

Mikroprocesorové a vestavěné systémy
Měření vzdálenosti laserovým senzorem

Obsah

1. Úvod	3
2. Hardwar	3
2.1. ESP32.....	3
2.2. Laserový senzor vzdálenosti GY-VL53LOX I2C	4
2.3. Modul displeje OLED 0,96`` I2C	4
2.4. Rotační encoder KY-040.....	4
2.5. Příklad připojení.....	5
3. Implementace	6
3.1. Popis.....	6
3.2. Struktura projektu.....	6
3.3. Funkčnost.....	6
3.3.1. Výpočet vzdálenosti.....	6
3.3.2. Menu.....	7
3.3.3. Statistiky	7
3.3.4. Mixin	8
3.3.5. Go out	8
3.4. Problémy.....	8
3.5. Spuštění.....	9
3.6. Videoukazka	10
4. Výsledek	11
5. Bibliografie	11

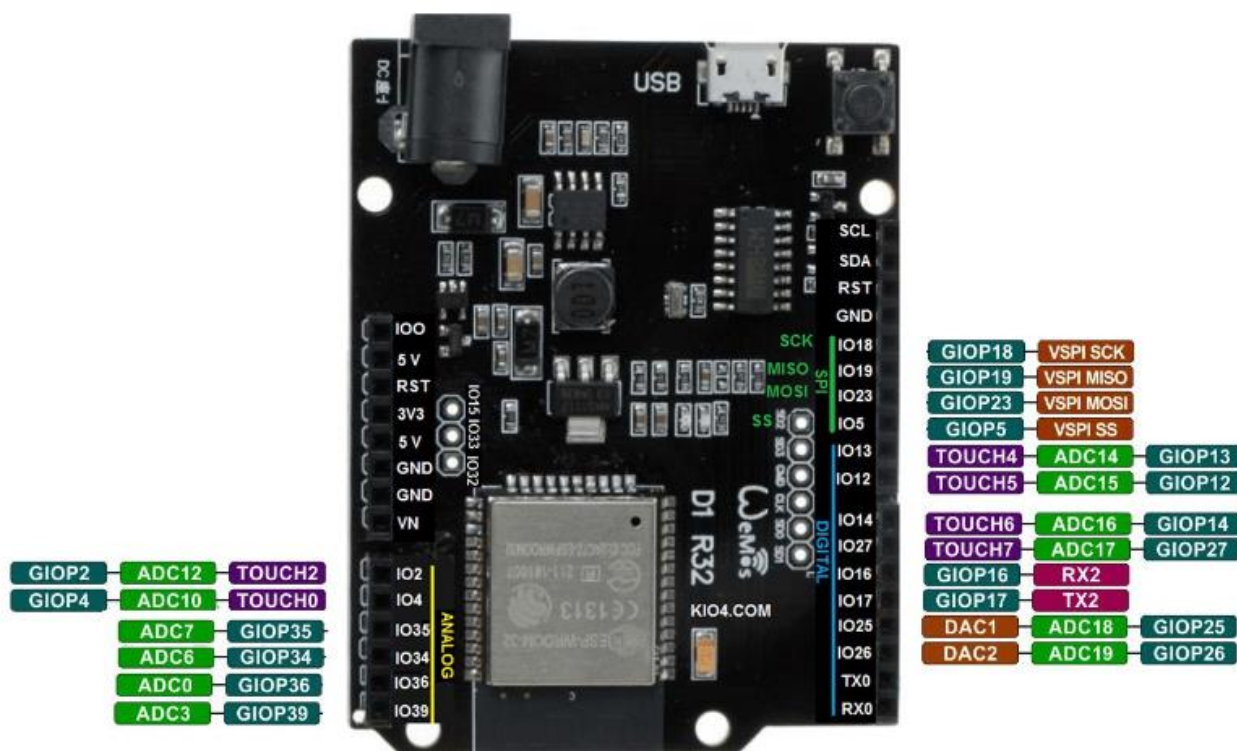
1. Úvod

Úkolem mého projektu ve kurzu IMP bylo implementovat vestavný program pro měření vzdálenosti laserovým senzorem na základě poskytnutého hardwaru (ESP-32, Laserový senzor vzdálenosti GY-VL53L0X I2C, OLED displej, KY-040 a deska Wemos D1).

2. Hardwar

K zadání byl připojen popis hardwarových jednotek projektu. Bez pochopení jejich obecné struktury a principu fungování není možné provádět další práci. Proto se budu zabývat každou z nich zvlášť a popíšu princip jejich propojení.

2.1. ESP32



Výkonný, energeticky úsporný dvoujádrový mikrokontrolér vyvinutý společností Espressif Systems. Je vybaven vestavěnými moduly Wi-Fi a Bluetooth, takže je ideální pro projekty internetu věcí (IoT). ESP32 podporuje různá rozhraní, jako jsou SPI, I2C, UART, PWM a další, což poskytuje flexibilitu pro vývoj různých aplikací. Mikrokontrolér má vysoký výkon, spoustu paměti a různé periferie, což umožňuje jeho použití v komplexních systémech, jako jsou chytrá zařízení, senzory a řídicí systémy. V kontextu tohoto projektu je ústředním prvkem, řídicí strukturou, ke které budou připojeny všechny níže uvedené periferie. [\[1\]](#)

2.2. Laserový senzor vzdálenosti GY-VL53LOX I2C

Modul obsahuje integrovaný laserový senzor vyrobený společností STMicroelectronics. Je založen na technologii Time-Of-Flight (ToF): měření vzdálenosti pomocí pulzu infračerveného světla a odečítání času do jeho zpětného odrazu. [\[2\]](#)

Protokol: I2C (výchozí 0x29)

I/O	Описание	Подключение к ESP-32
VCC	Napájení (3-5V)	VCC(5V)
GND	Uzemnění	GND
SCL	Rozhraní I2C pro časovač	SCL
SDA	Rozhraní I2C pro data	SDA

2.3. Modul displeje OLED 0,96`` I2C

Kompaktní displej na technologii organických LED OLED určený k zobrazování jednoduché grafiky a řadič SSD1306 k ovládání displeje. [\[3\]](#)

Velikost: 0,96".

Rozlišení: 128x64 px.

Protokol: I2C (výchozí 0x3C)

I/O	Ропис	Рріпожені к ESP-32
VCC	Napájení (3.3-5V)	VCC(5V)
GND	Uzemnění	GND
SCL	Rozhraní I2C pro časovač	SCL
SDA	Rozhraní I2C pro data	SDA

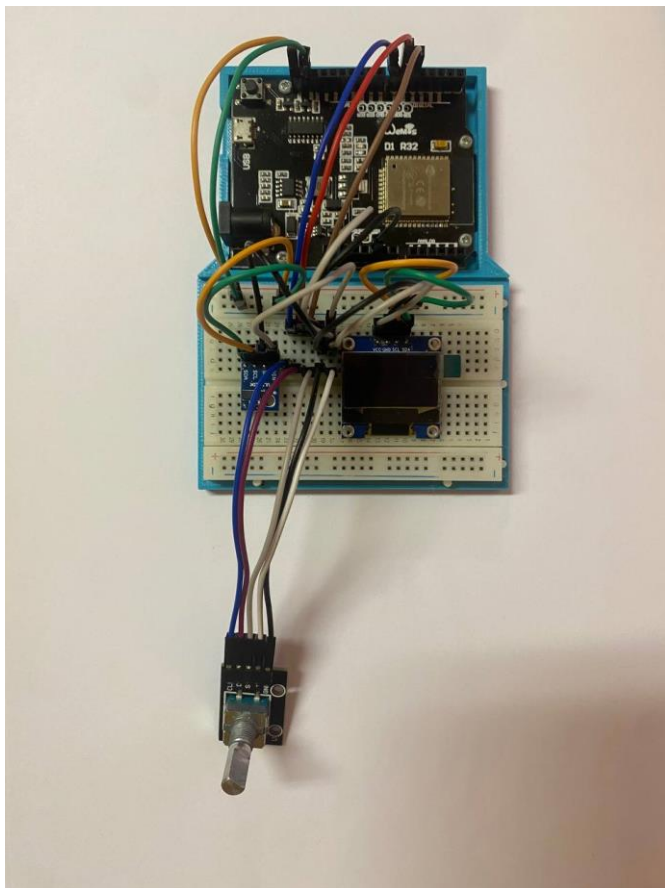
2.4. Rotační encoder KY-040

Poskytuje možnost snímání směru a úhlu natočení rotačního prvku encoderu ve formátu kroků, jakož i stisknutí vestavěného tlačítka pro další přenos do vnějšího prostředí. Na základě generování impulsů a fázového posunu s ohledem na dva kanály (A a B), jejichž rozdíl určuje směr otáčení. [\[4\]](#)

Protokol: není podporován. Výstupní signál realizován.

I/O	Ропис	Рріпожені к ESP-32
+	Napájení (3.3-5V)	VCC(5V)
GND	Uzemnění	GND
SW	Výstup tlačítka	digital pin IO16
DT	Kanál pro detekci směru	digital pin IO27
CLK	Kanál pro generování impulsů	digital pin IO14

2.5. Příklad připojení



3. Implementace

3.1. Popis

Pro svůj projekt jsem napsal softwarové řešení založené na frameworku Arduino, který je podporován na platformě ESP32. Framework se syntaxí podobnou jazyku C poskytuje pohodlná vysokoúrovňová řešení, díky nimž je proces implementace intuitivní a v mnoha ohledech flexibilní. Jako vývojové prostředí jsem zvolil PlatformIO (rozšíření pro VS Code), protože je pohodlně integrováno s IDE, které jsem již znal, a nevyžaduje složitou dodatečnou konfiguraci a instalaci. Níže jsou uvedeny knihovny použité při implementaci:

- **Adafruit_VL53L0X** – knihovna pro řízení činnosti senzoru VL53L0X. [\[5\]](#)
- **Adafruit_SSD1306** -- knihovna pro ovládání řadiče OLED displeje SSD1306. [\[6\]](#)
- **Adafruit_GFX** – univerzální knihovna pro práci s grafikou na displejích (propojeno s Adafruit_SSD1306). [\[7\]](#)
- **Wire** – knihovna pro I2C.
- **Adafruit_BusIO** – knihovna sběrnice I2C (nutná pro integraci Adafruit_GFX). [\[8\]](#)

3.2. Struktura projektu

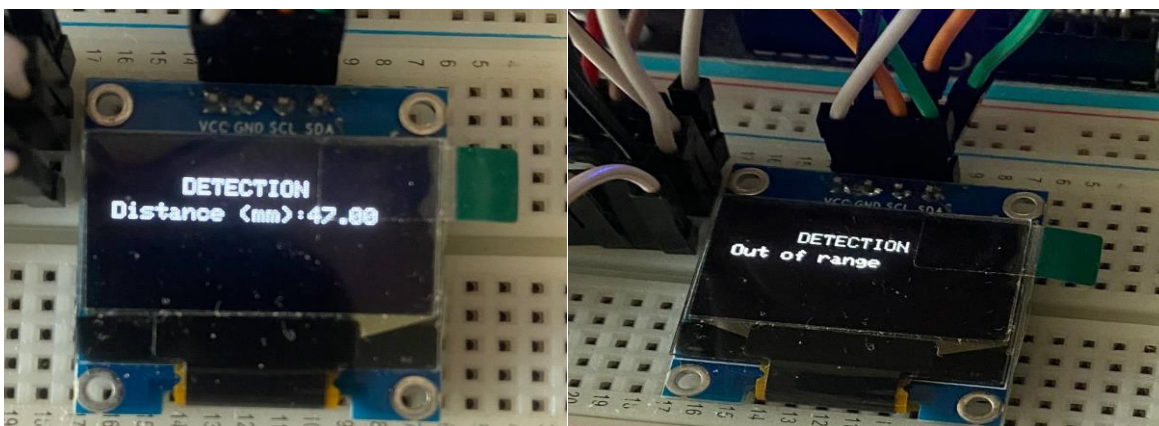
- **platformio.ini** – konfigurační soubor projektu v prostředí PlatformIO (VSCode).
- **./test** – testovací adresář. Nepoužívá se.
- **./src** – adresář se zdrojovým kódem. V mém projektu je zdrojový kód uveden v souboru *src/main.cpp*, který si můžete přečíst, abyste viděli konečnou implementaci..
- **./lib** и **./include** – adresáře pro ruční přidání knihoven třetích stran. Nepoužívá se.
- **./vscode** -- adresář pro integrační soubory PlatformIO a VS Code.
- **./pio** – automaticky vygenerovaný adresář po nastavení projektu ve vývojovém prostředí. Automaticky načte závislosti frameworku a projektu a obsahuje výsledky kompilace..

3.3. Funkčnost

Funkce, které můj program poskytuje, lze rozdělit do částí:

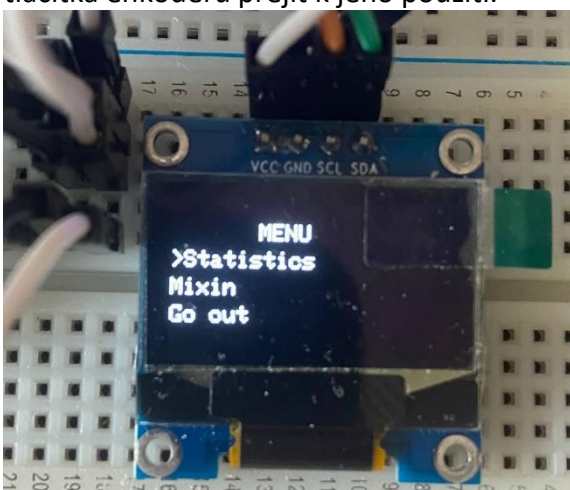
3.3.1. Výpočet vzdálenosti

Po úspěšném spuštění programu na hardwaru může uživatel komunikovat s laserovým senzorem, výstup výpočtu se zobrazí na displeji (standardně v milimetrech) nebo textem «Out of range» v případě nepodporované senzorem vzdalenností. Uživatel bude také moci otáčením encoderu změnit jednotku výpočtu vzdálenosti (milimetry, centimetry, metry): otáčení doprava - na větší jednotky, otáčení doleva - na menší jednotky



3.3.2. Menu

Stisknutím tlačítka enkodéru se laserové výpočty pozastaví a uživatel se bude moci seznámit s nabídkou na displeji, která mu poskytne další funkce na výběr: statistiky, mixáž a návrat do režimu výpočtu. Posouváním enkodéru může uživatel vybrat kterýkoli z těchto nástrojů a po stisknutí tlačítka enkodéru přejít k jeho použití.



3.3.3. Statistiky

Zobrazí uživateli na displeji nejmenší a největší vzdálenost detekování encoderem. Po stisknutí tlačítka encoderu v tomto režimu se uživatel vrátí do režimu výpočtu vzdálenosti. Uživatel bude moci do výpočtů přidat číselný šum podle své potřeby. Na displeji se zobrazí aktuální použitý šum a řádek pro doplnění nového šumu. Posouváním encoderu (doprava pro zvýšení, doleva pro snížení) lze nastavenou hodnotu upravit. Po stisknutí tlačítka encoderu se uživatel vrátí do režimu výpočtu a vzdálenost se zobrazí s číselným přídatkem.



3.3.4. Mixin

Stisknutím tlačítka enkodéru se laserové výpočty pozastaví a uživatel se bude moci seznámit s nabídkou na displeji, která mu poskytne další funkce na výběr: statistiky, mixáž a návrat do režimu výpočtu. Posouváním enkodéru může uživatel vybrat kterýkoli z těchto nástrojů a po stisknutí tlačítka enkodéru přejít k jeho použití.



3.3.5. Go out

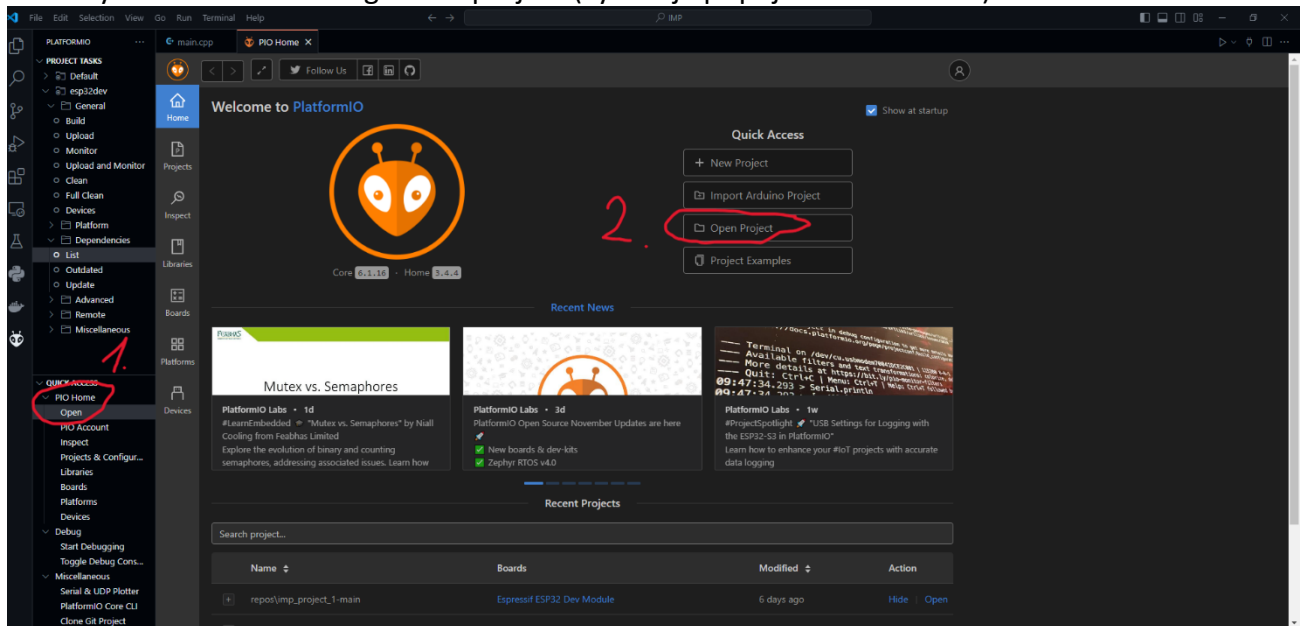
Vrátí uživatele z nabídky do režimu výpočtu.

3.4. Problémy

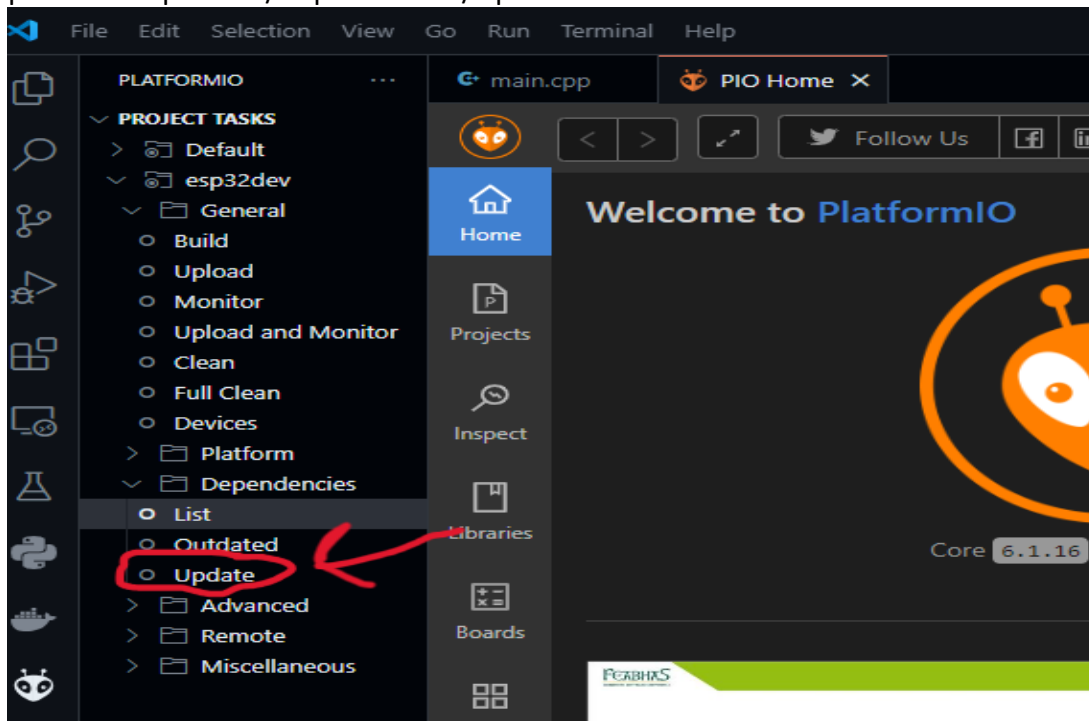
Hlavním problémem byla synchronizace čtení rotace a stisků tlačítek encoderu se změnami vnitřních stavů programu, způsobená citlivostí hardwaru a fázovým principem jeho fungování. Pro vyřešení tohoto problému jsem ve svém řešení spojil přerušení pro rotace a softwarové čekání na stisknutí tlačítka. Program jako celek je implementován odolný vůči tomuto problému, ale mohou se vyskytnout drobné chyby reakce na interakci se encoderem (není kritická), takže je třeba s ním komunikovat opatrně.

3.5. Spuštění

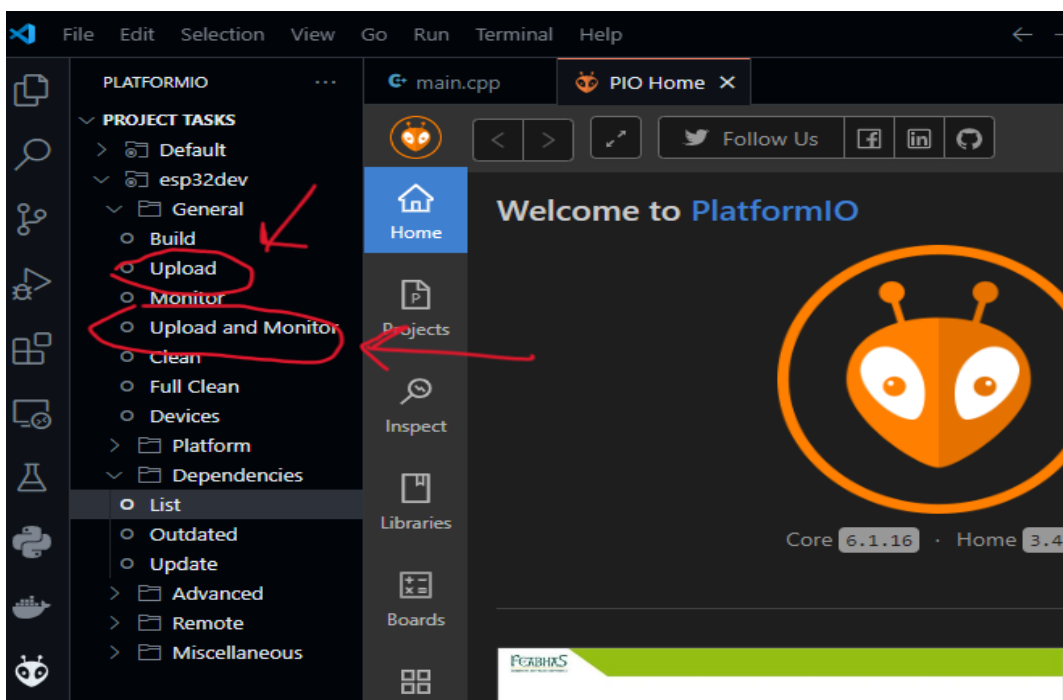
Projekt byl vyvinut a určen pro použití v rámci vývojového prostředí PlatformIO (VS Code). Po stažení projektu do počítače otevřete PlatformIO (PIO Home). V rozhraní vyberte tlačítko „Open Project“ a otevřete soubor platformio.ini. Poté by mělo vývojové prostředí automaticky načíst všechny závislosti a nakonfigurovat projekt (vyžaduje připojení k internetu!).



Zkontrolujte, zda byly závislosti úspěšně načteny. V opačném případě přejděte v rozhraní na položku «esp32dev /Dependencies/Update».



Po úspěšné instalaci bude projekt připraven k přenosu do zařízení. Za tímto účelem připojte hardware k počítači a v nabídce IDE vyberte «esp32dev/General/Upload» nebo «esp32dev/General/Upload and Monitor». Po stažení kódu do zařízení můžete začít program plně používat. Hodně štěstí!



3.6. Videoukazka

Pro názornější seznámení s principem fungování a použitím mého řešení jsem natočil další video s podrobným popisem. Níže uvádím odkazy, prostřednictvím kterých si jej můžete prohlédnout. Děkuji předem za pozornost.

YouTube:

<https://youtu.be/HLqoUfeyYAY>

GoogleDisk:

https://drive.google.com/file/d/1TJymZb3bRzboh4TQ8zLQ9ME7eGKNd8dA/view?usp=drive_link

4. Výsledek

Můj program napsaný v jazyce C s využitím frameworku Arduino realizuje snímání vzdálenosti laserovým senzorem VL53L0X a zobrazuje tyto hodnoty na displeji OLED, což plně odpovídá výchozímu stavu úlohy. Uživateli jsem také poskytl další funkce pomocí otočného snímače pro přidání číselných chyb a statistického sběru výpočtů.

Řešení má potenciální využití v oblasti automatizace a robotiky, např. pro podporu provozu manipulátorů apod.

Po přehledu celku, mám důvod tvrdit, že projekt je plně realizován a splňuje požadavky finalního hodnocení.

5. Bibliografie

- [1] ESP32 <https://en.wikipedia.org/wiki/ESP32>
- [2] VL53L0X Datasheet https://www.laskakit.cz/user/related_files/vl53l0x.pdf
- [3] OLED 0.96`` I2C Datasheet https://www.mouser.com/datasheet/2/1398/Soldered_333099-3395096.pdf
- [4] Encoder KY-040 https://elty.pl/cs_CZ/p/Impulsni-modul-snimace-KY-040/1155
- [5] Adafruit_VL53L0X https://github.com/adafruit/Adafruit_VL53L0X.git
- [6] Adafruit_SSD1306 https://github.com/adafruit/Adafruit_SSD1306.git
- [7] Adafruit-GFX-Library <https://github.com/adafruit/Adafruit-GFX-Library.git>
- [8] Adafruit_BusIO https://github.com/adafruit/Adafruit_BusIO.git