

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra řídicí techniky

Pulzní generátor pro testování a vývoj modulů pro průmyslovou automatizaci

Otakar Kodytek

Vedoucí: Ing. Josef Kopečný
Obor: Kybernetika a robotika
Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kodytek** Jméno: **Otakar** Osobní číslo: **503233**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Pulzní generátor pro testování a vývoj modulů pro průmyslovou automatizaci

Název bakalářské práce anglicky:

Pulse generator for testing and development of modules for industrial automation

Pokyny pro vypracování:

1. Navrhněte a realizujete pulzní generátor pro testování a vývoj modulů pro průmyslovou automatizaci.
2. Pro řízení pulzního generátoru použijte PLC od firmy Siemens (popřípadě použijte uživatelské rozhraní s použitím dotykového panelu)
3. Pulzní generátor by měl mít k dispozici alespoň dva digitální výstupy a dva výstupy s nastavitelnou hodnotou napětí s možností nastavit frekvenci a střihu signálu.

Seznam doporučené literatury:

- [1] Trevor Martin: The Insider's Guide to the STM2 ARM Base, Microcontroller - United Kingdom 2009
[2] Kolektiv autorů: MISRA C++; 200, Guidelines for the Use of the C++, Language in Critical Systems - United Kingdom 2008

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Josef Kopečný Siemens, s.r.o. , Siemens Advanta Development

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Ing. Martin Hlinovský, Ph.D. katedra řídicí techniky FEL

Datum zadání bakalářské práce: **27.01.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Josef Kopečný
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Tímto bych chtěl poděkovat vedoucímu této práce Ing. Josefovi Kopečnému za vstřícný přístup při dotazech ohledně praktické i teoretické části. Doktoru Martinu Hlinovskému, za zprostředkování této práce v rámci fakulty. Ing. Honzovi Vavroušovi za pomoc s frézováním desky plošných spojů. Dále také kolegům, kteří zařízení zkoušeli a přispěli tím k vylepšení praktické části práce, jmenovitě Ing. Petru Pařízkovi, Ing. Miroslavovi Šindelářovi a Richardu Rubášovi.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 25. května 2023

Podpis

Abstrakt

Tato práce pojednává o návrhu a vývoji programovatelného pulzního generátoru, který lze využít pro automatizaci integračních testů během vývoje vstupně-výstupních modulů pro PLC.

Práce pojednává nejprve o návrhu řešení z hlediska architektury zařízení, kde bylo využito jednodeskového počítače Raspberry Pi, jakožto rozhraní mezi generátorem a uživatelem. Samotné generování výstupních pulzů obstarává Nucleo32. Obě tyto komponenty jsou propojeny pomocí desky plošných spojů, která také zesiluje výstupní pulzy na potřebná napětí a obstarává napájení Raspberry a desky Nucleo. Následně se v práci probírá řešení z pohledu softwaru, ale také použité komunikační protokoly.

Navržené zařízení bylo realizováno a v praxi použito ve firmě Siemens, pro kterou byla práce zpracovávána. Generátor obsahuje 2 digitální a 2 analogové výstupy. Obsahuje také grafické rozhraní a možnost konfigurace složitějších průběhů pomocí souborů XML. Z hlediska automatizace tento generátor nahradil zastaralejší zařízení v několika testech, zároveň však rozšířil možnosti testování a nahradil několik původně manuálních testů.

Klíčová slova: Pulzní generátor, PLC, Snap7, Raspberry Pi, Nucleo32, XML

Vedoucí: Ing. Josef Kopečný
Siemens, s.r.o,
Siemensova 1,
Praha 13

Abstract

This thesis outlines the architecture and development of the programmable pulse generator, which can be used for the automation of integration testing during the development of input-output PLC modules.

First of all, the thesis inform about the device architecture, where a single-board computer Raspberry Pi was used as a interface between the user and the generator. Generation of the pulses is performed on the microcontroller development board Nucleo32. Both of these components were connected with the printed circuit board, which at the same time amplifies output signals from Nucleo32 and provides power with the designed voltage regulation circuit. Thesis then discusses the implementation from the software point of view, as well as the used communication protocols that were used.

The device was created and practically used at the company Siemens, for which this work was done. The generator has 2 digital and 2 analog outputs. It also includes a graphical user interface and the possibility to configure more complex waveforms with the usage of a XML file. This generator replaced obsolete device in a few tests, while expanding the test capabilities and replaced several of the manual tests.

Keywords: Pulse generator, PLC, Snap7, Raspberry Pi, Nucleo32, XML

Title translation: Pulse generator for testing and development of modules for industrial automation

Obsah

1 Úvod	1	3 Teorie	13
1.1 Testování modulu	1	3.1 Použitý jazyk	13
1.2 Vývoj modulu	2	3.2 Komunikační protokoly	13
1.3 Cíl práce	3	3.2.1 S7Comm	13
2 Návrh řešení	5	3.2.2 UART	14
2.1 Analýza problému	5	3.3 Nested Vector Interrupt Control (NVIC)	15
2.1.1 Generování výstupních signálů	5	4 Raspberry Pi	17
2.1.2 Komunikační rozhraní	6	4.1 Příprava počítače raspberry	17
2.1.3 Grafické rozhraní	6	4.2 Možnosti konfigurace výstupních pulzů	18
2.2 Výběr jednodeskového počítače	7	4.3 Grafické rozhraní	18
2.2.1 Protokol pro komunikaci PLC-Generátor	7	4.3.1 Použité nástroje a knihovny	18
2.2.2 Shrnutí požadovaných parametrů	7	4.3.2 Vzdálená konfigurace (Remote Control)	19
2.3 Výběr MCU	8	4.3.3 Manuální konfigurace (Manual Control)	20
2.4 Sběrnice Raspberry-MCU	11	4.3.4 Výběr z XML souborů (Program Control)	21
2.5 Použitý programovací jazyk	11	4.4 Uložení dat o požadované sekvenci	22

4.5 Konfigurace generátoru pomocí souborů XML	23	6.3 Vytvoření struktury sekvence z přijatého paketu.	39
4.5.1 Struktura XML souboru	24	6.4 Členění objektů generátoru	39
4.6 Komunikace s PLC	25	6.5 Generování pulzů	42
4.6.1 Přijímání dat z PLC	25	7 Hardware	45
4.6.2 Zaslání zpětné vazby do PLC	26	7.1 Schéma zapojení	46
4.7 Výměna dat s MCU	27	7.1.1 DC-DC měnič napětí	46
5 PLC a TIA Portal	29	7.1.2 Digitální PMOS spínač	47
5.1 Struktura dat v PLC	29	7.1.3 Analogový zesilovač	48
5.2 Kontrola výměny dat	30	7.1.4 Ochrany napájení obvodu ...	50
6 MCU a generování pulzů	31	7.1.5 Konektory DPS	52
6.1 Konfigurace MCU v STM32CubeIDE	31	7.2 Rozložení komponent na DPS ..	53
6.1.1 Nastavení hodinového signálu	32	7.3 Krabice	53
6.1.2 timerinterrupts	33	7.3.1 Čelní strana s displejem	53
6.1.3 Nastavení sběrnice UART ...	34	7.3.2 Hlavní rám	54
6.1.4 Nastavení výstupních periférií	35	7.3.3 Dvířka krabice	56
6.1.5 Konfigurace NVIC	36	7.3.4 Zvýšení pevnosti krabice ...	57
6.2 Zpracování přijatých dat	37	7.3.5 Zavěšení na lištu DIN	57

8 Měření signálů generátoru	59
8.1 Analogové výstupy	59
8.2 Digitální výstupy	61
8.3 Externí spuštění generátoru	62
9 Závěr	65
10 Slovník pojmů	67
Literatura	69
A Konfigurace TIA Portal	73
A.1 Nastavení komunikace v programu TIA Portal	73
A.2 Volání komunikačních funkcí v cyklu OB1	76

Obrázky

1.1 Digitální vstupní modul kategorie SIL3[1].	1	4.3 Ukázka grafického rozhraní - manuální mód.	20
1.2 Diagram testování modulu.	2	4.4 Ukázka grafického rozhraní - programový mód.	22
2.1 Diagram použití vyvíjeného generátoru.	5	4.5 Ukládání sekvencí v jednodeskovém počítači Raspberry.	22
2.2 Jednodeskový počítač Raspberry Pi 3B+, převzato z [2].	8	4.6 Struktura souboru konfiguruujícího průběh výstupů generátoru.	24
2.3 Sedmipalcový dotykový displej, převzato z [3].	8	4.7 Tok dat mezi PLC a Raspberry Pi uvnitř programu generátoru.	26
2.4 Rozřazení MCU podle typu aplikace a výkonu, převzato z [4].	9	4.8 Typy použitých paketů a jejich složení.	27
2.5 Rozdělení dostupných desek Nucleo, výřez z nabídky produktů [5].	9	5.1 Síťové zapojení zařízení k PLC a testovanému modulu.	29
2.6 DAC buffer v MCU firmy STM řady G4, upraveno z [6].	10	5.2 Struktura paketů zasílaných mezi PLC a Raspberry Pi	30
2.7 Výsledná architektura zařízení.	12	6.1 Vyvedení výstupních signálů mikrokontroleru STM32G431.	31
3.1 UART paket, převzato z [7].	14	6.2 Nastavení hodinového signálu - Phase-Locked Loop.	32
4.1 Diagram možností konfigurace generátoru.	18	6.3 Nastavení hodinového signálu periférií.	33
4.2 Ukázka grafického rozhraní - vzdálený mód.	19	6.4 Parametry sériové linky komunikující s Raspberry Pi.	35
		6.5 Nastavení DAC převodníků.	36

6.6 Sekvenční diagram znázorňující přijímání dat v MCU.	37	7.5 Schéma zapojení operačního zesilovače.	50
6.7 Změřené zpoždění po požadavku na start generátoru.	38	7.6 Schéma zapojení ochran napájení desky a externí spouště generátoru.	51
6.8 Rozhodovací strom tvorby sekvencí z přijatého paketu.	40	7.7 Schéma zapojení konektorů displeje, Raspberry Pi a desky Nucleo32.	52
6.9 Vazba objektů zajišťujících správu celého zařízení STM32.	41	7.8 Schéma zapojení výstupních konektorů a konektorů pro LED signalizaci stavu generátoru.	53
6.10 Vazba objektů spravujících jednotlivé kanály.	41	7.9 Návrh DPS desky.	54
6.11 Vazba objektů definujících sekvenci a její konfiguraci.	42	7.10 Schéma zapojení výstupních konektorů a konektorů pro LED signalizaci stavu generátoru.	55
6.12 Program MCU - řízení výstupních napěťových hladin.	43	7.11 Schéma zapojení výstupních konektorů a konektorů pro LED signalizaci stavu generátoru.	55
6.13 Dokončení sekvence generátorem po přijetí paketu Stop.	44	7.12 Krabíčka po zasunutí paciček, pohled shora.	56
7.1 Schéma hardware použitého v práci.	45	7.13 Krabíčka před zaháknutím háčků, pohled zdola.	56
7.2 Schéma zapojení spínaného regulátoru napětí.	46	7.14 Vnitřek krabíčky včetně distančních sloupků a šroubů pro zpevnění.	57
7.3 Schéma zapojení digitálního spínače výstupního signálu.	47	7.15 Generátor zavěšený na DIN liště a připojený k testovanému zařízení. .	58
7.4 Porovnání spínání digitálního výstupu pomocí dvou vyzkoušených metod.	49		

8.1 Analogové výstupy, primární a sekundární se zpožděním, časové základy používané pro testování. . .	60
8.2 Analogové výstupy, primární a sekundární, rychlé změny napěťových hladin.	60
8.3 Digitální výstupy, primární a sekundární, se zpožděním, ekvivalentní.	61
8.4 Digitální výstupy, primární a sekundární se zpožděním, neekvivalentní.	62
8.5 Digitální výstup primárního kanálu, externí spuštění náběžnou hranou.	63
8.6 Digitální výstup primárního kanálu, externí spuštění náběžnou hranou se zpožděním.	63
A.1 Síťové zapojení zařízení k PLC a testovanému modulu.	73
A.2 Použité datové a systémové bloky pro komunikaci mezi PLC a Generátorem.	74
A.3 Konfigurace komunikačních bloků S7 PUT a S7 GET.	75
A.4 Struktura paketů zasílaných mezi PLC a Raspberry Pi	75

Kapitola 1

Úvod

V průmyslové automatizaci se dnes běžně používají programovatelné logické automaty (PLC), které pomocí připojených vstupně-výstupních karet zpracovávají vstupy z připojených senzorů a následně ovládají připojené aktuátory nebo dále zpracovávají data pro uživatele.

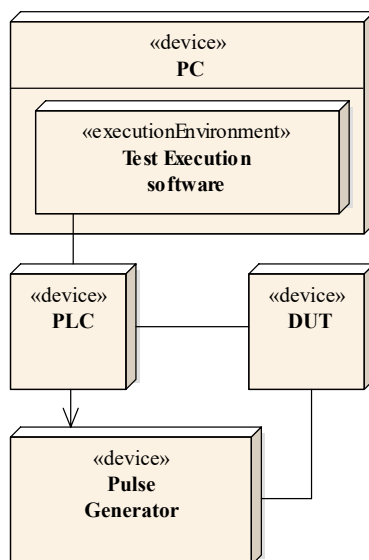
Moduly vyvíjené v rámci projektu podléhají certifikaci stupně integrity bezpečnosti 3. úrovně (SIL3) dle standardu IEC 61508. Správnost procesů během vývoje posuzuje externí dozorce, který mimo jiné posuzuje, zda byl modul dostatečně testován během vývoje.



Obrázek 1.1: Digitální vstupní modul kategorie SIL3[1].

1.1 Testování modulu

Během vývoje těchto vstupně-výstupních modulů je třeba testovat, zda modul reaguje na vnější vstupy dle očekávání a splňuje předem určenou specifikaci. Na začátku vývoje modulu se stanoví požadavky, které musí modul splňovat pro dosažení výše zmíněného stupně integrity bezpečnosti. Následně je během vývoje realizováno více typů testování. Dle standardu IEC 61508-2, tabulky B.5, je třeba realizovat takzvaný "Functional testing"(popsáno v IEC 61508-7 B.5.1), ten se provádí v



Obrázek 1.2: Diagram testování modulu.

rámci integračního testování. Jeho smyslem je paralelně s vývojem modulu testovat funkčnost z pohledu reakcí na očekávané vstupy a také kontrola splnění požadavků kladených na vyvíjený modul.

V integračním testu se také používá metody "Expanded functional testing" (IEC 61508-7 B.6.8), což je typ testování, kde se ověřují reakce modulu na méně očekávané stavy nebo stavy mimo specifikaci.

Testování probíhá na stojanu, kde se nachází PLC, ke kterému jsou připojené již vyvinuté vstupně-výstupní moduly. Testovaný modul (DUT) se zpravidla nachází za IM. V prostředí TIA Portal je naprogramován program pro PLC, pomocí kterého se poté vyvíjený modul testuje. Nadstavbou tohoto programu je dále interně vyvíjený program, který je schopen nastavovat proměnné v PLC a také nahrávat konfigurace vyvíjeného modulu.

Těchto testů je velké množství a během vývoje se realizují opakovaně, proto je třeba, aby byly co možná nejvíce automatizovány. Mnoho testů je možné realizovat za pomoci již schválených modulů, ovšem pro některé testy je třeba vlastností, které nemusí vstupně-výstupní moduly mít.

1.2 Vývoj modulu

Během samotného vývoje může chtít sám vývojář mít možnost ověřit si reakce modulu na vstupní napěťové pulzy. Má-li být tedy přístroj použitelný i pro vývoj modulu, je třeba vytvořit co možná nejintuitivnější prostředí, aby používání přístroje nevyžadovalo zdlouhavé přípravy a umožnilo vývojáři ověřit si reakce sám.

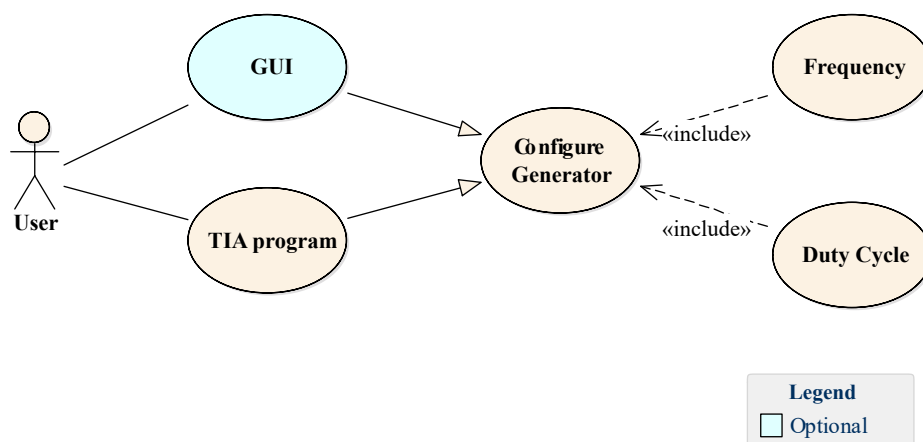
■ 1.3 Cíl práce

Cílem této práce je navrhnout a realizovat pulzní generátor, který bude schopen mimo jiné provádět automatizované testování vstupně-výstupních karet PLC, konkrétně například testování vstupního zpoždění modulu, diskrepance páru kanálů, nebo s již existujícím zesilovačem provádět testování vstupního napájení modulu. Konfigurace generátoru bude realizována pomocí PLC, přičemž program pro PLC bude vytvořen v prostředí TIA Portal. Pulzní generátor bude mít případně také uživatelské prostředí, eliminující nutnost připojení k PLC.

Kapitola 2

Návrh řešení

2.1 Analýza problému



Obrázek 2.1: Diagram použití vyvíjeného generátoru.

2.1.1 Generování výstupních signálů

Hlavní činnost, kterou bude muset vyvinuté zařízení zvládat, je generování výstupních signálů s co nejvyšší přesností v časové oblasti. Pokud by byly signály generovány přímo pomocí jednodeskového počítače, jakým je například

Raspberry Pi, bylo by obtížné se systémem Raspberry Pi OS (linuxové jádro) zajistit, že pulzy mají precizní délku a nejsou prodlouženy díky pozdržení procesu, který má změnu výstupního signálu na starosti. Důvodem je, že systém Raspberry Pi OS není RTOS systém, tudíž není zajištěno přednostní zpracování úkolů s vyšší prioritou. Hlavním problémem Raspberry Pi nicméně je, že neobsahuje DAC převodníky, tedy bychom potřebovali rozšiřující modul. Další možností by bylo využít výhody MCU, která nemají operační systém a jsou snadno dostupná jako vývojové desky. Nevýhodou těchto zařízení je nicméně nižší výkon.

2.1.2 Komunikační rozhraní

Dále pro zařízení potřebujeme zajistit komunikaci s PLC, pomocí kterého bude moci uživatel řídit chod generátoru. Nejsnazší cestou připojení je zde na fyzické vrstvě Ethernet. Má-li být zachováno v řešení MCU, je třeba ke komunikaci s PLC a případným grafickým rozhraním zvolit zařízení, které bude schopno všechna tato data zpracovat a unifikovaně zaslat MCU. Protože většina MCU nemá na vývojových kitech možnost připojení pomocí ethernetu a také mají mikrokontrolery nižší výkon, bude generátor tvořen jednodeskovým počítačem, který bude interagovat s uživatelem a to prostřednictvím PLC (TIA Portal).

2.1.3 Grafické rozhraní

Jednodeskové počítače již mohou mít dostatečný výkon pro snadnou implementaci grafického rozhraní. V rámci této práce lze takto tedy implementovat dotykový displej, na kterém by mohl uživatel nastavovat generátor bez nutnosti připojení k PLC. Takto získaná data od uživatele předá jednodeskový počítač mikrokontroleru, se kterým lze díky absenci operačního systému zajistit přesnost generovaného signálu v časové oblasti.

2.2 Výběr jednodeskového počítače

K automatizaci testů se využívá PLC řady Simatic S7-1500 firmy Siemens. Jednodeskový počítač bude od tohoto PLC přijímat data, která dále zpracuje a pomocí kterých spustí generátor s požadovaným nastavením.

2.2.1 Protokol pro komunikaci PLC-Generátor

Při výběru způsobu komunikace bylo vybíráno především mezi protokoly OPC UA a S7comm.

Protokol OPC UA je protokolem nezávislým na platformě, ve kterém si každé zařízení specifikuje sebou zasílaná data, která mohou být rozdělena do sémantických objektů[8].

Z důvodu jednoduchosti implementace byl upřednostněn protokol S7comm. Implementace může být totiž provedena s pomocí knihovny Snap7[9]. Tato knihovna je schopna virtualizovat na daném zařízení CPU Simatic S7-300 a připojit zařízení takto do dané sítě.

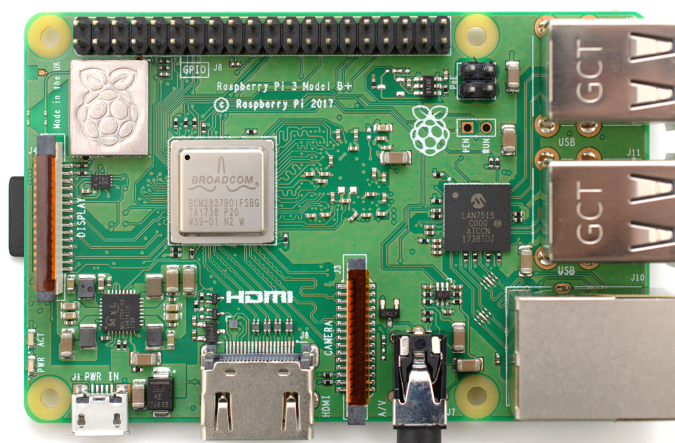
2.2.2 Shrnutí požadovaných parametrů

Zásadní parametry pro výběr jsou:

- Ethernetový port - podpora knihovny Snap7
- Snadná implementace dotykového displeje
- SPI/UART/I2C pro komunikaci s MCU
- Dostupnost

Výslednou volbou se stalo Raspberry Pi 3B+ (2.2).

Toto zařízení je jednodeskový počítač s 64-bitovým procesorem 1,4 GHz se 4 jádry[10]. K tomuto mikroprocesoru firma nabízí dotykový 7 palcový displej, na kterém budou zobrazovány údaje o generátoru a skrze který bude moci uživatel generátor konfigurovat. Tento displej je vybaven rozhraním DSI (Display Serial Interface), které podporuje také Raspberry Pi 3B+. Výhodou tohoto rozhraní je vysoký přenosový výkon při nízké náročnosti a malém počtu pinů[11]. Displej lze s tímto počítačem na fyzické vrstvě spojit pomocí konektorů, které se nalézají na obou deskách, přičemž kabel je dodáván spolu



Obrázek 2.2: Jednodeskový počítač Raspberry Pi 3B+, převzato z [2].



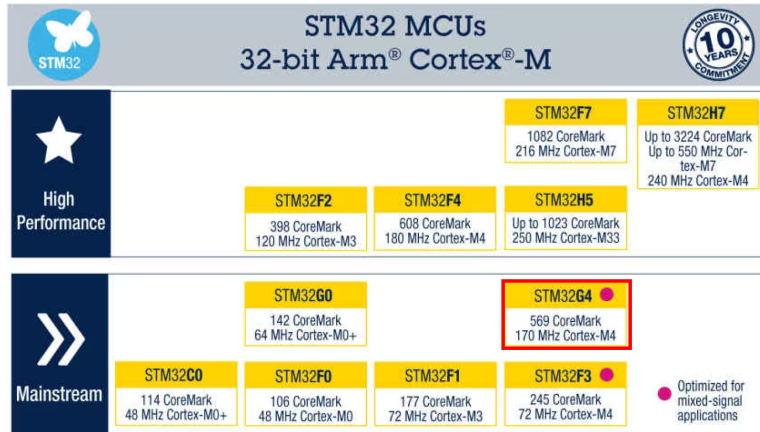
Obrázek 2.3: Sedmipalcový dotykový displej, převzato z [3].

s displejem. Raspberry Pi 3B+ také obsahuje ethernetový port, tudíž je možné se také připojit k síti, ve které se nachází PLC.

2.3 Výběr MCU

Protože hardware má být snadno dostupný a snadno zprovoznitelný, byla zvolena deska Nucleo, vyráběná firmou STMicroelectronics. Tyto desky v sobě zahrnují mikrokontroler, ale také další obvody pro jeho snadné použití, jako je například komponenta ST-Link[5], která slouží k programování a ladění

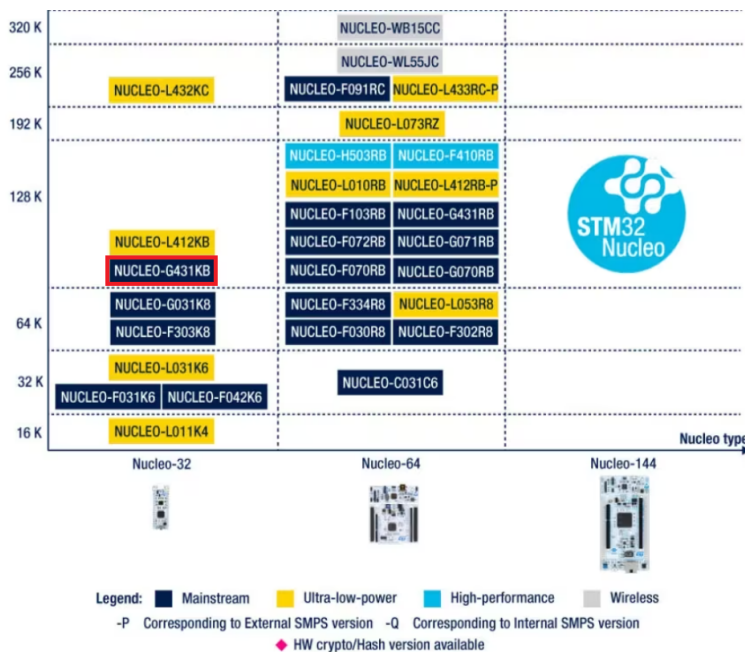
mikrokontroleru, nebo také regulátor napájecího napětí na 3,3 V. Pro výběr bylo využito tabulky 2.4 na stránkách výrobce STM, která srovnává MCU do skupin podle jejich hlavních předností. Celé zařízení bude napájeno



Obrázek 2.4: Rozřazení MCU podle typu aplikace a výkonu, převzato z [4].

z externího zdroje a energetická náročnost MCU nebude během aplikace rozhodujícím faktorem.

STM vyrábí mnoho desek Nucleo, používajících různé čipy a rozvržení. Zaměříme-li se na nejmenší Nucleo desky s 32 piny, je nejvýkonnější volbou deska STM32G431KB, jak lze vidět na obrázku 2.5. V generátoru budou tedy pulzy generovány na této desce.



Obrázek 2.5: Rozdělení dostupných desek Nucleo, výřez z nabídky produktů [5].

Základní parametry použité desky Nucleo (a osazeného MCU) jsou: [12, 13]

- 170 MHz
- 128 KB Flash
- 24 MHz krystalový oscilátor
- STLINK-V3 debugger/programmer
- 2x 12-bit DAC s nízkou impedancí

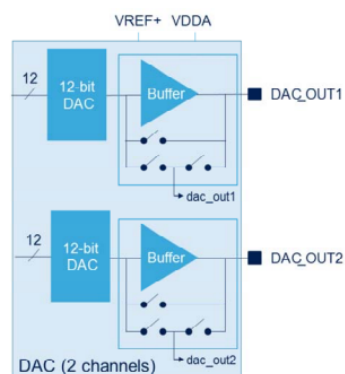
■ Řízení přerušení

Další z výhod STM32 s procesorem ARM-Cortex M4 oproti Raspberry Pi je využití periferie NVIC pro řízení přerušení. Tato periferie nám zvládne zajistit precizní časování výstupních signálů 3.3.

■ DAC převodníky

Deska disponuje dohromady 4 převodníky DAC, které mohou být použity pro analogový výstup, nicméně 2 z nich jsou takzvaně "unbuffered", neboli mají vysokou výstupní impedanci.

Výstup převodníku má výstupní impedanci v případě vybrané desky 25 k Ω . Mikrokontroler umožňuje interně připojit výstup DAC převodníku do operačního zesilovače, pomocí kterého se sníží výstupní impedance. Nevýhodou tohoto řešení je snížení vzorkovací frekvence převodníku z 15 Msps na 1 Msps [14].



Obrázek 2.6: DAC buffer v MCU firmy STM řady G4, upraveno z [6].

2.4 Sběrnice Raspberry-MCU

MCU i Raspberry Pi mohou mezi sebou použít pro komunikaci sběrnice I²C, UART nebo SPI. Sběrnice I²C je polo-duplexní, tedy umožňuje v jednu chvíli přenos pouze jedním směrem, proto ho nevyužijeme. Výhodou sběrnice SPI je velmi vysoká možná přenosová rychlost. V práci byla zvolena, z důvodu snazší implementace v počítači Raspberry Pi, sběrnice UART.

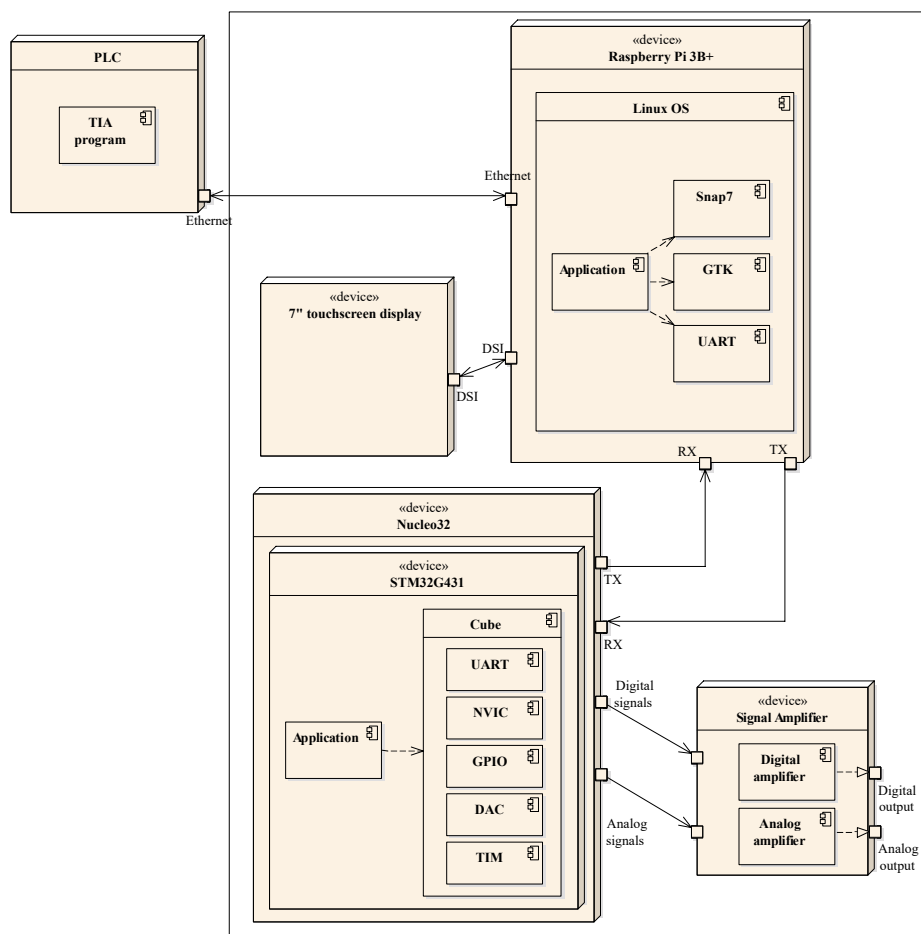
2.5 Použitý programovací jazyk

Výběr programovacího jazyka probíhal s ohledem na podporu jazyků ze strany knihovny Snap7 implementující komunikaci jednodeskového počítače s CPU. Z programovacích jazyků, které knihovna podporuje bylo vybíráno z jazyků C#, Python a C++, které jsou ve firmě nejpoužívanější.

Vzhledem k použití MCU bez operačního systému byl zvolen jazyk C++. V tomto jazyce je naprogramováno jak MCU, tak počítač Raspberry Pi, z důvodu zachování jednotného programovacího jazyka celé práce.

Objektově orientovaný jazyk C++ byl zvolen oproti procedurálnímu jazyku C z důvodu přehlednosti, kterou objekty přináší. Z pohledu efektivity bude podle principu "Zero overhead principle" kód napsaný v jazyce C++ pomalejší oproti kódu v C, pouze při nevhodném používání funkcionalit C++[15].

Psaní kódu práce bylo inspirováno pravidly pro psaní kódu pro kritické systémy [16]. Generátor nicméně není sám kritickým systémem, proto nebylo dbáno všech pravidel, aby se práce stihla zhotovit rychleji.



Obrázek 2.7: Výsledná architektura zařízení.



Kapitola 3

Teorie



3.1 Použitý jazyk

V práci se nalézají obrázky a grafy, jejichž obsah je v anglickém jazyce. Po konzultaci s vedoucími práce bylo rozhodnuto, že tyto grafy zůstanou ponechány v jazyce anglickém. Důvodem pro toto rozhodnutí je, že obrázky jsou používány také v pracovní dokumentaci, a většina těchto grafů je vázána k realizovanému kódu, který je také psán v angličtině. Pokud by jména objektů byla přeložena do českého jazyka, jejich vazba na kód by již nebyla zřejmá.



3.2 Komunikační protokoly



3.2.1 S7Comm

S7comm je protokol navržený firmou Siemens, který umožňuje realizovat výměnu dat mezi dvěma PLC[17]. Protokol pokrývá relační, prezentační a aplikační vrstvu OSI modelu (vrstvy 5-7). Tento protokol zajišťuje výměnu dat mezi různými PLC. Komunikace probíhá formou paketů TCP na transportní vrstvě. Fyzická a linková vrstva je realizována pomocí Ethernetu[17].

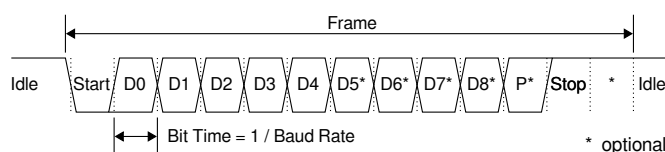
■ Snap7

Pro obsluhování komunikace mezi PLC a vytvářeným generátorem byla využita knihovna Snap7. Knihovna dovoluje na jednom vlákně virtualizovat Raspberry Pi tak, aby ho ostatní prvky v lokální síti považovali za skutečné PLC. Virtuální PLC se může chovat jako server, klient nebo partner. Detailní chování objektů popisuje dokumentace knihovny. Generátor se chová jako Server, jsou mu tedy zasílána data od klientů[9], v tomto případě od skutečného PLC 1516F-35.1.

■ 3.2.2 UART

Uart (Universal asynchronous receiver transmitter) je asynchronní komunikační protokol na linkové vrstvě OSI modelu. Protokol umožňuje přenos dat v obou směrech zároveň. Přijímač i vysílač musí mít předem určenu stejnou přenosovou rychlost. Neprobíhá-li přenos dat je na fyzickou vrstvu přivedeno kladné napětí. Přenos začíná start bitem, při kterém je napětí staženo na nízkou napěťovou hladinu po dobu jednoho hodinového cyklu (doba pro přenos jednoho bitu). Přijímač při detekci sestupné hrany napětí začne vzorkovat přijímaný signál s přednastavenou přenosovou rychlostí. Následuje datový rámeček, který může obsahovat 5 až 8 bitů (v případě vynechání paritního bitu až 9). Následuje paritní bit a na závěr 1 až 2 stop bity[18].

Po přijetí paketu jsou nastaveny v registrech takové hodnoty, které signalizují přijatý paket. Jestliže program nestihne tyto registry opět nastavit do stavu signalizujícího možnost přijetí nových dat, UART po přijetí nového datového rámce nebude schopen obsah užitečná data z předat přes registry do programu. Mikrokontroler v tomto případě nastaví v registru error bit "Overrun". Datový rámeček je po sběrnici odeslán od nejméně významného bitu (LSB) po nejvíce významný (MSB)[18]. Obecná forma paketu je zakreslena na obrázku 3.1.



Obrázek 3.1: UART paket, převzato z [7].

■ 3.3 Nested Vector Interrupt Control (NVIC)

Řízení MCU bude probíhat pomocí přerušení. Pro řízení těchto přerušení používají procesory ARM-Cortex M4 interní periférii procesoru zvanou NVIC. NVIC umožňuje dynamicky nastavit prioritu jednotlivých přerušení a následně tato přerušení volá, přičemž zajistí, že ta přerušení, jejichž priorita je vyšší (nižší číslo vyšší priorita), mohou být pozastavena dokud nejsou vykonána přerušení s vyšší prioritou. Takto je zajištěno, že časově náročné instrukce, nezpozdí časově kritická přerušení, která mají vyšší prioritu[19].

Kapitola 4

Raspberry Pi

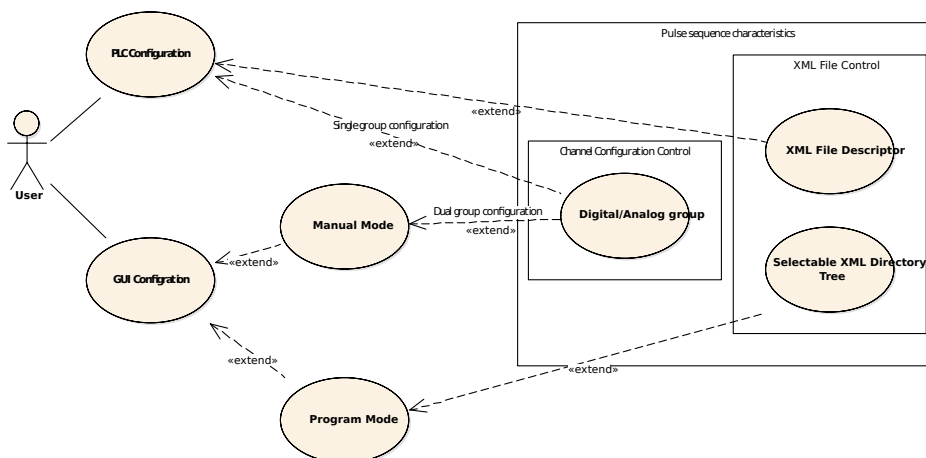
Raspberry Pi spravuje interakce ze 3 dalších zdrojů, kterými jsou PLC, uživatelská interakce skrze grafické rozhraní a použitý mikrokontroler pro samotné generování pulzů. Architektura programu je založena na 2 vláknech. První vlákno je virtualizace PLC pomocí knihovny Snap7. Druhé vlákno je hlavní aplikační vlákno, které obsluhuje interakce uživatele s grafickým rozhraním a data, která jsou určena ke konfiguraci generátoru zasílá pomocí sériové linky do připojeného mikrokontroleru.

4.1 Příprava počítače raspberry

Nejprve je nutné připravit si SD kartu, ze které bude Raspberry bootovat. To je možné učinit například pomocí programu Raspberry Pi Imager. Nahrajeme systém Raspberry Pi OS a můžeme počítač po zasunutí SD karty zapnout. Po nabofování otevřeme konfiguraci zařízení a v kolonce Interfaces zaškrtneme pouze Serial Port, přičemž Serial Console musí být vypnuto. Takto povolíme naši aplikaci používat UART a zároveň tuto možnost zakážeme konzoli Raspberry, čímž zajistíme, že se budou zasílat pouze požadovaná data.

4.2 Možnosti konfigurace výstupních pulzů

Generátor je možné ovládat pomocí PLC, nebo grafického rozhraní. Oba tyto přístupy umožňují generátor konfigurovat pomocí manuálního nastavení hodnot neboli konfiguraci kanálu. Dále je možné vybrat soubor XML s konfigurací generátoru uložený v Raspberry Pi. V grafickém prostředí lze toto učinit pomocí výběru ze stromové struktury adresáře, zatímco v prostředí TIA Portal je třeba znát jedinečné číslo charakterizující daný soubor. Výběr souboru XML je dále probrán v kapitole 4.5.



Obrázek 4.1: Diagram možností konfigurace generátoru.

4.3 Grafické rozhraní

Grafické rozhraní je tvořeno 3 záložkami, mezi kterými se dá vybírat v horní části obrazovky. Záložky jsou pojmenovány "Remote", "Manual" a "Program" Control. Názvy jsou odvozeny od způsobu, kterým je generátor nastavován.

4.3.1 Použité nástroje a knihovny

Pro implementaci grafického rozhraní byla využita open source sada knihoven GTK3. Výhodou této sady je její velká rozšířenost. Tato sada knihoven je

naprogramována v jazyce C, nikoli C++. Pokud bychom chtěli použít sadu pro C++, je k dispozici již upravená sada gtkmm. V této práci je pracováno se standardní sadou knihoven GTK. Důvodem pro toto rozhodnutí byla znalost této sady. Sada knihoven GTK byla preferována z důvodu předchozí zkušenosti autora.

Dále bylo využito nástroje Glade, umožňujícího rozvržení objektů v grafickém prostředí. Pomocí tohoto nástroje byly rozloženy komponenty grafického rozhraní, nicméně propojení signálů probíhalo již v programovacím jazyce C++. Vzhledem k přehlednosti je zachováno ošetření signálů v naprogramovaném kódu. Výstupem tohoto nástroje je soubor XML, které si program načte při spuštění aplikace.

4.3.2 Vzdálená konfigurace (Remote Control)

Remote Control		Manual Control		Program Control	
1001	Period 1	100 ms	Digital	Analog	
	Period 2	200 ms	Period	Frequency	
1002eq	Frequency	3.333 Hz	Toggle T2 high	DJ 50%	
	Analog Voltage	15 V	Start Trigger	External Trigger	
1002neq	Discrepancy	0 us	Start Digital	Start Analog	
	Delay	0 us			

Generator is configured through PLC

Obrázek 4.2: Ukázka grafického rozhraní - vzdálený mód.

Při výběru záložky Remote Control se uvažuje, že generátor je ovládán za pomoci PLC. Z tohoto důvodu se uživateli výrazně omezí možnosti změny nastavení generátoru. Jediným tlačítkem, které může uživatel využít je tlačítko Stop, neboli zastavení generátoru.

Do této záložky je generátor přepnut automaticky při přijetí dat z PLC. Konfigurace skrze PLC má tedy prioritu před uživatelským rozhraním, čímž se zajistí správná funkčnost nastavení v případě nechtěného přepnutí záložek do jiného druhu nastavení generátoru.

4.3.3 Manuální konfigurace (Manual Control)

Remote Control		Manual Control		Program Control		
1001	Period 1	500 ms	Digital	Analog		
	Period 2	500 ms	Period	Frequency		
1002eq	Frequency	1 Hz	Toggle T2 high	DF 50%		
	Analog Voltage	24 V	Start Trigger	External Trigger		
1002neg	Discrepancy	0 us	Start Digital	Start Analog		
	Delay	0 us				

s			ms			us			1	2	3
0	0	0	0	0	0	0	0	0			
Period 1			0 us			↑			4	5	6
Period 2			0 us						7	8	9
Frequency			1 Hz			↓			enter	0	->
Analog Voltage			24 V								
Discrepancy			0 us								
Delay			0 us								

Obrázek 4.3: Ukázka grafického rozhraní - manuální mód.

Záložka manuální konfigurace dovoluje uživateli konfigurovat výstupní kanály v reálném čase. Analogové i digitální kanálové páry jsou konfigurovány separátně, přičemž každá konfigurace je také separátně uložena pro případ, že mezi nimi chce uživatel přepínat a zachovat si uložené nastavení. Párový (sekundární) výstupní kanál nelze konfigurovat separátně, lze ho pouze zapnout, invertovat jeho výstupní hodnotu a oproti primárnímu ho v čase posunout.

V horní pravé části jsou bitové parametry.

- Druh konfigurovaného kanálu - Digitální/Analogový
- Typ nastavení periody 2 - Perioda/Frekvence
- 50% střída signálu (DF50)
- Prohození Period (Toggle T2 High)
- Druh spuštění - Externí/Tlačítkem Start
- Start generátoru - Digitální/Analogový

Dle těchto parametrů se zpřístupní příslušné číselné konfigurační parametry (viz níže v této kapitole). Dle druhu konfigurovaného parametru se přepínají ostatní parametry, aby korespondovaly s naposledy vybranými hodnotami. Typ nastavení periody 2 značí způsob nastavení časové základny druhé části pulzu. Tlačítko DF50 nastaví dle zadané frekvence obě periody rovnoměrně, aby hodnoty odpovídaly frekvenci signálu. Chce-li uživatel signál spustit pomocí náběžné hrany vstupního signálu generátoru, poté musí vybrat tlačítko

External Trigger.

V dolní části záložky se nachází číselné konfigurační parametry, kterými jsou:

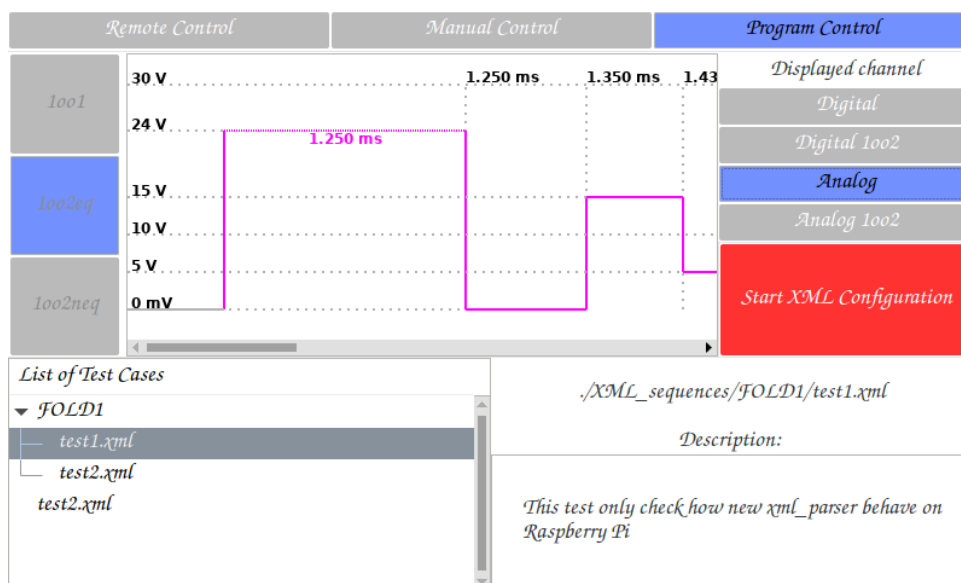
- Perioda 1
- Perioda 2
- Frekvence
- Výstupní napětí
- Diskrepance kanálů
- Zpoždění kanálu

Při kliknutí na daný parametr se nad parametry přepne číslicové zobrazení dané hodnoty. Toto zobrazení je tvořeno tlačítky, kterými si uživatel může vybrat nastavovanou číslici. Napravo od vybíraných parametrů je zobrazen číselník, kde uživatel nastaví vybranou číslici na zvolené číslo. Při výběru čísla se výběr automaticky přepne na následující číslici. Po nastavení požadované hodnoty se výběr potvrdí stiskem "Enter". Číslicové výběry se mění podle vybraného parametru tak, aby uživatel vždy mohl vybrat pouze jednotky, které lze u daného parametru konfigurovat. Stejně tak se blokují takové parametry, které nemají v daném nastavení bitových parametrů smysl (Při výběru Frekvence se blokuje Perioda 2, v případě Digitální konfigurace se blokuje výběr nastavení výstupního napětí).

■ 4.3.4 Výběr z XML souborů (Program Control)

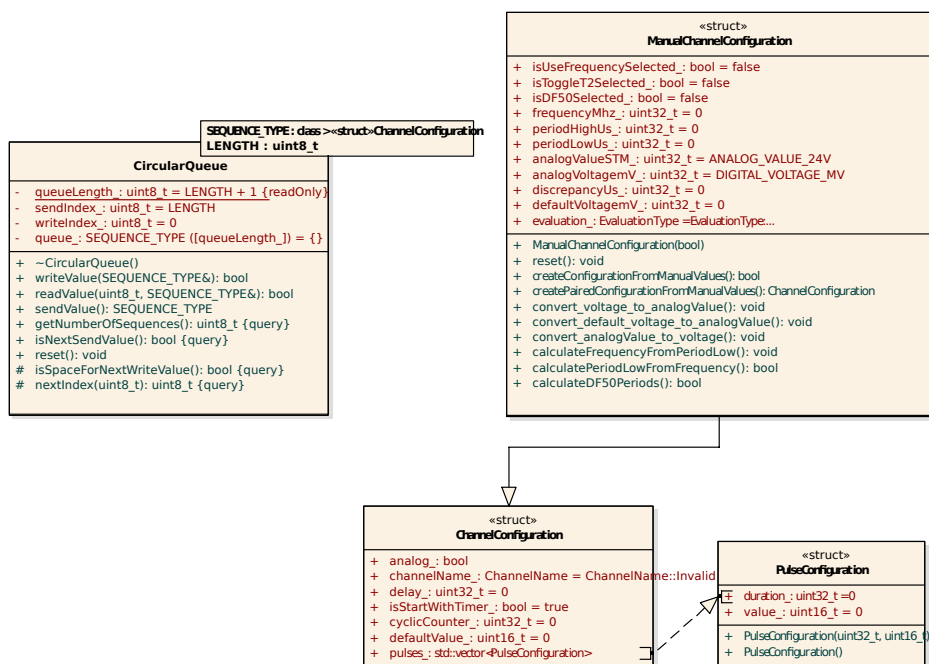
Tato záložka slouží k výběru souboru, ve kterém je uložena XML konfigurace požadovaných výstupních pulzů. Na dolní půlce obrazovky může uživatel vidět obsah adresáře "XML_Sequences", který je součástí projektu a do kterého se ukládají zadané soubory.

Po vybrání testu se uživateli zobrazí jeho název a popis vlevo od stromové struktury adresáře. V horní části obrazovky se také zobrazí náhled výsledného generovaného signálu. Mezi náhledy jednotlivých výstupních kanálů lze přepínat pomocí přepínače vpravo od náhledu.



Obrázek 4.4: Ukázka grafického rozhraní - programový mód.

4.4 Uložení dat o požadované sekvenci



Obrázek 4.5: Ukládání sekvencí v jednodeskovém počítači Raspberry.

Hotové sekvence jsou v Raspberry Pi ukládány stejně jako v MCU do

kruhové fronty. Implementace a velikost fronty je shodná v obou projektech. Samotná data se ukládají do struktury "Channel Configuration", charakterizující celou sekvenci, tak jak bude zaslána do MCU. Rodičem této třídy je třída "Manual Channel Configuration", která obsahuje navíc s přepínače, které může uživatel ovládat v záložce "Manual Control" a také parametry, kterými jsou před odesláním nastaveny délky trvání signálu.

Z manuální konfigurace je vždy před přidáním do fronty vytvořena kanálová konfigurace (Channel Configuration), která je již tvořena pouze daty, která jsou dále zaslána MCU (viz podkapitola 4.7).

4.5 Konfigurace generátoru pomocí souborů XML

Pro potřeby testování je vhodné mít možnost měnit napěťové hladiny výstupu nejen periodicky mezi dvěma hodnotami, ale také být schopen vygenerovat obtížnější sekvence napěťových hladin a délek trvání pulzů. Definování takovýchto požadavků by bylo na použitém displeji velmi nepřehledné, proto je toto grafické rozhraní nedovoluje nastavit. Nastavovat jednotlivé pulzy sekvencí by šlo skrze TIA Portal, ale výrazně by se zvýšil počet nastavovaných parametrů a snížila se přehlednost v parametrech.

Z výše zmíněných důvodů byla implementována možnost konfigurace pulzů v externím souboru XML. V těchto souborech uživatel přednastavuje přesné průběhy pro každý výstupní kanál zvláště pomocí parametrů uvedených v 4.6. Ke každé sekvenci se tedy zapíše její konfigurace (zpoždění, použití externího spuštění, atp.) a dále její jednotlivé pulzy, které se následně budou vykonávat od prvního k poslednímu navolenému. Ke každému kanálu může být přiřazeno až 20 sekvencí, přičemž jejich vykonávání proběhne dle jejich řazení v souboru. V souboru je možno konfigurovat všechny kanály, které generátor implementuje.

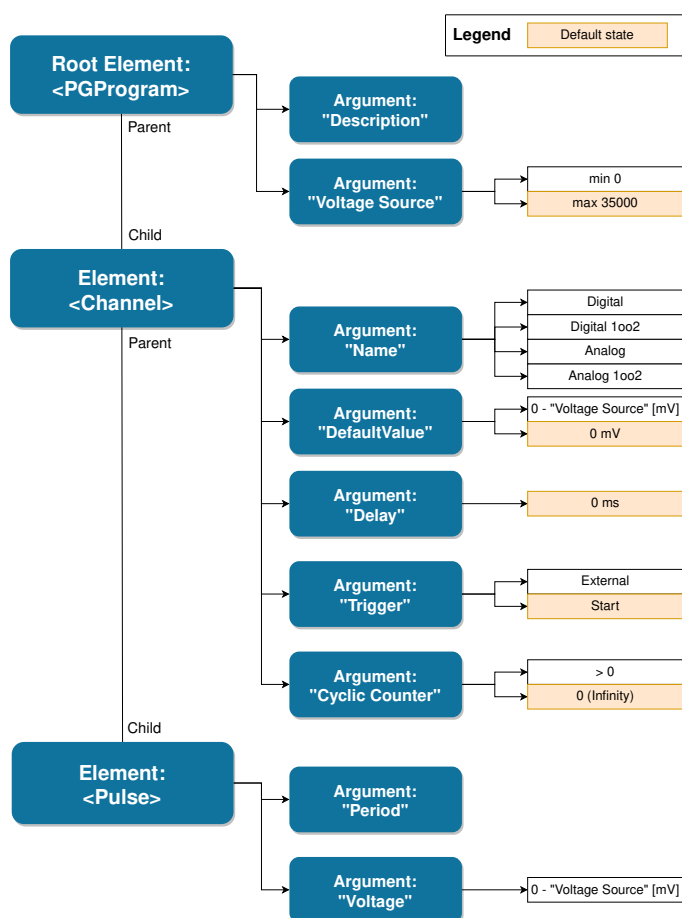
V souboru lze také napsat více sekvencí pro stejný kanál. V takovém případě se budou sekvence vykonávat postupně jedna po druhé. Generátor v takovém případě nemůže zajistit synchronizaci mezi kanály, protože není zřejmé, zda mají být sekvence synchronizovány, či nikoliv. Synchronizace musí být zajištěna správnou konfigurací souboru, například použitím zpoždění pro daný kanál, tak aby sekvence končily ve stejný moment.

V grafickém rozhraní je v režimu "Program Control" zobrazen ve stromové struktuře kompletní obsah složky XML_Sequences. Zde si uživatel může vybrat požadovaný soubor XML. Detailněji je výběr a jeho zobrazení rozebíráno v kapitole 4.3.4.

Chce-li uživatel využít možnosti výběru přes PLC, musí dané soubory ukládat do složek, jejichž jméno začíná číslem v rozsahu 0-255. V těchto složkách musí soubory opět začínat číslem ve stejném rozsahu. Číslo adresáře charakterizuje kapitolu testů a číslo souboru daný test.

Tato čísla tvoří následně dvoubytový identifikátor souboru. Prvním bytem je XML_ Chapter, druhým pak XML_ TestCase. Vždy za číslem na počátku jména souboru nebo adresáře je třeba použít alespoň 1 znak, který číslo oddělí od zbytku jména. Na výběr je tedy z 2^8 možností souborů a každý z nich může mít opět 2^8 možných souborů. Při výběru souboru XML z TIA Portal je dále je také nutné bit "Use_ PredefinedXML" na logickou úroveň 1, čímž se přepne z manuálního módu výběru sekvence do módu výběru souboru.

4.5.1 Struktura XML souboru



Obrázek 4.6: Struktura souboru konfigurujiho průběh výstupů generátoru.

Struktura souboru XML je znázorněna na obrázku 4.6. Každý soubor musí nejprve obsahovat element "PGProgram", který může přijmout argumenty "Description" a "Voltage Source". Argument "Voltage Source" uvádí, na jaký zesilovač máme připojené analogové výstupy zesilovače. V základu nabývá hodnoty 35000. Tato hodnota se dále v souboru uvažuje jako maximální hodnota pro analogové výstupy. Pro tuto hodnotu bude na DAC převodníku

MCU nastavena maximální hodnota.

Hlavní element "PGProgram" může dále obsahovat pro každý ze 4 kanálů až 20 sekvencí. Každá sekvence je charakterizována pomocí elementu "Channel" s argumentem "Name" charakterizujícím o který kanál se jedná. Dále se zde mohou vyskytovat argumenty "Trigger", "Delay", "Default Value" a "Cyclic Counter".

Nakonec každá sekvence musí obsahovat alespoň 1 pulz. Každý element charakterizující pulz musí obsahovat požadovanou délku signálu ("Period") a požadované výstupní napětí v rozsahu 0-"Voltage Source", přičemž u digitálních kanálů se rozlišuje pouze nulová a nenulová hodnota.

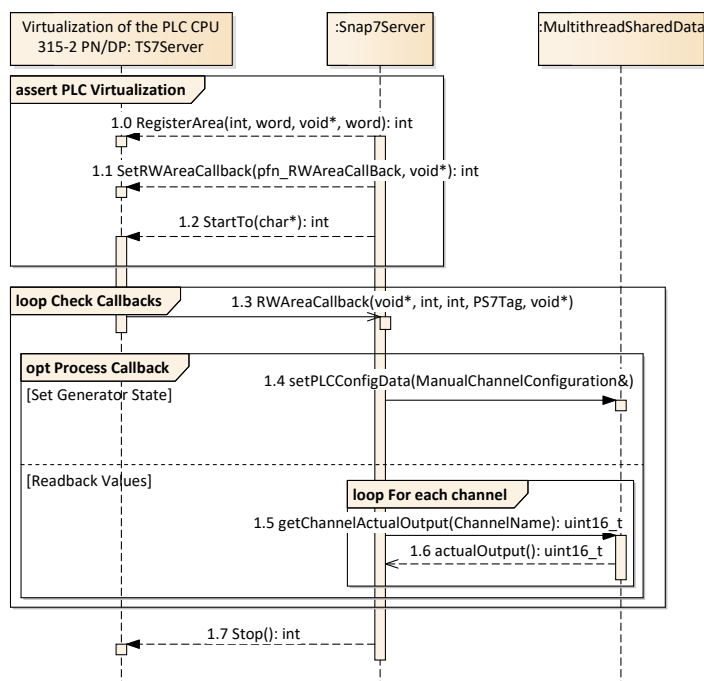
4.6 Komunikace s PLC

Protože knihovna virtualizuje CPU, i v C++ je získávání dat podobné práci v TIA Portalu. Nejprve je třeba zaregistrovat nový datový blok, který bude mít shodné číslo shodné s konfigurací instrukce PUT, tedy bude mít číslo 1 a bude začínat od prvního bytu. K tomuto datovému bloku je připojeno volání, které se spustí vždy při zavolání funkce PUT lokálním PLC. Data v PLC jsou ukládána v endianitě - Big-endian -[9] tedy na nejnižší adrese je uložen nejvýznamější bit. V Raspberry Pi se data nicméně ukládají od nejvíce významného bitu (Little-endian). Je tedy třeba udělat konverzi endianity před přijetím nebo odesláním dat.

4.6.1 Přijímání dat z PLC

Při zaslání dat od skutečného PLC je zavolána funkce "RWAreaCallback". Server je vytvořen na separátním vlákně, je tedy nutné data předat do hlavního vlákna a zajistit synchronizaci dat. Pro předávání dat slouží třída "Multi-threadSharedData". Každá funkce této třídy při zavolání zamkne mutex nad daty. Komunikační vlákno vždy při přijetí nové zprávy nastaví "PLCDataReceivedFlag_" na "false", čímž zablokuje čtení dat z hlavního vlákna. Následně dle konfigurace vytvoří strukturu z uživatelem nastavených dat, případně připraví informaci o požadavku na načtení souboru XML, ve kterém jsou obsaženy instrukce pro generátor. Nakonec povolí čtení dat hlavnímu vlákně změnou stavu "PLCDataReceivedFlag_" na "true". Na obrázku 4.7 můžeme vidět řízení komunikace s PLC v programu pro počítač Raspberry. Pokud je vlákno virtualizující PLC zavoláno, jsou (dle typu volání) zaslána data jako zpětná vazba (směrem z Raspberry do PLC), nebo jsou přijata data o konfiguraci generátoru (z PLC do Raspberry Pi), která jsou uložena do

sdílené struktury s hlavním vláknem.



Obrázek 4.7: Tok dat mezi PLC a Raspberry Pi uvnitř programu generátoru.

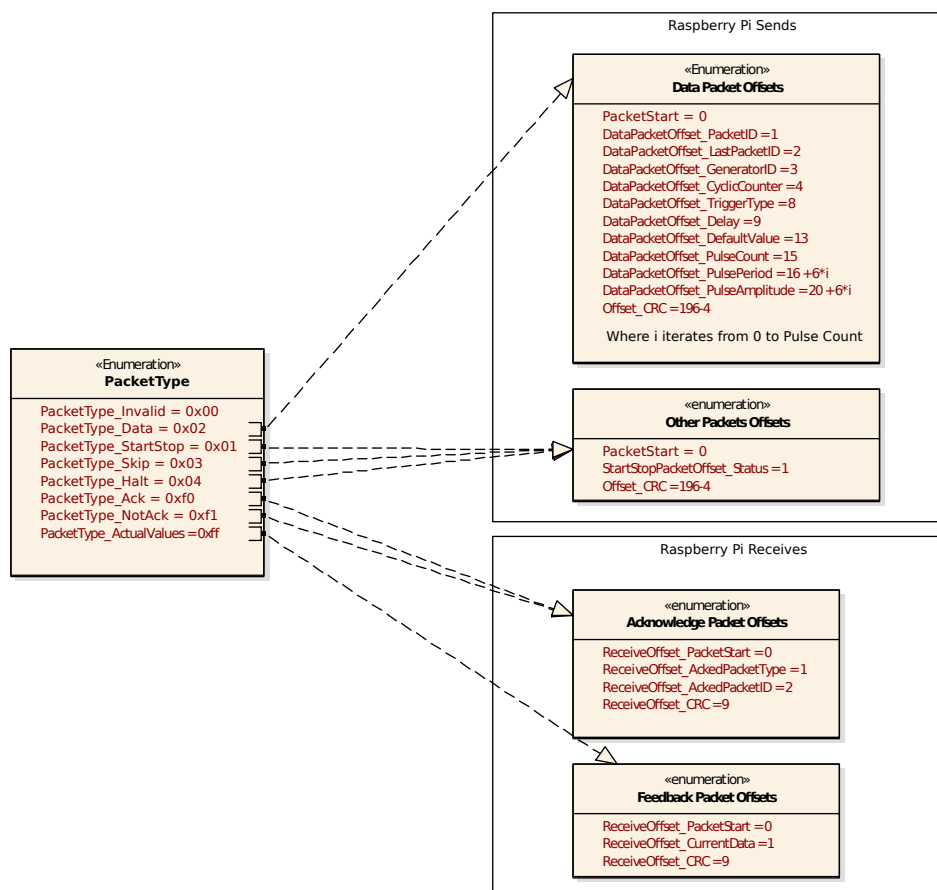
4.6.2 Zasílání zpětné vazby do PLC

Komunikace mezi PLC a Raspberry probíhá také v opačném směru. K tomuto účelu slouží funkční blok GET, který načte data z Raspberry do PLC. Nicméně je nutné zdůraznit, že Raspberry Pi se chová jako server, je proto schopné pouze odpovídat na přijaté požadavky. Tímto požadavkem je v tomto případě žádost o vyplnění datového bloku daty. Raspberry nemůže samovolně zasílat data PLC. Data která v CPU umí generátor vyplnit podávají uživateli zpětnou vazbu o stavu generátoru, je nicméně třeba brát v potaz zpoždění, které vzniká v každé z komponent. Pro rychle měnící se signály je tato zpětná vazba tedy prakticky nepoužitelná. Důvodem její implementace v práci byl požadavek mít zpětnou vazbu k testům, kde nejkratší použitý čas je 500 ms, tedy toto zpoždění lze akceptovat.

4.7 Výměna dat s MCU

Zasílání dat z Raspberry Pi do mikrokontroleru STM32 probíhá pomocí sériové linky s rychlostí 460800 Baud. Data jsou zasílána vždy v paketu s fixní délkou, jehož délka je přímo závislá na maximálním počtu pulzů, které je možné do jedné sekvence nakonfigurovat. Nastavená délka paketu dokáže pojmout až 30 pulzů. Tato délka je zvolena tak, aby s rezervou byla schopna přenést konfiguraci použitou v dosud nejkompexnějších testech. Jestliže je počet pulzů překročen, sekvence nebude spuštěna. Na toto je třeba brát zřetel pouze při vytváření konfiguračních souborů XML. Při manuální konfiguraci je počet pulzů roven 2, tedy tato situace nastat nemůže.

Zasílání dat z Raspberry Pi probíhá metodou "Stop-and-Wait". Raspberry



Obrázek 4.8: Typy použitých paketů a jejich složení.

tedy musí ke každému odeslanému paketu přijmout paket potvrzovací. Jestliže je počítačem Raspberry přijat paket signalizující chybu přenosu, nebo není přijat žádný paket a uběhne nastavený časový limit, stejný paket je odeslán

znovu.

Konec odesílaných i přijímaných paketů obsahuje CRC32, konkrétně je použit polynom 0xF4ACFB13. Je-li CRC přijaté na konci paketu shodné s CRC spočteným ze zbytku přijatého paketu, jsou přijatá data z polynomu zpracována.

Složení jednotlivých paketů je vidět na obrázku 4.8. Komunikace je implementována obousměrně. V TIA Portal je totiž výhodné zobrazovat hodnoty generátoru v aktuální moment. Zpětná komunikace eliminuje nutnost připojení osciloskopu pro každé měření. Zobrazování aktuálních hodnot je důležité při spuštění generátoru s XML konfigurací a spuštěním pomocí externího signálu, neboť v takovém případě nemáme v reálném čase informaci o aktuální výstupní napěťové hladině generátoru.

Data jsou z STM32 čtena cyklicky s periodou 5 ms (nastavení je rozebíráno v kapitole 6.1.3), po které je v STM32 zavoláno přerušení, která uloží aktuální výstupní hodnoty. Toho zasílání není nutné volat co nejrychleji dokola, neboť použití této funkcionality je používáno pouze k zobrazení hodnot v TIA Portal, kterému není možné zasílat data z Raspberry Pi s dostatečnou rychlostí, proto nemá smysl zasílat data z MCU nižší periodou.

Kapitola 5

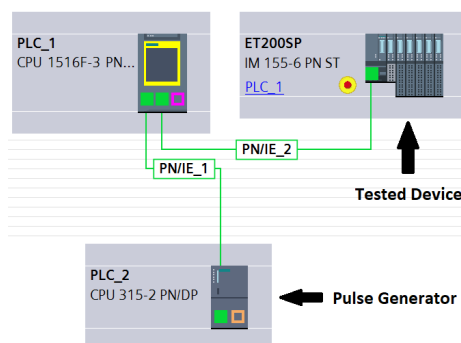
PLC a TIA Portal

5.1 Struktura dat v PLC

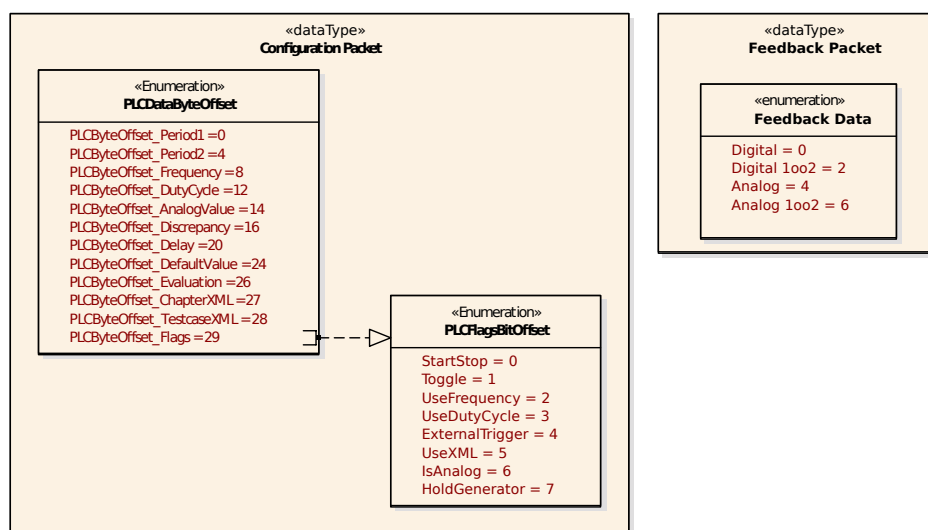
Generátor je v prostředí TIA Portal zobrazen jako CPU 315-2 PN/DP. Zařízení, které představuje generátor je zvoleno dle manuálu knihovny Snap7[9]. Komunikace je realizována pomocí protokolu S7comm, konkrétně funkcí PUT a GET, které se starají o zaslání a přijetí dat mezi dvěma PLC. Z PLC jsou do generátoru zasílána data o požadované konfiguraci generátoru.

Funkcí GET poté PLC může vyčítat aktuální stav výstupních hodnot generátoru. Pro obě tyto funkce jsou v PLC uloženy datové bloky, jejichž data reflektují stav komunikace. Datové bloky předávané mezi PLC a Generátorem musí mít strukturu dle obrázku 5.2.

Přesný způsob nakonfigurování PLC je rozepsán v příloze A.



Obrázek 5.1: Síťové zapojení zařízení k PLC a testovanému modulu.



Obrázek 5.2: Struktura paketů zasílaných mezi PLC a Raspberry Pi

5.2 Kontrola výměny dat

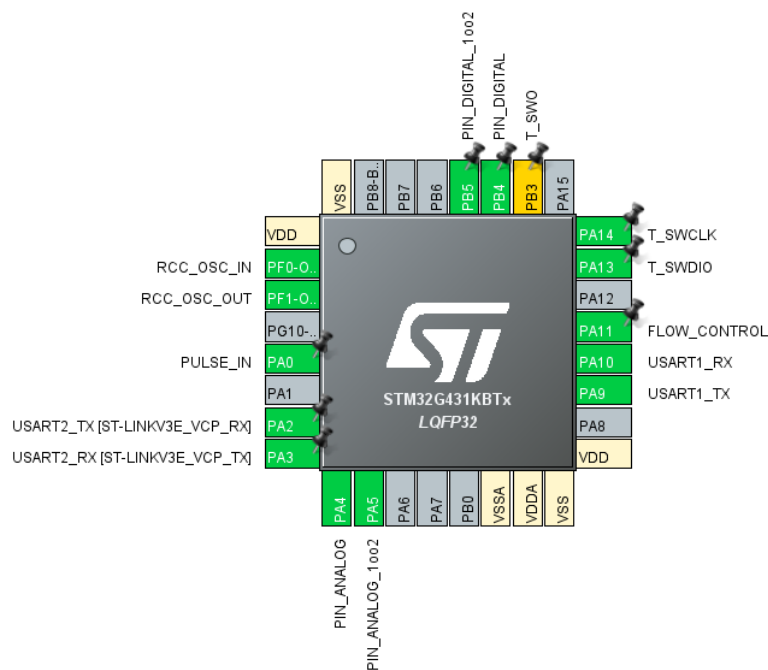
Výše zmíněné systémové datové bloky funkcí PUT a GET jsou využity k zahájení komunikace. Oba tyto bloky obsahují bit, který při náběžné hraně odešle/přijme data (bity Start/REQ).

Pro instrukci PUT je dále v systémovém bloku bit indikující provedení zaslání dat, případně chybu komunikace. U instrukce GET je možné načítat bit NDR (New Data Ready), nastavený vždy po přijetí nových dat a ERROR signalizující chybu přenosu. V obou instrukcích se dále nachází 2 bytová informace o detailnějším stavu přenosu.

Kapitola 6

MCU a generování pulzů

6.1 Konfigurace MCU v STM32CubeIDE



Obrázek 6.1: Vyvedení výstupních signálů mikrokontroleru STM32G431.

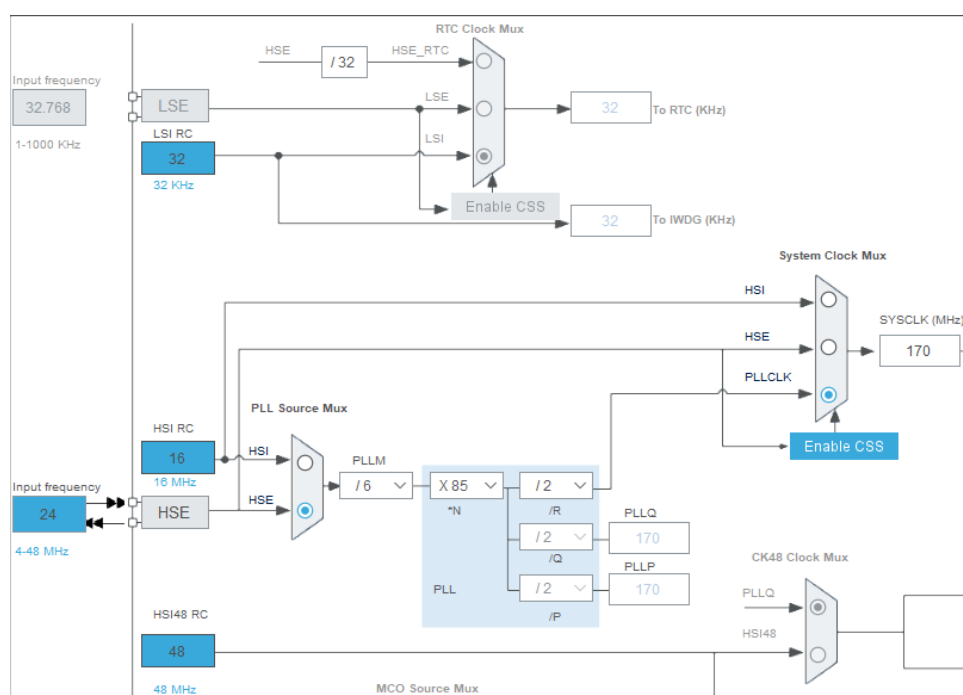
6.1.1 Nastavení hodinového signálu

Pro precizní načasování výstupních pulzů je nutné využít co nejstabilnější způsob generování hodinového signálu. Nejstabilnější způsob generování hodinového signálu, který se na použité desce Nucleo nachází, je externí krystalový oscilátor o frekvenci 24 MHz, který je v manuálu produktu ([12]) označen jako X2.

Dle manuálu vybrané Nucleo desky je třeba pro použití externího krystalového oscilátoru spojit piny SB9 a SB10, zároveň nesmí být spojeny piny SB8, SB11 a SB13. Takto zajistíme připojení oscilátoru na piny PF0 a PF1.

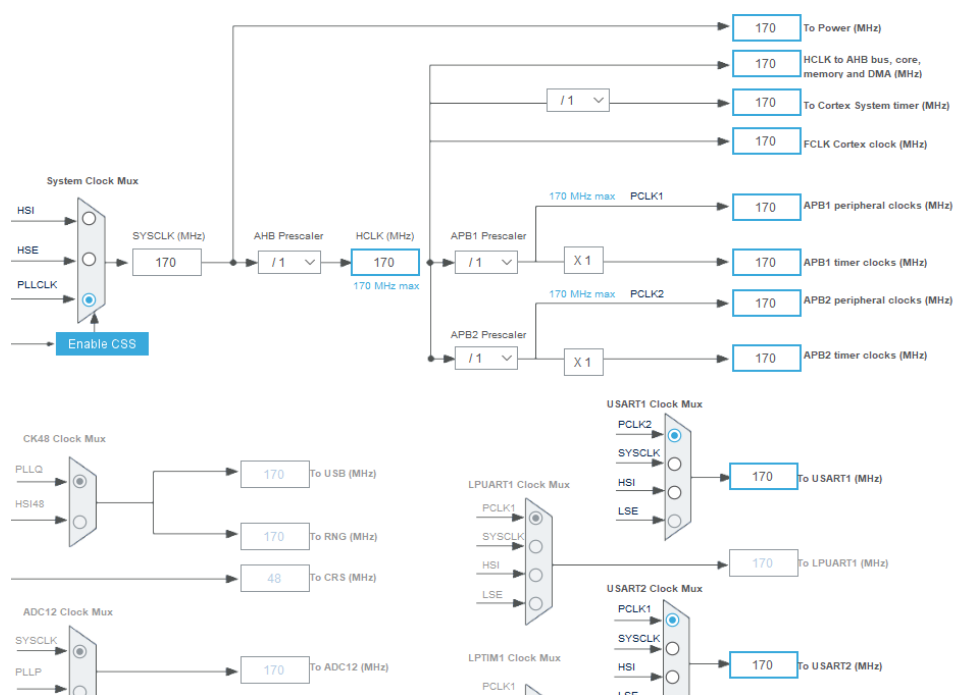
V prostředí STM32CubeIDE můžeme konfigurovat děličky a násobičky hodinového signálu uvnitř MCU, přičemž rovnou v prostředí vidíme, jaká frekvence hodin bude v různých částech MCU.

Výsledná konfigurace hodin je vidět na obrázcích 6.2 a 6.3



Obrázek 6.2: Nastavení hodinového signálu - Phase-Locked Loop.

Při generování kódu z konfigurace (funkcionalita STM32CubeIDE) se data vyplněná v záložce Clock Configuration zapíše do registrů "Reset and Clock Control (RCC)"[20].



Obrázek 6.3: Nastavení hodinového signálu periférií.

6.1.2 Konfigurace časovačů přerušení

V práci jsou využity 2 časovače. TIM2, který se volá s periodou $10 \mu s$ a obsluhuje hlavní smyčku programu, tedy nastavování pulzů. Z této hodnoty lze tedy zjistit maximální časové rozlišení generátoru, které je právě $10 \mu s$. Nyní si můžeme spočítat periodu čítače:

$$t = \frac{x}{f} \quad (6.1)$$

$$x = \frac{170 \cdot 10^6}{10^5} \quad (6.2)$$

$$x = 1700 \quad (6.3)$$

, kde f = frekvence, t = čas a x = nastavení čítače.

Po kontrole nastavení čítače hodin na osciloskopu byla tato hodnota snížena na hodnotu 1699, která blíže odpovídá specifikovaným $10 \mu s$. Vzhledem k nastaveným hodinám je třeba časovač nastavit následovně:

- Prescaler = 0
- Counter Mode = Up
- Counter Period = 1699
- auto-reload preload = Enable

Druhým časovačem je TIM4, který ukládá nastavené hodnoty generátoru a v případě, že je nastaven bit FLOW_CONTROL a přechází paket byl již odeslán, data odešle do Raspberry Pi. Toto časování není nutné navolit na přesně stanovenou hodnotu, vzhledem o pomalé komunikaci mezi STM32 a Raspberry Pi nemá smysl nastavovat přerušování příliš často a vytěžovat MCU. Zvolené nastavení odpovídá periodě přibližně 5 ms mezi přerušováními.

- Prescaler = 1024
- Counter Mode = Up
- Counter Period = 850
- auto-reload preload = Enable

Toto přerušování může odeslat také potvrzovací paket o přijatých datech, pokud tento nemohl být odeslán ihned po správném přijetí. Více k problematice v kapitole 6.2.

■ 6.1.3 Nastavení sběrnice UART

Na vybrané desce Nucleo je již připojen UART2 a to k programátoru STLinkV3, proto využijeme druhou sériovou linku UART1. Nejprve nastavíme mód sériové linky jako asynchronní. Následně ověříme, že se v Pinout view nakonfiguroval UART na piny PA9 a PA10, přičemž pin PA10 je označen jako RX (příjímač) a pin PA9 jako TX (vysílač). Následně otevřeme parametrizaci UART1 a nastavíme následující parametry

Především se zaměříme na parametry kategorie "Basic parameters", Data Direction. Nezvyklý je výběr parametru TX and RX Pins Swapping, který bude nastaven na enabled, neboli prohození pinů TX a RX. Tato konfigurace je zvolena z důvodu snazšího rozložení desky plošných spojů.

Pro odesílání dat je využito DMA (Direct Memory Access). Konkrétně kanál DMA1_Channel6, který je připojen na USART1_TX.

MCU zasílá periodicky data do Raspberry Pi, tato data nicméně jsou zasílána pouze pokud je pin PA11 nastaven na logickou hodnotu jedna. Proto pin PA11 nastavíme jako GPIO_Input a nazveme pin "FLOW_CONTROL". Tento pin se tedy chová jako CTS (Clear to Send).

<ul style="list-style-type: none"> <ul style="list-style-type: none"> Basic Parameters 	
Baud Rate	460800 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
<ul style="list-style-type: none"> <ul style="list-style-type: none"> Advanced Parameters 	
Data Direction	Receive and Transmit
Over Sampling	16 Samples
Single Sample	Disable
ClockPrescaler	1
Fifo Mode	Disable
Txfifo Threshold	1 eighth full configuration
Rxfifo Threshold	1 eighth full configuration
<ul style="list-style-type: none"> <ul style="list-style-type: none"> Advanced Features 	
Auto Baudrate	Disable
TX Pin Active Level Inversion	Disable
RX Pin Active Level Inversion	Disable
Data Inversion	Disable
TX and RX Pins Swapping	Enable
Overrun	Enable
DMA on RX Error	Enable
MSB First	Disable

Obrázek 6.4: Parametry sériové linky komunikující s Raspberry Pi.

6.1.4 Nastavení výstupních periférií

Generátor je schopen produkovat 2 analogové a 2 digitální výstupní signály. Digitální výstupní signály jsou připojeny na piny PB4 a PB5, v sekci Digitální výstupy nastavíme v sekci Pinout->System Core->GPIO(General-purpose input/output) je k oběma těmto výstupům přiřazeno následující nastavení:

- GPIO output level = Low
- GPIO mode = Output Push Pull
- GPIO Pull-up/Pull=down = No pull-up and no pull-down
- Maximum output speed = Very High

Analogové výstupy nastavíme na PA4 a PA5 jako DAC_OUT 1 a 2. Opět v sekci Pinout->Analog->DAC1 nastavíme u pinů:

- OUT1 a OUT2 mode = Connected to external pin only

Konfigurace jednotlivých výstupů DAC převodníku je znázorněna na obrázku 6.5.

▼ DAC Out1 Settings	
Output Buffer	Enable
DAC High Frequency	Mode Automatic
DMA Double Data	Disable
Signed Format	Disable
Trigger	None
Trigger2	None
User Trimming	Factory trimming
Sample And Hold	Sampleandhold Disable
▼ DAC Out2 Settings	
Output Buffer	Enable
DAC High Frequency	Mode Automatic
DMA Double Data	Disable
Signed Format	Disable
Trigger	None
Trigger2	None
User Trimming	Factory trimming
Sample And Hold	Sampleandhold Disable

Obrázek 6.5: Nastavení DAC převodníků.

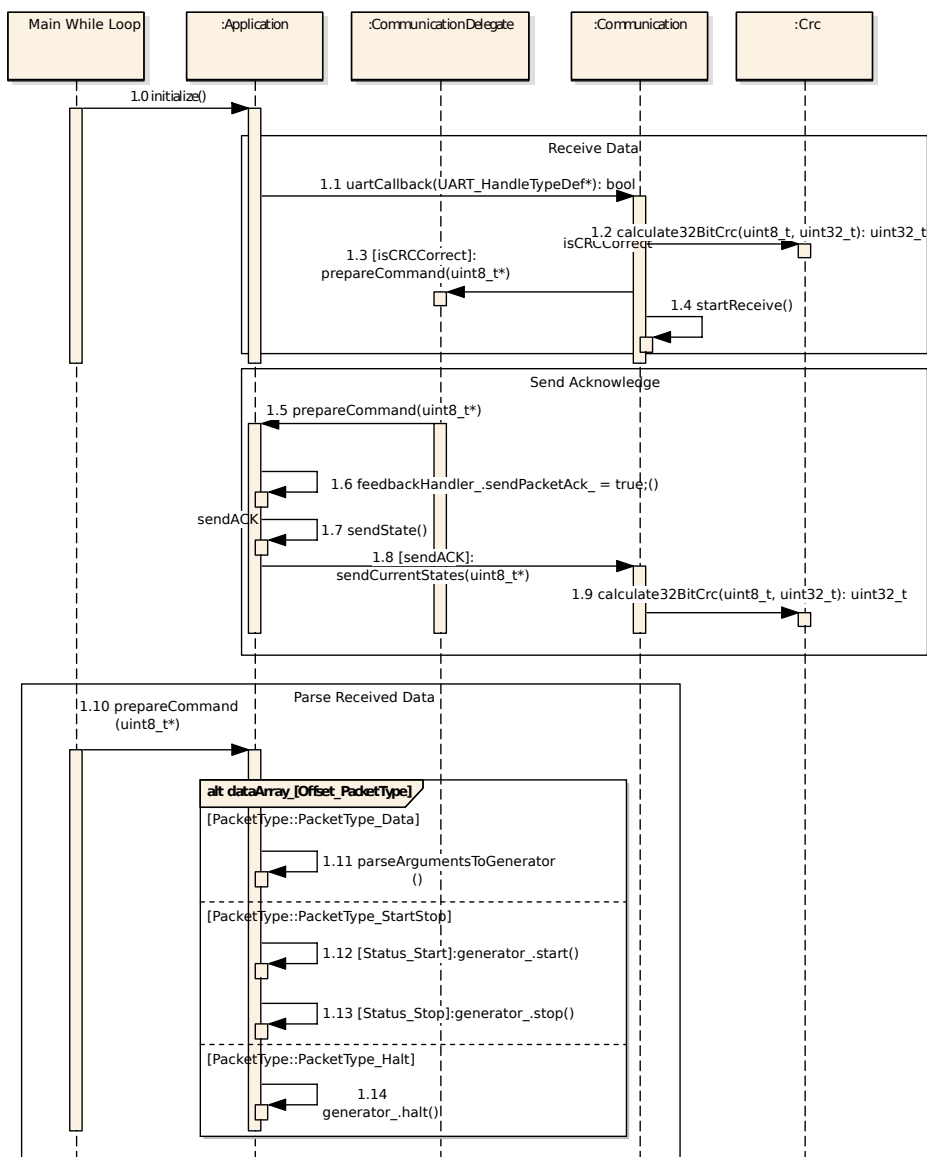
Nakonec nakonfigurujeme pin PA0 jako GPIO_Input, nastavíme pull-down pinu. Mód pinu nastavíme na "External Interrupt Mode with Rising edge trigger detection", neboli volání přerušení při detekci náběžné hrany signálu. Pin pojmenujeme PULSE_IN.

■ 6.1.5 Konfigurace NVIC

Z důvodu potřeby vyšší priority pro přerušení ovládající délku pulzu nejprve zvýšíme počet bitů určujících prioritu na 4 bity. Čím nižší číslo má přerušení nastaveno jako "Preemption priority", tím je jeho priorita vyšší. Priority přerušení tedy nastavíme následovně:

- TIM2 global interrupt = 0
- EXTI line0 interrupt = 1
- USART1 global interrupt = 2
- DMA1 channel6 global interrupt = 3
- TIM4 global interrupt = 4
- Jiné = 0

6.2 Zpracování přijatých dat



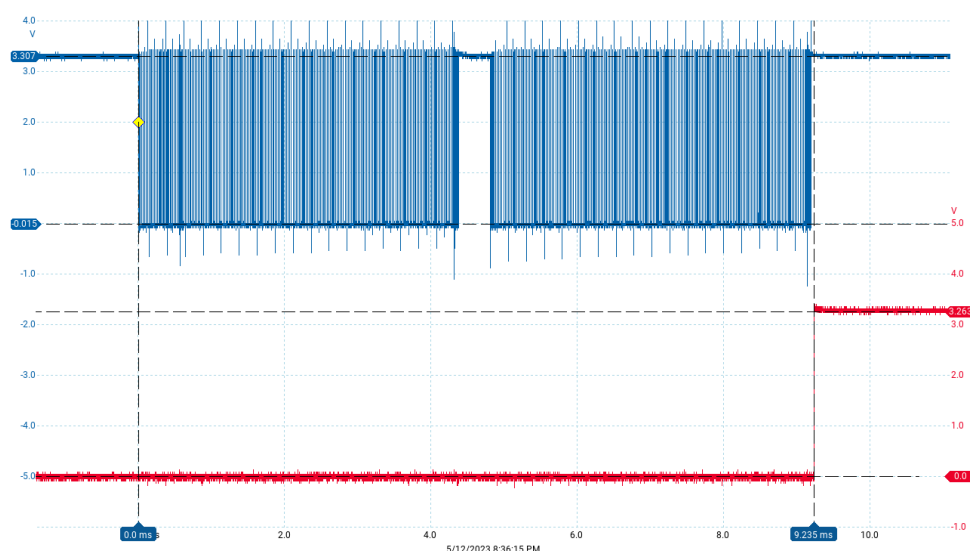
Obrázek 6.6: Sekvenční diagram znázorňující přijímání dat v MCU.

Je-li přijat nový paket, třída "Communication" započne příjem nového paketu a přijatá data předá třídě "Application", která data uloží a nastaví vlajku značící, že jsou nová data k dispozici. V samotném přerušení nejsou data hlouběji zpracovávána, aby se předešlo chybě "Overrun" - nepřijetí následujícího start bitu před zpracováním aktuálního přerušení. V hlavní smyčce programu je volána funkce "processData", která při nastavení vlajky nově přijatých dat tato data zpracuje. Po zpracování je opět vlajka zresetována.

Je-li přijatý paket typu Start/Stop nebo Hold, je ihned změněn stav generátoru.

Je-li přijatý paket typu "Data", zkontroluje se jeho identifikační číslo. Pokud je toto číslo shodné s počtem paketů, které mají být přijaty, všechny přijaté sekvence jsou připojeny na konec fronty generátoru. Přijaté sekvence jsou přiloženy až při posledním přijatém paketu z důvodu synchronizace mezi přijatými sekvencemi. Pokud by byly sekvence přikládány rovnou, došlo by mezi výstupními kanály k opoždění následující přijímané sekvence o čas, po který byl paket přijímán a zpracováván. Byty se posílají po jednom, bez parity, přičemž začínají start bitem a končí jedním stop bitem. Tedy pro každý přenesený byte dat je nutné odeslat 10 bitů. Vydělíme-li desetinásobek počtu bytů rychlostí přenosu, získáme dobu přenosu jednoho paketu, která je tedy $\approx 4,28ms$. Pro start jednoho kanálu s manuální konfigurací je třeba odeslat 1 datový paket a následně paket start.

Důvodem pro zvolení tohoto přístupu byl požadavek na zachování synchro-



Obrázek 6.7: Změřené zpoždění po požadavku na start generátoru.

nizace mezi kanály. Naopak nevýhodou je nutnost přijetí všech paketů před odstartováním sekvence. Zpoždění mezi konfigurací startu v Raspberry Pi a reálným startem na výstupu mikrokontroleru STM32 roste lineárně s počtem zaslaných sekvencí. Po manuálním odstartování bude v nejhorsím případě, bez uvažování chyb v přenosu, sekvence spuštěna za:

$$t = t_{data} \cdot 2 + t_{TIM4} \quad (6.4)$$

$$\approx 4,28ms \cdot 2 + 5,12ms \quad (6.5)$$

$$\approx 13,68ms \quad (6.6)$$

Po přijetí paketu je odeslán pomocí DMA a UART paket obsahující potvrzení o správnosti přijatého paketu. Není-li po přijetí paketu zaneprázdněno DMA,

je potvrzovací paket odeslán ihned. Pokud DMA při přijetí odesílá právě data, je potvrzovací paket zaslán až v následujícím přerušení, tedy po zhruba 5 ms.

6.3 Vytvoření struktury sekvence z přijatého paketu.

Strukturu přijatého datového paketu můžeme vidět na obrázku 4.8. První 3 byty paketu určují jeho typ a identifikační číslo. Tato data jsou pro sestavení sekvence nepotřebná, nebudeme se jimi v této části tedy dále věnovat. Byty 3 až 15 obsahují konfiguraci dané sekvence, tedy:

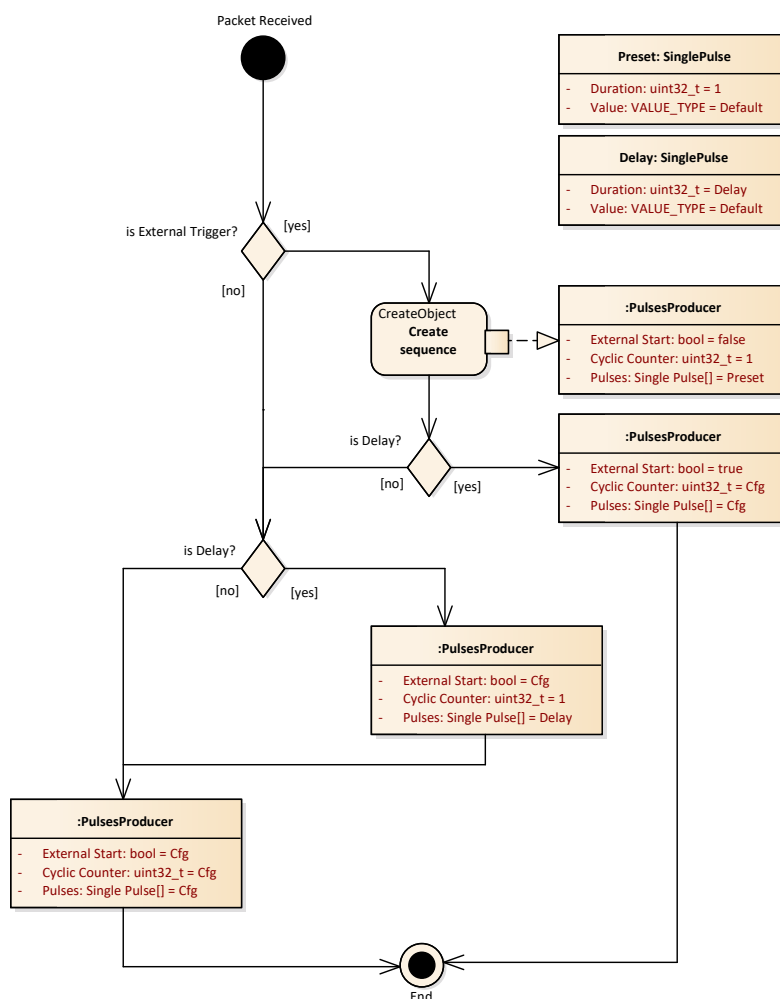
- GeneratorID -> identifikace nastavovaného kanálu
- Cyclic Counter -> Počet iterací přes pulzy
- Trigger Type -> Typ spuštění sekvence
- Delay -> Zpoždění kanálu po startu
- Default Value -> Výstupní napěťová hladina, která se nastaví při čekání na externí vstup a během posunutí sekundární sekvence
- Pulse Count -> Počet pulzů v sekvenci

Každá sekvence může obsahovat až 30 různých pulzů, které jsou charakterizovány časovým trváním a napěťovou hladinou. Tyto pulzy jsou uloženy v kruhové frontě. Každá iterace přes tuto frontu zvýší počítadlo cyklů. Je-li tato hodnota shodná s proměnnou "Cyclic Counter", spustí generátor následující sekvenci.

Jak je vidět na rozhodovacím stromě 6.8, jeden datový paket může vytvořit jednu až tři sekvence. Dodatečně vytvořené sekvence se v MCU starají o zpoždění signálu, je-li nastaven parametr "Delay" nebo "Discrepancy", nebo o čekání na detekci externího spuštění generátoru přerušením EXTI.

6.4 Členění objektů generátoru

Při spuštění MCU se inicializuje třída "Application", která zpracovává všechna přerušení a propaguje je následně ostatním třídám. Tato třída také zajišťuje analyzování přijatých dat a ukládání výstupních napěťových hodnot pro zpětnou vazbu. Jejím rodičem je třída "CommunicationDelegate", která deleguje volání o přijatém paketu od třídy "Communication" pomocí svých virtuálních metod. Třída "Communication" jak název napovídá spravuje komunikaci

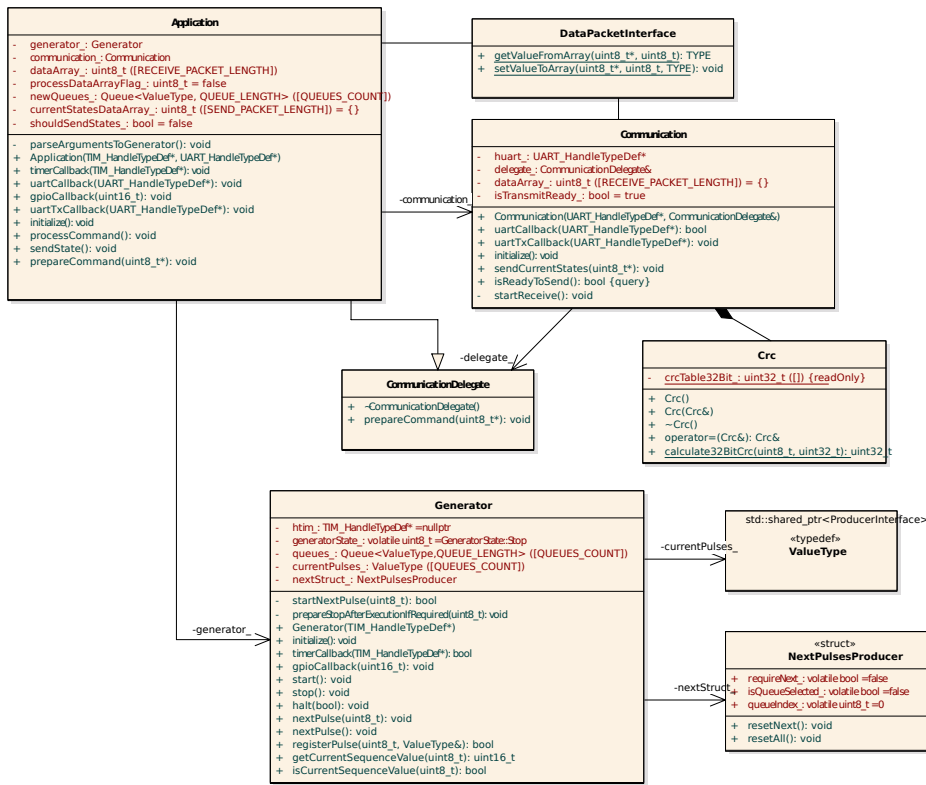


Obrázek 6.8: Rozhodovací strom tvorby sekvencí z přijatého paketu.

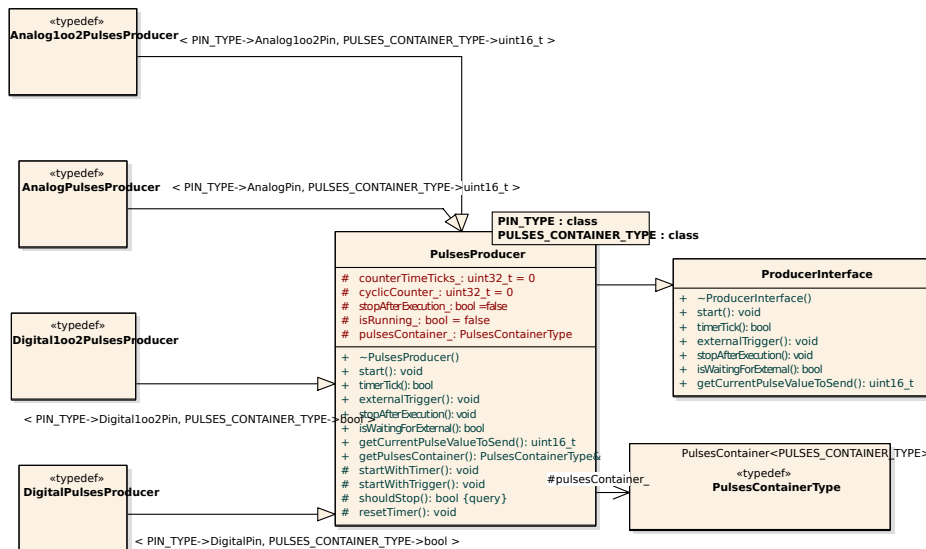
s Raspberry Pi (detailněji rozebíráno v kapitole 6.2).

Třída "Generator" obsluhuje již samotné generování výstupních pulzů a to pomocí svých členů "PulsesProducer". "Generator" propaguje přerušení TIM2 do jednotlivých kanálů podle aktuálního stavu (Start, Stop, ...) a přepíná aktuálně běžící sekvence.

Každý kanál generátoru je potomkem třídy "PulsesProducer". Při inicializaci tohoto objektu se již kanálu přiřazuje výstupní Pin. Třída "PulsesProducer" zajišťuje třídě "Generator" přístup k výstupním kanálům přes své rozhraní "ProducerInterface". Tato třída také obsahuje konfiguraci aktuálních sekvencí a spravuje sekvence dle počtu jejich iterací a aktuálního stavu generátoru.

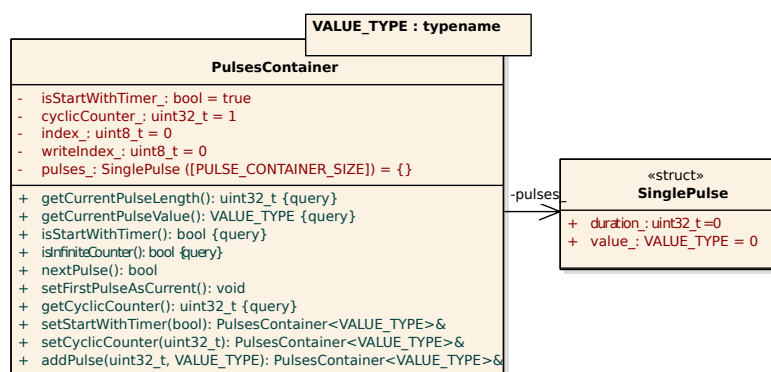


Obrázek 6.9: Vazba objektů zajišťujících správu celého zařízení STM32.



Obrázek 6.10: Vazba objektů spravujících jednotlivé kanály.

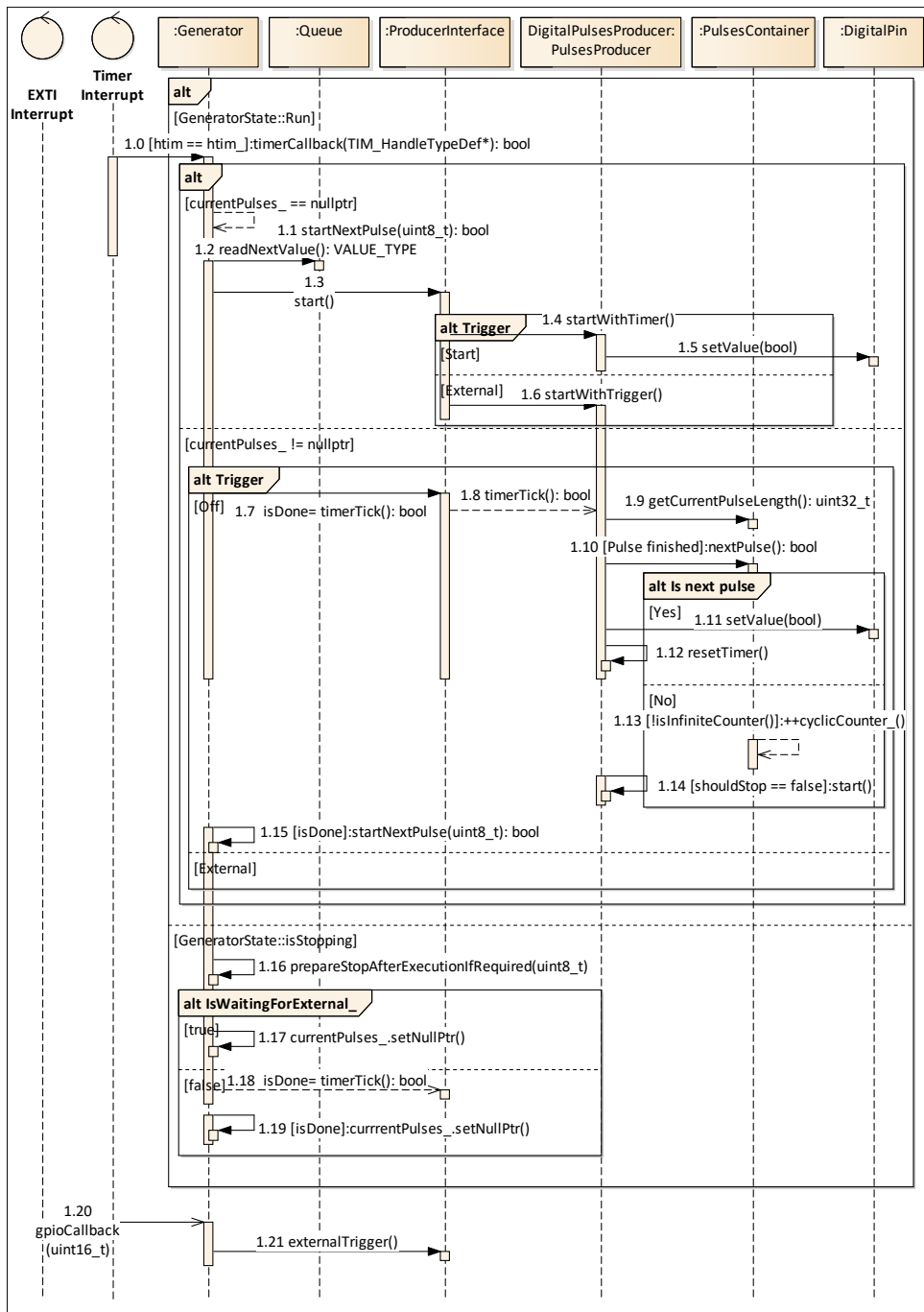
Samotné sekvence jsou implementovány třídou "PulsesContainer", která se stará o přepínání pulzů dle aktuální doby, kterou je pulz nastaven. Pulzy jsou implementovány strukturou definovanou uvnitř této třídy. Každý pulz je již tvořen pouze délkou trvání, která je charakterizována délkou trvání v jednotkách $10\mu s$ a napětovou hladinou, což je proměnná dle typu kanálu.



Obrázek 6.11: Vazba objektů definujících sekvenci a její konfiguraci.

6.5 Generování pulzů

Kód v STM32 musí být schopen především v cyklickém přerušeni běžet co nejrychleji tak, abychom přešli situaci, kdy přerušeni nestihne být obslouženo, což zapříčiní ztrátu následujícího přerušeni a tím dojde k protažení pulzu. Jedním ze zvolených přístupů je tedy omezení dynamické alokace paměti. Vzhledem k požadavku na zachování obecnosti kódu pro více druhů MCU byla využita hardwarová abstrakční vrstva (HAL). Toto rozhraní umožňuje využívat stejné metody pro různé typy MCU.



Obrázek 6.12: Program MCU - řízení výstupních napěťových hladin.

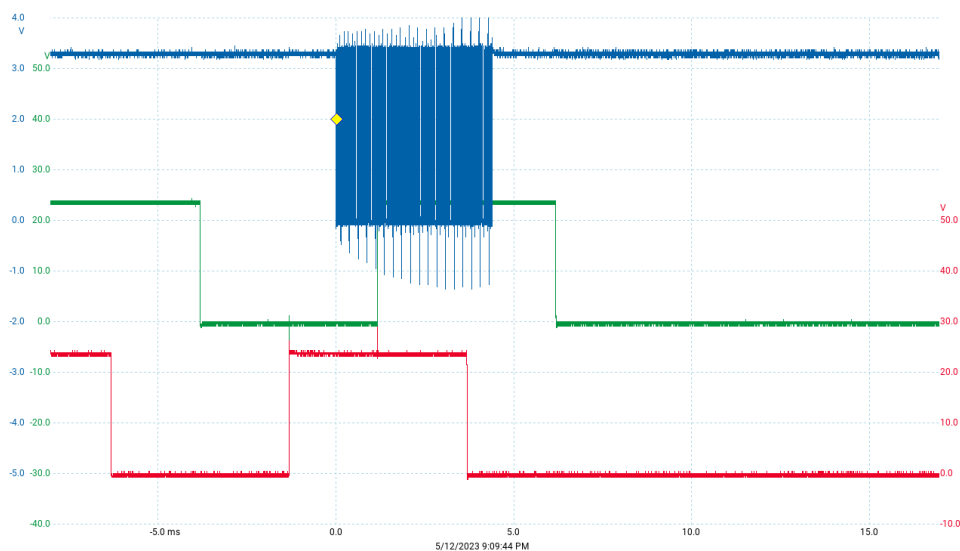
Generátor manipuluje s celými uloženými sekvencemi. Zjednodušený sekvencí diagram řízení generátoru je vidět na obrázku 6.12. V diagramu se pro přehlednost neobjevují všechny rozhodovací podmínky, které program během běhu provádí. Generátor neměří přímo uběhlý čas každého pulzu, ale počítá zavolaná přerušení TIM2. V každém přerušení se zkontroluje, zdali byl

s aktuálně běžícím pulzem přijat počet přerušení odpovídající délce pulzu. Je-li tomu tak, je nastavena napěťová hladina následujícího pulzu. Pokud není další pulz v poli a sekvence již byla opakována tolikrát, kolikrát požadoval uživatel v konfiguraci, je nastavena další sekvence.

Tento přístup umožňuje připravit si následující sekvence do fronty a umožňuje v budoucnu doimplementovat již připravený paket "Skip", který bude schopen mezi sekvencemi přepnout, aniž by se na výstupu vyskytly nežádoucí pulzy nepredikovatelné délky.

Jak můžeme vidět na obrázku 6.13, generátor musí dokončit aktuální běh sekvence i pokud je přijat paket s instrukcí k zastavení generátoru. Aktuální cyklus pulzů se tedy dokončí vždy, čímž je zajištěno, že po zastavení se aktuální pulz nezakončí dříve a jeho nepředvídatelná délka trvání zapříčiní nepredikovatelnou reakci připojeného testovaného modulu.

Je-li nutno generátor vypnout okamžitě, může uživatel uzavřít na Raspberry



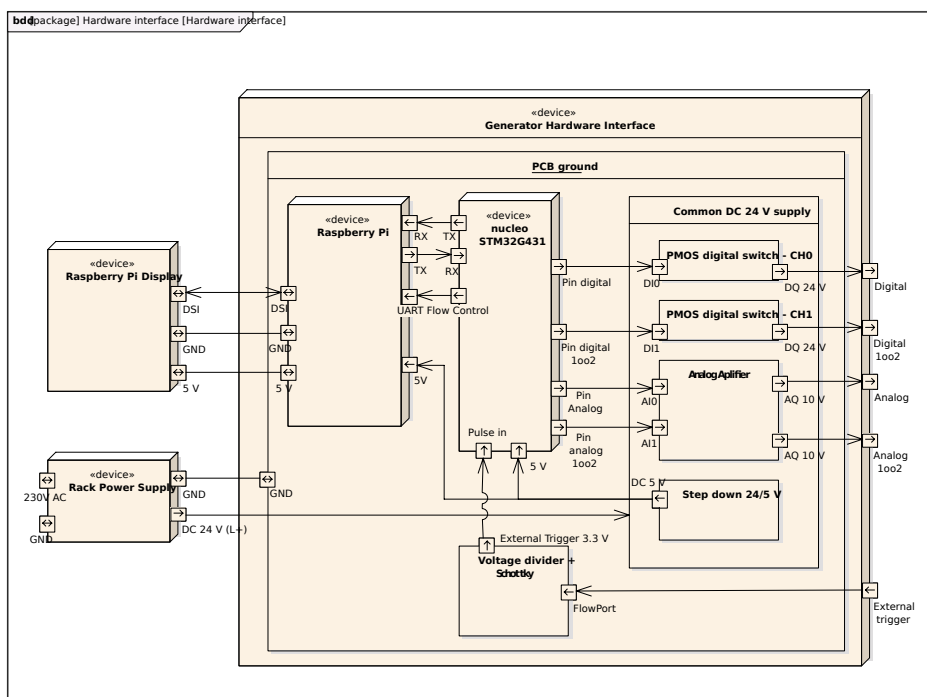
Obrázek 6.13: Dokončení sekvence generátorem po přijetí paketu Stop.

Pi aplikaci generátoru. V takovém případě se odešle příkaz Hard Stop, který generátor vypne ihned. Grafické rozhraní a tedy i aplikace se uzavře opět až po přijetí potvrzení doručení paketu.

Generátor obsahuje také pin s nakonfigurovaným přerušením (EXTI) na náběžnou hranu vstupního signálu. Je-li konfigurace aktuální frekvence nastavena na externí spuštění, při detekci náběžné hrany signálu se spustí daná sekvence. Příklad takového průběhu je vidět v kapitole 8.3.

Kapitola 7

Hardware



Obrázek 7.1: Schéma hardware použitého v práci.

Zařízení slouží jako generátor napěťových hladin pro IO moduly PLC, jejichž digitální moduly fungují na napěťové hladině 24 V, vstupy analogových modulů mají rozsah do 10 V. Výstupní signály z generátoru je třeba převést na tyto napěťové hladiny. K tomuto účelu byla navržena a realizována deska plošných spojů, která v sobě také integruje napájení MCU a jednodeskového

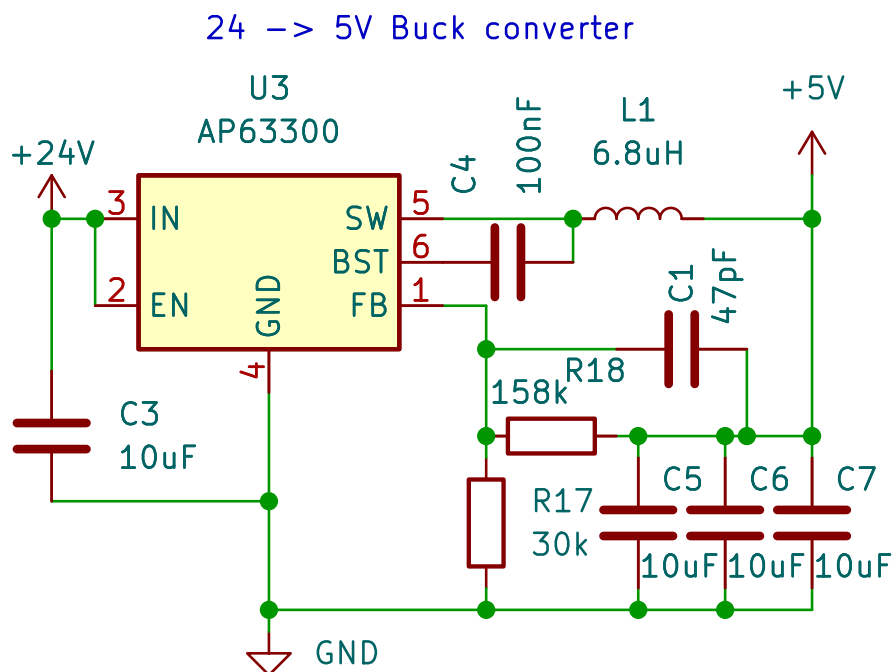
počítače Raspberry, včetně jeho semipalcového displeje. Dále se na desce nalézají níže zmíněné ochrany MCU a Raspberry Pi.

Propojení periférií Raspberry Pi a Nucleo desky je realizováno pomocí kolíkových zásuvek s roztečí 2,54 mm, což odpovídá standardu rozložení desek Nucleo i Raspberry Pi.

Návrh zapojení a rozvržení desky plošných spojů bylo realizováno pomocí software KiCad V6.0[21].

7.1 Schéma zapojení

7.1.1 DC-DC měnič napětí



Obrázek 7.2: Schéma zapojení spínaného regulátoru napětí.

Raspberry Pi i použité STM32 není možné napájet přímo z 24 V. Obě desky nicméně podporují napájení z 5V. Samotné Raspberry Pi má doporučený napájecí proud 2,5 A, což je schopna dodat oficiální nabíječka dodávaná v Raspberry Pi kitu. Po vyzkoušení připojení Raspberry, displeje a desky STM32 k této nabíječce bylo ověřeno, že tato nabíječka je schopna dodávat dostatečný výkon všem těmto zařízením dohromady. Výstupní napětí oficiálně

prodávané nabíječky je udáváno jako 5.1 V[22]. Při výběru regulátoru napětí budeme tedy hledat takovou součástku, která je schopna dodat při napětí 5.1 V alespoň 2,5 A. Pro stabilizaci napájecího napětí byl využit integrovaný spínaný obvod (buck converter). Lineární regulátor napětí by měl vzhledem k velkému rozdílu napětí a vysokému výstupnímu proudu příliš vysoké ztráty, které by vedly k potenciálním problémům s přehříváním zařízení.

Zvoleným spínaným regulátorem napětí je AP63300WU-7 od firmy Diodes Incorporated, označen jako U3. Důvodem pro výběr této součástky je vysoký dodaný výstupní proud a spínací frekvence 500 kHz, díky které není nutné používat cívky s vysokou indukčností. Díky vysoké spínací frekvenci je možné, i s použitím cívky s nízkou indukčností, splnit požadavek na nízké zvlnění výstupního signálu. Menší cívka poté dovoluje navrhnut deskou plošných spojů na menší ploše, díky čemuž se zařízení lépe vkládá do krabičky.

Dalším důležitým parametrem cívky je RMS proud, který musí být alespoň 3A, které je schopen dodávat spínaný regulátor. Vzhledem k výše zmíněným požadavkům byla vybrána cívka SRP6530A-6R8M, jejíž RMS proud je 5.5 A a indukčnost $6.8\mu H$.

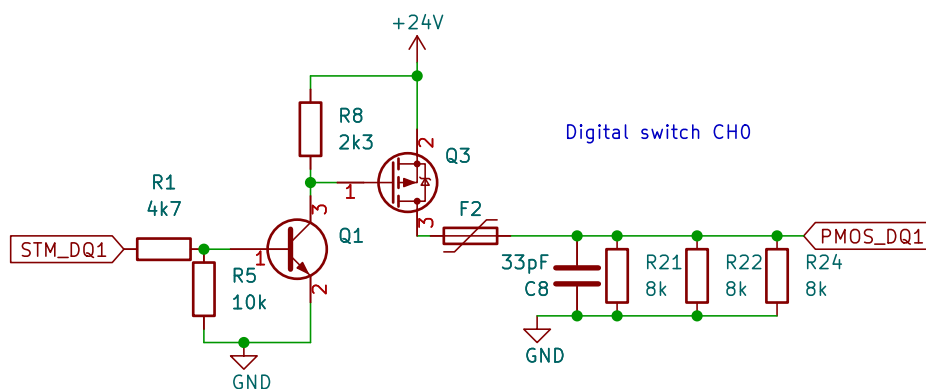
Výstupní napětí regulátoru je nastavováno pomocí děličem napájecího napětí v obvodu zpětné vazby. Odpory použité v děliči napětí lze spočítat dle požadovaného výstupního napětí. Dle vzorce z manuálu výrobce[23] tedy spočteme výsledné odpory:

$$R18 = R17 \cdot \left(\frac{U_{out}}{0,8V} - 1 \right) \quad (7.1)$$

$$= 30k\Omega \left(\frac{5,1V}{0,8V} - 1 \right) \quad (7.2)$$

$$= 158k\Omega \quad (7.3)$$

7.1.2 Digitální PMOS spínač



Obrázek 7.3: Schéma zapojení digitálního spínače výstupního signálu.

Digitální výstupní kanály mají výstupní napětí 24 V. Obvod pro zesílení digitálního výstupu MCU je složen s tranzistoru Q3 (FDN5618P - PMOS tranzistor), který může připojit na výstup kladné napětí 24 V a to v případě, že je na gate tranzistoru přivedeno napětí nižší než:

$$U_{gs} < U_{VCC} - U_{th} \quad (7.4)$$

$$\approx 24V - 1.6V \quad (7.5)$$

$$\approx 22,4V \quad (7.6)$$

Vzhledem k nízkému výstupnímu napětí MCU není možné spínat PMOS Q3 přímo, proto je na gate připojen NPN tranzistor Q1, který je již ovládán digitálním pinem MCU, přivedeným na bázi tranzistoru. Při kladném napětí výstupního pinu MCU je NPN tranzistor otevřen a gate tranzistoru PMOS je připojen k zemi. V tomto případě je napětí na gate 24 V, přičemž dle datasheetu[24] je třeba napětí alespoň 20 V. V případě nízkého výstupního napětí MCU je NPN tranzistor uzavřen a na gate Q3 je saturační napětí kolektor-emitor použitého NPN tranzistoru.

V případě sepnutí je náboj dodáván z kolektoru použitého PMOS tranzistoru, v případě rozepnutí tranzistoru Q1 a tedy i Q3 je náboj kapacitní zátěže disipován skrze připojené rezistory R21, R22 a R24. Důvod pro připojení 3 rezistorů paralelně je rychlejší disipace náboje a tedy i rychlejší pokles napětí při rozpojení tranzistoru Q3. Pokud by byl použit pouze jeden SMD odpor, disipoval by více energie, než je přípustné.

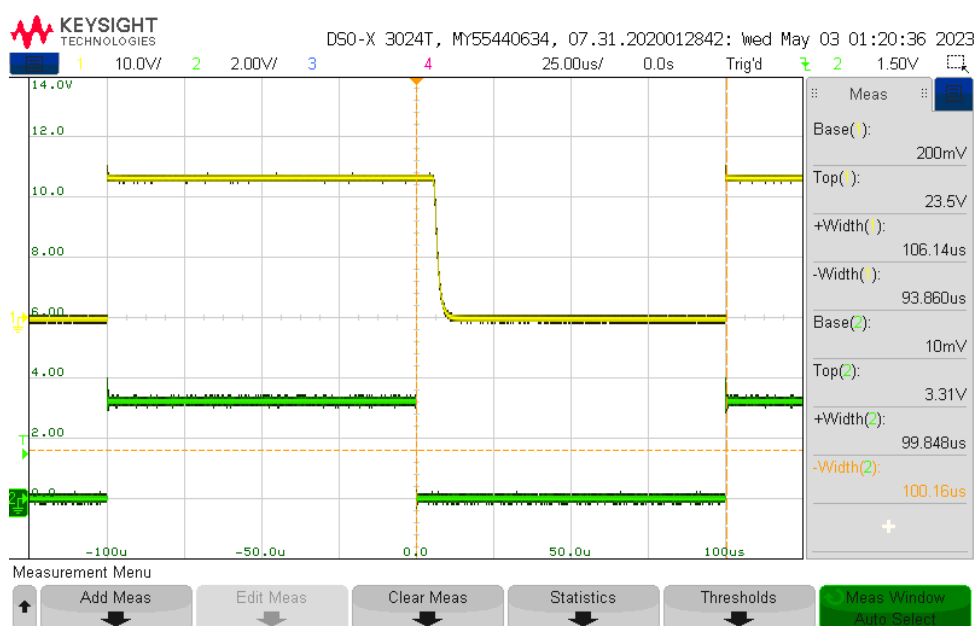
Bohužel v případě, kdy by byl PMOS uzavřen, by byl výstup nedefinovaný a napětí by klesalo velmi pomalu pouze skrze připojenou zátěž. Z tohoto důvodu jsou za drain tranzistoru Q3 připojeny paralelně k zemi 3 odpory o hodnotě 8,2 kΩ. Při připojení zátěže, která je připojena dlouhým vodičem dojde k přechodovému jevu z důvodu indukčnosti vodiče a tím k překmitu napětí. K omezení tohoto jevu slouží kondenzátor C8.

Pro spínání byl vyzkoušen také integrovaný obvod FOD3182 od výrobce onsemi. Tento obvod je velmi rychle (viz 7.4b) schopen dlouhodobě spínat proudy okolo 1 A, jeho nevýhodou se ale ukázala malá schopnost zvládat elektrostatické výboje, vzhledem ke zničení 4 kusů a delší dodací lhůtě byla nakonec zvolena varianta spínání PMOS tranzistorem.

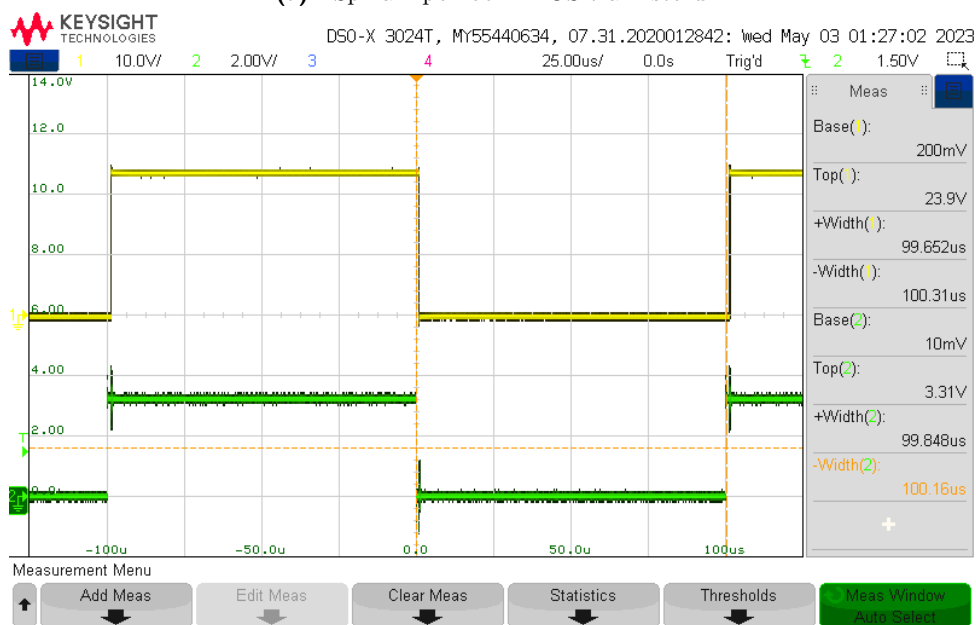
Tento integrovaný obvod je velmi schopen velmi rychle přepínat mezi napájecím napětím a zemí (stovky nanosekund[25]), bohužel je obvod velmi citlivý na elektrostatické výboje. V praxi se tedy ukázalo toto řešení bez použití externích ochran jako problematické. Z tohoto důvodu bylo zvoleno robustnější řešení se spínáním pomocí PMOS tranzistoru.

7.1.3 Analogový zesilovač

Na operační zesilovač zesilující analogový výstup MCU jsou kladeny následující požadavky:



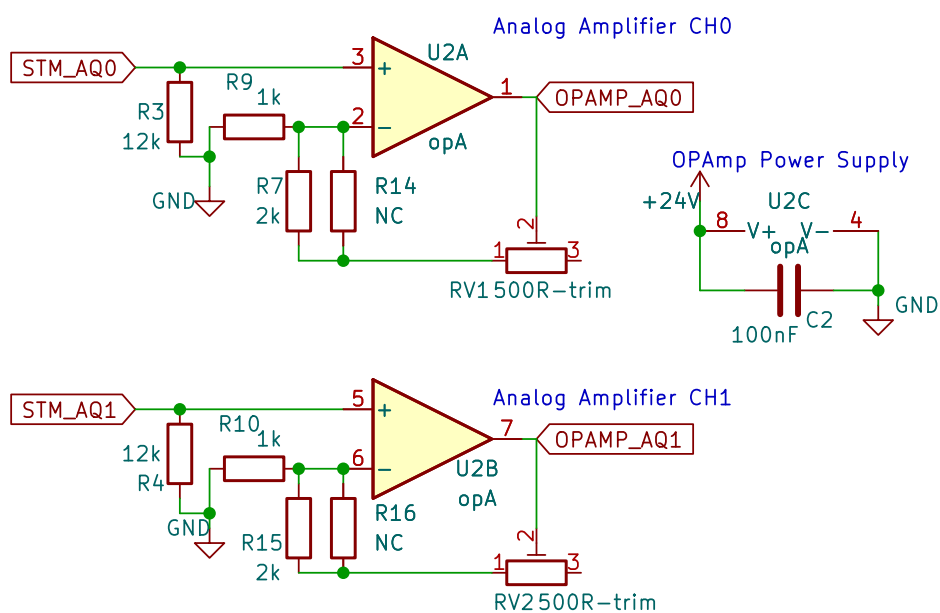
(a) : Spínání pomocí PMOS tranzistoru



(b) : Spínání s pomocí integrovaného obvodu FOD3182

Obrázek 7.4: Porovnání spínání digitálního výstupu pomocí dvou vyzkoušených metod.

- Maximální napájecí napětí > 30 V
- Výstupní napětí > 10 V
- Schopnost přiblížit se zemnicímu napětí (rail-to-rail)
- Rychlost přeběhu < 5 μs (10 μs doba pulzu)
- 2 kanály



Obrázek 7.5: Schéma zapojení operačního zesilovače.

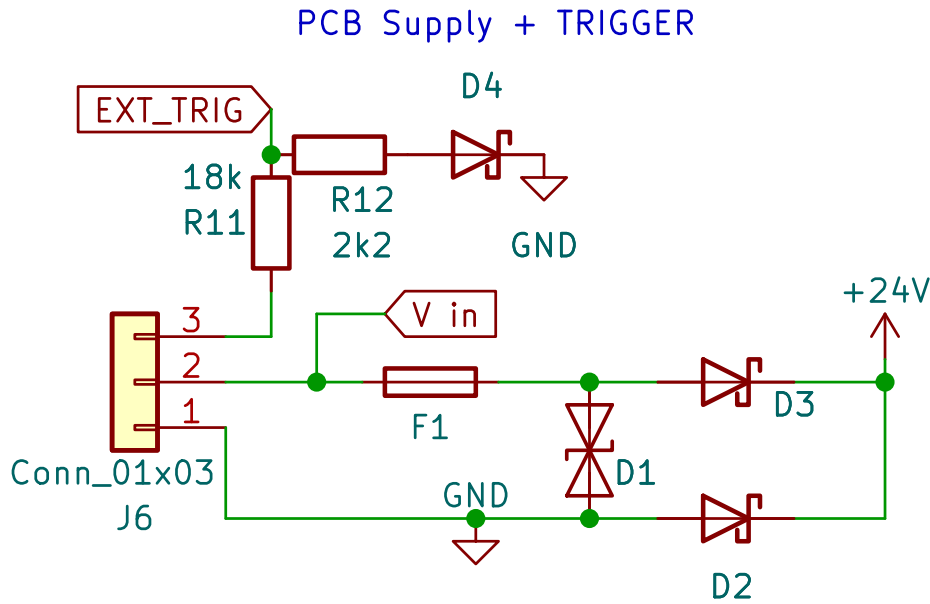
Pro zesílení analogových výstupů bylo požadavkem zvolit takový zesilovač, který může být napájen pomocí napájecího napětí použité desky plošných spojů, tedy 24-30 V, přičemž bude schopen dosáhnout co nejbližší zemnicímu napětí.

Operačním zesilovačem, který splňuje dané požadavky je OPA2172ID od společnosti Texas Instruments. Tento operační zesilovač má vysokou frekvenci 10 MHz a rychlost přeběhu (slew rate) $10\text{V}/\mu\text{s}$, čímž bude zajištěna dostatečná rychlost nastavení napěťové hladiny. U samotného výstupu DAC převodníku MCU je čas ustálení výstupu s přesností 1 LSB typicky $1,6\mu\text{s}$ ([13] strana 133). Zapojení operačního zesilovače je neinvertující a ve zpětné vazbě byl zapojen potenciometr z důvodu možnosti kalibrace výstupního napětí.

7.1.4 Ochrany napájení obvodu

Deska obsahuje v sérii za napájecím konektorem připojenou vratnou pojistku PPTC (F1), která při proudu větším než 1,1 A zvýší svůj odpor, čímž omezí proud protékající obvodem. Během této chvíle dojde k vypnutí jednodeskového počítače Raspberry Pi a MCU, čímž se resetují výstupní hodnoty generátoru.

Generátor je dále chráněn proti přepětí a to TVS diodou D1 připojenou mezi konektor GND a 24 V za PPTC pojistkou, tedy paralelně napájenému zařízení.



Obrázek 7.6: Schéma zapojení ochran napájení desky a externí spouště generátoru.

Při provozním napětí se TVS dioda chová jako otevřený spínač. Je-li ale na TVS diodě větší napětí než specifikované průrazné napětí, dochází k lavinovému průrazu a TVS diodou začne procházet proud[26], který je následně omezen pojistkou F1, která rozpojí daný obvod a tedy nedojde k přepětí. PMOS tranzistory a operační zesilovače zvládnou napájecí napětí 60 a 40 V, nicméně spínaný regulátor napětí je dimenzován maximálně na 32 V, proto je třeba zvolit TVS diodu, která zabrání napětí vyššímu než zmíněných 32 V. Požadavky na diodu splňuje SMBJ24CA od Littelfuse, jejíž průrazné napětí je 26.7 až 29.5 V[27].

U napájení desky plošných spojů se dále nacházejí celkem 3 Schottkyho diody. Dioda D4 zamezuje připojení signálu opačné polaroty na vstup MCU, který je pojmenován na schématu jako EXT_TRIG. Dělič napětí sníží vstupních 24V na napětí:

$$U_{MCU} = U_{PLC} \cdot \frac{R_{12}}{R_{11} + R_{12}} \quad (7.7)$$

$$= 24V \cdot \frac{2,2k\Omega}{18k\Omega + 2,2k\Omega} \quad (7.8)$$

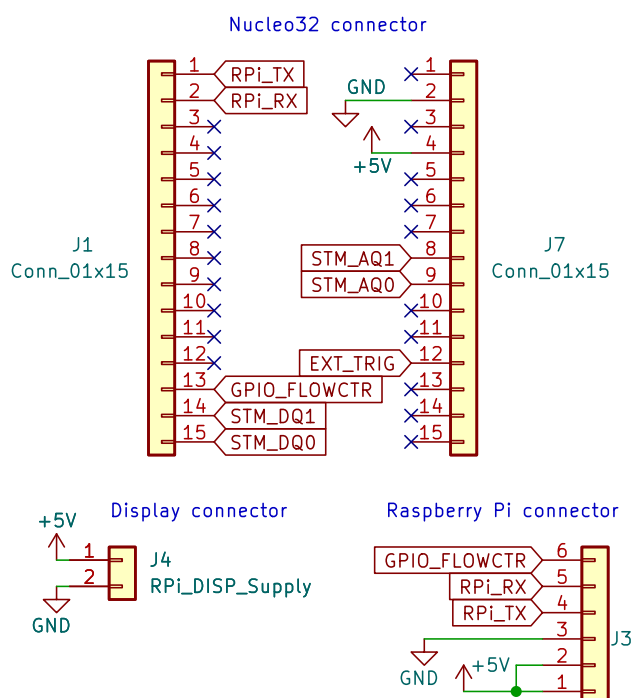
$$\approx 2,6V \quad (7.9)$$

Dioda D3 chrání obvod proti připojení opačné polaroty na napájecí svorky. V závěrném směru diody nicméně stále může protékat proud 1 mA[28]. Pokud by obvod měl velmi vysoký odpor, mohlo by být stále v obvodu dost vysoké napětí na poškození obvodu. Z tohoto důvodu se zde nachází druhá dioda. Ta v případě opačné polaroty dovolí proudu protéci skrz tuto diodu a v obvodu

se tedy nebude nacházet vyšší napětí než to, které má dioda v propustném směru, tedy zhruba 0,3 V.

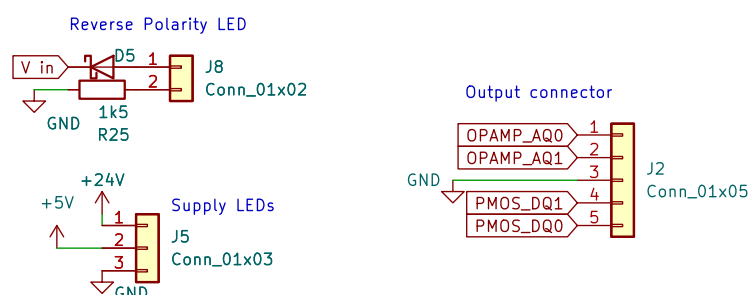
7.1.5 Konektory DPS

Poslední částí schématu jsou konektory, do kterých se připojí Raspberry Pi, deska Nucleo32, displej a výstupní konektory krabičky



Obrázek 7.7: Schéma zapojení konektorů displeje, Raspberry Pi a desky Nucleo32.

Ve schématu se nachází také příprava na LED signalizaci stavu generátoru. Připravena je signalizace o přítomnosti napájení 24V a 5V a dále se zde nachází LED signalizující připojení opačné polaroty napájení. Při připojení opačné polaroty bude diodou v propustném směru protékat proud dioda bude svítit. Při správném zapojení bude LED vyšší napětí na závěrné straně diody a LED bude zhasnutá. Bohužel tento obvod nestihl být zahrnut v poslední verzi desky plošných spojů.



Obrázek 7.8: Schéma zapojení výstupních konektorů a konektorů pro LED signalizaci stavu generátoru.

7.2 Rozložení komponent na DPS

Rozložení komponent na desce můžeme vidět na obrázku 7.9, přičemž byla zvolena dvouvrstvá deska plošných spojů.

7.3 Krabíčka

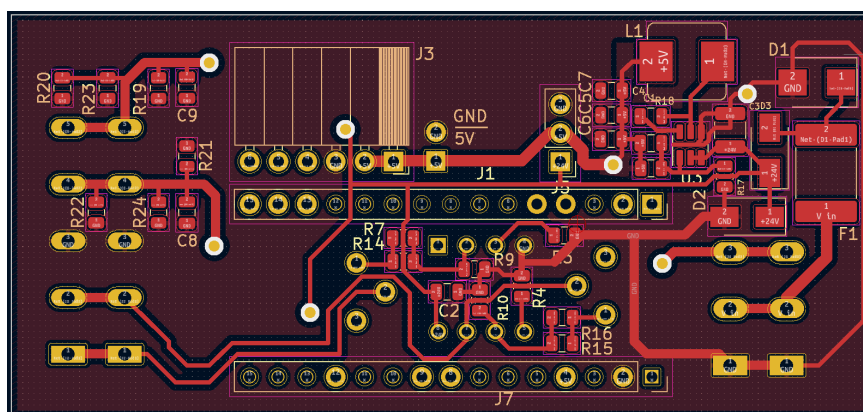
Pro generátor byl vytvořen 3D model krabíčky, do které uživatel připojí pomocí 4 šroubů displej. K displeji je připojeno Raspberry Pi a pomocí navržené desky plošných spojů také STM32. Všechny tyto části jsou připevněny k čelnímu dílu krabíčky. Dále se k do tohoto dílu připevní 13 banánových konektorů a konektor pro napájení generátoru. Je připravena také zdířka pro Ethernetový konektor, ta však nyní bohužel není použita, protože její rozměry nesouhlasí s rozměry pořízeného konektoru. Touto zdířkou je tedy pouze protažen ethernetový kabel do Raspberry Pi.

Na krabíčce se nalézá 8 konektorů pro digitální výstup, neboť je plánováno zvýšit počet digitálních kanálů generátoru na 8 ze současných 2.

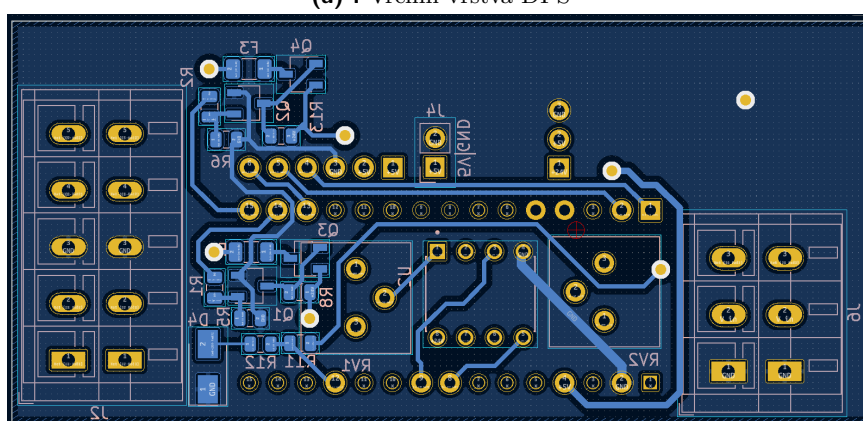
Krabíčka je tvořena ze 3 dílů tisknutých na 3D tiskárně Prusa i3 MK3S+ materiálem PETG.

7.3.1 Čelní strana s displejem

Tato část byla založena na modelu převzatého z [29], čímž se eliminovala nutnost měření rozměrů displeje a rozložení šroubů pomocí kterých se displej



(a) : Vrchní vrstva DPS



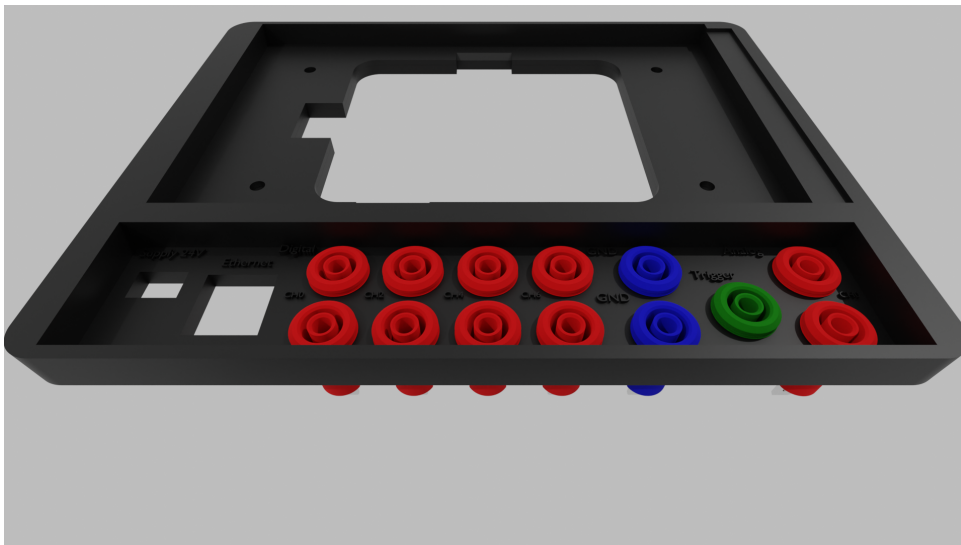
(b) : Spodní vrstva DPS

Obrázek 7.9: Návrh DPS desky

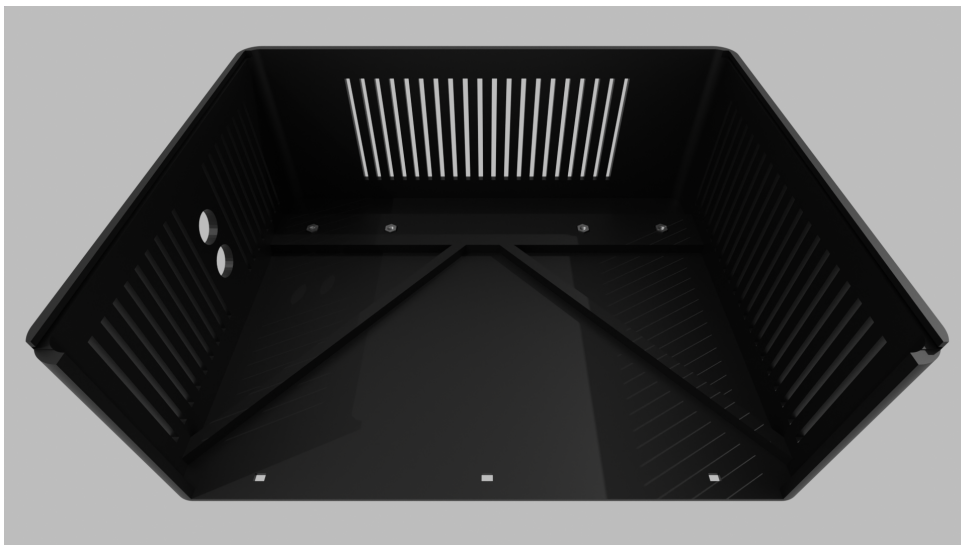
přípevní. Převzatý model byl navržen pouze pro připevnění displeje, z tohoto důvodu byl dále zvětšen, aby mohly být přidány konektory. Dále byly přidány výřezy pro snazší manipulaci s Raspberry Pi a byla zvýšena tloušťka podstavu. Čelní strana byla tisknuta s podporami, neboť spodní část dílu je tvořena výřezem, který pasuje do rámu krabičky. Do čelní strany se pomocí 4 šroubů připevní displej spolu s Raspberry Pi, navrženou deskou plošných spojů a MCU. Dále se zde připevní banánové konektory, napájení a vyvede se ethernetový kabel.

7.3.2 Hlavní rám

Hlavní rám byl tisknut bez podpor, ačkoli obsahuje kolejnice do kterých pasuje čelní strana. Do vrchních kolejnic se vsune displej s již osazeným displejem, Raspberry Pi, deskou plošných spojů a MCU. Hlavní rám (7.3.2) krabičky obsahuje uvnitř místa na 4 matice kterými se přichytí pomocí šroubu velikosti



Obrázek 7.10: Schéma zapojení výstupních konektorů a konektorů pro LED signalizaci stavu generátoru.

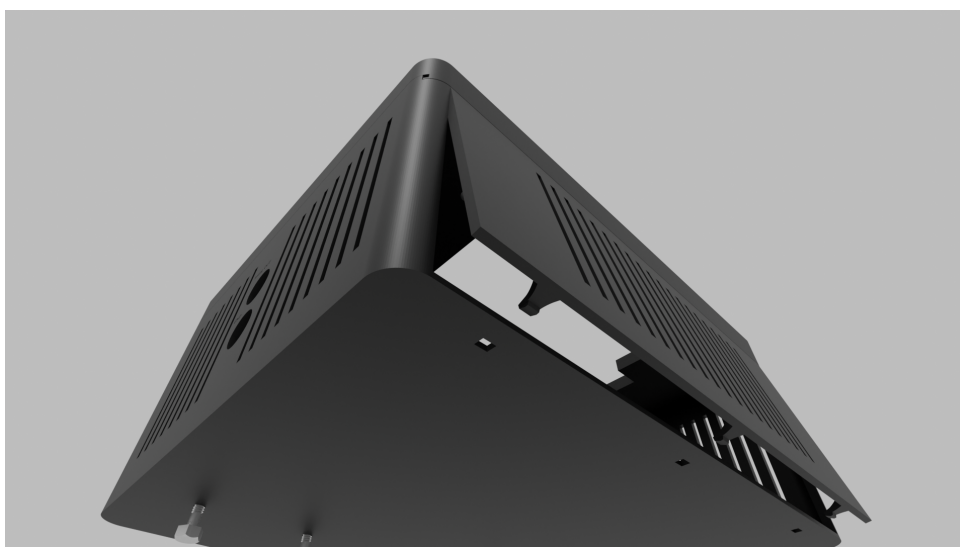


Obrázek 7.11: Schéma zapojení výstupních konektorů a konektorů pro LED signalizaci stavu generátoru.

M3 držáky, pomocí kterých lze zavěsit zařízení na lištu DIN. Zvolené byly držáky Bopla TSH 35[30].



Obrázek 7.12: Krabička po zasunutí paciček, pohled shora.



Obrázek 7.13: Krabička před zaháknutím háčků, pohled zdola.

■ 7.3.3 Dvířka krabičky

Posledním dílem jsou dvířka, která se po zavěšení na DIN lištu nachází v horní části krabičky. Tento díl byl tisknut naležato s podporami. Tento díl je nejprve třemi pacičkami zasunut do čelní strany a následně třemi háčky na opačné straně zaháknut do rámu krabičky.

7.3.4 Zvýšení pevnosti krabíčky

Krabíčka je navržena tak, aby díly pasovaly do sebe a držely celou krabíčku pohromadě bez dalšího spojovacího materiálu. Z čelní strany nicméně přechází displej. Při pádu čelní strany by se tedy displej snadno mohl zničit, proto byla z důvodu zabezpečení displeje, pomocí dvou distančních sloupků a dvou šroubů, čelní strana připevněna k hlavnímu rámu.



Obrázek 7.14: Vnitřek krabíčky včetně distančních sloupků a šroubů pro zpevnění.

7.3.5 Zavěšení na lištu DIN



Obrázek 7.15: Generátor zavěšený na DIN liště a připojený k testovanému zařízení.

Kapitola 8

Měření signálů generátoru

8.1 Analogové výstupy

Na obrázku 8.1 je zobrazen průběh signálu obou výstupních analogových kanálů s následujícími parametry:

- Perioda 1 = 1,6 ms
- Perioda 2 = 1,6 ms
- Výstupní napětí = 10 V
- Diskrepance kanálů = 0,4 ms

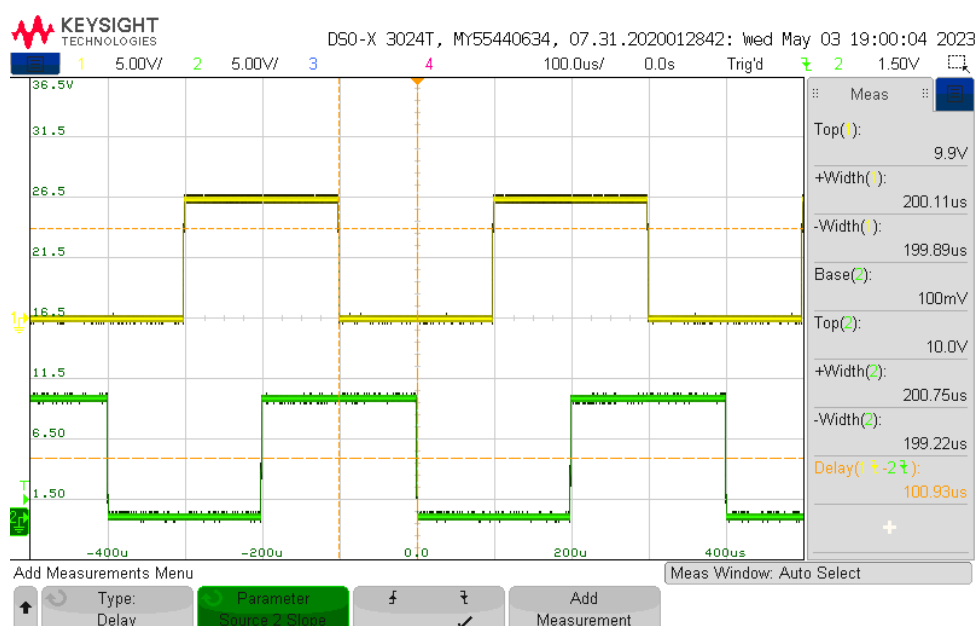
Parametry měření 8.2

- Perioda 1 = 200 μs
- Perioda 2 = 200 μs
- Výstupní napětí = 10 V
- Diskrepance kanálů = 100 μs

8. Měření signálů generátoru



Obrázek 8.1: Analogové výstupy, primární a sekundární se zpožděním, časové základy používané pro testování.



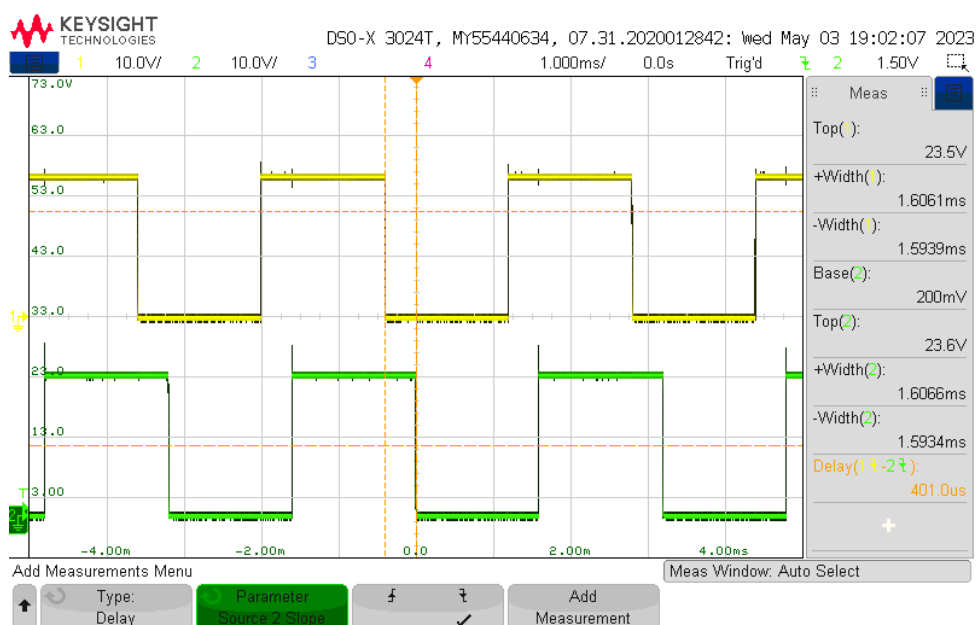
Obrázek 8.2: Analogové výstupy, primární a sekundární, rychlé změny napěťových hladin..

8.2 Digitální výstupy

Jak můžeme vidět na obrázcích 8.3 a 8.4, výstupní napětí generátoru není napájecích 24 V. Důvodem je Schottkyho dioda, použitá z důvodu ochrany obvodu proti připojení opačné polarity napájení 7.1.4.

Parametry měření 8.3

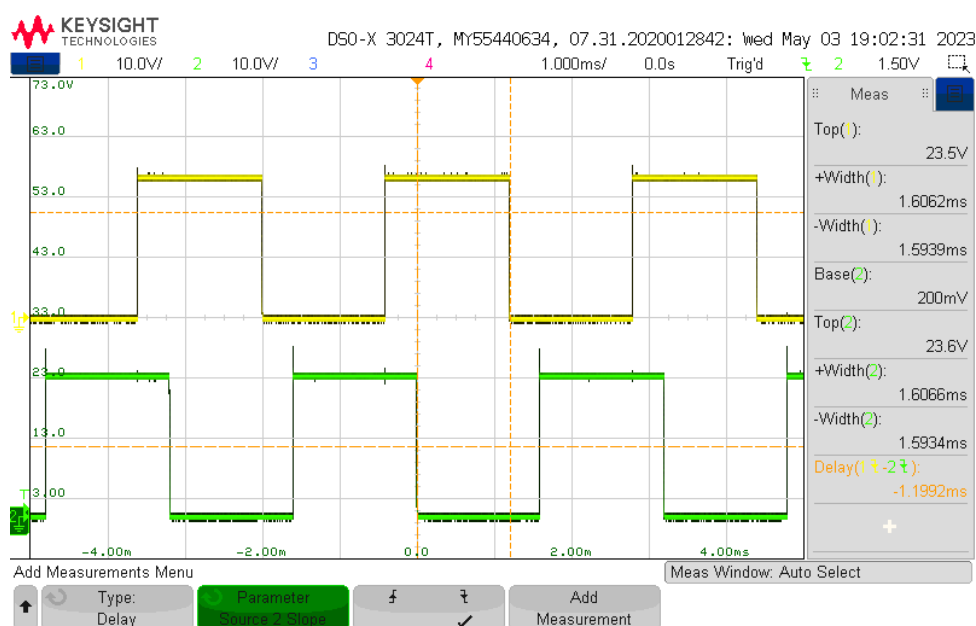
- Perioda 1 = 1,6 ms
- Perioda 2 = 1,6 ms
- Diskrepance kanálů = 0,4 ms
- Ekvivalence výstupů



Obrázek 8.3: Digitální výstupy, primární a sekundární, se zpožděním, ekvivalentní.

Parametry měření 8.4

- Perioda 1 = 1,6 ms
- Perioda 2 = 1,6 ms
- Diskrepance kanálů = 0,4 ms
- Negování výstupů

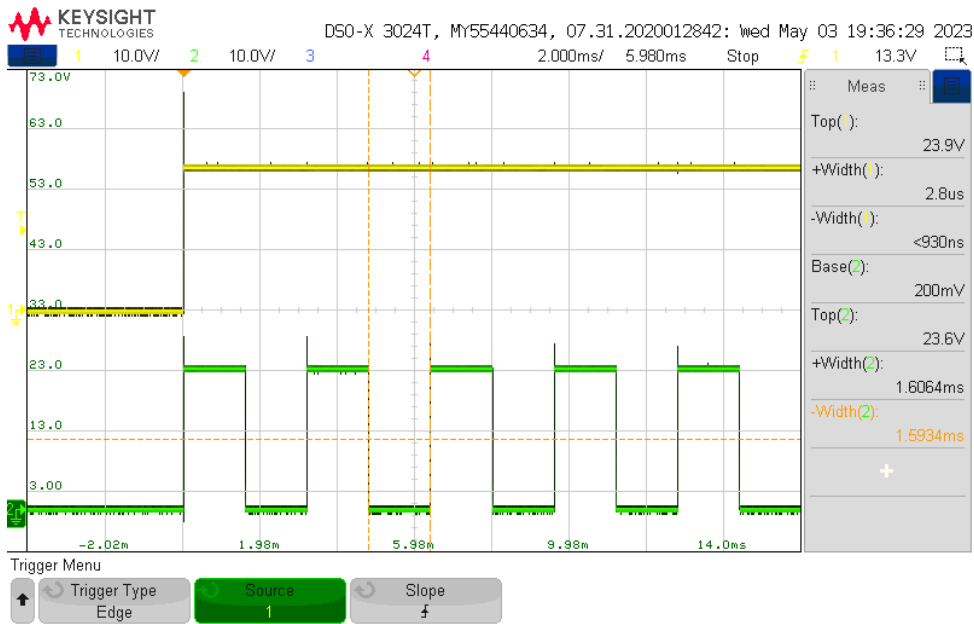


Obrázek 8.4: Digitální výstupy, primární a sekundární se zpožděním, neekvivalentní.

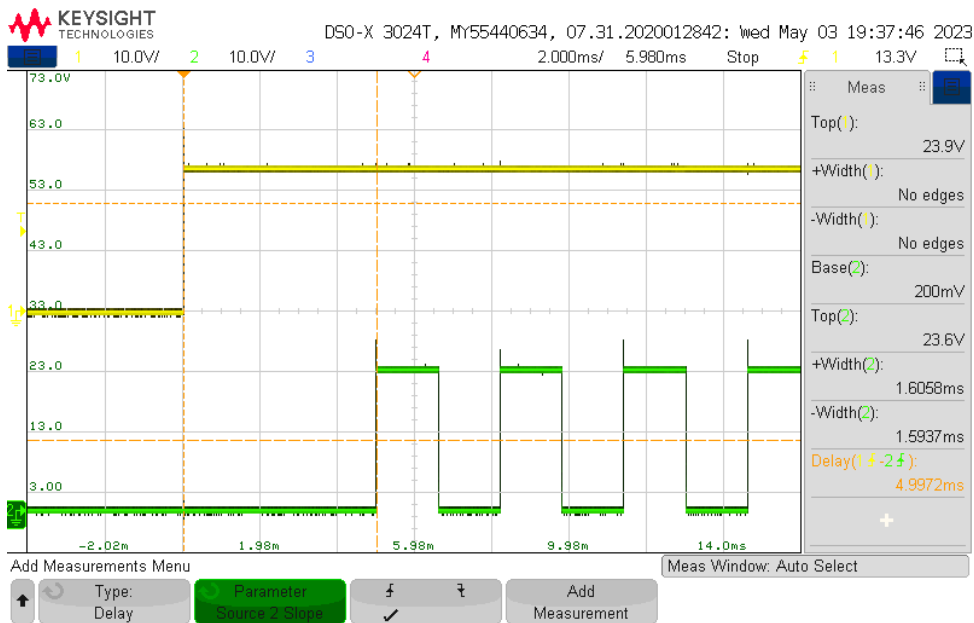
8.3 Externí spuštění generátoru

Generátor může být spuštěn pomocí náběžné hrany vstupního konektoru. Na obrázku 8.5 můžeme vidět osciloskopické měření vstupního signálu (žlutý kanál CH1) a výstup generátoru (zelený kanál CH2). Obrázek 8.5 ukazuje spuštění ihned po detekci náběžné hrany.

V případě spuštění externím signálem může být navoleno zpoždění startu generátoru. Na obrázku 8.6 můžeme vidět stejné výstupní pulzy pouze s nakonfigurovaným zpožděním 5 ms.



Obrázek 8.5: Digitální výstup primárního kanálu, externí spuštění náběžnou hranou.



Obrázek 8.6: Digitální výstup primárního kanálu, externí spuštění náběžnou hranou se zpožděním.

Kapitola 9

Závěr

V této práci byla nejprve rozmyšlena architektura vyvíjeného produktu. Návrh s jednodeskovým počítačem a mikrokontrolerem se osvědčil a stal se proto také finálním řešením. Možnost konfigurovat generátor skrze používané PLC byla úspěšně implementována pomocí knihovny Snap7. Nad rámec této možnosti konfigurace bylo implementováno také uživatelské rozhraní pomocí 7 palcového dotykového displeje. Poslední možností konfigurace generátoru se stala možnost vytvořit si soubor XML, který obsahuje požadovanou konfiguraci generátoru. Tento soubor, který uživatel uloží do adresáře projektu generátoru, poté může být načten a dovoluje uživateli realizovat složitější průběhy signálu. Generátor je schopen generovat signály v násobcích desítek μs , přičemž při testování jsou časové základny měněny po $50 \mu s$. Rychlost zařízení tedy splňuje zadané požadavky.

Pro generátor byla navržena a vyrobena deska plošných spojů zesilující výstupní signály generátoru. Na desce se nachází také napěťový regulátor, který umožňuje celé zařízení napájet z 24 V zdroje, který je běžně používán na testovaných zařízeních. Všechny části generátoru se nachází v navržené krabičce, která byla vtištěná na 3D tiskárně.

V současné době jsou již prováděny práce na nové verzi, která upraví nedostatky zjištěné na současné verzi a mimo jiné přidá další 3 páry digitálních kanálů.

Hlavním nedostatkem současné verze je použití vlastního zapojení digitálního spínače pomocí PMOS tranzistoru. Zapojení není schopno spínat obvod k zemi, tedy trvá než je náboj disipován skrze 3 připojené rezistory. Tento obvod bude nahrazen integrovaným obvodem s dvojčinným výstupem, který je schopen rychle přepínat mezi oběma stavy výstupu. Vyvinuté zařízení je nicméně funkční a aktuálně se používá pro testování fail-safe vstupně-výstupních modulů. Rozšířením oproti původně využívanému generátoru funkcí je přítomnost druhého synchronizovaného kanálu, díky kterému je

možné testovat reakci zařízení na posunuté signály kanálového páru. Možnost využití externího spuštění při detekci náběžné hrany také umožnila automatizovat jednu z testovaných kategorií. Analogový výstup je využíván především pro testování reakce vstupně-výstupního modulu na změny napájecího napětí. Pro tyto testy je využit externí zesilovač, který je konfigurován právě pomocí napětí 0 - 10 V. Tyto testy mohly být generátorem automatizovány právě s pomocí XML konfigurace. Takto byly nahrazeny aktuálně 2 kategorie testů, které dříve byly prováděny manuálně.

Kapitola 10

Slovník pojmů

- Sekvence - Označuje souhrn jednotlivých pulzů
- Konfigurace Sekvence - Proměnné definující chování sekvence, mimo její pulzy
- Pulz - Atomický objekt konfigurovaný délkou trvání a napětovou hladinou
- Kanál -> Výstup generátoru
 - analogový - nastavitelná délka pulzu a napětová hladina.
 - digitální - nastavitelná délka pulzu a jednobitový výstup.
- PLC (Programmable logic controller) - Programovatelný logický automat
- IM (Interface module) - zařízení zprostředkávající komunikaci mezi vstupně-výstupními moduly a PLC
- MCU - mikrokontroler
- DPS - deska plošných spojů
- XML (Extensible Markup Language) - značkovací jazyk
- DUT (Device under test) - testované zařízení



Literatura

- [1] Siemens AG, “Et 200sp fehlersichere digitaleingabe f-di.” [Online]. Dostupné z: https://www.parmley-graham.co.uk/image/cache/data/products/P_ST70_XX_07162P-500x500-watma.png
- [2] G. Halfacree, “Raspberry pi 3 b+,” Bradford, UK. [Online]. Dostupné z: https://commons.wikimedia.org/wiki/File:Raspberry_Pi_3_B%2B_%2839906369025%29.png
- [3] R. Ltd, “Raspberry pi touch display.” [Online]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-touch-display/>
- [4] STMicroelectronics N.V., “Stm32 32-bit arm cortex mcus.” [Online]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>
- [5] S. N.V., “Stm32 nucleo boards.” [Online]. Dostupné z: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>
- [6] STMicroelectronics N.V., *STM32G4 - DAC, Revision 1.0*. [Online]. Dostupné z: https://www.st.com/resource/en/product_training/STM32G4-Analog-DAC.pdf
- [7] Wireshark Wiki, “Elements of a uart frame including optional bits,” Květen 2017. [Online]. Dostupné z: https://commons.wikimedia.org/wiki/File:UART_Frame.svg
- [8] OPC Router, “What is opc ua? a practical introduction.” [Online]. Dostupné z: <https://www.opc-router.com/what-is-opc-ua/>
- [9] D. Nardella, “Snap7 overview.” [Online]. Dostupné z: <https://snap7.sourceforge.net/>

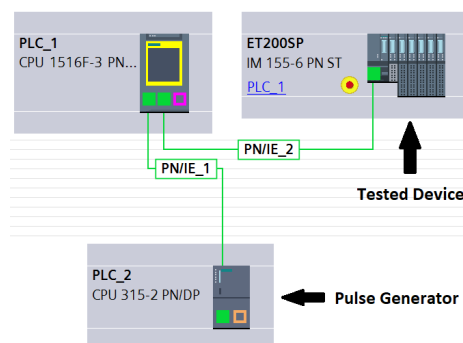
- [24] Fairchild Semiconductor International, *FDN5618P*, Zář 2000. [Online]. Dostupné z: <https://www.farnell.com/datasheets/2303914.pdf>
- [25] ON Semiconductor Corporation, *3 A Output Current, High Speed MOSFET Gate Driver Optocoupler*, Zář 2021. [Online]. Dostupné z: <https://www.onsemi.com/pdf/datasheet/fod3182-d.pdf>
- [26] Components101, “How to use tvs diodes for transient voltage suppression.” [Online]. Dostupné z: <https://components101.com/articles/how-to-use-tvs-diodes-for-transient-voltage-suppression>
- [27] Littelfuse Inc, “Smbj series.” [Online]. Dostupné z: https://www.littelfuse.com/~media/electronics/datasheets/tvs_diodes/littelfuse_tvs_diode_smbj_datasheet.pdf.pdf
- [28] multcomp PRO, “Schottky barrier rectifier.” [Online]. Dostupné z: <https://www.farnell.com/datasheets/2861410.pdf>
- [29] L. Eeckelaert, “Raspberry pi 7 inch touchscreen display case.” [Online]. Dostupné z: <https://www.thingiverse.com/thing:1585924>
- [30] Bopla Gehäuse Systeme GmbH, “Kabelverschraubungen / allgemeines zubehör | universeller tragschienenhalter tsh 35, polyamid.” [Online]. Dostupné z: <https://www.bopla.de/gehaeusetechnik/kabelverschraubungen-allgemeines-zubehoer/hutschienenadapter/universeller-tragschienenhalter-tsh-35-polyamid/tsh-35>

Příloha A

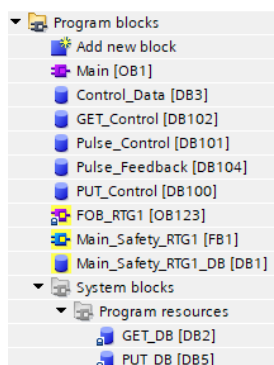
Konfigurace TIA Portal

A.1 Nastavení komunikace v programu TIA Portal

Pro naprogramování programu pro PLC bylo využíváno jazyka SCL. Nastavení komunikace mezi PLC a generátorem probíhá v prostředí TIA Portal V18. Nejprve si uživatel navolí zvolenou konfiguraci hardwaru, jako je CPU, IM a vstupně-výstupní moduly. CPU použité dále má označení CPU 1516F-3 PN/DP. Nyní nastavíme IP adresu zařízení na 192.168.2.127 a masku sítě na 255.255.255.0. Následně otevřeme PLC a nastavíme jeho IP adresu v síti, pomocí které bude spojen s generátorem. Adresu PLC volíme tak aby měla stejnou síťovou adresu, tedy 192.168.2.XXX, kde znaky "XXX", lze nahradit dle potřeby číslem (1-254), kromě 127 (IP adresa generátoru). Maska bude nastavena opět na 255.255.255.0.



Obrázek A.1: Síťové zapojení zařízení k PLC a testovanému modulu.



Obrázek A.2: Použité datové a systémové bloky pro komunikaci mezi PLC a Generátorem.

Pro posílání je nutné v nastavení CPU v záložce "Protection" vypnout zabezpečení CPU a potvrdit možnost použití "PUT/GET" bloku. Komunikace směrem z PLC do generátoru je realizována pomocí instrukce S7 PUT. Ta je schopna posílat data mezi 2 propojenými PLC. CPU odesílající data je označováno jako "Local" a přijímající kontroler jako "Remote". Knihovna, zajišťující komunikaci na straně Raspberry Pi, vytvoří virtuální CPU, které není pro ostatní prvky v síti rozpoznatelné od skutečného CPU. Virtualizovaným kontrolerem je CPU 315-2 PN/DP. Vybereme tedy také toto zařízení, konkrétně verzi V3.2.18. Po výběru zařízení musí být v "Network View" generátor připojen k Local CPU. IP adresu nového CPU, která musí být 192.168.2.127 (výběr zařízení dle instrukcí knihovny [9], část "Snap7 Communication - Snap7 Server").

Pro zřízení komunikace je nutné vytvořit další 2 datové bloky. Blok, který bude obsahovat přenášená data, ten v práci bude dále pojmenován "Pulse_Control" a jeho číslo je 101, ve vlastnostech bloku je třeba zrušit "Attributes->Optimized block access", aby se data v paměti automaticky nepřeskládávala a jejich adresa zůstala konstantní. Datový blok, do kterého bude instrukce PUT zaznamenávat svůj aktuální stav a pomocí kterého bude započato přenášení dat, pojmenujeme "PUT_Control" s číslem 100.

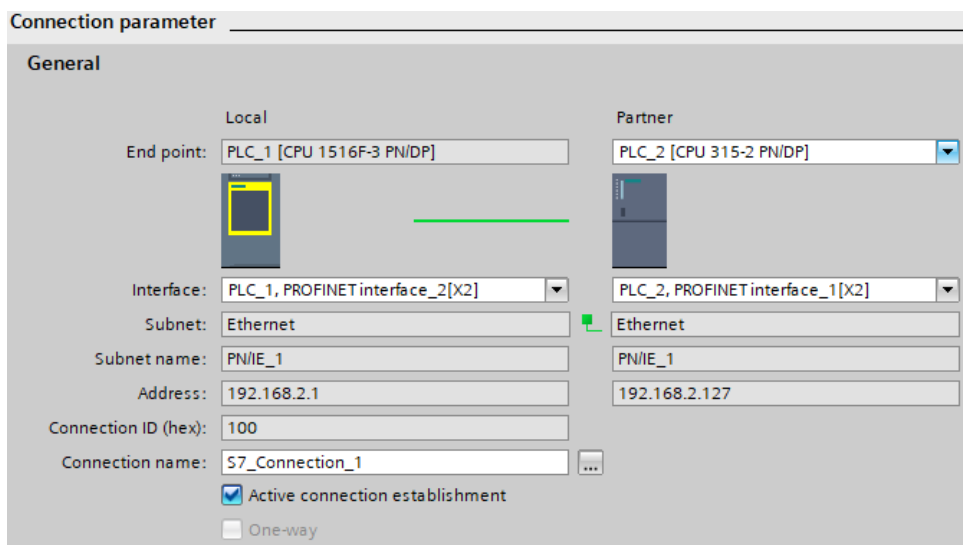
V seznamu instrukcí si nyní nalezneme výše zmíněnou instrukci S7 PUT (Instructions ->Communication->S7 Communication->PUT). Při přetažení této instrukci do cyklu OB1 se otevře vyskakovací okno, ve kterém zvolíme jméno datového bloku přiřazeného instrukci. V této práci je název tohoto bloku zachován, tedy "PUT_DB" s číslem 5.

Funkce PUT zašle data virtuálnímu CPU vždy při detekci náběžné hrany signálu START. Po zaslání dat druhému kontroleru se nastaví bit DONE na logickou úroveň 1. V případě, že byla data odeslána, opět přerušíme zaslání dat pomocí nastavení bitu START na úroveň 0.

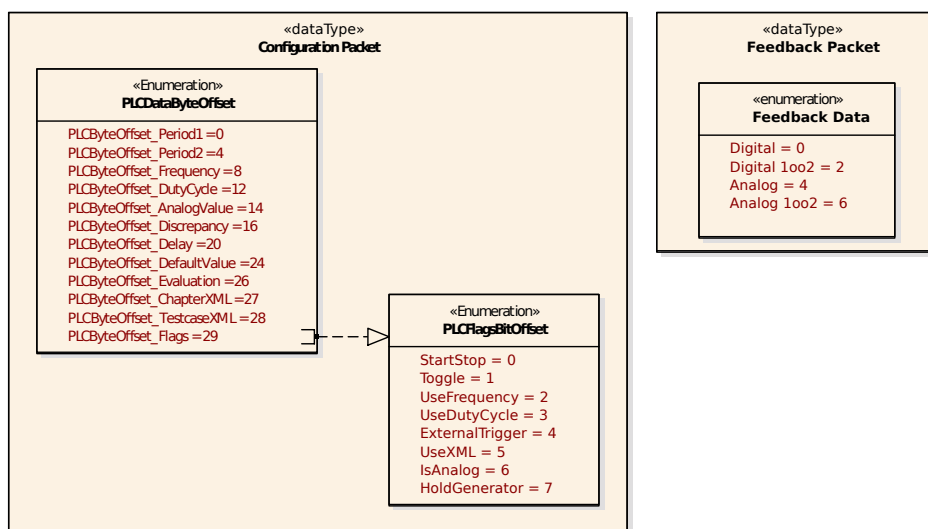
Kliknutím na instrukci v cyklu OB1 otevřeme konfiguraci komunikace, zde vybereme druhé nakonfigurované CPU a zaškrtneme "Active connection establishment", aby navazoval s generátorem spojení PLC kontroler. Pod záložkou Block Parameter nastavíme do "Inputs" a "Outputs" proměnné z bloku "PUT_Control" a do "InOut" nastavíme absolutní adresy datového bloku "Pulse_Control" do kolonky posílaných dat a počátek DB1 do kolonky přijímaných dat. Správná konfigurace je vidět na obrázku A.3.

Stejný proces zopakujeme také s instrukcí S7 GET, kde číslo datového bloku nastavíme na 2.

Bloky "Pulse_Control" a "Pulse_Feedback" musí obsahovat strukturu dat podle obrázku A.4. Názvy proměnných jsou libovolné, ale jejich pořadí a da-



Obrázek A.3: Konfigurace komunikačních bloků S7 PUT a S7 GET.



Obrázek A.4: Struktura paketů zasílaných mezi PLC a Raspberry Pi

tové typy musí zůstat zachovány.

■ A.2 Volání komunikačních funkcí v cyklu OB1

Konfigurace generátoru se odešle vždy při detekci náběžné hrany bitu "PUT_Control.Start", tento signál nastavíme na logickou úroveň 1 v případě, že uživatel nastaví bit "Control_Data.UpdateGenerator" a "PUT_Control.Start" na úroveň 0.

Dále pokud je nastaven požadavek na vypnutí generátoru, nastavíme proměnnou "Pulse_Control.HoldGenerator" := 0, abychom přerušili přidržení aktuálně běžících sekvencí generátoru.

Po zavolání bloku PUT zkontrolujeme, zdali nebyl funkcí nastaven bit "PUT_Control.Done". Pokud byl nastaven na úroveň 1, přerušíme požadavek na zaslání dat ("PUT_Control.Start" := 0 a "Control_Data.UpdateGenerator" := 0).

Nakonec po zavolání funkce S7_GET nastavíme bit "GET_Control.REQ" na logickou úroveň 0 v případě, že příjem proběhl v pořádku, tedy pokud bit "GET_Control.NDR" je roven 1 a "GET_Control.ERROR" je roven 0.