat dolů. Registr `STK_VAL` obsahuje informaci o aktuální hodnotě čítače, při nastavení se do toho registru zapíše 0. V řídicím registru `STK_CTRL` se dá zjistit příznak dopočítání `COUNTFLAG`, určit zdroj hodin AHB nebo AHB/8 v `CLKSOURCE`, povolit generování přerušení v `TICKINT`, povolit čítání SysTick čítače v `ENABLE`. Toto je všechno co se musí udělat pro nastavení SysTicku.

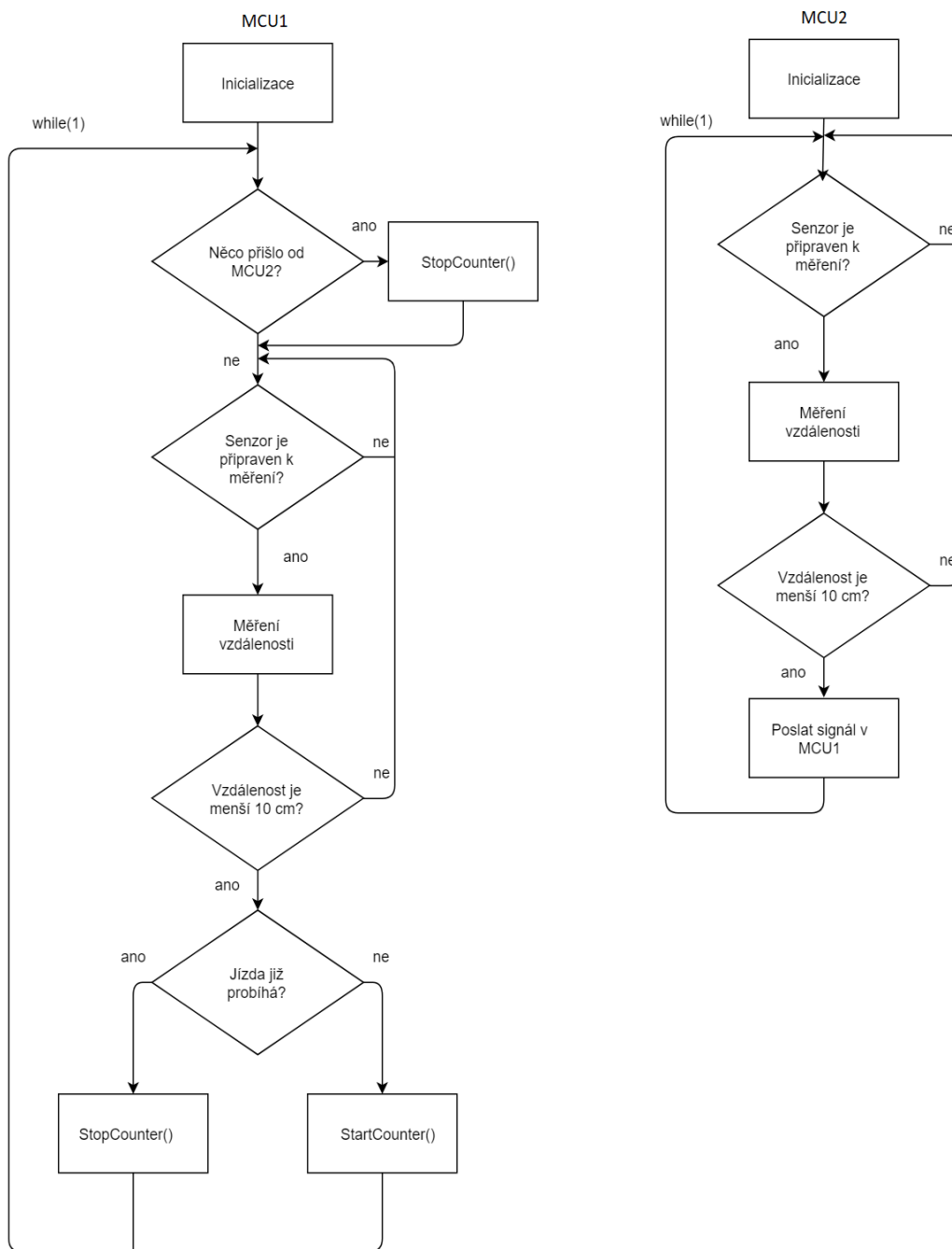
Pro obsluhu přerušení v hlavním souboru se musí nadefinovat funkce `SysTick_Handler()`. Jméno funkce musí být stejné jako u defaultní funkce pro obsluhu přerušení SysTicku. V tabulce vektorů adresa defaultní funkce označená příznakem `.weak`, znamená, že linker použije adresu funkce s hlavního souboru.

Ve funkci `SysTick_Handler()` probíhá implementace globální proměnné `_ticks`, která je nadefinována jako `volatile uint32_t`. `Volatile` znamená, že proměnná může být změněná kdekoliv v kódu, `uint32_t` znamená, že 32bitový čítač bez znaménka, při implementaci každou milisekundu, čítač nepřekročí maximální hodnotu přibližně 50 dní.

## 4 Návrh SW

### 4.1 Popis principu SW realizace časomíry

Blokové schéma na Obr. 21 ukazuje algoritmus kódu pro MCU1 a MCU2.



Obr. 21 Blokové schéma algoritmu

V MCU1 se nejdřív musí provést inicializace časovače, USART2 pro komunikaci s počítačem, USART6 pro komunikaci s MCU2, I2C pro komunikaci s displejem a senzorem, displej, senzor, použité GPIO. Poté co senzor je připravený provádět měření, začne měřit vzdálenost v hlavní nekonečné while smyčce. Až změřená vzdálenost bude menší 10 cm, zkontroluje se stav globální proměnné competition. Proměnná competition znamená, jestli je jízda již probíhá. Když je competition nastavená do stavu false, znamená, že to byl první průchod bránou a zavolá se funkce StartCounter(). V této funkci

se vynuluje globální proměnná `SysTicku _ticks` a její hodnota se bude inkrementovat s každou milisekundou prostřednictvím přerušení, `competition` se nastaví do `true`, signál o začátku jízdy se pošle do PC. Když je `competition true`, znamená to, že jízda již probíhá. Zavolá se funkce `StopCounter()`. Aktuální hodnota `_ticks` se запиše do pomocné proměnné `time`, `competition` a nastaví se do stavu `false`. Pomocná proměnná je zapotřebí protože `SysTick` pořád počítá a musí se zachytit hodnota `_ticks` v určitý okamžik. Hodnota, která je zapsaná do `time` znamená čas jízdy a data se pošlou do PC.

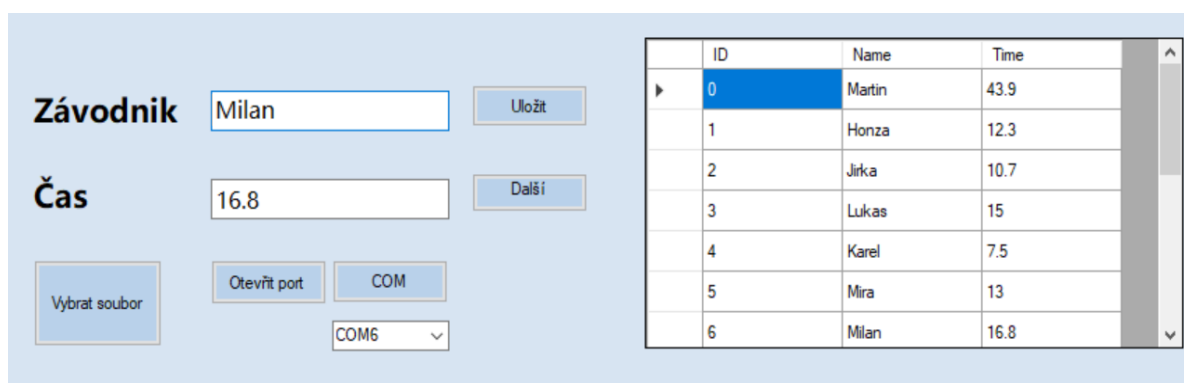
V MCU2 se musí inicializovat `USART6` pro komunikaci s MCU1 a `I2C` pro komunikaci se senzorem, sensor. Princip je obdobný jako v MCU1. Poté co se všechno inicializovalo, sensor začne měřit vzdálenost. Při zmenšení změřené vzdálenosti pod 10 cm do MCU1 se pošle znak přes `USART6`, a až MCU1 znak obdrží, zavolá se funkce `StopCounter()` v MCU1.

Nastavit variantu “okruh” nebo “start-cíl” se dá přepnutím tlačítka na MCU1. Když je nastavená varianta “okruh”, v MCU1 se pak nebude provádět kontrola, jestli něco přišlo přes `USART6`.

Vytvořený algoritmus je jednoduchý, kód je přehledný, rozdělený do různých souborů a funkcí podle funkcionality, kterou vykonává. Obsahuje minimální počet `if` podmínek, nemá smyčky, které by blokovaly provedení jiných částí kódu, a je univerzální pro obě varianty.

## 4.2 Uživatelské rozhraní

Pro přehledné zobrazení informací o závodě byla udělaná formulářová aplikace v jazyce C#. Snímek obrazovky aplikace je na *Obr. 22*.



ID	Name	Time
0	Martin	43.9
1	Honza	12.3
2	Jirka	10.7
3	Lukas	15
4	Karel	7.5
5	Mira	13
6	Milan	16.8

Obr. 22 Snímek obrazovky formulářové aplikace

Do textového pole se zadá jméno závodníka, v poli “Čas” bude zobrazen aktuální čas jízdy. Je možné uložit data o závodě do CSV souboru. Nejdřív se pomocí tlačítka “Vybrat soubor” se musí zvolit do kterého souboru se data budou ukládat. Data z nahraného souboru se zobrazí v tabulce. Při stisknutí tlačítka “Uložit” do tabulky se přidá další řádek s jménem a časem jízdy aktuálního závodníka. Tlačítkem “Další závodník” se vyčistí políčka “Závodník” a “Čas” a pak může být začata nová jízda. Důležitým je ovládací prvek USART, právě díky němu je umožněná komunikace s MCU1. Pro začátek komunikace se musí prozkoumat dostupné COM porty pomocí tlačítka “COM”, možné varianty se objeví v seznamu pod tlačítkem. Tlačítkem “Otevřít port” se otevře zvolený COM.

## Závěr

Úkolem této práce bylo navrhnout zařízení, které umožňuje měřit čas, po který roboti závodí. Časomíra umožňuje provedení dvou druhů závodů: “okruh” a “start-cíl”. To je realizováno pomocí dvou mikrokontrolérů, ke kterým jsou připojeny senzory vzdálenosti. Senzory jsou schopné měřit vzdálenost dostatečně rychle, aby detekovali průjezd robotů. Přesnost měření není závislá na okolních podmínkách, protože senzory jsou založené na technologii “Time of flight”. Všechny prvky časomíry odpovídají požadavkům, kromě displeje. Displej je malý a je špatně umístěn. Navíc, během vývoje jsem se setkala s problémem nastavení displeje, z tohoto důvodu je používán pouze pro zobrazení informace o tom, jestli momentálně probíhá jízda. Pro přehledné zobrazení informace o závodu a její zpracování byla vyvinutá formulářová aplikace v jazyce C#. Tato aplikace umožňuje zobrazení aktuálního času jízdy, zadání jména závodníka, uložení výsledků do souboru a zobrazení výsledků všech závodníků v tabulce přímo v okně aplikace. Aplikace by mohla být doplněná o možnost zobrazení informace o stavu senzorů a provedení jejich resetu.

Mikrokontrolér se zachycuje v takové poloze, aby paprsek ze senzoru směřoval na robota ze strany, a právě proto byly udělané 3D modely stojánku pro MCU1 a MCU2. Senzory jsou umístěny v takové výšce, že zařízení je schopné detekovat i nejmenší z robotů, které se používají ve výuce. Stojánky mají jednoduchou monolitickou konstrukci.

Mezi dvěma MCU a MCU1 a PC je nastavená USART komunikace pro sdílení dat o průběhu závodu. V realizovaném modelu tato komunikace probíhá přes kabel. Pro vylepšení konstrukce časomíry by mohla být použita komunikace přes Bluetooth nebo WiFi a to by umožňovalo PC a obou MCU, aby se mohly nacházet několik metrů daleko od sebe, konstrukce by byla snadněji přemístitelná.

Vyvinutá časomíra může být používána ve výuce a soutěžích spolupořádaných FEL KAE.

## Seznam literatury a informačních zdrojů

- [1] UM1724. STM32 Nucleo-64 board user manual. [online]. [citováno 2019-06-10]. Dostupné z:  
[https://www.st.com/content/ccc/resource/technical/document/user\\_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf](https://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf)
- [2] Pinker, J.: “Mikroprocerory a mikropočítače”, Praha: BEN - technická literatura, 2004, ISBN 80-7300-110-1
- [3] UM1852. User manual X-NUCLEO-6180XA1 proximity and ambient light sensor expansion board based on VL6180X for STM32 Nucleo . [online]. [citováno 2019-06-10]. Dostupné z:  
[https://www.st.com/content/ccc/resource/technical/document/user\\_manual/5a/d3/5b/bd/14/82/46/1f/DM00154104.pdf/files/DM00154104.pdf/jcr:content/translations/en.DM00154104.pdf](https://www.st.com/content/ccc/resource/technical/document/user_manual/5a/d3/5b/bd/14/82/46/1f/DM00154104.pdf/files/DM00154104.pdf/jcr:content/translations/en.DM00154104.pdf)
- [4] VL6180X Datasheet. [online]. [citováno 2019-06-10]. Dostupné z:  
<https://www.st.com/resource/en/datasheet/vl6180x.pdf>
- [5] UM2046. VL53L0X ranging and gesture detection sensor. [online]. [citováno 2019-06-10]. Dostupné z:  
[https://www.st.com/content/ccc/resource/technical/document/user\\_manual/group0/44/17/0e/ed/f7/51/44/67/DM00285084/files/DM00285084.pdf/jcr:content/translations/en.DM00285084.pdf](https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/44/17/0e/ed/f7/51/44/67/DM00285084/files/DM00285084.pdf/jcr:content/translations/en.DM00285084.pdf)
- [6] NOVOBILSKÝ, Petr. Bakalářská práce. Systém elektronické časomíry s bezdrátovým připojením terčů a hlavního zobrazovače. BRNO 2009. 42 s.
- [7] USART vs UART: Know the difference. [online]. [citováno 2019-06-10]. Dostupné z: <https://www.radiolocman.com/review/article.html?di=164400>
- [8] KAE/MPP – domácí příprava. [online]. [citováno 2019-06-10]. Dostupné z: <https://portal.zcu.cz/portal/studium/courseware/kae/mpp/cviceni/index.html>
- [9] AN4545. Application note. [online]. [citováno 2019-06-10]. Dostupné z: [https://cdn.sparkfun.com/datasheets/Sensors/Proximity/VL6180\\_ApplicationNote.pdf](https://cdn.sparkfun.com/datasheets/Sensors/Proximity/VL6180_ApplicationNote.pdf)
- [10] Kudera , Tomáš. Bakalářská práce. Zařízení pro měření reálného času s mikrokontrolérem. Plzeň 2015. 60 s.
- [11] VL53L0X Datasheet sensor. [online]. [citováno 2019-06-10]. Dostupné z: <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>
- [12] STMPE1600. Expander datasheet. [online]. [citováno 2019-06-10]. Dostupné z: <https://www.st.com/resource/en/datasheet/stmpe1600.pdf>
- [13] PM0214. STM32F3, STM32F4 and STM32L4 Series, Cortex®-M4 programming manual. [online]. [citováno 2019-06-10]. Dostupné z:  
[https://www.st.com/content/ccc/resource/technical/document/programming\\_manual/6c/3a/cb/e7/e4/ea/44/9b/DM00046982.pdf/files/DM00046982.pdf/jcr:content/translations/en.DM00046982.pdf](https://www.st.com/content/ccc/resource/technical/document/programming_manual/6c/3a/cb/e7/e4/ea/44/9b/DM00046982.pdf/files/DM00046982.pdf/jcr:content/translations/en.DM00046982.pdf)
- [14] NUCLEO-F401RE. [online]. [citováno 2019-06-10]. Dostupné z: <https://os.mbed.com/platforms/ST-Nucleo-F401RE/>