



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**Návrh a vytvoření diagnostické obrazovky pro
řídící systém Heidenhain**

Design and creation of a diagnostic screen for the Heidenhain control system

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michal Navrátil

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. et Ing. Stanislav Lang, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	Bc. Michal Navrátil
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	Ing. et Ing. Stanislav Lang, Ph.D.
Akademický rok:	2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh a vytvoření diagnostické obrazovky pro řídicí systém Heidenhain

Stručná charakteristika problematiky úkolu:

Práce je zaměřena na možnosti diagnostiky technologických procesů spravovaných řídicími systémy Heidenhain. Pro diagnózu PLC jsou k dispozici výrobcem dodávané nástroje, avšak při tvorbě pokročilých diagnostických obrazovek mohou být s výhodou použita i vlastní řešení postavená na komunikaci se zařízením prostřednictvím některého z nabízených protokolů. Tvorbě vlastního diagnostického rozhraní se bude student v rámci práce věnovat.

Cíle diplomové práce:

Proveďte stručnou rešerši v oblasti programovatelných řídicích systémů.

Nastudujte práci s řídicími systémy Heidenhain, shrňte základní poznatky.

Prozkoumejte a popište dostupná uživatelská komunikační rozhraní, diagnostické, servisní a podpůrné nástroje.

Navrhněte vlastní diagnostický nástroj umožňující vyčítání hodnot proměnných z PLC, uvažujte i měření rychle se měnících signálů. Neopomeňte rozměr uživatelské přívětivosti.

Implementujte navržený nástroj (včetně základního popisu jednotlivých kroků implementace).

Zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

FINLAY, John. PyGTK 2.0 Reference Manual [online]. Version 2.5.2. 2005 [cit. 2021-10-15].
Dostupné z:
<https://wiki.python.org.br/DocumentacaoPython?action=AttachFile&do=get&target=ref.pdf>

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Předmětem diplomové práce je návrh a realizace sady obrazovek k oživovacím, diagnostickým a servisním účelům. Obrazovky budou vytvářeny pro CNC systémy Heidenhain a měly by poskytovat základní informace o stavu stroje a jeho částech. Cílem je usnadnit a urychlit práci techniků při stavbě, seřizování a servisních zásazích.

ABSTRACT

The subject of the thesis is the design and implementation of a set of screens for adjustment, diagnostic and service purposes. The screens will be created for Heidenhain CNC systems and should provide basic information about the state of the machine and its parts. The aim is to make the work of technicians easier and faster during build, set-up and service interventions.

KLÍČOVÁ SLOVA

Heidenhain, TNC, diagnostické obrazovky, servisní obrazovky, Python, gtk

KEYWORDS

Heidenhain, TNC, diagnostic screens, service screens, Python, gtk



ÚSTAV AUTOMATIZACE
A INFORMATIKY



2022

BIBLIOGRAFICKÁ CITACE

NAVRÁTIL, Michal. *Návrh a vytvoření diagnostické obrazovky pro řídicí systém Heidenhain*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky, 2022, Diplomová práce. Vedoucí práce: Ing. et Ing. Stanislav Lang, Ph.D.

PODĚKOVÁNÍ

Rád bych zde zmínil vedoucího této práce Ing. et Ing. Stanislava Langa, Ph.D, který po celý průběh tvorby této práce poskytoval cenné rady, připomínky a výborné návrhy na zlepšení.

Další, které bych rád zmínil jsou Ing. Vlastimil Bravenec a Ing. Dalibor Odstrčilík, kteří mi poskytli podklady, vhodné návrhy a dokázali poradit, čímž mi umožnili se danou problematikou zabývat.

Děkuji.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 20. 5. 2022

.....

Bc. Michal Navrátil

OBSAH

1	ÚVOD.....	15
2	PROGRAMOVATELNÉ ŘÍDÍCÍ SYSTÉMY	17
2.1	Řídicí systémy firmy Siemens	17
2.2	Řídicí systémy firmy Heidenhain	21
3	PROGRAMOVÁNÍ PLC	27
3.1	Instruction list	27
3.2	Structured text.....	29
3.3	Function Block Diagram	30
3.4	Ladder diagram.....	32
3.5	Programovací software pro HEIDENHAIN TNC 640.....	33
4	VLASTNÍ ŘEŠENÍ.....	37
4.1	Programové prostředky.....	38
4.2	Řídicí operační systém reálného času HEROS 5 a Python	39
4.3	Aktuální stav diagnostiky a první zadání	40
4.4	Koncept a návrh.....	42
4.5	Vytvoření vlastní obrazovky	43
4.6	Tvorba hlavní diagnostické obrazovky.....	48
4.7	Vznik dalších obrazovek a rozšíření.....	57
4.8	Návrhy pro budoucí obrazovky	60
5	ZÁVĚR	63
6	SEZNAM POUŽITÉ LITERATURY.....	65
7	SEZNAM OBRÁZKŮ	67
8	SEZNAM TABULEK.....	69
9	SEZNAM PŘÍLOH.....	71

1 ÚVOD

CNC stroje pro obrábění kovových i nekovových materiálů prošly během posledních let mnoha inovacemi. Nepožaduje se od nich, tak jako dříve, jen přesný řízený pohyb os a obrábění. S průmyslem 4.0 do procesů vstupuje digitalizace a další nároky, jako například sledování procesní historie každého výrobku, hlídání taktu linek, funkční bezpečnost nebo digitální ekonomika. Tyto inovace jsou z technického hlediska výhody pro zákazníka, který má podrobné informace a přehled o tom, co se na každém stroji děje a může tak zvyšovat kvalitu, bezpečnost a efektivitu výroby. Na straně výrobce však inovace obnáší znalosti širšího spektra oborů, hlavně informačních technologií. Zvyšují se nároky jak na konstrukci, tak na techniky, kteří stroj uvádí do provozu, nebo provádí následné opravy, což obnáší komplexní znalosti o stroji, umístění a funkci jednotlivých komponent. Důležitý je také čas, který každý technik se strojem stráví.

Řídicí systémy Heidenhain pro zmiňované uvádění do provozu a servis umožňuje zobrazit pro kontrolu vstupů a výstupů pouze tabulky s adresami celého paměťového prostoru, kde lze filtrovat vstupní i výstupní hodnoty. Doposud se v řídicím systému nevyskytuje žádná jiná, příznivější možnost zobrazení. Tato skutečnost je dána tím, že každý výrobce obráběcích center může ke konstrukci a zapojení přistupovat jinak.

Pro případ sériové výroby vertikálních obráběcích center by vznik souboru diagnostických obrazovek mohl ušetřit hodiny práce při uvádění stroje do provozu, diagnostice i servisu.

První část práce bude věnována popisu řídicích systémů pro obráběcí centra firem Siemens a Heidenhain, jelikož s těmito se autor práce setkal. Ze srovnání vyplyne variabilita Siemens i Heidenhain řízení, kde Heidenhain je lehce pozadu v rámci digitalizace, ale disponuje možností programování v dialogovém prostředí Klartex, které je u obsluhy roky oblíbený. Vznikne řešerše běžných programovacích jazyků PLC, kde bude podrobněji popsán jazyk používaný pro PLC Heidenhain spolu se softwarem nutným k jeho programování a ladění.

Druhá část práce se bude zabývat tvorbou a programováním obrazovek pro systémy Heidenhain, kde hlavním předmětem bude diagnostická obrazovka, která by měla nahradit aktuální dostupné tabulky paměťového prostoru PLC, které nejsou příliš přehledné. Navíc vzniknou další obrazovky usnadňující úkony na řízeních TNC.

2 PROGRAMOVATELNÉ ŘÍDÍCÍ SYSTÉMY

Řídicí systém je software, který je obsažený v každém CNC stroji a je jeho nezbytnou součástí. Zajišťuje přesný, synchronizovaný pohyb os, včetně, a vytváří prostředí pro tvorbu programů CNC. Dnešní řídicí systémy s uzavřenou smyčkou jsou velmi přesné, výkonné a jsou schopny kompenzovat odchylky v reálném čase. [1] Řídicí systémy, které se v České republice vyskytují jsou převážně od firem Fanuc, Siemens, Heidenhain a Mitsubishi.

Systém, který na CNC stroji bude použit je volen na základě charakteristiky obrábění, nebo preferenci zákazníka. Pro tří- a víceosé frézování je preferováno řízení Siemens Sinumerik nebo Heidenhain TNC 640. Konkrétní modelové řady se volí podle faktorů, mezi které patří například:

- 1) Počet řízených os
- 2) Počet vřeten
- 3) Počet kanálů obrábění
- 4) Kinematika obrábění
- 5) Velikost panelu
- 6) Programové prostředí

2.1 Řídicí systémy firmy Siemens

Řídicí systém Sinumerik je produktem německé společnosti Siemens AG. Jako největší průmyslová společnost v Evropě působí na trhu průmyslu, energetiky a zdravotnictví. Sinumerik nabízí modelové řady se širokou škálou funkcí a je tak vhodný prakticky pro kterékoliv aplikace frézování, soustružení nebo 5osé obrábění [6]. Rozdíly v modelových řadách jsou často velmi malé a je nutné pečlivě zvážit, které funkce bude koncový uživatel využívat. Jako příklad lze uvést systém Sinumerik 808D, na kterém nelze v základní konfiguraci nalézt funkce jako je kompenzace tření, automatické ladění servopohonů, ani ethernetové rozhraní pro přenos programů a analýzu dat [2].

Tab. 1: Přehled modelových řad Sinumerik [2, 3, 4, 5, 6]

	808	828	840	ONE	MC
Umístění řízení	panel	panel	driver	driver	PC CNC
Počet os/vřeten	5	10 + 2*	93	31	8
OPC UA	-	✓	✓	✓	✓
Obráběcí kanály	1	2	30	10	4
Panel	8.4"	10.4"/15.6"	Do 24"	Do 24"	(DisplayPort)
PLC	S7-200	S7-200	S7-300	S7-1500F	S7-1500F

Sinumerik 808

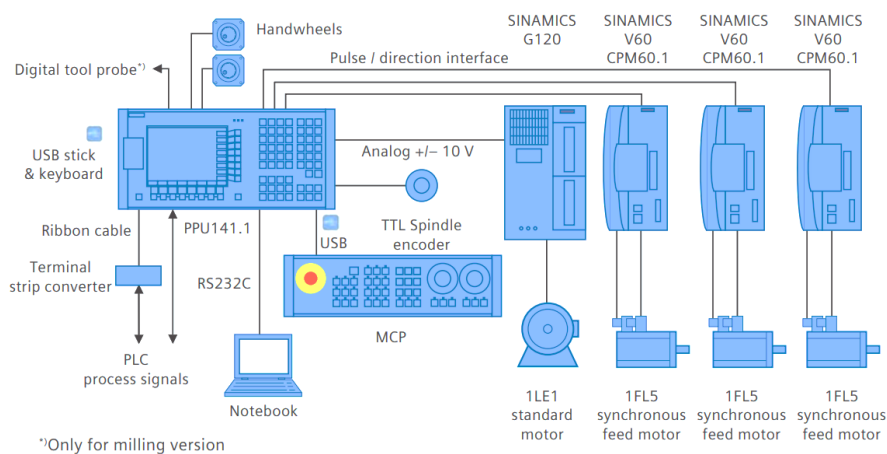
Sinumerik 808 je základní řadou CNC řízení panelové konstrukce s uživatelskou obrazovkou. Je doporučován do náročnějšího prostředí. Prostorově nenáročné rozměry pak umožňují použití v kompaktních zařízeních. [6]



Obr. 1: Sinumerik 808: Panel řízení [18]

Řada je optimalizována pro základní frézovací a soustružnické aplikace. Nabízí se tedy použití v základních frézovacích obráběcích center a soustruzích. Je k dispozici dostatečný výkon a přiměřené nástroje například pro výrobu forem. Sinumerik 808 je spolehlivým systémem, kdy uživatel preferuje robustnost a jednoduchost před moderními myšlenkami sběru dat, vizualizacemi a funkcemi pro analýzu výrobního procesu. Proto zde nelze najít funkce jako jsou kompenzace (teplotní, třecí, mechanické), předcházení kolizí, hlídání životnosti nástroje a podobně. [2]

Modulární architektura neomezuje výrobce v umístění komponent, kde příkladem je odnímatelný ovládací panel MCP připojený skrze sběrnici USB.



Obr. 2: Sinumerik 808: Příklad možné konfigurace [2]

Sinumerik 828

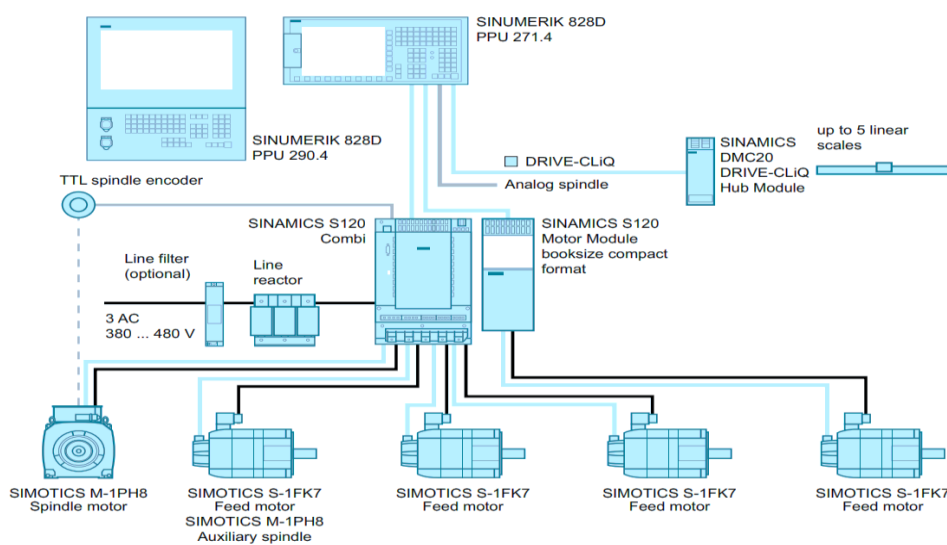
Sinumerik 828 je bezúdržbovou, vyšší verzí panelového řízení, které počítá s dnešní vahou automatizace. Mezi přednostmi vůči předchozí řadě systému je ethernetové rozhraní, vertikální nebo horizontální 10.4“ nebo 15.6“ dotykový displej a plnohodnotná QWERTY klávesnice. Tyto prvky umožňují přehledné dílenské programování v různých standardech. Systém se dále dělí na výkonnostní řady. [6]

Sinumerik 828 má zajistit možnou automatizaci CNC frézek a soustruhů. Poskytuje obsluhu prostředí k využívání široké řady funkcí pro podporu obrábění, ke kterým Sinumerik 808 nemá prostředky.



Obr. 3: Sinumerik 828: Panel řízení [3]

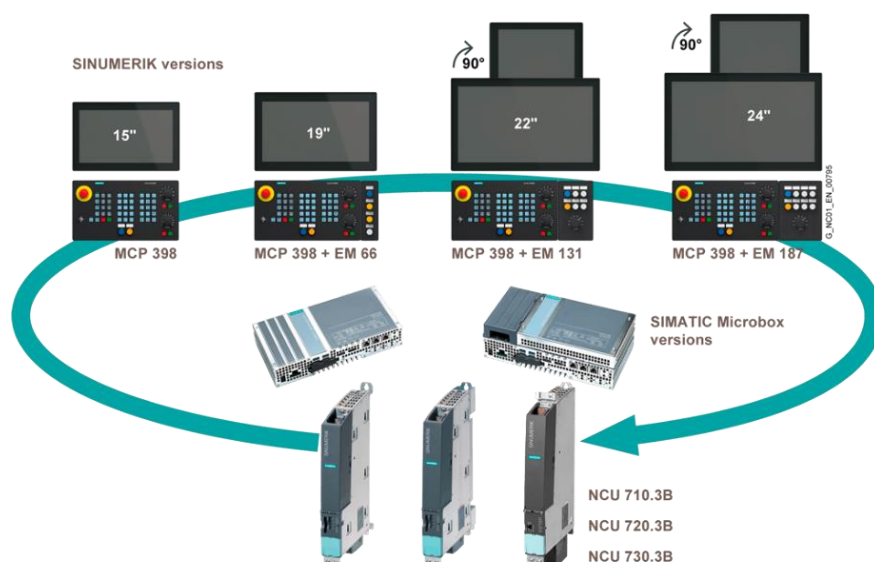
Architektura řízení je podobná, jako u nižší verze, opět je CNC, PLC a HMI vestavěno v panelu, který je schopen v závislosti na variantě řídit až 8 os celkem ve 2 obráběcích kanálech.



Obr. 4: Sinumerik 828: Příklad možné konfigurace [3]

Sinumerik 840

Řada řízení Sinumerik 840 nejkvonnější řadou řízení, které firma nabízí. Nabízí nejvyšší výkon, variabilitu a otevřenost, která je vhodná pro širokou škálu koncepcí obráběcích center. Řízení s PLC již není obsaženo v HMI, ale v řídicí jednotce NCU. Jedná se o modulární systém panelů, řídicích jednotek, pohonů a dalších komponent, které lze skládat dle požadavků na výkon a činnost zařízení. HMI již nemusí být umístěno společně s jednotkou PCU. [4]



Obr. 5: Sinumerik 840: Varianty řízení [4]

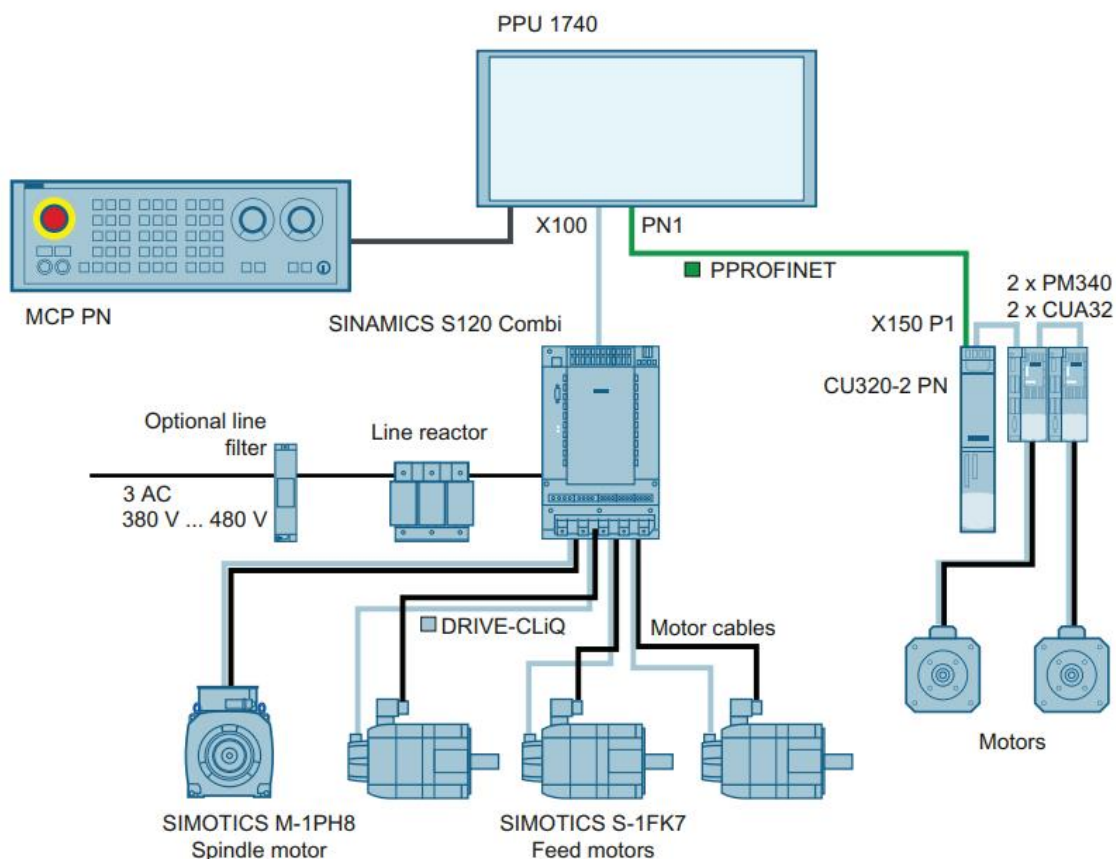
Architektura řízení je, jak již bylo zmíněno, postavena tak, aby bylo dosaženo co možná největší variability dle potřeb výrobce. Neměl by být problém zařízení rozšířit například o mechanické zakladače, dopravníky, nebo roboty.

Sinumerik ONE

Aktuálně nejnovější řízení Sinumerik ONE je od základu postaveno na myšlenkách digitalizace a automatizace. Řízení je pro obráběcí centrum navrhováno jako digitální dvojče, kde mohou být simulovány reálné podmínky a vytvářeny virtuální procesy. Díky těmto prostředkům má být zkrácen procesní čas, jelikož každý krok je možné simulovat na digitálním dvojčeti, kde se předejde mnoha problémům jako jsou kolize, nedostatek prostoru a potvrdí se proveditelnost na reálném zařízení. Sinumerik ONE by měl mít pro zákazníka díky plánování výroby, virtuálního obrábění a optimalizací za následek snížení časů, kdy je obráběcí centrum neproduktivní.

Řízení je dle firmy vhodné také pro takzvané „Retrofit“ obráběcích center, kde nové řízení je odladěno na simulacích a modelech stávajícího centra. Umožňuje tedy vytvořit a přednastavit systém tak, aby bylo možné jím efektivně a rychle nahradit ten stávající s minimálním následným laděním. [5]

Architektura řízení je opět modulární, ale HMI a PCU se stalo jedním hardwarovým prvkem nazývaným panelová procesní jednotka. Konfigurace se 4 osami a 2 pomocnými může vypadat například tak, jak je uvedeno na následujícím obrázku.



Obr. 6: Sinumerik ONE: Příklad hardwarové konfigurace [5]

2.2 Řídicí systémy firmy Heidenhain

Německá firma DR. JOHANNES HEIDENHAIN GmbH se zabývá vývojem a výrobou úhlových, rotačních a polohových snímačů společně s aplikacemi pro jejich řízení a polohování [19]. Heidenhain řízení se (na rozdíl od Siemens) zaměřuje primárně na obráběcí stroje a zařízení, takže nenabídne možnosti pro připojení desítek os a ani nenabízí samostatné PLC systémy.

Heidenhain, podobně jako Siemens nabízí několik modelových řad řízení pro soustružení a pro frézování. Řady pro samotné soustružení, kterými jsou CNC PILOT 640 a MANUALplus 620 nejsou v České republice příliš rozšířené, proto jsou v následujících kapitolách popsány pouze systémy podporující frézování, nebo frézování a soustružení.

U řídicích systémů TNC je zásadní odlišnost, kterou je používání operačního systému HEROS 5 (Heidenhain Real-time Operating System), který je založen na systému Linux. Jedná se tedy o linuxový systém, ve kterém je současně spuštěno PLC s aplikací HMI. [17]

Heidenhain poskytuje velký počet softwarových a hardwarových opcí, o které může zákazník žádat. Příkladem může být na řídicím systému TNC 128 připojení až 2 dalších os nebo obrobkových sond.

Heidenhain TNC 128

Základní řadou od firmy Heidenhain je panel řízení TNC128, který je určen k pravoúhlému řízení tří os a vřetene. Je tedy vhodný k frézovacím a vrtacím operacím. Zásadní rozdíl oproti vyšším modelovým řadám je fakt, že TNC128 je aplikován spíše na již existující obráběcí stroje, jsou tedy zpravidla vyžadovány testy celého systému bez ohledu na specifikace obráběcího stroje. Heidenhain dané řízení doporučuje pro centra určená ke kusové nebo malosériové výrobě. Řídicí jednotky, motory a snímače Heidenhain nejsou pro TNC 128 použitelné. [7]

Řízení je směřováno k jednoduché obsluze zařízení a dílensky orientovanému programování. Pro frézování čela není dokonce nutné psát programy, jelikož lze využít přímo posuvného kolečka či tlačítek na panelu. Programování na stroji se stává se složitějšími výrobky náročnější, TNC 128 proto nabízí přizpůsobené prostředí *Klartext* pro offline dílenské programování, kde je obsluha naváděna pomocí grafických oken a nápovědy.



Obr. 7: TNC 128: Panel řízení [21]

Uvedený systém řízení je základní řadou, ale i tak v základní konfiguraci obsahuje gigabitové ethernet rozhraní a tři porty USB, takže není žádným problémem i online programování. Protože, jak už bylo zmíněno, Heidenhain je řízení založené na linuxovém systému a díky tomu je například možné přistupovat k NC i PLC části z Windows zařízení jako k serverové službě Samba. [7]

Architektura řízení TNC 128 není specifikována, jelikož se může na základě aplikace zásadně lišit.

TNC 320

Vyšší řadou řízení hned po TNC 128 je řízení TNC 320. Koncept řízení i aplikace je v podstatě totožný, takže stejně jako TNC 128 je vhodný pro analogové osy a „retrofity“ obráběcích zařízení. Nabízí ale větší, 15“ displej, větší procesní výkon, cykly pro obrobkové sondy a řízení až 6 os, s čímž souvisí i schopnost jejich interpolace. [8]

Vyšší verze systému také podporuje funkce pro výrobce obráběcích center, jako je například výměna palet, gantry osy v momentové vazbě, nebo vzdálený přístup k panelu z PC.



Obr. 8: TNC 320: Panel řízení [22]

TNC 620

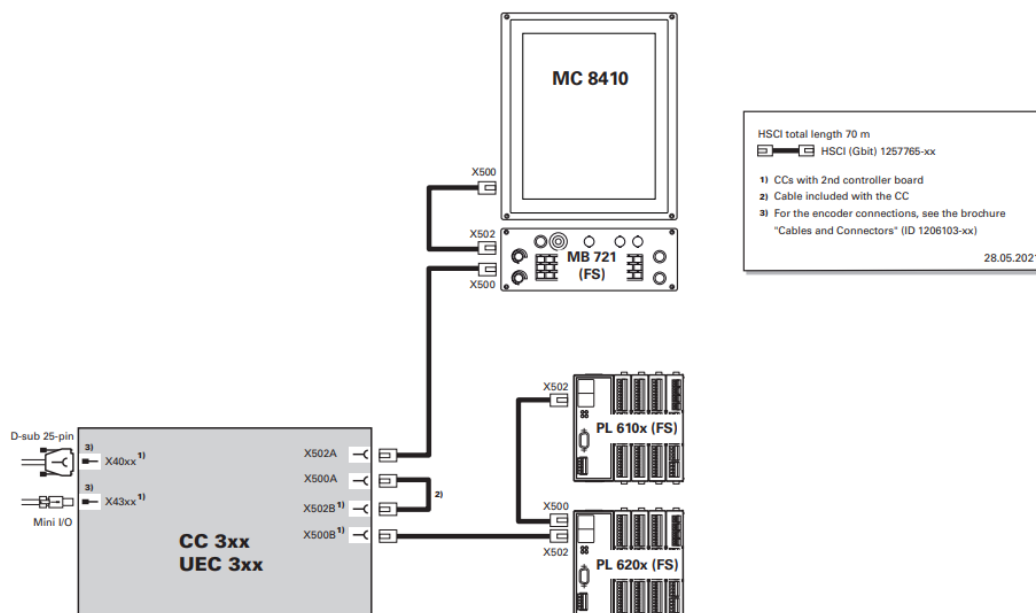
TNC 620 je modulární řízení až pro 5 os dostupné jako 19“ multidotykový panel s integrovanou klávesnicí v displeji, nebo jako 15“ multidotykový displej s klávesnicí zvlášť. Tak jako u nižších řízení je přímo v panelu zabudováno PC se systémem HEROS.



Obr. 9: TNC 620: Panel řízení [23]

Systém TNC 620 je postaven pro řízení Heidenhain driverů a ideálně i motorů. Podporována je ale široká škála motorů jiných výrobců i motorů Siemens, jelikož původně firma Heidenhain pracovala hlavně s motory a drivery od firmy Siemens. Kompletní sestava pro řízení, tj. řídicí jednotka, napájení, motory a enkodéry je dostupná jako celek, což znamená, že výrobce obráběcího centra vytipuje komponenty, Heidenhain pak potvrdí sestavu a nabídne ji výrobcovi jako standartní paket. [9]

Architektura řízení je oproti předchozím řadám odlišná už jen díky faktu, že hlavní komponenty (Panel, PLC vstupně-výstupní karty, klávesnice a rozšiřující moduly) řízení komunikují na sběrnici HSCI (HEIDENHAIN Serial Controller Interface). HSCI komunikace je v poslední generaci řízení postavena na gigabitovém ethernetovém hardwaru. HSCI sběrnice umožňuje tvorbu flexibilních a škálovatelných systémů, má vysokou odolnost vůči rušení a jednoduše se zapojuje prostřednictvím optických kabelů. Hardwarově splňuje požadavky pro funkční bezpečnost, podporuje celou řadu bezpečnostních funkcí (SS0 – SS2, STO, SOS, SLS, SLP, SBC) a bezpečné operační režimy pro skládání, seřizování a produkci obráběcích center. [9]



Obr. 10: TNC 620: Základní hardwarová konfigurace [9]

TNC 640

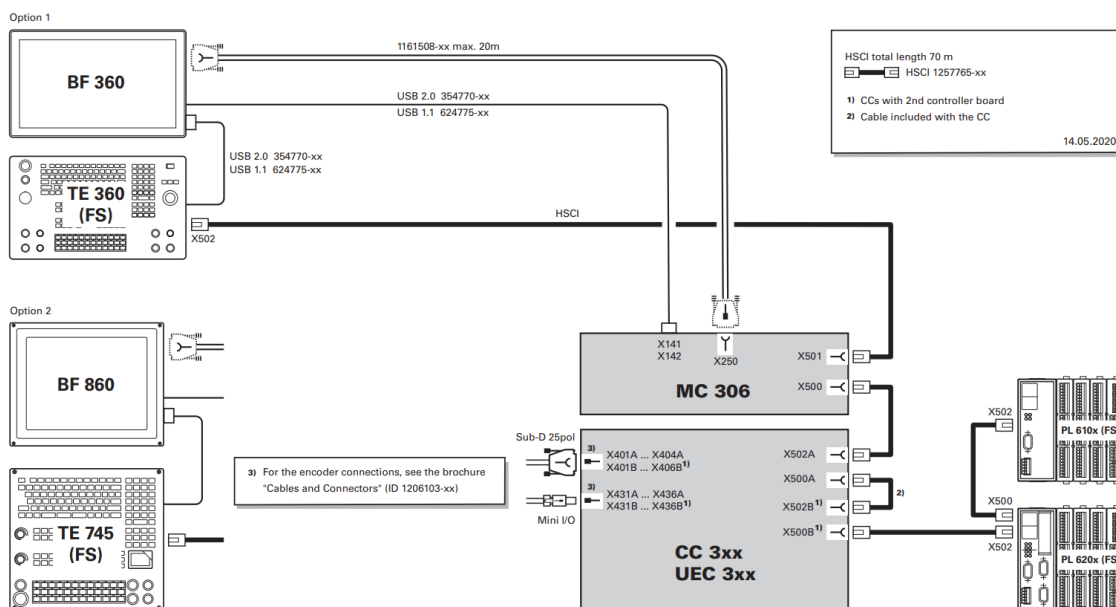
Univerzální řízení TNC 640 je nejvyšší řadou řízení, kterou firma Heidenhain nabízí. Umožňuje jak frézovací, tak soustružnické operace a pokud výrobce stroje umožní, tak lze vytvořit i obráběcí centrum, které je schopno fungovat přepnutím kinematiky v kterémkoliv z těchto režimů. Tak jako TNC 620 je propojení hlavních komponent řešeno skrze sběrnici HSCI. Systém se od TNC 620 liší v podstatě pouze výkonem, možností přepínání kinematik a podporou 5-osého obrábění až s 24 osami. [10]

Vyšší řada řízení je charakteristická 24“ displejem BF 360 s rozlišením 1920 x 1024 pixelů, kde řídicí PC s názvem MC 306 je určeno k instalaci do elektrorozvaděče. Větší panel přináší další možnosti, jak usnadnit práce a rozšířit možností systému, kterou je například opce „Extended Workspace Compact“, kde v panelu během obrábění lze prohlížet modely z CAD dat, PDF soubory, nebo se vzdáleně připojit k Windows zařízení a pracovat vzdáleně z panelu. [10, 17]



Obr. 11: TNC 640: Panel a standardní rozložení klávesnice [10]

Hardwarová sestava řízení je obdobná, jako u TNC 620, což znamená, že pro chod systému je nezbytná karta vstupů a výstupů PL 6000 a další komponenty, které jsou voleny v závislosti na charakteru systému. Úložný prostor oproti TNC 620 zajišťuje SSD disk, ve kterém je uložen řídicí systém, PLC i NC programy.



Obr. 12: TNC 640: Schéma zapojení panelu a řídicího PC [10]

3 PROGRAMOVÁNÍ PLC

Výrobci jednotlivých řídicích systémů umožňují výrobcům obráběcích center přizpůsobovat chování a vlastnosti stroje pomocí PLC vestavěných v řízení. PLC pak může komunikovat s řízením, elektronickými vstupy a výstupy a přizpůsobovat tak chování centra. Příkladem takové aplikace může být řazení rychlostních stupňů vřeten s převodovkami, pravidelné mazání os, obsluha dotykových a nástrojových sond, automatické výměny nástroje, hlídání teplot, nebo i záznam klíčových událostí pro následné odladování. [11]

PLC je tedy prvkem řízení, ke kterému výrobce řízení zpravidla poskytuje i software, ve kterém lze jeho programování provádět. Jazyků programování PLC existuje široká řada a závisí na výrobci řídicího systému, kterým z nich programování PLC umožní. Dle normy IEC 61131-3 pro programovatelné logické kontroléry je lze dělit například na textové a grafické tak, jak ukazuje následující tabulka. Standard IEC 61131-3 definuje programovací režimy PLC bez ohledu na hardware, kde výrobcem PLC ale nemusí být nutně využit, jak tomu je například u systémů od firmy Heidenhain. [12,17]

Tab. 2: Programování PLC: Rozdělení programovacích standardů [13, 20]

Jazyk	Zkratka	Typ
Ladder diagram	LD	Grafický
Function block diagram	FBD	Grafický
Sequential function chart	SFC	Grafický
Instruction list	IL	Textový
Structured text	ST	Textový
Object oriented programming	OOP	Textový

3.1 Instruction list

Instrukční list byl mezi prvními jazyky pro programování PLC spolu s ladder diagramem. Jedná se o nízkoúrovňový programovací jazyk PLC a je přirovnáván k programování v jazyku Assembler. V mezinárodním standardu IEC 61131-3 byl podporován jako jeden z pěti jazyků, ale ve 3. vydání normy je označen za zastaralý a nebude dále podporován. [13]

Samotný program je tvořen posloupností jednoduchých instrukcí jako je sčítání, násobení, přesuny hodnot v paměťovém prostoru a podobně. Všechny PLC řídicího systému Heidenhain je doposud možné programovat jen v jazyku IL. Každá instrukce programu je na samostatném řádku a každá instrukce se skládá z operátoru, operandu a případně modifikátoru. I když standard IEC 61131-3 jasně definuje sady instrukcí, výrobci PLC se stále rozcházejí minimálně v sadách dostupných instrukcí. Heidenhain definuje zhruba 74 instrukcí, které obsahují načítání a zápisy do datových typů, logické, aritmetické, srovnávací operace a speciální operace pro práci s funkcemi jako je limitování momentu

vřetene nebo vyčítání dat z komunikace PROFINET. [17] Několik instrukcí pro PLC Heidenhain je v následující tabulce.

Tab. 3: Instruction list: Přehled vybraných instrukcí [17]

Syntax	Význam
LB	Load BYTE
=N	Assign NOT
RN	Reset not
BS	Bit set
JT	Jump if true
<=	Less or equal to

Ukázka programu pro detekci zaseknutí se třískového dopravníku v systému TNC 640 je na následujícím obrázku.

```

LBL Overload_1
L    MG_chip_conveyor_1_running_sensor
IFT
M_start_time(TR_chip_conveyor_positive_signal,KL_Chip_Conveyor_1)
ELSE
M_start_time(TR_chip_conveyor_negative_signal,KL_Chip_Conveyor_1)
ENDI

L    TR_chip_conveyor_positive_signal[KL_Chip_Conveyor_1]
XO   TR_chip_conveyor_negative_signal[KL_Chip_Conveyor_1]
=    MG_chip_conveyor_1_stand

L    O_chip_conveyor
O    O_chip_conveyor_back
IFF
M_start_time(TR_chip_conveyor_started,KL_Chip_Conveyor_1)
ENDI
EM

```

Obr. 13: Instrukční list: Ukázka struktury jazyka

Výhodou Instrukčního listu je jeho rychlost a velikost v porovnání s grafickými jazyky. Programátor má naprostou kontrolu nad strukturou, chováním a možnými optimalizacemi programu. [13]

Na druhou stranu z ukázky [obr. 12] je patrné, že jazyk není vhodný k složitým aplikacím, protože program se s větším počtem instrukcí stává nepřehledným. Hledání chyb se s nárůstem instrukcí stává zdlouhavým a pro údržbu bez znalosti jazyka instrukčního listu pak prakticky nemožným procesem.

3.2 Structured text

Jazyk strukturovaného textu je vysokoúrovňovým, textovým programovacím jazykem pro PLC syntakticky se podobající jazyku Pascal. Jazyk podporuje většinu prvků moderních programovacích jazyků jako jsou interační smyčky (WHILE, FOR, REPEAT) nebo datové struktury. Stejně jako instrukční list není náročný na procesní čas a velikost programů je malá ve srovnání s grafickými jazyky. [25]

Hlavní předností strukturovaného textu je schopnost provádět komplexní matematické kalkulace s využitím funkcí jako například SIN() nebo LOG(), proto jej mnoho funkčních bloků i v grafických programovacích jazycích využívá jako svůj základ. Další výhodou je podobnost s moderními jazyky, což usnadňuje přechod programátorů z jiných jazyků, zajišťuje dobrou čitelnost a snadné porozumění programu. [26] Strukturovaný text nabídne úsporu řádků oproti instrukčnímu listu i dalším jazykům z čehož plyne jeho další výhoda, kterou je zobrazení většího množství informací na stejném pracovním prostoru, jak je zobrazeno v následujícím obrázku.

```

IF ( bStartFlag AND NOT(bBusy) ) THEN
  nStatus := 1;
  (* Not executed if data is already being read. *)
END_IF;
(* Read processing *)
CASE nStatus OF
  1 : (* Read command to bar code reader *)
    IF SendEnableCPUPort = TRUE THEN
      (* Send if RS-232C port can send data. *)
      bBusy := TRUE; (* Turn ON Vision Sensor reading flag. *)
      TXD_CPU('MEASURE'); (* Send "Measure once" command. *)
      nStatus := 2;
    END_IF;
  2 : (* Get data read from bar code reader. *)
    IF EndRecvCPUPort = TRUE THEN
      (* If the reception completed flag is ON *)
      RXD_CPU(strXYPosition, 25); (* Read reception data to strXYPosition. *)
      nStatus := 3;
    END_IF;
  3 : (* Processing after the read *)
    (* Analyze the string from the Vision Sensor into X and Y coordinates. *)
    nLen := LEN(strXYPosition); (* String length *)
    nCommaPos := FIND(strXYPosition, ','); (* Delimiter position *)
    strXPos := LEFT(strXYPosition, nCommaPos - 1);
    (* Extract X-coordinate string. *)
    strYPos := MID(strXYPosition, nCommaPos + 1, nLen - nCommaPos);
    (* Extract Y-coordinate string. *)
    (* Convert strings to numbers and extract the command values. *)
    nXDiff := nXTargetPos - STRING_TO_DINT(strXPos);
    (* Command value := Target value - Present value *)
    nYDiff := nYTargetPos - STRING_TO_DINT(strYPos);
    (* Command value := Target value - Present value *)
    nStatus := 0;
    bBusy := FALSE; (* Turn OFF Vision Sensor reading flag. *)
END_CASE;

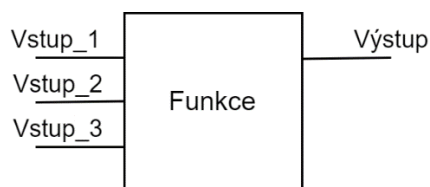
```

Obr. 14: Jazyk strukturovaného textu: detekce pozice vision senzorem [27]

Nevýhodou jazyka je skutečnost, že člověk bez předchozích zkušeností s programováním může mít problém s porozuměním programu a pro osobu provádějící servis na stroji může být náročné hledat příčinu závady.

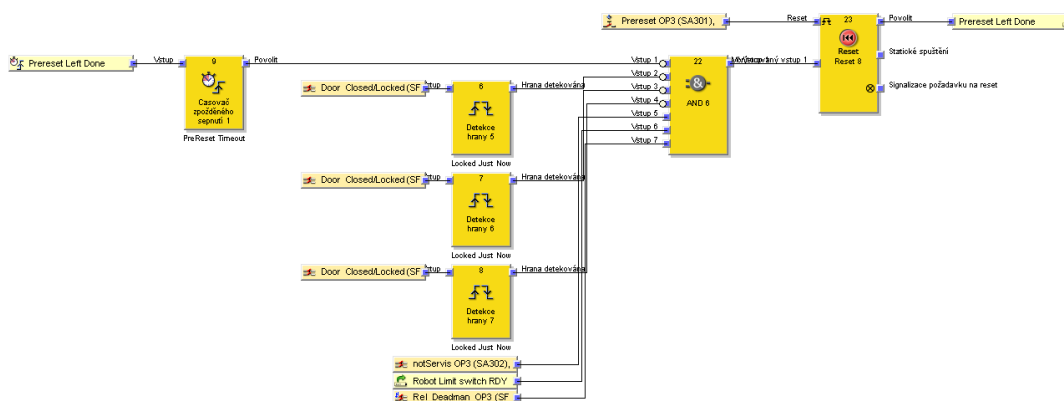
3.3 Function Block Diagram

Funkční blokové schéma (FBD) je grafický programovací jazyk pro PLC, kde je program tvořen za pomoci vedení signálů do grafických bloků. Z těchto bloků pak může být vyveden jeden nebo více signálů. Funkčním blokem se rozumí softwarový prvek, který při spuštění jako výsledek vrací jednu nebo více hodnot. Blok je v programu reprezentován tak, jak je zobrazeno na obrázku níže s názvem funkce, kterou provádí. [15]



Obr. 15: Funkční blok

Vstupem i výstupem funkčního bloku nemusí být jen binární hodnota, ale klidně numerická, textová, nebo i celá datová struktura. Funkční blok může mít podobu standardních operací jako jsou čítače, časovače, logické operace, hledání maxima a minima, speciální funkce definované výrobcem jako jsou PSD regulátory, nebo funkce vytvořené programátorem. Skládáním funkčních bloků pak vzniká funkční blokové schéma.



Obr. 16: Funkční blokové schéma

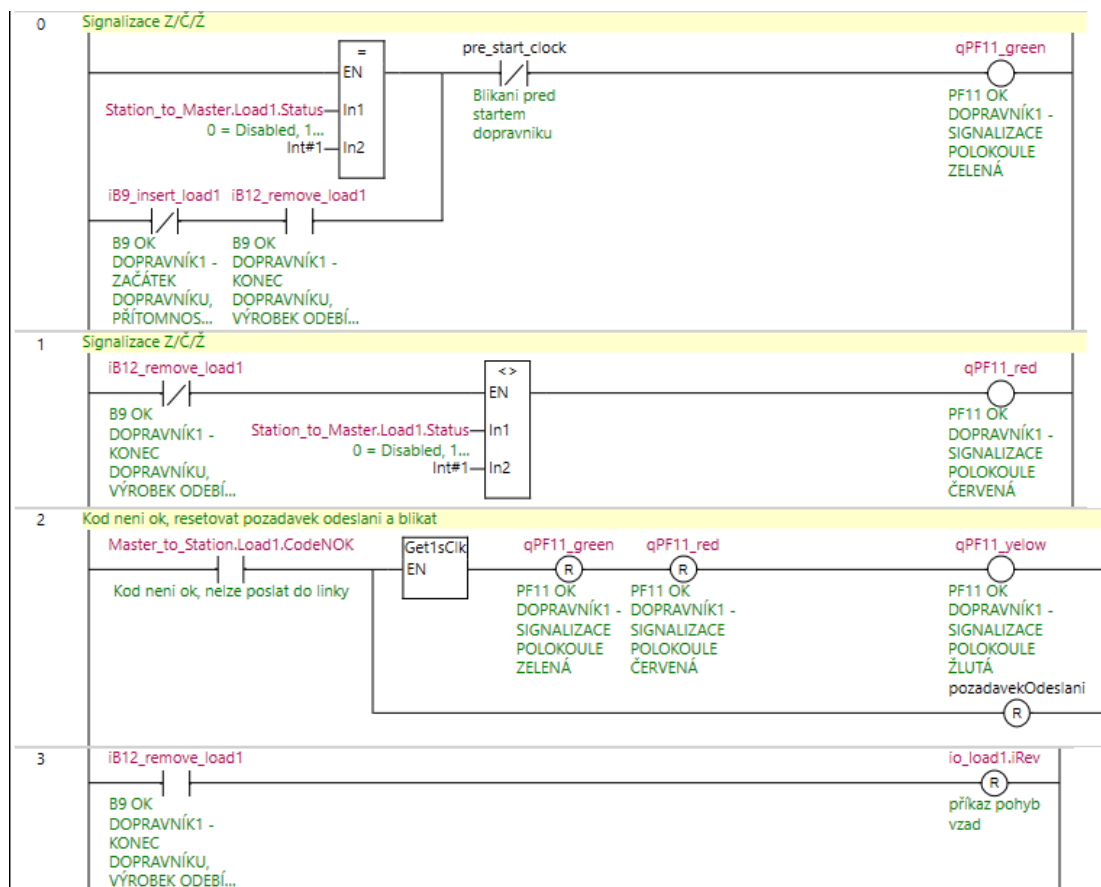
Mezi výhody FBD patří jednoduchost programování díky vizuální reprezentaci logiky programu. Z toho důvodu není potřeba náročných softwarových školení pro programátora. Další kladnou vlastností je, že bloky FBD lze v programu opakovaně

používat bez rizika ovlivnění vnitřních proměnných hodnot, jelikož s každým novým použitím bloku vzniká i další databáze proměnných pro tento blok. Jazyk FBD má také výborný základ pro vykonávání paralelních operací díky přehledné reprezentaci paralelní logiky. S přehlednou reprezentací kódu je usnadněna také diagnostika, kde většina softwaru podporující programování ve FBD podporuje také simulaci s vyznačeným vedením signálů. [24]

Algoritmy matematických a nízkoúrovňových operací jsou tradičně vytvořeny pomocí textového programovacího jazyka kvůli rychlosti a robustnosti, proto tvorba nových funkčních bloků může vyžadovat rozsáhlejší znalosti programování i v nižších jazycích. Zachování historie a verzí vývoje programu je složitější než u textových, jelikož už nestačí jen porovnání programů jako textových souborů a pro zobrazení změn je potřeba využít přímo softwaru výrobce. Grafická reprezentace kódu s sebou nese také složitější nativní integraci do moderních systémů IT jako jsou databáze nebo záznamy historie. Kdyby PLC systém HEIDENHAIN byl programován v jednom z grafických jazyků, řadu diagnostických funkcí z této práce (jako je vyhledávání vstupu v elektroschématech dle názvu proměnné PLC) by nebylo možné implementovat.

3.4 Ladder diagram

Jazyk příčkového diagramu (LD) je velmi rozšířená metoda programování PLC, kde program je tvořen prakticky kreslením spínacího obvodu mezi dvěma vertikálními čarami reprezentujícími napájecí sběrnici. Levá sběrnice reprezentuje logickou jedničku a „napájí“ prvky navazující vpravo, které signál mohou blokovat nebo pustit dále. Posledními prvky bývají výstupy a ty už jsou spojeny s pravou sběrnicí. Kód se provádí po řádcích zleva doprava. [16] Na obrázku níže je uvedena funkční ukázka pro signalizaci pásového dopravníku.



Obr. 17: Ladder diagram: struktura jazyka

Jazyk LD vyniká výbornou čitelností díky dobře se chápající reprezentaci programu jako obvodu. Výhodou je také možnost živě vidět tok signálů, což usnadňuje simulaci a ladění. Osobám provádějící servisní zásahy a diagnostiku se programovací jazyk LD dobře chápá. Jazyk je vhodný pro práci hlavně s binárními signály, ale je možné využívat i funkční bloky.

Stejně jako FBD ale není jednoduchá práce s verzemi ani integrace do IT. Přesto je ale LD oblíbený jazyk, díky čitelnosti. Není tedy nutno rozsáhlých školení pro technické a servisní pracovníky.

3.5 Programovací software pro HEIDENHAIN TNC 640

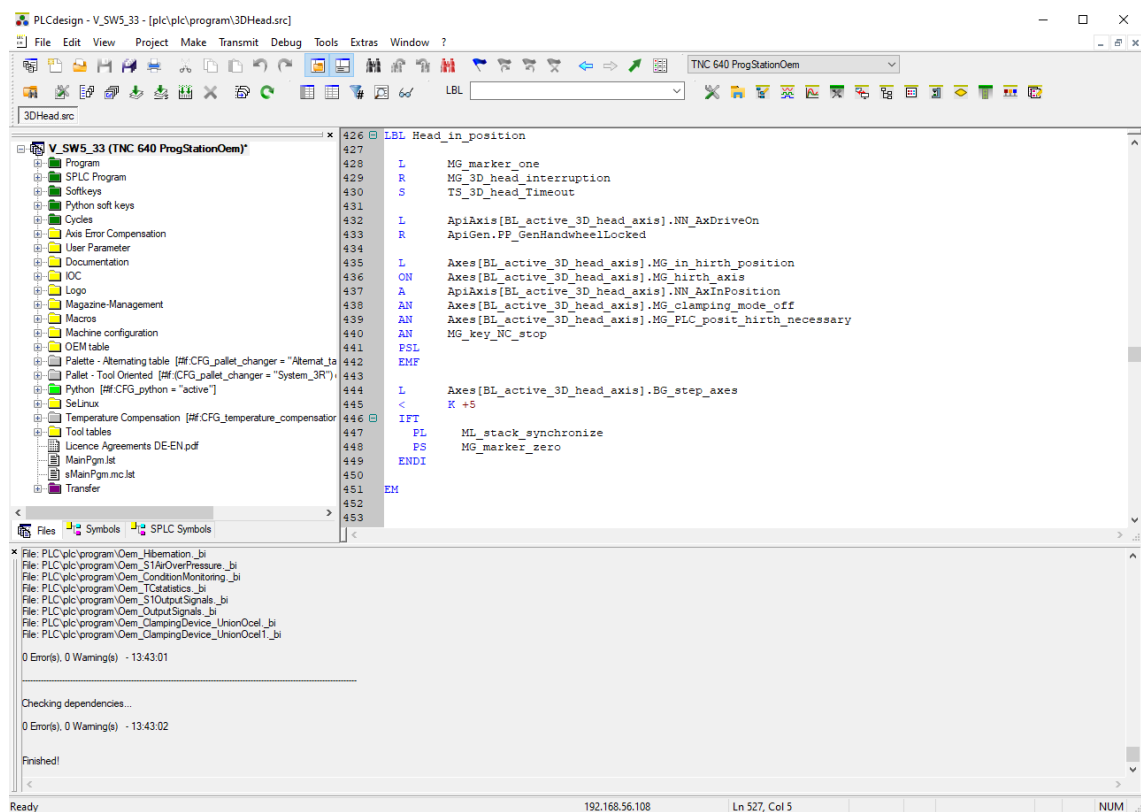
Výrobce řídicích systémů TNC poskytuje pro výrobce obráběcích center sadu softwarových nástrojů. Základním softwarem je PLCdesign, který slouží k programování PLC, uvádění stroje do provozu a základní diagnostiku PLC i kontrolérů. Přehled dalších nástrojů, které Heidenhain nabízí jsou popsány v tabulce níže.

Tab. 4: Seznam softwaru dodávaného k řídicím systémům TNC [17]

Software	Použití
PLCdesign	Vývojové prostředí pro programování a diagnostiku PLC
BMXdesign	Návrh softwarových tlačítek s podporou jazykových variant
ConfigDesign	Konfigurace parametrů stroje
CycleDesign	Umožňuje výrobcí stroje tvorbu vlastních cyklů, které lze použít při NC programování
IOconfig	Konfigurace sběrnic, ručních kol a pohonů
KinematicsDesign	Tvorba interaktivní kinematiky včetně kolizních objektů
PLCtext	Zpracovávání a správa jazykově závislých textů
TeleService	Umožňuje dálkovou diagnostiku, monitorování i obsluhu
TNCanalyzer	Vyhodnocení a analýza servisních souborů
TNCdiag	Analýza diagnostických informací pohonů stroje
TNCkeygen	Umožňuje vytvoření denního hesla, které umožní zákazníkovi přístup do parametrů výrobce
TNCOpt	Zavádění digitálních regulačních smyček do provozu
TNCremo	Zajišťuje komunikaci mezi PC a řídicími systémy HEIDENHAIN
TNCscope	Externí osciloskop se záznamem i vyhodnocováním hodnot os i PLC
TNCtest	Tvorba a vyhodnocení přejímacích testů
virtualTNC	Simulace stroje včetně řízení na PC

Software zmíněný v tabulce výše je dodáván výrobcům obráběcích center a má umožnit přizpůsobení vlastností obráběcího centra nejen při obrábění, ale i graficky pro programování a ovládání obsluhou.

PLCdesign není jen softwarem k programování PLC, jedná se o celé vývojové prostředí včetně správy projektů, kompilace, nahrávání, sledování zdrojů řízení a možnosti integrace externích softwarů třetích stran. Mimo jiné také sdružuje i ostatní nástroje Heidenhain pro vývoj a oživení stroje. Obrazovka prostředí PLCdesign je na následujícím obrázku.



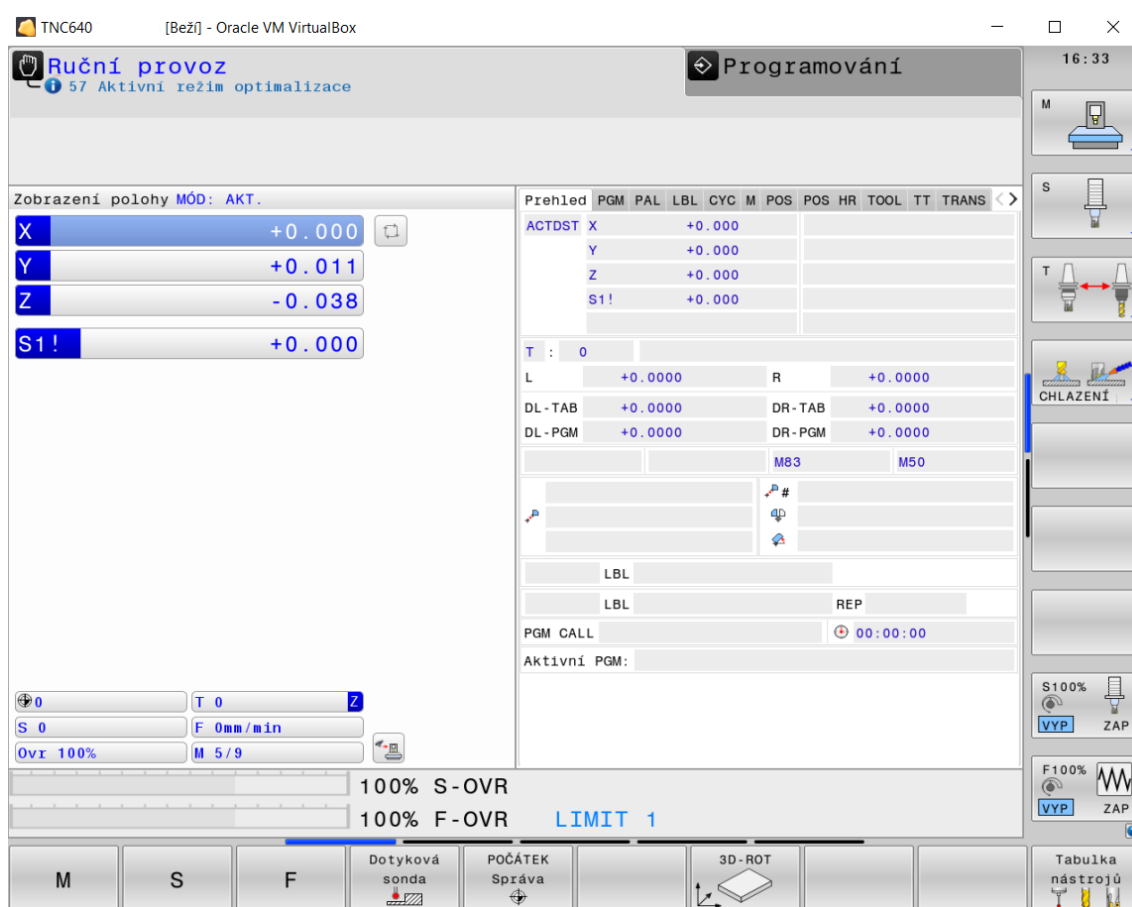
Obr. 18: Vývojové prostředí HEIDENHAIN: PLCdesign

Prvním horizontálním řádkem hned pod názvem softwaru a projektu je programové menu, které zajišťuje správu, kompilaci a další nástroje pro ladění řízení. Vybrané volby z těchto záložek jsou dostupné i jako ikony na dalším řádku v podobě sad widgetů, kde je zleva několik položek věnováno správě projektu a práci s textem. Poslední položkou řádku je volba konfigurace, kde obdobně jako u jiných vývojových prostředí je na výběr ze seznamu konfigurací, které pak mají vliv na kompilaci programu a případně simulaci vstupů nebo na funkční bezpečnost, když se program testuje a ladí ve virtuální stanici. Poslední řádek je tvořen vybranými nástroji pro kompilaci a ladění programu jako je tabulka paměťového prostoru, nebo trasování zdroje proměnných. Poslední částí řádku je výše zmiňovaný seznam softwarových nástrojů přístupných přímo z prostředí PLCdesign.

Prostřední sektory vývojového prostředí spolu tvoří okno stromové struktury projektu a okno pro kód programu PLC. Stromová struktura otevřeného projektu shromažďuje všechny podprojekty ze softwarových nástrojů pro daný projekt, mezi které patří například projekty hardwarové konfigurace z programu *IOconfig* nebo projekty softwarových tlačítek z programu *BMXdesign*. Při otevření podprojektu ve stromu se otevře příslušný software a lze provádět změny pro celý projekt v externím softwaru.

virtualTNC (virtuální stanice)

Software virtualTNC má nejčastější využití při školení obsluhy, jelikož se jedná o simulaci kompletního stroje včetně kinematiky. Je tedy možné simulovat například obrábění, ruční najíždění a psaní NC programu přesně tak, jak probíhá i na reálném stroji. Simulace ale nekončí jen u obrábění a panelu, základem virtualTNC je (v předchozích kapitolách zmiňovaný) HEROS, tedy stejný operační systém, který běží na reálných obráběcích centrech. Tato skutečnost dělá z virtualTNC ideální nástroj i pro programátory PLC, kteří mohou na virtuální stanici testovat všechny změny programu PLC i obrazovek. Virtuální stanice umožňuje i simulaci reálné strojní klávesnice. Ukázka prostředí TNC640 je na obrázku níže. I když se jedná o obrázek z virtuálního prostředí, prakticky nelze rozeznat od skutečného.



Obr. 19: virtualTNC: Hlavní obrazovka panelu TNC64

4 VLASTNÍ ŘEŠENÍ

Jak bylo zmíněno v předchozí kapitole, TNC 640 je nejvyšší řadou řízení od firmy Heidenhain a v následujících částech práce bude software vytvářen a testován pro tuto řadu řízení. Nutnými prekvizitami jsou systémy TNC (640, 620, 320 a 128) a doporučenou opcí pro tyto systémy je opce s ID 46 – *Python OEM process*, která umožní procesům Python využívat více než 10MB paměti RAM, což je pro obrazovky s více grafickými prvky nezbytností.

Strojírenská firma TAJMAC-ZPS a.s. působí ve vývoji CNC obráběcích strojů od roku 1980. Sortiment výrobního programu zahrnuje vertikální, horizontální, portálová, víceosá obráběcí centra, vícevřetenové a dlouhotočné soustružnické automaty.

Pro účely práce je zvolena nejprodávanější řada obráběcích center, kterou je MCV. Vertikální obráběcí centra jsou vysoce produktivní stroje pro komplexní třískové obrábění. Vertikální centra mohou být vybavena 4. a 5. osou, výměníkem palet a dalšími opcemi jako je například přídatný výměník nástrojů, chlazení nástroje středem vřetena nebo třeba adaptace stroje pro obrábění grafitu.



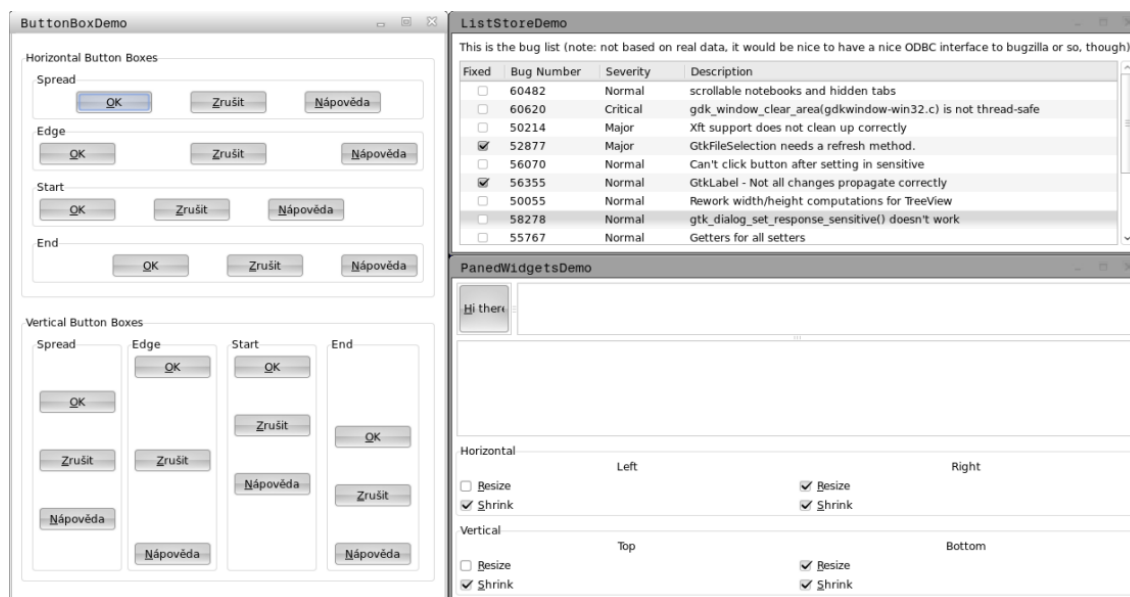
Obr. 20: Vertikální obráběcí centrum: MCV1260i [28]

4.1 Programové prostředky

Všechny vývoj programů v instrukčním listu probíhá přednostně v softwaru dodávaném výrobcem řídicího systému, což není podmínkou, v opačném případě ale programátor nemusí mít k dispozici sadu nástrojů pro práci s kódem, mezi které patří zvýrazňování proměnných nebo našeptávání. Instrukční list zde slouží zejména pro programování PLC, tvoří tedy menší část této práce, protože práce je zaměřena na grafickou interpretaci. Grafické prostředí vychází hlavně z hodnot PLC operandů, a proto v rámci práce bude podrobněji popsáno i programování PLC, kde vzniknou a budou popsány nové programy v PLC a právě na nich bude vysvětleno propojení s obrazovkami a vznik obrazovek samotných.

Vývoj obrazovek na systémech TNC probíhá v jazyce Python, tedy v textovém objektově-orientovaném programovacím jazyku, za použití knihoven GTK (systémy od firem Siemens, Fanuc, Omron, Schneider dnes běžně nabízí grafické editory obrazovek).

GTK je sada knihoven napsaná v jazyce C. Shromažďuje standartní widgety pro tvorbu rozložení a designu grafického prostředí. Knihovny GTK jsou zdarma, multiplatformní a open-source. Jedná se o velice rozšířené knihovny, které původně vznikly pro software GIMP, což je editor rastrových obrázků původně pro linuxové systémy. Později byly tyto knihovny přepsány do objektově-orientované podoby a následně adoptovány nadací GNOME. Tato nadace je známá hlavně díky open-source desktopovému prostředí pro systémy Linux, které je postaveno na výše zmíněných grafických knihovnách GTK, se kterými je toto prostředí současně i vyvíjeno. Grafické prostředí operačního systému HEROS5 je taktéž postaveno na knihovnách GTK, stejně tak i NC panel řízení TNC. Ukázka některých předdefinovaných widgetů GTK přímo ze systému HEROS5 je na obrázku níže.



Obr. 21: GTK demo: ukázka grafických prvků

Python je v dnešní době populárním interpretačním objektově-orientovaným programovacím jazykem, který vznikl v roce 1991. Popularitě začal nabývat až v posledních letech díky své jednoduchosti, flexibilitě a nárůstem výpočetního výkonu.

PyGTK je knihovna pro programovací jazyk Python, která zajišťuje propojení výše zmíněných sad knihoven a umožňuje tak jazyku Python používání prvků knihoven GTK psaných v jazyce C.

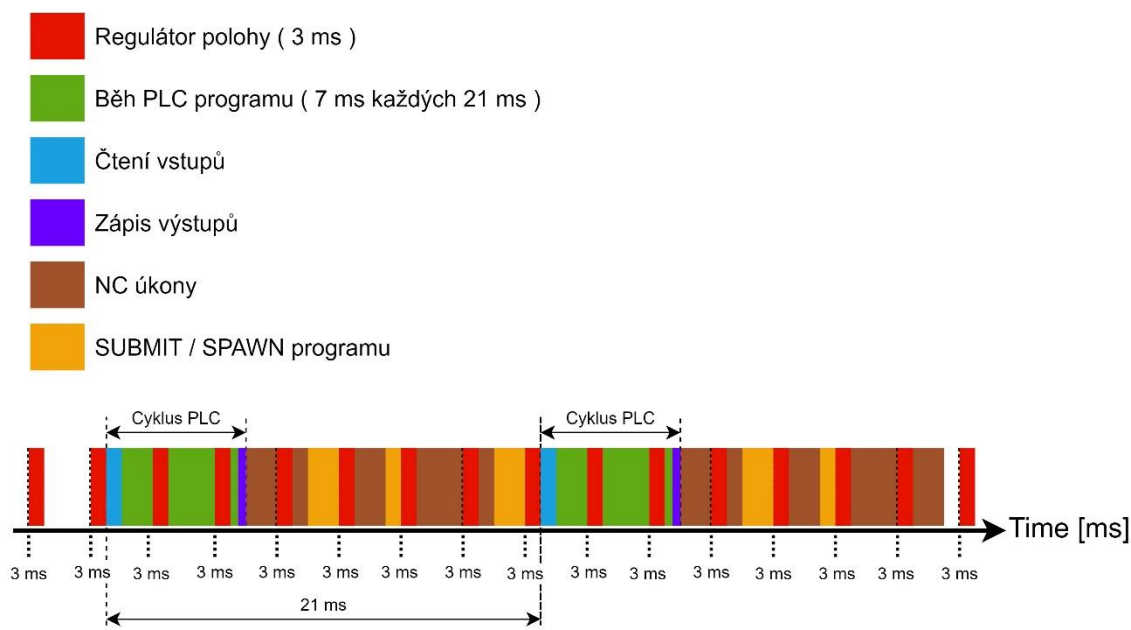
4.2 Řídicí operační systém reálného času HEROS 5 a Python

HEROS 5 je základem každého řídicího systému TNC a jak již bylo zmíněno, jedná se o typ linuxového systému, který je adaptován pro účely řízení NC. Využívá grafického prostředí XFCE, který je také založen na knihovnách GTK stejně jako GNOME, ale disponuje pouze nezbytnými funkcemi, není tedy tak náročný.

Prostředí XFCE v řízení nabízí pouze lištu s aplikacemi, ale díky tomu, že se jedná o plnohodnotné grafické prostředí, lze využívat aplikace jako například Firefox, Remote desktop Manager, editory tabulek a další grafické nástroje. Bohužel nelze běžně instalovat další aplikace (kromě kolekce aplikací podporovaných firmou HEIDENHAIN, což je ale pochopitelné z hlediska stability systému).

Jednou z aplikací je NCcontrol, která spouští ovládací panel HMI pro ovládání NC a umožňuje uživateli na stroji pracovat.

Výměna dat mezi NC a PLC probíhá v cyklických intervalech skrze datové typy (MARKER, BYTE, WORD...) nebo přes speciální Q-parametry používané v cyklech výrobce. U systémů TNC 640 je jeden cyklus PLC spuštěn každých 21ms, ale trvá běžně pouze 7ms. Cyklus se neskládá pouze z cyklu PLC, ale i z cyklů NC a dalších úkonů a jejich časová posloupnost lze vidět na obrázku níže.



Obr. 22: HEIDENHAIN PLC: Časové rozložení cyklu

Časové posloupnosti výše je nutné brát v úvahu při tvorbě programování PLC i obrazovek. Procesní výkon není neomezený a procesy Python, i když jsou spuštěny v samostatném vláknu, volají funkce, které jsou spuštěny ve vláknu PLC, proto jejich časté volání může mít nežádoucí dopad a může zapříčinit zastavení programu PLC. Z obrázku výše je také patrné, že nemá v žádném případě cenu vyčítat jakoukoliv proměnnou z PLC rychleji, než každých 21 ms, kde vzhledem k charakteru této práce až na jednu výjimku (která vznikne během testování obrazovek v práci dále) nebude nutné přistupovat k PLC proměnným častěji, než jednou za 200ms

4.3 Aktuální stav diagnostiky a první zadání

Hlavním podmětem pro vznik této práce je dosavadní stav nástrojů pro ladění a diagnostiku. První uvedení obráběcího centra do provozu vyžaduje kontrolu všech vstupů a nastavení stovek parametrů. Kontrola všech vstupních členů probíhá v tabulce diagnostiky, která je dostupná pouze pod heslem PLC výrobce stroje, což prakticky znemožňuje diagnostiku zákazníkem. K dispozici jsou 4 druhy tabulek rozdělené podle typu dat, které zobrazují. První z tabulek umožňuje binárně zobrazit všechny proměnné typu MARKER, INPUT, OUTPUT a TIMER, druhá tabulka slouží k zobrazení všech proměnných typu BYTE, WORD a STRING a poslední dvě tabulky umožňují zobrazit všechny vstupy a výstupy včetně analogových. Jedná se o užitečné nástroje, ale práce a pohyb v těchto tabulkách je zdoluhavý a složitý. Příkladem může být kontrola čidla zavřených dveří, kdy pro zobrazení hodnoty je nutné stisknutí klávesy MOD, zadání PLC hesla výrobce stroje, stisknutí tlačítka pro tabulky, kde je nutné vybrat typ tabulky pro binární přehled hodnot, vybrat záložku se vstupy, což uživatele navede na tabulku zobrazenou níže.

Ruční provoz 57 Aktivní režim optimaliza... TABULKY I/O/C/T/M/B/W/D 21:41

INPUT

	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
280	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
320	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
340	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
360	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
380	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
420	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
440	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
460	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I689 = Symbol not defined

M MARKER I INPUT O OUTPUT T TIMER C COUNTER SET/RESET NAVIGATE

Obr. 23: TNC 640: Tabulka přehledu binárních hodnot vstupů

Tabulka zobrazená výše zobrazuje všechny binární vstupy, kde ale pro příklad čidla zavřených dveří nepomůže ani znalost názvu vstupní proměnné, kterým je *I_guard_1_closed_locked*. Je nejdříve nutné najít číslo vstupu, které lze po kompilaci programu PLC nalézt v programu PLCdesign, což většinou vyžaduje kontaktování programátora, nebo projekt a školení pro osobu provádějící ověření. Po zjištění adresy vstupu (v tomto případě *I163*) lze v tabulce dohledat, že nabývá hodnoty 1, tedy dveře jsou dle vstupu zavřeny a zamčeny.

Proces popsaný výše je jen pro jeden vstup a už tak je bez pochyby časově náročný, vyžaduje proškolené osoby, zákazník nemá možnost bez vzdáleného přístupu diagnostiku provést sám a prodlužují se servisní zásahy. Při uvádění stroje je nutné projít všechny relevantní vstupy, což při sériové výrobě různých variací stroje enormně prodlužuje proces.

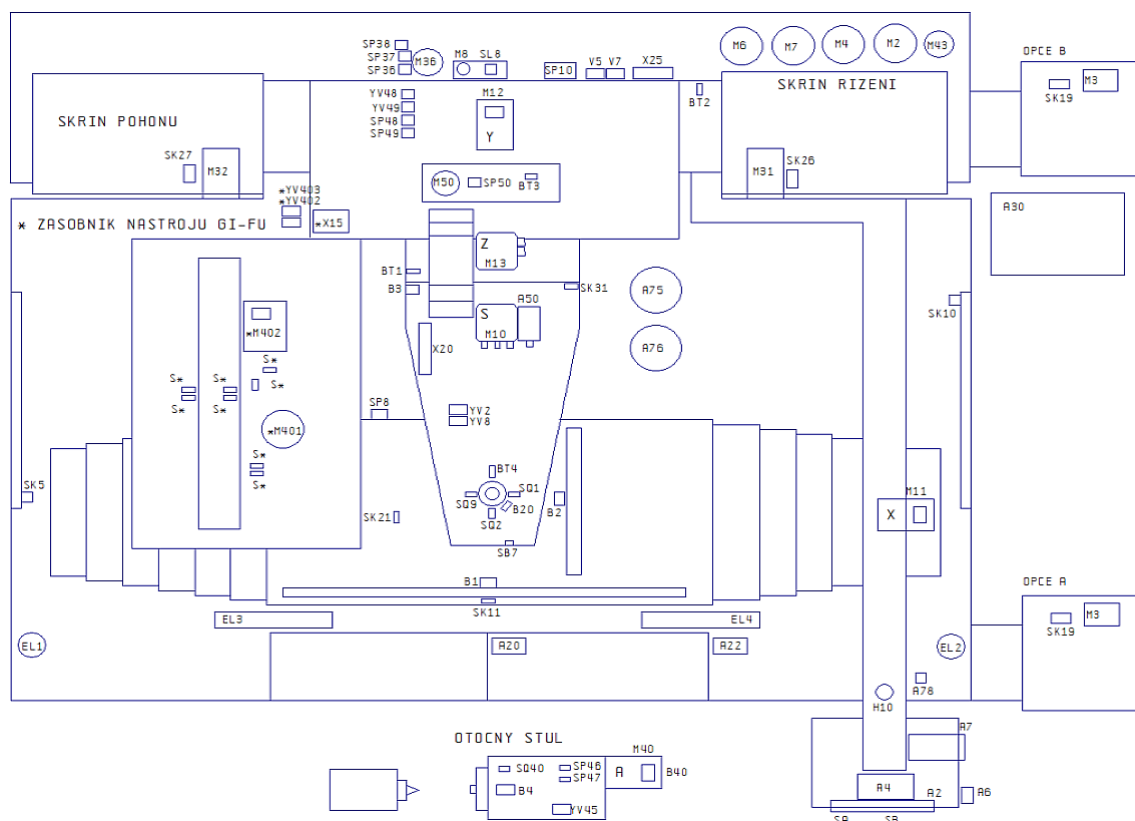
Uvádění stroje do provozu a v podstatě každý servisní zásah vyžaduje programátora, který s nárůstem počtu strojů přichází o čas k programování a místo toho dělá podporu při těchto úkonech. Z těchto důvodů vznikla původní myšlenka vytvoření seznamu vstupů a výstupů používaných na stroji. Nelze však generovat tabulku vstupů a výstupů, protože po určitých úpravách stroje může být adresní prostor odlišný a tabulky by neseděly. Bylo by možné ke každému stroji vytvořit seznam vstupů a výstupů, ale opět by se jednalo o zdlouhavou manuální práci programátora. Ideální by byl grafický náhled stroje, kde by byl vidět rovnou stav vstupu nebo výstupu. Kdokoliv by mohl živě vidět

kritické vstupní a výstupní hodnoty i s přibližným umístěním, což by usnadnilo oživení i servis stroje.

Původním zadáním tedy bylo libovolným možným způsobem docílit zobrazení požadovaného seznamu vstupů, výstupů a popřípadě PLC operandů pro odhalování chyb v programu.

4.4 Koncept a návrh

Návrh na řešení výše zmíněných problémů vznikl ještě když byla aktuální verze řízení TNC 530, tedy řízení, kde nebylo možné tvořit obrazovky v jazyce Python, ale jen v jazyku podobajícím se XML, bylo tedy prakticky nemožné používat jakékoliv jiné prvky než textové pole a nebylo možné zadání splnit. S příchodem aktuální verze, která je založena na grafickém rozhraní XFCE, se ale rozšířily možnosti tvorby obrazovek na používání všech dostupných widgetů i tvorby vlastních, odvozených, jak bude v dalších částech předvedeno. Bylo tedy možné vytvořit první náhled toho, co by v diagnostické obrazovce mohlo být vyobrazeno. První náhled vytvořený elektroprojekcí je půdorys s prvky, které jsou zapojeny do vstupních nebo výstupních karet, jak lze pozorovat na obrázku níže.



Obr. 24: První koncept: Půdorys obráběcího centra

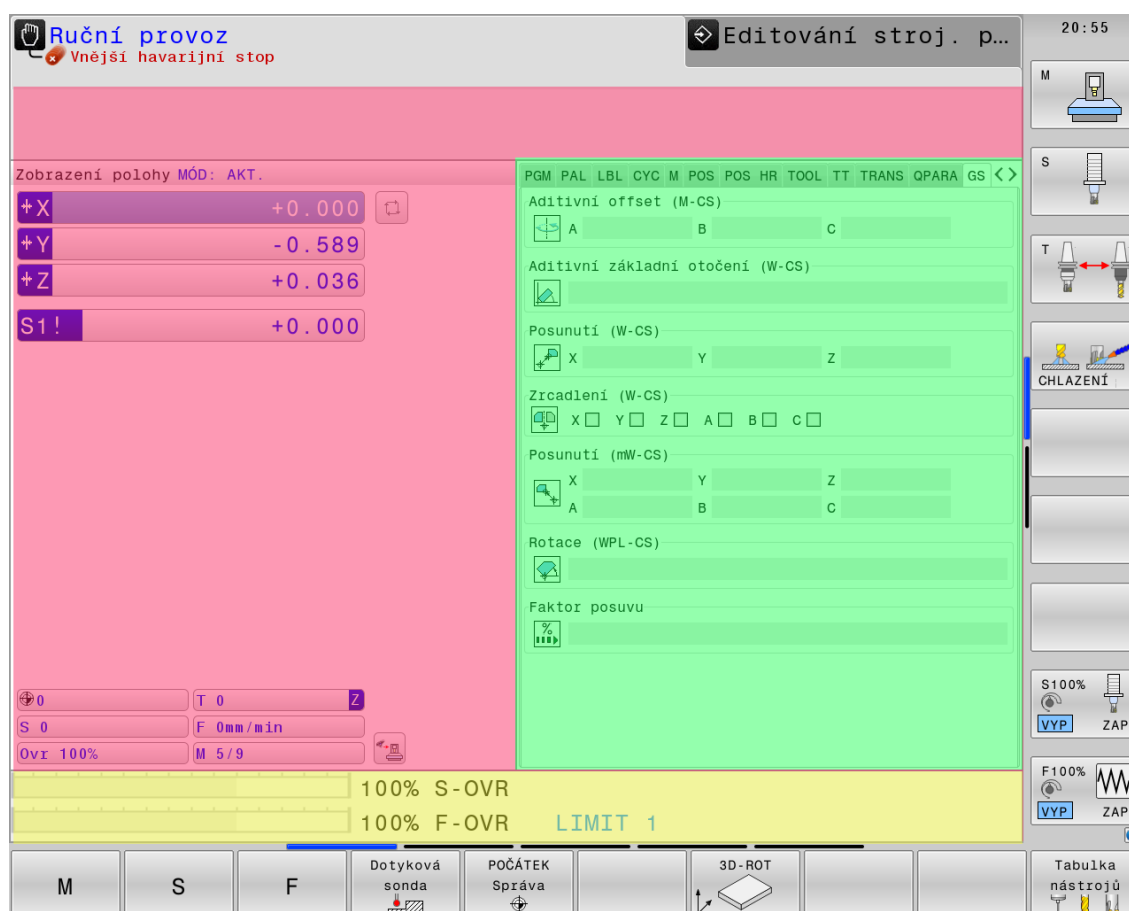
Z obrázku je patrné, že se už uvažuje o zobrazování volitelných opcí a náhled by měl být univerzální, nebo dynamický dle reálného stroje.

4.5 Vytvoření vlastní obrazovky

Tak jako každé grafické rozhraní, je i Heidenhain *NCcontrol* (Hlavní obrazovka řízení TNC) software složený z hlavního objektu (programového okna), který poskytuje prostor pro widgety (prvky a podokna grafického prostředí), stará se o jejich seskupení a zobrazení. Jedním z těchto podoknem bude i hlavní diagnostická obrazovka.

Předdefinované rozvržení

Programátoři HEIDENHAIN usnadnili tvorbu obrazovek tím, že umožnili využívat stávající rozvržení programu *NC Control* a rozdělili jej do částí, které může programátor výrobce stroje využít a není nutné vytvářet další plovoucí okna, se kterými by mohlo být problematické v průmyslovém prostředí manipulovat. Tyto části lze vidět na obrázku níže.



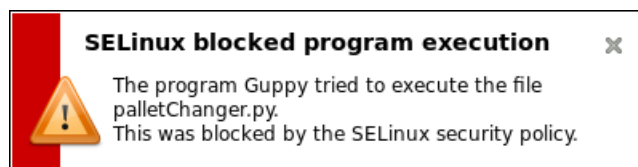
Obr. 25: NC Control: rozvržení využitelných oken

Řízení je složeno z mnoha částí a widgetů, jak je zmíněno i znázorněno výše. Barevně zvýrazněné jsou ty části, které lze využívat pro obrazovky výrobce. Červeně vyznačená část obrazovky se nazývá pro účely programování *PLClarge* a obsahuje zeleně zvýrazněnou (*PLCmedium*) i žlutě zvýrazněnou (*PLCsmall*) část.

Předpoklady pro běh obrazovek

Jak bylo zmíněno v předchozí kapitole, je doporučena aktivace opce ID 46 – *Python OEM process*. Dalším krokem je pak povolení Python programů a masek (tak se nazývají v TNC obrazovky, které využívají předdefinované rozvržení). Povolení se provádí v konfiguračním souboru výrobce (zpravidla soubor *Oem.mcg*), kde je nutné přepsat hodnoty předdefinovaných parametrů *CFG_python* a *CFG_python_masks* na „*active*“.

Prerekvizitou pro spuštění kterékoliv vlastní obrazovky je vypnutí ochrany *SELinux*, což je software, který má za úkol bránit systém před elevací práv malware. Důvodem nutnosti vypnout *SELinux* je vlastnost tohoto softwaru, která brání ve spuštění jakéhokoliv souboru, který nebyl součástí instalace systému, tedy i python programů. Vypnutí lze provést najetím kurzoru do spodní části řízení a po zobrazení hlavního panelu kliknout v sekci *Nastavení* na *SELinux*. Pokud *SELinux* nebude deaktivován, bude se chyba projevovat při spouštění obrazovek jako pokus o spuštění programu s neexistujícím názvem, což může být samo o sobě zavádějící, navíc se ale zobrazí v pravém spodním rohu řízení hlášení, které je na obrázku níže.

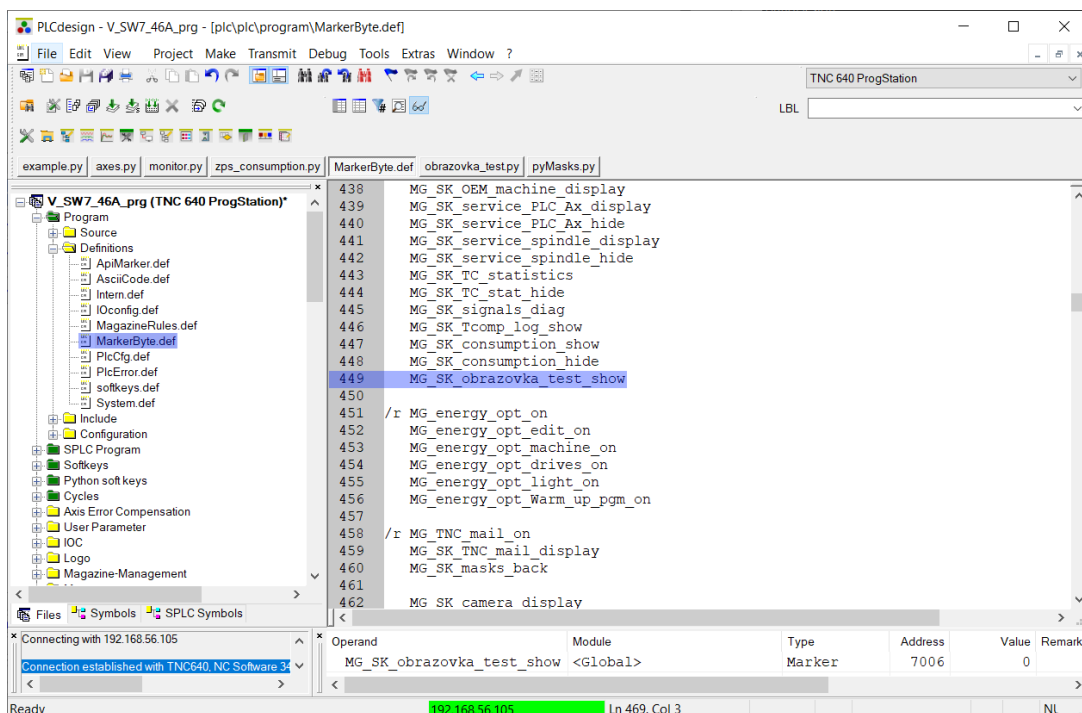


Obr. 26: SELinux: projev blokace programu výrobce stroje

Aby obrazovka nezabírala prostor řízení nebo nezakrývala důležité informace o výrobním procesu na pracovní ploše, musí být zobrazena jen ve chvíli, kdy je vyžádána a musí být možné ji skrýt, když se vyvolá jiná, nebo se přepne okno řízení. Knihovna GTK umožňuje u každého widgetu vyvolání funkcí *gtk_widget_show_all()* a *gtk_widget_hide_all()*, které budou pro tento účel využívány a budou propojeny se signálem, který bude generovaný na základě změny vybraného PLC operandu pro danou obrazovku. Pro čtení hodnot PLC operandů je k dispozici od výrobce řízení TNC knihovna *jh*, která umožňuje přístup do paměťového prostoru vyhrazenému programu PLC.

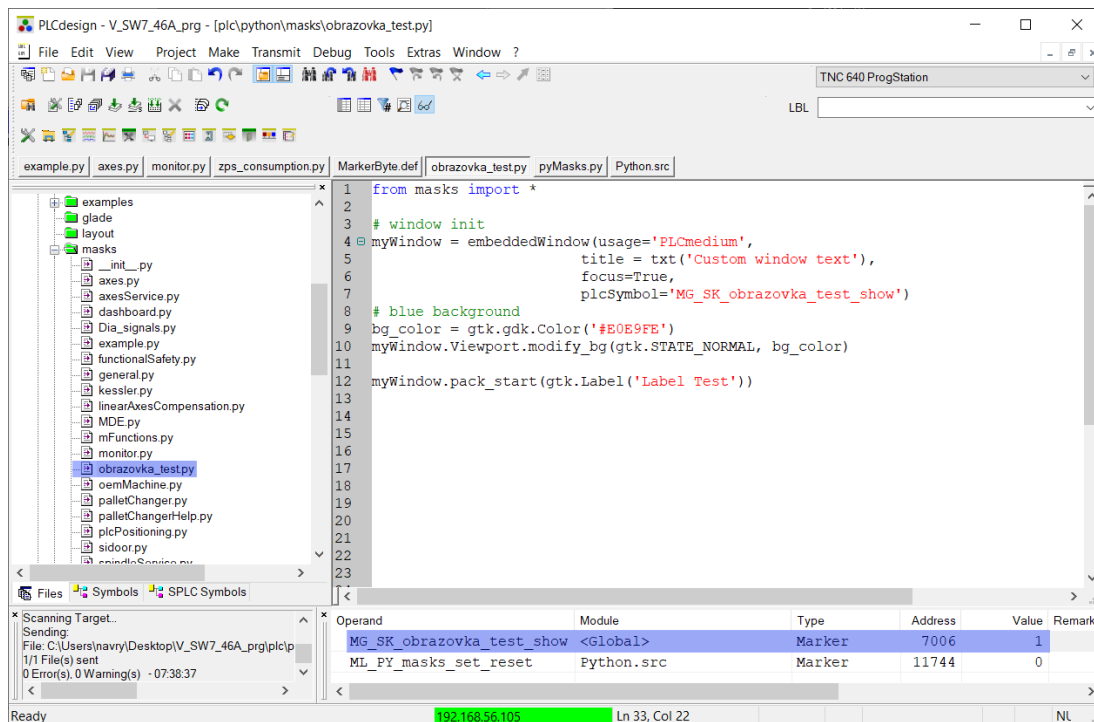
Postup vytvoření obrazovky v TNC640

Za dodržení předpokladů pro spuštění Python programů je dalším krokem určení názvu obrazovky. První testovací obrazovka se bude jmenovat *obrazovka_test* a marker (typ vytvořeného PLC operandu), který k jejímu zobrazování bude sloužit bude pojmenován *MG_SK_obrazovka_test_show* (může být pojmenován libovolně, ale dodržení této konvence usnadňuje případné ladění) a musí být ručně definován jako globální marker spolu s ostatními v souboru *<Název_projektu>/Program/Definitions/MarkerByte.def*, kde bude dopsán do bloku s markery pro grafické obrazovky.



Obr. 27: PLCdesign: Vytvoření markeru pro novou obrazovku

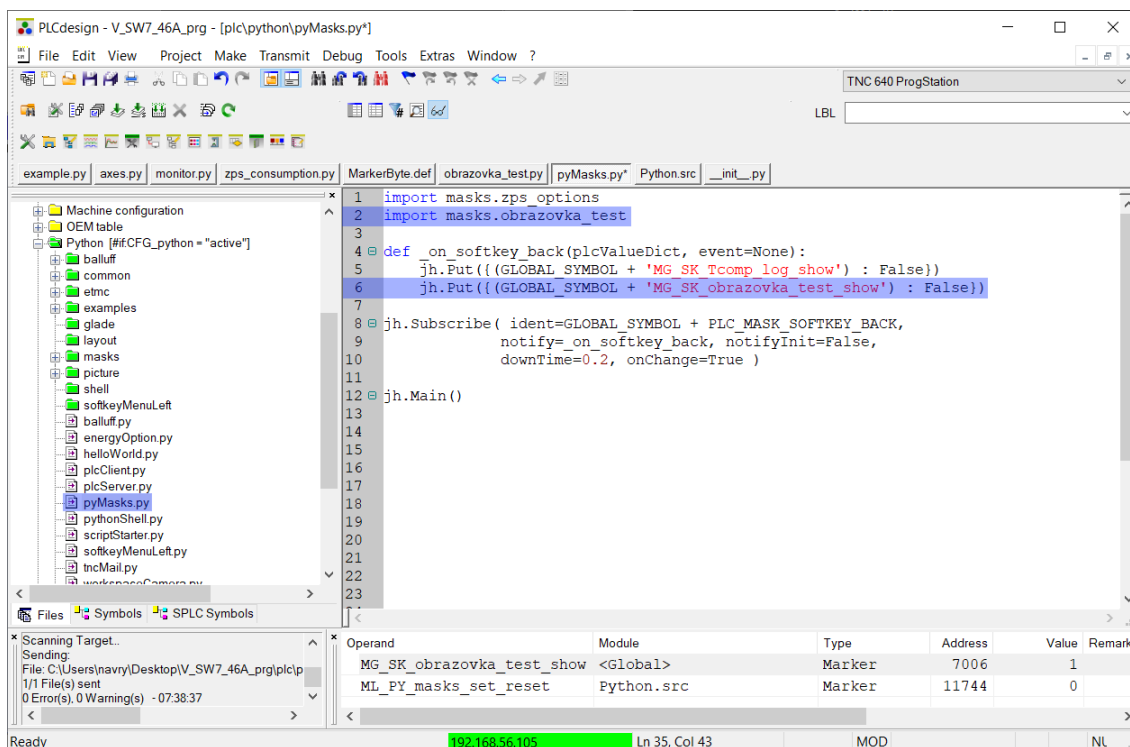
Dalším krokem je vytvoření souboru Python programu, dle zvoleného názvu pro tento případ bude jméno souboru *obrazovka_test.py* a ve stromové struktuře musí být umístěno v adresáři *<Název_Projektu>/Python/masks/* tak, jak je možno vidět na obrázku níže, kde už je do souboru vepsán i kód pro obrazovku, který umožní už z PLC obrazovku zobrazit.



Obr. 28: PLCdesign: umístění a kód první obrazovky

Každá obrazovka vyžaduje pro práci s objekty GTK přístup k GTK knihovnám, což může být provedeno importováním každé části knihovny zvlášť (tato možnost je vhodná k šetření paměti RAM), nebo, jak je na obrázku výše, využít import všech základních knihoven k tvorbě grafických obrazovek, což je praktičtější z hlediska použití, ale za cenu většího využití paměti RAM.

Posledním krokem před kompilací PLC programu a nahráním obrazovek je zařazení vytvořeného python souboru obrazovky mezi volané programy. Obrazovky se načítají hned po kompilaci a spuštění programu PLC, a aby nebylo nutné pro každou obrazovku vytvářet samostatný proces, vytváří se jen jeden, který spouští program *pyMasks.py*. Vytvořená obrazovka se tedy musí do tohoto souboru zavést tak, jak je zobrazeno níže, tedy importovat obrazovku do souboru spuštěného jako hlavní proces a vložit pod funkci *_on_softkey_back()*. Tato funkce je volána vždy, když uživatel stiskne na panelu TNC klávesu zpět a zajistí tak skrytí obrazovky. Další níže uvedená funkce *Put()* z knihovny *jh* vkládá při každém vyvolání do operandu PLC požadovanou hodnotu. Pro tento konkrétní případ to znamená, že po každém stisku klávesy zpět se do PLC markeru vytvořené obrazovky *MG_SK_obrazovka_test_show* zapíše hodnota *False*, čímž se zajistí skrytí obrazovky, je-li aktivní. Funkce, která vytváří signál pro vyvolání jiné vytvořené funkce při změně PLC operandu je pojmenována *Subscribe()* a je opět z knihovny *jh*, kde je i velice podrobně popsána.



```

1 import masks.zps.options
2 import masks.obrazovka_test
3
4 def _on_softkey_back(plcValueDict, event=None):
5     jh.Put((GLOBAL SYMBOL + 'MG_SK Tcomp log show') : False)
6     jh.Put((GLOBAL SYMBOL + 'MG_SK obrazovka_test show') : False)
7
8 jh.Subscribe( ident=GLOBAL SYMBOL + PLC_MASK_SOFTKEY_BACK,
9              notify=_on_softkey_back, notifyInit=False,
10             downTime=0.2, onChange=True )
11
12 jh.Main()
13
14
15
16
17
18
19
20
21
22
23

```

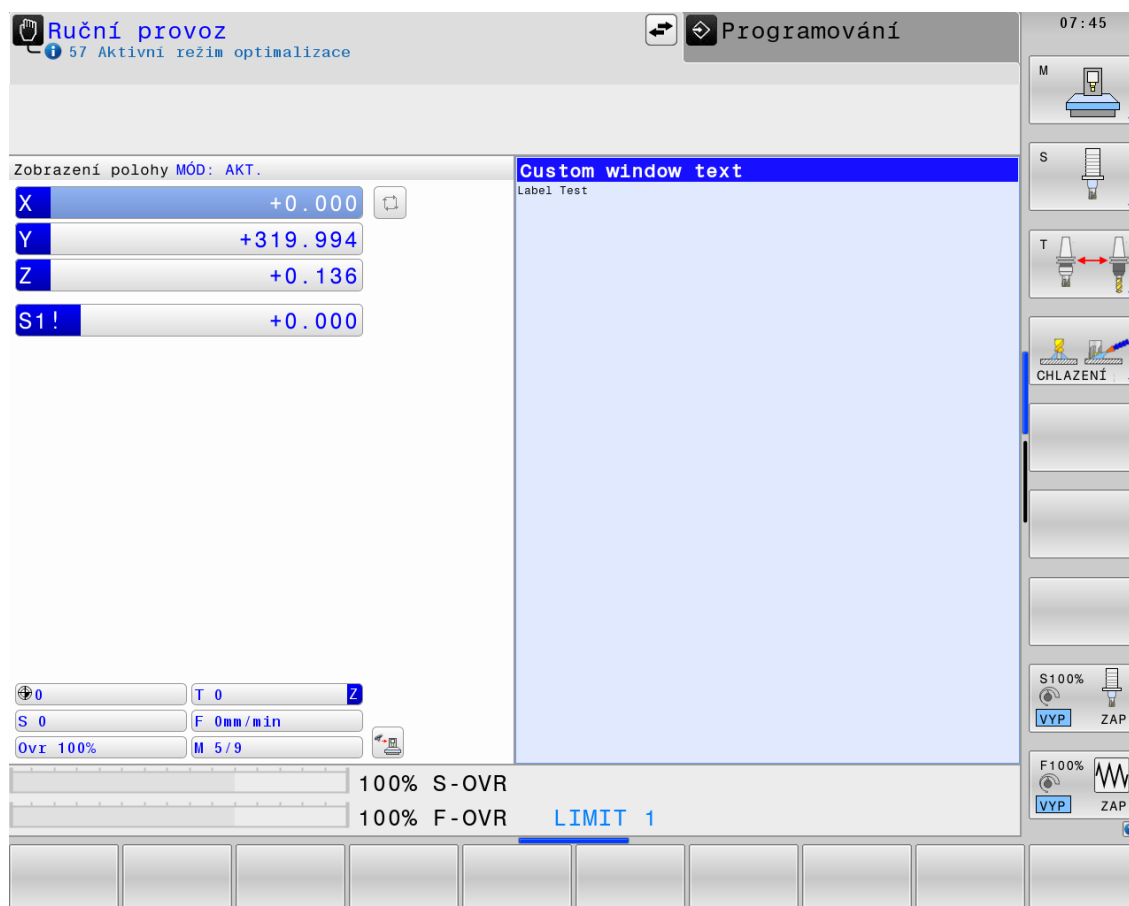
Operand	Module	Type	Address	Value	Remark
MG_SK_obrazovka_test_show	<Global>	Marker	7006	1	
ML_PY_masks_set_reset	Python.src	Marker	11744	0	

Obr. 29: PLCdesign: Hlavní program obrazovek

Posledním krokem je kompilace PLC programu kvůli vzniku markeru obrazovky a nahrání změněných Python souborů do systému TNC. Kompilaci programu PLC je možné provést v záložce *Make* kliknutím na možnost *Compile* čímž by mělo současně

dojít i k nahrání PLC programu do systému (pokud nedojde k nahrání, je nutné upravit konfiguraci projektu, nebo nahrát manuálně). Nahrávání python souborů lze provádět několika způsoby, kde nejbezpečnější je kliknutí pravým tlačítkem myši v okně vytvářeného programu, který má být odeslán do řízení a vybrat možnost *Transmit File to Control*. Soubor s obrazovkou *obrazovka_test.py* vždy půjde nahrát bez obtíží, ale soubor *pyMasks.py* nejde nikdy za běhu PLC programu nahrát do systému, jelikož nelze přepsat soubor, který je využíván procesem řízení. Je tedy nutné pro nahrání souboru ukončit řízení PLC, nebo program nahrát hned po spuštění řízení, ještě před přeložením a spuštěním programu PLC (PLC program je přeložen a spuštěn vždy po spuštění grafického prostředí po stisknutí fyzického tlačítka na klávesnici *CE*).

Po kompilaci PLC programu a nahrání upravených python programů lze znovu spustit řízení. Stisknutím tlačítka *CE* se spustí program PLC a vyvolají se procesy pro python obrazovky. Pokud jsou obrazovky bez chyb a nebude tedy přítomno žádné chybové hlášení, tak jsou python programy již v běhu na pozadí a změnou hodnoty operandu (nyní pouze v online prostředí skrze watchlist v programu *PLCdesign*), který byl vytvořen výše, by obrazovka měla být zobrazena ve stavu jako na následujícím obrázku.



Obr. 30: TNC 640: První obrazovka v systému

4.6 Tvorba hlavní diagnostické obrazovky

Po ověření funkčnosti konceptu výše je možné začít s tvorbou hlavní obrazovky, kdy je patrné, že bude využíván největší dostupný prostor pro obrazovku *PLCLarge* a na pozadí bude vykreslen obrázek půdorysu stroje. Zavádění obrazovky bude probíhat s následujícími parametry.

Tab. 5: Zvolené parametry obrazovky

Název parametru	Hodnota parametru
Název souboru obrazovky	Dia_signals.py
Název markeru PLC	MG_SK_signals_diag
Využívané okno řízení	PLCLarge
Adresář bitmap	<Název_projektu>/Python/picture/signal_diag/

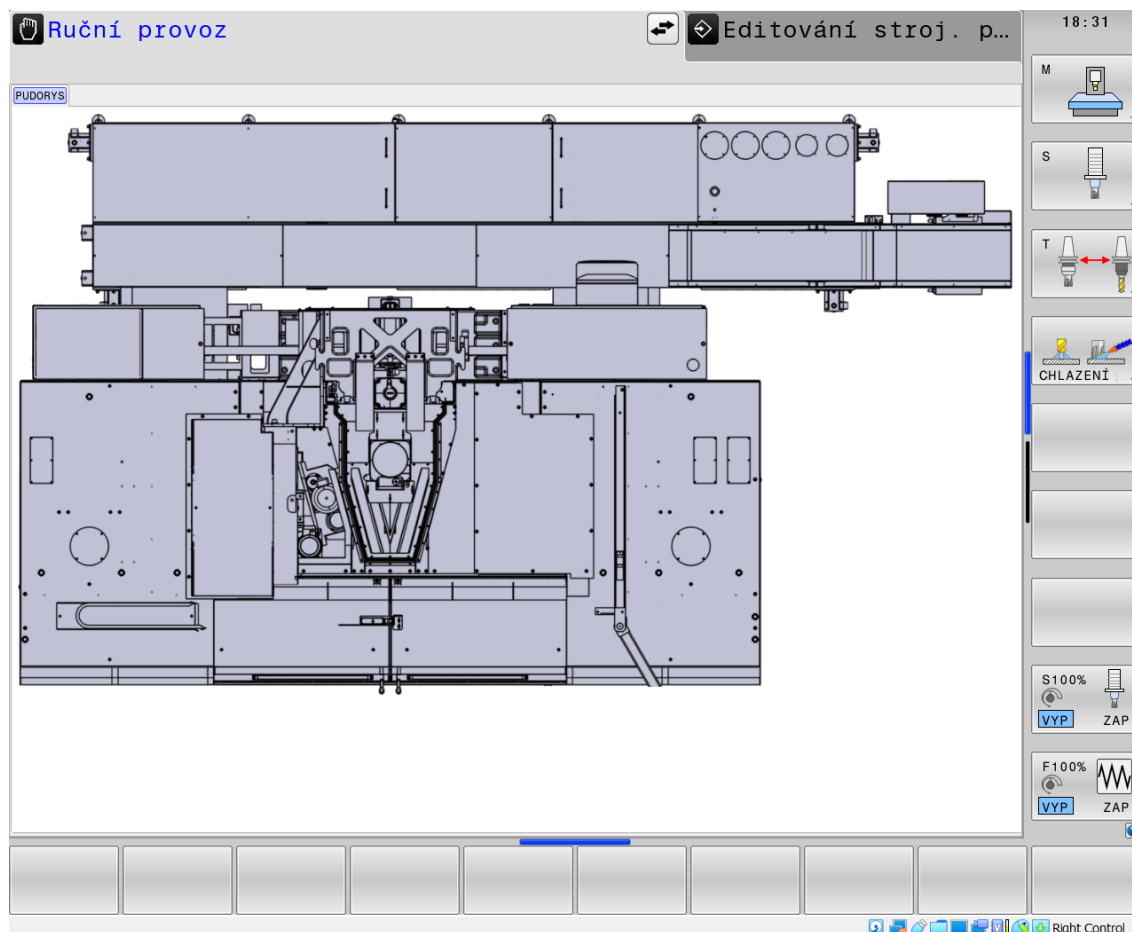
Je nutné brát na vědomí, že v momentu vložení kteréhokoliv prvku do obrazovky se vkládaný prvek vykreslí vždy přes prvky vložené dříve, což znamená, že je nutné začít obrázkem pozadí. Pro tento účel bude využit widget `gtk.Image`, který je schopen vyobrazovat bitmapovou grafiku. Nevýhodou je, že jednoduše nelze měnit velikost widgetu, jelikož ta se odvíjí od rozměrů bitmapy a pokud je bitmapa větší, než je plocha k dispozici, zobrazí se posuvníky, které zasahují do obrazovky. Testováním bylo zjištěno, že ideální je rozlišení 1138x818 pixelů pro *PLCLarge* okno. Těsně před dokončením praktické části práce byla vydána nová verze manuálu pro výrobce pracujícími s obrazovkami, kde jsou stanoveny přesné rozměry oken a dle tabulky níže lze vidět, že rozměry pro tuto obrazovku jsou 1142x849 pixelů, což v budoucnu umožňuje využití větší bitmapy za předpokladu, že widget `gtk.Image` nerozšiřuje bitmapu o tloušťku ohraničení.

Tab. 6: Rozměry předdefinovaných oken (TNC s rozlišením 1280x1024)

Název okna	rozměry okna (manuální režim / automatický režim)
PLCsmall	1142x79 / 570x79
PLCmedium	570x687 / 570x685
PLCLarge	1142x849

Vytvoření půdorysu a jeho zobrazení

Prvním krokem je tedy zobrazení nového půdorysu stroje, který již není doplněn o předznačené vstupní a výstupní členy, protože členy budou zobrazovány programově. Obrazovka navíc pro budoucí použití a možnosti rozšíření bude organizována do záložek, které knihovny GTK také dle dokumentace podporují skrze widget pod jménem `gtk.Notebook`. Náhled na vytvořenou obrazovku s první záložkou obsahující půdorys stroje MCV1680 s dopravníkem třísek je na obrázku níže.



Obr. 31: Diagnostická obrazovka: Půdorys pro vyobraze

Interpretace digitálních signálů v obrazovce

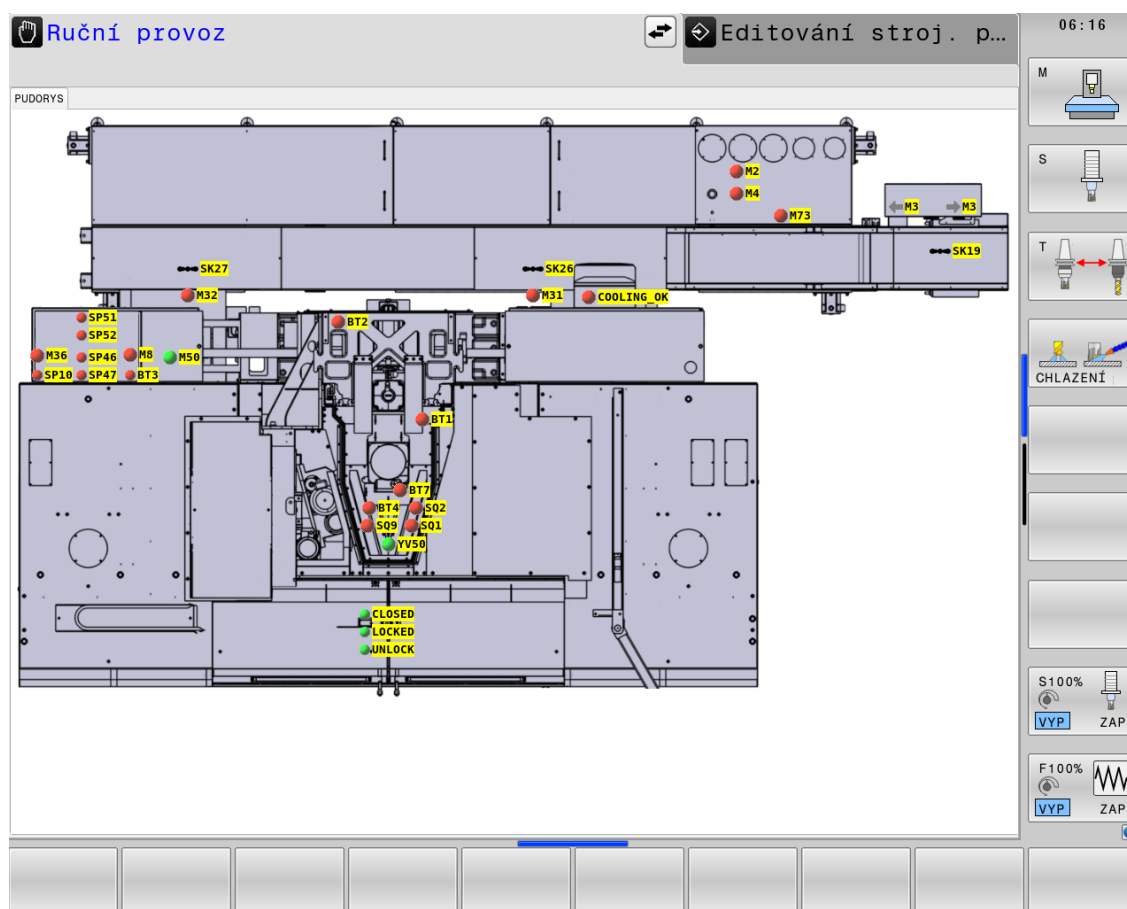
Půdorys zařízení se podařilo úspěšně zobrazit, nyní je třeba graficky interpretovat vstupní nebo výstupní signál a umístit jej na obrazovku. Prvním zvoleným řešením vyobrazení byl barevný dynamický text s názvem členu dle schématu, což se jevilo jako matoucí, protože červený text na sebe poutal pozornost jako nežádoucí stav namísto hodnoty *False*. Nevhodnost řešení utvrzoval také problém reprezentace analogových hodnot a čitelnost. Lepší možné řešení je takové, kde se pro vstupy i výstupy bude zobrazovat barevná ikona spolu s názvem prvku dle elektroschématu. Pro vyobrazení proměnných hodnot PLC operandů je pak zobrazena také ikona, ale s názvem proměnné pro usnadnění odladování programátorem.

Pokud by programátor měl na diagnostickou obrazovku pro každý signál vkládat zvlášť název a zvlášť ikonu, bylo by to nepraktické a zdlouhavé pro umísťování nových signálů. Vznikne tedy nový widget, který bude výše zmíněnou problematiku usnadňovat a jeho použití bude praktické i pro budoucí vkládání nových signálů do obrazovky. Název widgetu bude *Signal_icon* a bude vyžadovat jen parametr s adresou vstupu nebo názvem PLC operandu. Widget bude možné použít samostatně a programově umístit, ale bude také součástí funkce *add()* třídy *Diag*, která mimo jiné řeší i umístění tohoto widgetu. Další parametry widgetu zobrazeny níže jsou volitelné a vznikly ze zpětné vazby v průběhu testování. Tyto parametry jsou popsány v následující tabulce.

Tab. 7: Volitelné parametry třídy Signal_icon

parametr	popis
Sig_name	Jméno prvku (obvykle dle elektroschématu)
Img_active	Ikona zobrazená, pokud je hodnota operandu jiná, než False, nebo pokud je teplota ve specifikovaném teplotního čidla
Img_inactive	Ikona zobrazená, pokud je hodnota operandu False, nebo pokud je teplota mimo specifikovaný rozsah teplotního čidla
Description	Poznámka k danému prvku (pro použití v dalších částech práce)
Help_img	Obrázek prvku nebo jeho umístění (bude využito v dalších částech práce)
Size	Velikost ikony signálu (bylo nutno dodělat z důvodu velkého počtu signálů na malém prostoru)

Po vytvoření widgetu, doplnění funkce *add()* a použití widgetu pro některé vstupy a PLC operandy obrazovka pokročila na stav zobrazený níže, kde jsou na obrazovce umístěny klíčové vstupy a výstupy s přibližným umístěním.



Obr. 32: Diagnostická obrazovka: Použití vytvořeného widgetu

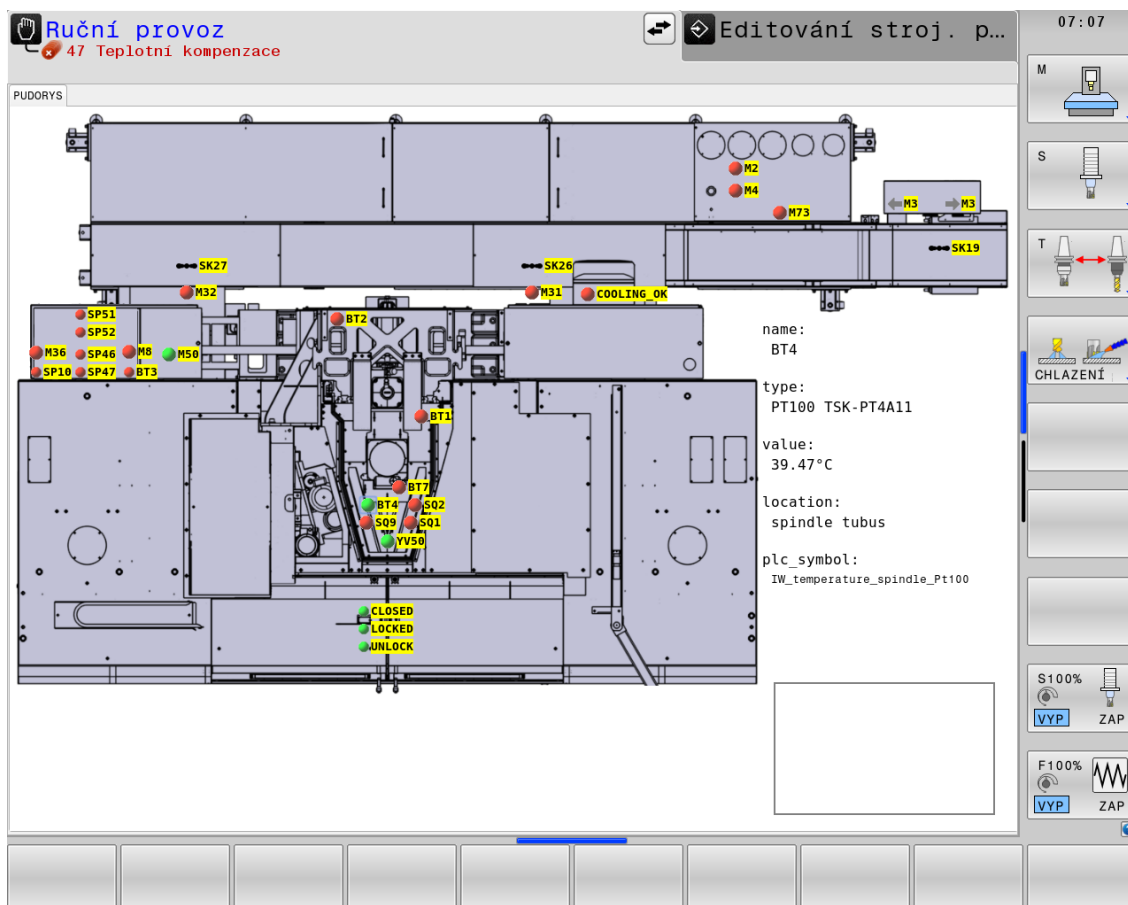
Volitelné parametry widgetu, jako jsou ikony stavů zde již byly využity za účelem zobrazení zpětných vazeb z třískových dopravníků, kde zpětnou vazbu pohybu pásu zajišťuje čidlo, které překlápí svůj stav při běžné operaci. Volitelné parametry widgetu byly použity i pro tlakové senzory z důvodu úspory místa ve skříni hydraulických prvků.

Rozšíření obrazovky pro analogové signály

Dalším krokem je zajištění reprezentace analogových hodnot. Pro tyto účely bude vytvořena tabulka ve volné části obrazovky, která bude zobrazovat dostupné informace o každém zobrazeném signálu.

Pro splnění požadavku a zajištění je podmínkou, aby ikony byly stisknutelné a aby se tabulka aktualizovala. Ikony signálů jsou tedy přepracovány na tlačítka, která neobsahují text, ale pouze bitmapy, které se mění v závislosti na stavu operandů, stejně jako u (do teď používaných) ikon, aby byl zajištěn co nejpodobnější vzhled.

Tabulka informací o signálu je dalším vytvořeným widgetem, který má schopnost kontinuálního obnovování zobrazovaných hodnot a v závislosti na typu vstupu provádět konverzi hodnot do čitelného formátu. Tabulka na základě typu čidla mění vyobrazení hodnot, jedná-li se tedy o teplotní snímač, jak je tomu na následující ukázce, převádí analogovou hodnotu na °C. Další typy chování lze definovat ve třídě tabulky. Spolu s tabulkou vznikl i prostor pro případný volitelný obrázek ke každému signálu, jak je zmíněno v parametrech výše. Rychlost obnovy každého signálu lze libovolně nastavit, ale z důvodu časové prodlevy zobrazení a zvyšování nároků na procesní čas není doporučeno na řízeních TNC přistupovat k hodnotám PLC operandům častěji, než každých 200 milisekund.



Obr. 33: Diagnostická obrazovka: Přidání tabulky detailu signálu

Další rozšíření obrazovky

Signály a tabulka vyobrazená výše byly základ, který byl v rámci této práce umožněn vložit do řízení CNC a na základě dalších zpětných vazeb od techniků uvádějících stroj do provozu, servisních techniků a programátorů PLC vznikly návrhy na zlepšení a doplnění obrazovky. Během této periody také vznikaly stroje, se kterými je dodávána i paletová výměna, bylo ideální možností pro otestování rozšiřitelnosti obrazovky doplněním záložky se signály pro paletovou výměnu. Doplnky a úpravy, které jsou provedeny v následujících částech práce jsou:

- **Univerzálnost** obrazovky pro veškeré typy strojů založeny na TNC, kde byly vyzkoušeny na dostupných obráběcích centrech (MCV1060, 1260, 1680, 2080), kde obrazovka načte základní půdorys stroje dle parametru typu stroje
- Přidána podpora pro **výměníky nástrojů**, kde obrazovka z parametrů stroje vyčte, které jsou aktivní a podle toho přidá signály a případně upraví půdorys
- Přidáno tlačítko pro **vkładání aktivního signálu do systémového watchlistu** TNC, který lze později otevřít a sledovat zvolené PLC operandy
- Přidána záložka pro **výměníky palet**, kde lze sledovat PLC operandy během výměny, stejně jako informace o kontaktování palet. Záložka také graficky zobrazuje aktuální půdorys s rozmístěním palet.

- PLC symbol aktivního signálu lze nyní v tabulce označit a ukládat do schránky pro další práci
- Přidána **podpora pro externí zařízení opcí**, jako je chlazení vnitřkem nástroje a několik typů agregátů chlazení, kde se na základě kombinace parametrů detekuje aktivní jednotka a zobrazí se box s klíčovými signály pro danou opci
- Přidán konfigurovatelný semafor jako widget pro diagnostiku semaforu na reálném stroji
- **Umožnění vyhledávání signálů ve schématech přímo v řízení**
- Zobrazení data, kdy byl naposled nahrán do stroje program PLC (slouží pro ověření, zda je verze PLC v systému shodná a kterou má zákazník i programátor pro případ, že by zákazník prováděl změny PLC bez vědomí výrobce a poté chtěl nahrát zálohu, která vznikla při přejímce, což by mohlo mít nežádoucí dopad)

Doplnění obrazovky o možnost hledání v elektroschématu

Podpora hledání v elektrotechnických schématech představuje asi nejkomplexnější úkol zpracovaný autorem práce. Cílem bylo zobrazovat schémata na stroji, pokud možno jednoduchým a efektivním způsobem.

První překážkou je fakt, že ve schématech je označována přímo adresa vstupu, nikoliv název prvku, jak je tomu v obrazovce. Bylo tedy nutné najít způsob, jak zjistit adresu vstupu podle požadovaného operandu. Za tímto účelem byl využit soubor *IOconfig.def*, který je produktem softwaru *IOconfig* a obsahuje názvy operandů spolu s adresami vstupů. Byla tedy vytvořena funkce, která uvedený konfigurační soubor prohledá a podle názvu PLC operandu vrátí hodnotu s adresou, která je použita v elektroschématu.

Z bezpečnostních důvodů nelze na hlavní obrazovce řízení zobrazit jiné okno (v tomto případě prohlížeč PDF). Prakticky zobrazit lze, ale při jakékoliv akci se přesune do pozadí, jelikož aplikace *NCcontrol* vyžaduje běh na popředí obrazovky. To znamená, že obrazovka s dokumentem PDF zůstane otevřena na pozadí a vždy, když se ji uživatel pokusí otevřít, vznikne nová instance programu, čímž započne zahlcování operační paměti, jelikož prohlížeč nelze jednoduše zavřít.

K řešení pomohl průzkum způsobu funkčnosti oken v řízení TNC, kdy každá obrazovka (hlavní řízení, okno editování programu, okno editování parametrů) je v podstatě vždy další pracovní plochou, čímž je zajištěno, že nedojde k překrývání, protože nedochází k přepínání oken, ale k přepínání celých pracovních ploch, jak linuxové i jiné desktopové systémy podporují. Bylo tedy třeba zjistit, jak a jestli vůbec lze v jazyce python automaticky spustit systémový PDF prohlížeč na další pracovní ploše, automaticky přepnout na požadovanou plochu a jak se vrátit zpět po dokončení prohlížení.

Po zkoumání bylo zjištěno, že PDF prohlížeč používaný v operačním systému HEROS nese název *evince* (prohlížeč GNOME korporace) a podporuje spouštění s parametry, které umožňují otevření PDF v režimu celé obrazovky. V dokumentaci sice nebyla nalezena možnost vyhledávání v dokumentu na základě textu, ale při vypsání nápovědy pro program bylo zjištěno, že je možné použít parametr „*--find*“. Parametr tedy byl použit při otevření a načtení dokumentu. Nelze ale bohužel otevřít dokument na jiné pracovní ploše než na právě aktivní, což znamená, že je nutné změnit pracovní plochu ještě před otevřením souboru.

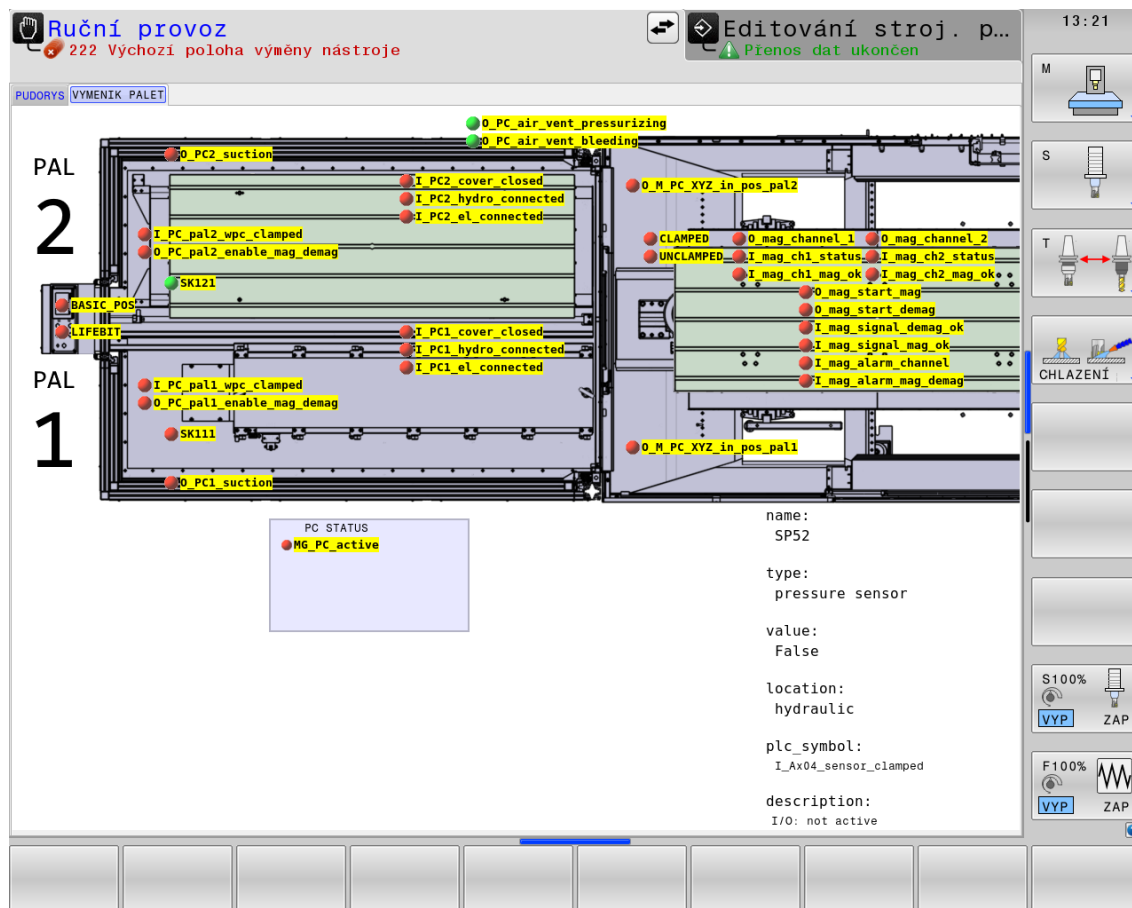
Testování běžných metod se ukázalo za nefunkční, nebylo možné tedy použít žádné systémové příkazy ani klávesové zkratky volané z programů python. Nebylo umožněno ani použití modulů pro python, jelikož většina z nich pro práci vyžaduje instalace, které ale v HEROS systému nejsou umožněny ani výrobcům stroje.

Během dalšího zkoumání možností, při procházení všech dostupných spustitelných programů z terminálu, byl nalezen v systému HEROS software *xdotool* a dle popisu se jedná o open-source prostředí příkazové řádky, které je určeno k testování klávesových zkratk. Při testování fungoval korektně pro většinu klávesových zkratk. Nefungovaly však výchozí zkratky používané v systémech GNOME ani XFCE, ale protože systém HEROS, jak je výše uvedeno, využívá grafické rozhraní GNOME, tak by měl být dostupný software pro přenastavení klávesových zkratk, který se volá z příkazové řádky příkazem *xfce4-keyboard-shortcuts*, což se ale ukázalo také jako nefunkční. Zajímavostí je, že všechny používané nástroje pro xfce4 desktopové prostředí, jako je *xfce4-mouse-settings*, *xfce4-display-setting* nebo i *xfce4-session-settings* jsou dostupné, jen nastavení klávesových zkratk chybí (nejspíš je využívá i jiný software Heidenhain a jedná se o způsob, jak zamezit změně těchto zkratk). Nicméně v této části práce není třeba nastavení měnit, jen zobrazit, aby bylo možné zkratky použít v programu. Dle dokumentace XFCE je možné zobrazit přehled nastavených hodnot, i když jsou uzamčeny proti přepsání v softwaru *xfce-settings-editor*. V tomto softwaru byly zjištěny klávesové zkratky pro přepínání pracovních ploch. Zkratky byly implementovány do části programu simulující stisk kombinací kláves prostřednictvím externího softwaru *xdotool*. Část programu zajistí, že bude kombinace kláves simulována těsně před otevřením dokumentu PDF, kvůli kterému je celý tento proces nutný. Řešení se osvědčilo a je stabilní. Při zavření okna PDF na pracovní ploše si systém TNC sám vyžádá okamžité přepnutí na pracovní obrazovku, ve které se nachází.

Doplnění záložky pro výměníky palet

Při tvorbě práce, jak je zmíněno na začátku kapitoly, vznikal stroj, který měl být vybaven paletovou výměnou. Při této příležitosti mohla být vytvořena další praktická záložka diagnostické obrazovky pro paletové systémy. Tato záložka bude v diagnostické obrazovce dostupná vždy, když bude zvolený parametr stroje pro přítomnost výměníku palet aktivní. Obrazovka v závislosti na koncových čidlech palet mění bitmapu půdorysu tak, aby představovala aktuální stav polohy palet. Po zadání hesla je obrazovka schopna vyobrazit celý adresní prostor i s hodnotami z protokolu PROFINET, po kterém stroj s výměníkem komunikují.

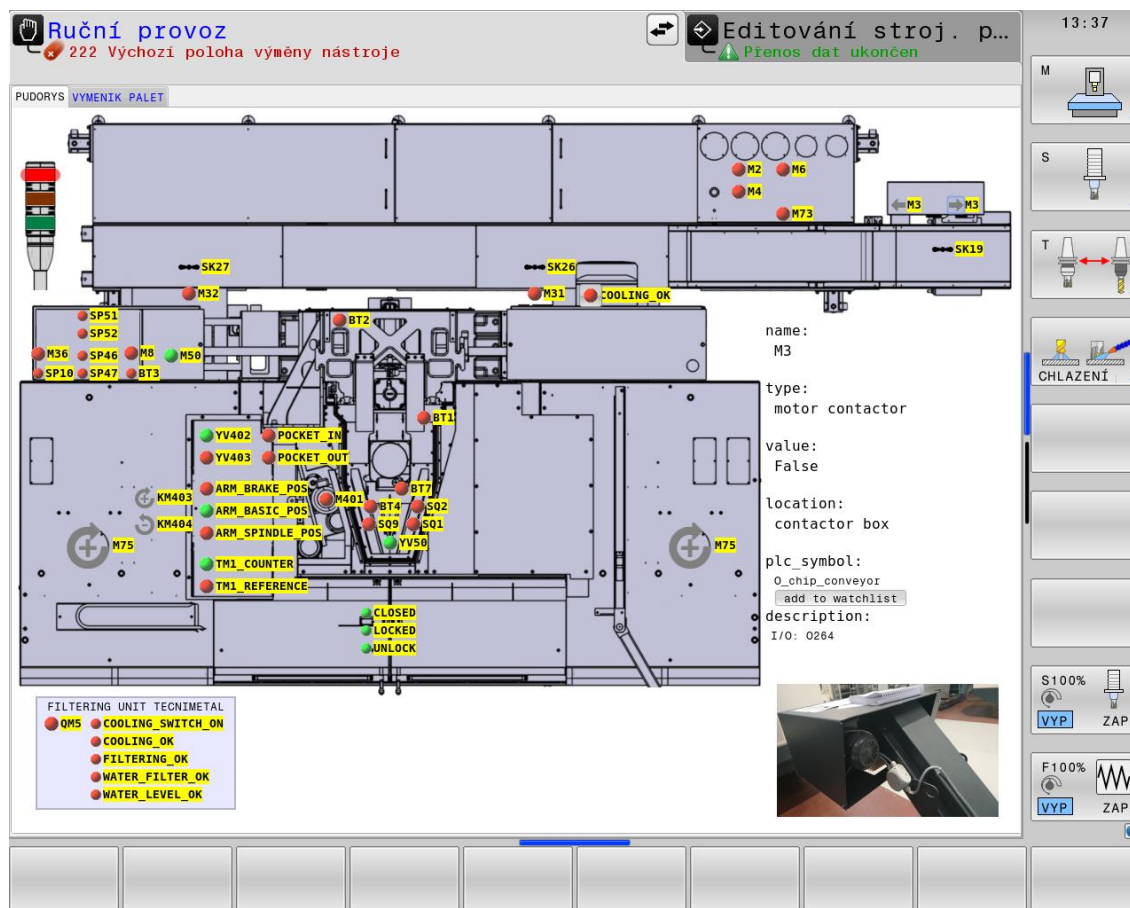
Tabulka detailu signálu je zobrazena v pravém spodním rohu, a protože tabulku je potřeba mít vždy k dispozici, tak byla upravena tak, že se vždy přesune do předem definované pozice v aktivní záložce, je tedy tím samým widgetem, který je použit i v první záložce s půdorysem stroje. Tato tabulka totiž nelze kopírovat kvůli struktuře knihoven GTK, která neumožňuje, aby jeden widget měl současně více nadřazených widgetů.



Obr. 34: Diagnostická obrazovka: Záložka pro výměníky palet

Dokončení hlavní diagnostické obrazovky

Finální diagnostická obrazovka by měla sloužit pro přehled vstupních a výstupních signálů, PLC operandů, stavu stroje, usnadňovat ladění programu PLC i zkracovat následné servisní zásahy. Obrazovka je možné libovolně přizpůsobit a adaptovat na kterékoliv řízení, jelikož obrazovka a její části jsou vždy vázány k operandům PLC. Náhled na okna finální diagnostické obrazovky jsou zobrazeny níže.



Obr. 35: Diagnostická obrazovka: Příklad finálního rozhraní pro stroj

Z výše uvedených obrázků je patrné, že diagnostickou obrazovku je možné prakticky libovolně upravovat, vkládat vlastní dynamické widgety, jako je semafor na obrázku výše, který lze při vkládání konfigurovat skrze argumenty. Tyto widgety mohou být vykreslovány kdekoli na volném místě obrazovky, nebo mohou překrývat již stávající.

Vznik knihovny funkcí pro budoucí obrazovky

Během tvorby hlavního předmětu této práce vznikla řada widgetů a funkcí včetně takových, které nebyly ve finální verzi použity. Bylo vytvořeno také mnoho widgetů, které jsou vzhledově či funkčně podobné se stávajícími dodávanými firmou Heidenhain, ale bylo je nutné pro použití v obrazovkách stávající widgety upravit, tak, aby nebyly ovlivněny již stávající systémové. Veškeré tyto funkce a widgety byly umístěny do samostatného souboru a je možné je využívat i v jiných obrazovkách, nebo je doplnit o vlastní.

Možnosti rozšíření

Každý výrobce obráběcích center, který využívá řízení TNC od firmy HEIDENHAIN má k dispozici manuál pro práci s obrazovkami v jazyce python s jednoduchými příklady použití. Pokud programátor pracuje s linuxovými systémy a případně některými grafickými knihovnami jako GTK, QT nebo jiné, může obrazovky libovolně adaptovat.

4.7 Vznik dalších obrazovek a rozšíření

Během tvorby hlavní diagnostické obrazovky, kdy bylo mnoho možností programovat PLC, účastnit se prvních spuštění a ožívování obráběcích center, vznikaly další nápady na obrazovky, které by mohly usnadnit vybrané činnosti. Některé z těchto pomocných obrazovek jsou popsány níže.

Program pro záznam historie PLC operandů

Systémy TNC umožňují samozřejmě logování proměnných, k čemuž je používána výše zmiňovaná aplikace TNCscope. Aplikace umožňuje vybrat rychlost vzorkování signálů a pomocí počítače pak zaznamenávat stavy libovolných operandů. Takový záznam na PC je ale spíše vhodný pro krátkodobé logování, protože dlouhodobý záznam vyžaduje ke své funkci PC a aktivní aplikaci TNCscope, což obnáší přípravu a někdy nelze ani přibližně detekovat, kdy problém nastává. Typický případ, kdy dojde ke stavu, který přeruší automatický režim. Operátor pak zpravidla provede smazání chyb, nebo restart řízení, aby mohl v práci pokračovat a následně se vytvoří report o problému na stroji. Tento poměrně častý případ se později velmi špatně analyzuje, protože nejsou k dispozici informace o tom, v jaké části a ve kterém cyklu chyba nastala.

Software by měl zjednodušit proces zpětné analýzy stavu řízení TNC tak, že u vybraných PLC operandů bude se zvolenou časovou periodou (výchozí je 1 sekunda) ukládat jejich stavy do souborů.

Program využívá k ukládání a formátování záznamů vestavěný modul jazyka python jménem logging, který umí ukládat zprávy třeba i na základě závažnosti. V tomto příkladu bude ale sloužit pro zpracování všech zpráv. Modul *logging* je doplněn o rozšíření *RotatingFileHandler*, které je využito pro kontrolu a odstranění starých záznamů, čímž předchází vzniku enormně velkých souborů záznamu. Program vždy při prvním spuštění vytvoří požadavky pro PLC, aby byly pravidelně posílány hodnoty vybraných operandů i s jejich názvy a pokud se tak nestane do určitého časového intervalu, jsou prohlášeny za neaktivní a záznam hodnot začne.

Záznamy jsou ukládány tak, aby byly kompatibilní se softwarem TNCscope, je tedy možné je jednoduše importovat a není třeba dalšího softwaru na jejich analýzu. Pro přípravu souboru záznamů stačí manuálně nebo přiloženým jednoduchým programem sjednotit vytvořené soubory. Náhled na software TNCscope, ve kterém je otevřen dlouhodobý záznam, který byl pořízen vytvořeným softwarem, je zobrazen na obrázku níže.



Obr. 36: TNCscope: import záznamu z vytvořeného softwaru

Obrazovka pro parametrizaci zařízení SIMOTION

Obráběcí centrum je poměrně komplexní zařízení, i přesto ale systém TNC nedokáže jednoduše pokrýt některé části stroje. Konkrétním příkladem je vysokokapacitní rotační zásobník, který slouží pro rychlou výměnu nástrojů (výměna nástroje za nástroj trvá pod 3,5 sekundy). Když zásobník má kapacitu 60 nástrojů a do každé pozice je možné vložit nástroj s váhou 15kg, je poměrně složité zásobník uřídit. K tomuto účelu se využívá řízení SIMOTION, které s řízením TNC komunikuje pomocí protokolu PROFINET.

V případě, kdy se zákazník rozhodne zásadně změnit konfiguraci nástrojů v zásobníku, může systém SIMOTION vyžadovat změnu některých parametrů řízení osy. Pro takový zásah bývá nutno výjezdu osoby, která má patřičný software, aktuální projekt a znalosti dělat zásahy do parametrů řízení.

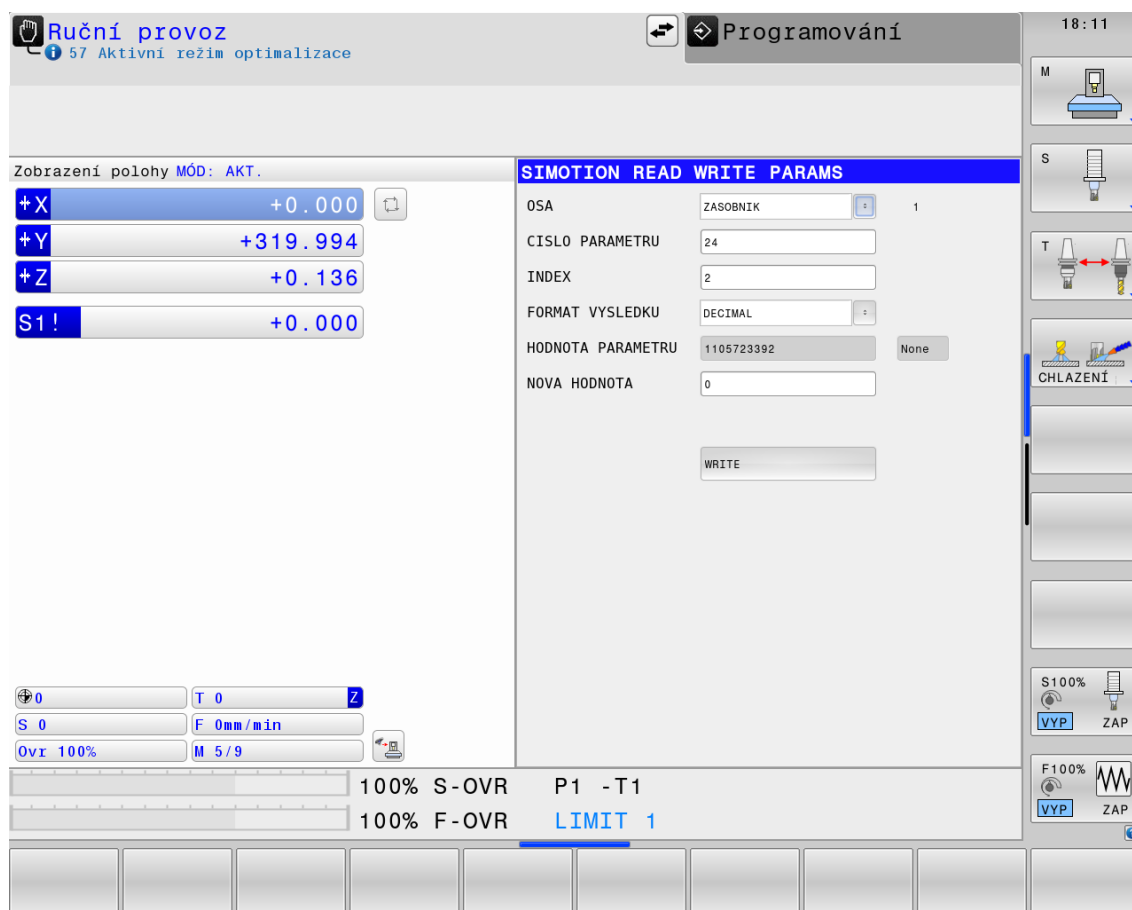
Obrazovka, která by umožnila parametrizování řízení SIMOTION přímo skrze řízení HEIDENHAIN by mohla pomoci časově a ušetřit cestování. K řízení TNC se lze většinou připojit skrze software TeleService, který disponuje i VNC připojením k obrazovce TNC. Tím by bylo možné technikem pomocí vzdálené správy a této obrazovky měnit potřebné parametry SIMOTION.

Při aplikaci obrazovky je nutné mezi programátory systémů TNC a SIMOTION stanovit v komunikaci PROFINET adresní rozsah, který bude obsahovat prostor na hodnoty parametrů, signály pro zahájení přepsání stávajících parametrů a signály na potvrzení přepsání těchto parametrů.

Je nutné brát na vědomí, že SIMOTION podporuje datové typy označované DWORD (double word), ale na TNC nebyl tento typ předchozími verzemi podporován, proto ještě nebyl tento typ v řízení použit. V systému TNC je tedy nutné pracovat s dvěma PLC operandy typu WORD. Při dalších adaptacích této obrazovky je možné přizpůsobit komunikace, parametry i datové typy.

Jedním z řešených problémů v této obrazovce je komunikace. Problém spočívá při přechodu u datového typu DINT přes hodnotu 0. K rozpoznání záporné hodnoty u těchto typů se využívá bitu umístěného vlevo. Bohužel v obrazovce TNC je přístup vždy pouze k typu WORD, tedy jedné polovině hodnoty, která je po vyčtení vždy reprezentována jako INT. Pokud tedy v datovém typu DINT odesílané SIMOTION řízením bit vlevo nabývá hodnoty 1, u všech bitů ve druhém WORDu TNC je nutné změnit hodnotu. Proto musel být vytvořen postup, jak se s touto zvláštní konverzí vypořádat, tedy jak pracovat s oběma INT na základě toho, jaké decimální hodnoty každý z nich představuje, dle jejich stavu převést hodnoty na binární, spojit je a zobrazit jako čitelnou hodnotu. Ta samá aplikace platí reverzně pro odeslání hodnoty parametru do externího řízení pro protokolu PROFINET.

Maska umožňuje měnit kterékoliv parametry řízení SIMOTION, což zahrnuje parametry zadávané různými datovými typy a formáty (FLOAT, HEXADECIMAL, DECIMAL), což platí i pro jejich zobrazení. Náhled této obrazovky lze vidět níže. Obrazovka lze libovolně přizpůsobit pro účely změny parametrů externích zařízení. Podle toho, který typ zobrazení je pro daný parametr v obrazovce vybrán (DECIMAL, HEXADECIMAL nebo FLOAT) je pak nutné zadávat ve stejném formátu i nový.

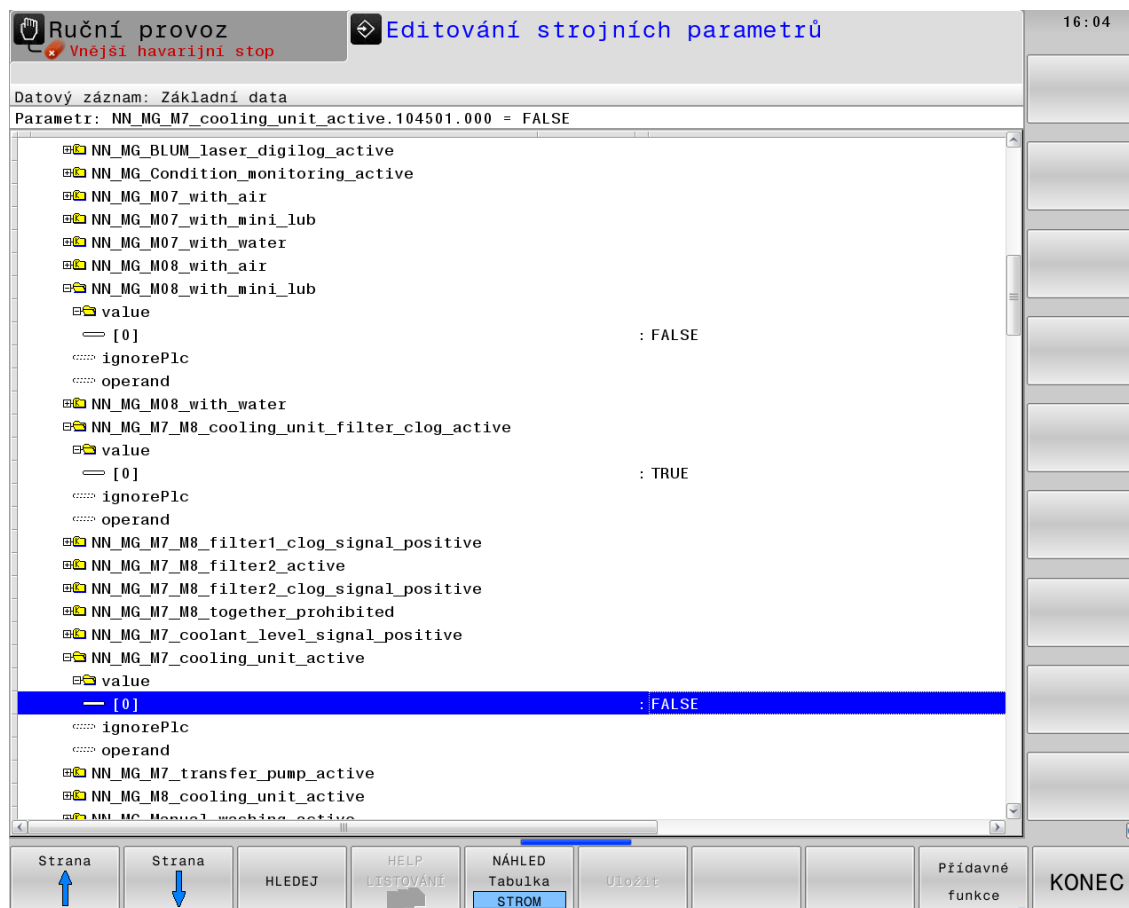


Obr. 37: Obrazovka pro parametrizaci řízení SIMOTION skrze PROFINET

4.8 Návrhy pro budoucí obrazovky

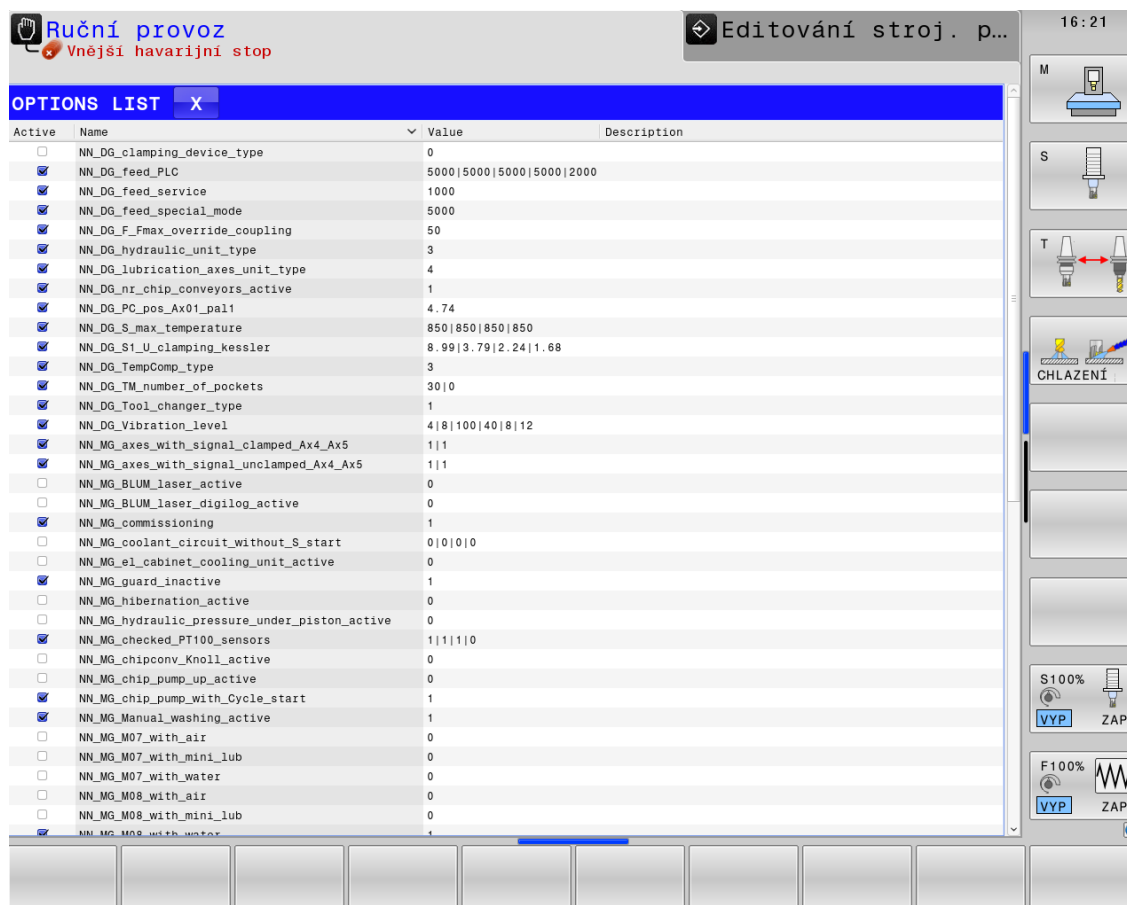
Samotný nápad na obrazovku zpravidla vzniká, když se vyskytne nový, nebo opakovaný problém, na kterém se tráví více času, než by bylo nutné, jak tomu bylo i u obrazovek vytvořených výše. S výskytem nových problémů je pak na úvaze, jestli by vhodná obrazovka v systému TNC mohla problému předejít, nebo alespoň ušetřit čas jeho řešení. Počet obrazovek není prakticky omezen, je omezena pouze operační paměť, která pro procesy python může být až 128MB. Pro představu, pokud by na jednom řízení byly aktivní všechny obrazovky, které autor vytvořil v této práci, využití operační paměti by se pohybovalo dle testování okolo 8MB, z čehož jde usoudit, že je zde prostor pro mnohem náročnější aplikace. Samozřejmě je nutno počítat i s využitím procesorového času, který s dalšími aplikacemi bude narůstat a proto není vhodné opakovaně rychle vyčítat či zapisovat hodnoty operandů PLC. Obecně by mělo být zajištěno, aby byl systém na vytvořených obrazovkách v automatickém režimu nezávislý, tedy umožnit při zhroucení procesu Python stále používat stroj v automatickém cyklu.

Jedna z obrazovek, na které autor práce začal pracovat, by měla sloužit pro přehledné zobrazení důležitých parametrů stroje. Dle aktuálního stavu jsou všechny parametry zobrazeny ve stromových strukturách, kde každým parametrem je struktura. Postupné procházení většího počtu parametrů vyžaduje procházení stromové struktury, kde k samotným hodnotám se uživatel dostane až po rozkliknutí stromu parametru, kde je nutné dále rozkliknout položku *value* a pod touto položkou je teprve možné otevřít pozici hodnoty, kterou chce měnit. Tento aktuální stav je zobrazen na obrázku níže.



Obr. 38: TNC 640: Aktuální editování parametrů stroje

Autor navrhuje jako řešení obrazovku, která by umožňovala definovat libovolné, nebo nejčastěji používané parametry. Hodnoty těchto parametrů by byly přehledně zobrazeny v tabulce. Parametry by bylo umožněno řadit podle hodnot nebo jiných skupin. Dalším usnadněním by mohla být možnost měnit jejich hodnoty přímo v seznamu (samozřejmě pod patřičným heslem). Aktuální rozpracovaný koncept (nepodporuje některé struktury parametrů a není doladěno řazení a rozložení) lze vidět na obrázku níže, kde je patrné, že na jednu plochu se vejde několikanásobek hodnot parametrů a pokud parametr obsahuje více hodnot, jsou zobrazeny také.



Obr. 39: Koncept obrazovky pro zobrazení vybraných parametrů

Kdyby bylo možné parametry i měnit, zcela jistě by se zkrátil čas uvádění stroje do provozu, kde je během výstupní kontroly nutno většinou zkontrolovat hodnoty důležitých parametrů.

Jistě existuje několik dalších možností, jak podobnými obrazovkami zkrátit nebo usnadnit některé z činností, kde uvedený koncept obrazovky výše by mohl být jednou z nich.

5 ZÁVĚR

Cílem práce bylo vytvoření diagnostické obrazovky pro řídicí systémy TNC firmy Heidenhain. V první části práce byla provedena rešerše používaných řídicích systémů pro vertikální obráběcí centra, kde byly shrnuty dostupné řady řízení firem Heidenhain a Siemens. Stručně byly popsány metody programování PLC dle normy IEC-EN-61131-3. Kapitola přiblížila jazyk pro programování PLC Heidenhain a byla zakončena popisem obsahujícím seznam všech softwarových prostředků nutných k programování i ladění systémů firmy Heidenhain.

Následoval popis samotného řídicího systému TNC640, jehož základem je linuxový operační systém reálného času HEROS, na kterém jsou programy PLC i grafické rozhraní provozovány.

Další část práce popisuje aktuální stav prostředků řízení se zaměřením na části určené k jeho diagnostice. Z hlediska pohybu v grafickém prostředí systému TNC byly popsány úkony technických pracovníků, které musí být absolvovány, aby bylo možné obráběcí centrum uvést do provozu či servisovat. Závěr této kapitoly upozorňuje na časovou náročnost a nepohodlnost vybraných stávajících procesů společně s návrhy pro jejich zlepšení.

Hlavní část práce se zabývá konkrétním návrhem a programováním samotné obrazovky doprovázené o popis problematiky jejího vzniku. Produktem je funkční diagnostická obrazovka s půdorysem přizpůsobujícím se konfiguraci stroje, na kterém jsou zobrazeny dynamicky měnící se ikony stavů vstupů a výstupů vybraných prvků stroje, a tabulky pro zobrazení případných analogových hodnot. Obrazovka dokáže dle typu čidla převádět analogové hodnoty na jednoduché, člověkem čitelné a umožňuje vytváření seznamů operandů pro pozorování. V obrazovce je možné využívat funkci, která po výběru pozorovaného prvku zajistí otevření elektroschématu přímo v systému a vyhledá list s připojením tohoto prvku do karty PLC. Vytvořená obrazovka je variabilní, umožňuje vkládání nových prvků a je rozšiřitelná o další typy strojů.

Rozšířením hlavní části práce je kapitola, kde jsou vytvořeny a popsány další obrazovky, které mají usnadňovat a urychlovat specifické činnosti a umožnit zpětnou analýzu událostí obráběcího centra v softwaru TNCscope.

Aby praktická část této práce mohla být užívána a dále rozšiřována, zdrojové kódy obrazovek jsou zpřístupněny pod open-source licencí MIT online přístupnou službou Github.

Obrazovky byly několik měsíců v rámci testování používány na reálných obráběcích centrech, a dle zpětné vazby během této doby dokázaly ušetřit několik servisních výjezdů, rychle odhalit vadná čidla, nalézt příčinu občasných nedokončení výměny nástroje a zkrátit čas oživení obráběcího centra.

6 SEZNAM POUŽITÉ LITERATURY

- [1] ZHANG, Peng. *Advanced Industrial Control Technology. Advanced Industrial Control Technology: Industrial control engineering* [online]. William Andrew Publishing, 2010, s. 41-70 [cit. 2022-05-20]. ISBN 9781437778076. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B9781437778076100026>
- [2] Siemens AG. *SINUMERIK 808 Equipment for Machine Tools* [online]. In: . 2018, [cit. 2022-05-20]. Dostupné z: <https://assets.new.siemens.com/siemens/assets/api/uuid:4646f7be-f0bf-4e4b-abca-97e87a3846c2/motion-control-catalog-nc81-1-sinumerik-808-complete-us-edition-.pdf>
- [3] Siemens AG. *SINUMERIK 828: Equipment for Machine Tools* [online]. In: . 2022 [cit. 2022-05-20]. Dostupné z: https://cache.industry.siemens.com/dl/files/977/109746977/att_1090185/v1/motion-control-NC82-complete-English-2022-01.pdf
- [4] Siemens AG. *SINUMERIK 840: Equipment for Machine Tools* [online]. In: . 2019 [cit. 2022-05-20]. Dostupné z: https://cache.industry.siemens.com/dl/files/845/109746845/att_984913/v1/motion-control-NC62-complete-English-2019.pdf
- [5] Siemens AG. *SINUMERIK ONE: Equipment for Machine Tools Function overview* [online]. In: . 2021 [cit. 2022-05-20]. Dostupné z: https://cache.industry.siemens.com/dl/files/631/109776631/att_1082286/v1/motion-control-NC63-complete-english-2021-10-glossary.pdf
- [6] Siemens Industry Inc. *SINUMERIK: INTELLIGENT SOLUTIONS FOR MACHINE TOOLS* [online]. In: . 2021 [cit. 2022-05-20]. Dostupné z: <https://assets.new.siemens.com/siemens/assets/api/uuid:4b2bdc67-5f58-40e4-9e24-64e4d3844f3e/dimc-b10079-00-7600sinumerikdachschrift-72.pdf>
- [7] DR. JOHANNES HEIDENHAIN GmbH. *TNC 128: Information for the Machine Tool Builder* [online]. In: . 2019 [cit. 2022-05-20]. Dostupné z: https://www.heidenhain.com/fileadmin/pdf/en/01_Products/Prospekte/PR_TNC128_OEM_ID827228_en.pdf
- [8] DR. JOHANNES HEIDENHAIN GmbH. *TNC 320: Information for the Machine Tool Builder* [online]. In: . 2019 [cit. 2022-05-20]. Dostupné z: https://www.heidenhain.com/fileadmin/pdf/en/01_Products/Prospekte/PR_TNC320_OEM_ID1113513_en.pdf
- [9] DR. JOHANNES HEIDENHAIN GmbH. *TNC 620: Information for the Machine Tool Builder* [online]. In: . 2021 [cit. 2022-05-20]. Dostupné z: https://www.heidenhain.com/fileadmin/pdf/en/01_Products/Prospekte/PR_TNC620_HSCI_OEM_ID895922_en.pdf
- [10] DR. JOHANNES HEIDENHAIN GmbH. *TNC 640 HSCI: Information for the Machine Tool Builder* [online]. In: . 2020 [cit. 2022-05-20]. Dostupné z: https://www.heidenhain.com/fileadmin/pdf/en/01_Products/Prospekte/PR_TNC640_HSCI_GEN3_OEM_ID1303179_en.pdf
- [11] OPC Foundation, *OPC UA Online Reference: General information to CNC systems and OPC UA* [online]. [cit. 2022-05-20]. Dostupné z: <https://reference.opcfoundation.org/v104/CNC/v100/docs/4.1.1/>
- [12] TIEGELKAMP, Michael, Karl Heinz a Michael. *IEC 61131-3: Programming Industrial Automation Systems* [online]. In: . 2010 [cit. 2022-05-20]. Dostupné z: doi:10.1007/978-3-642-12015-2

- [13] *PLC Programming Languages – Short Overview* [online]. 2019 [cit. 2022-05-20]. Dostupné z: https://isd-soft.com/tech_blog/plc-programming-languages-short-overview/
- [14] TECO. *Programování PLC podle normy IEC 61 131-3* [online]. In: . [cit. 2022-05-20]. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00321_01_mosaic_progiec_cz
- [15] BOLTON, W. *Programmable Logic Controllers* [online]. 4rd. Newnes, 2006 [cit. 2022-05-20]. ISBN 9780750681124. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B9780750681124500188>
- [16] URBAN, Luboš. *Programování PLC podle normy IEC EN 61131-3* [online]. In: . 2005 [cit. 2022-05-20]. Dostupné z: <https://coptel.cz/mod/page/view.php?id=6737>. Automa: časopis pro automatizační techniku.
- [17] DR. JOHANNES HEIDENHAIN GMBH. *TNC 640 Technical Manual: NC Software 34059x-11*. 83301 Traunreut, Germany, 2021.
- [18] *Sinumerik 808D Advanced* [online]. In: . [cit. 2022-05-20]. Dostupné z: <https://flexicam.com/sinumerik-808d-advanced/>
- [19] *Pracujte produktivněji s HEIDENHAIN* [online]. [cit. 2022-05-20]. Dostupné z: https://www.heidenhain.cz/cs_CZ/o-firme/
- [20] BOSCH REXROTH CORPORATION. *Understanding the IEC61131-3 Programming Languages* [online]. In: . [cit. 2022-05-20]. Dostupné z: https://dc-us.resource.bosch.com/media/us/products_13/product_groups_1/electric_drives_and_controls/_pdfs_1/BRC_Controller_Programming.pdf
- [21] *TNC 128 straight-cut control* [online]. In: . [cit. 2022-05-20]. Dostupné z: <https://www.heidenhain.com/products/cnc-controls/tnc-128>
- [22] TNC 320 contouring control. In: *Heidenhain* [online]. [cit. 2022-05-20]. Dostupné z: <https://www.heidenhain.com/products/cnc-controls/tnc-320>
- [23] TNC 620 contouring control. In: *Heidenhain* [online]. [cit. 2022-05-20]. Dostupné z: <https://www.heidenhain.com/products/cnc-controls/tnc-620>
- [24] VEERAMANI, ARUN a TODD WALTER. *Function block diagrams* [online]. In: . 2007 [cit. 2022-05-20]. Dostupné z: <https://www.controleng.com/articles/function-block-diagrams/>
- [25] PETER. *Structured Text Programming: A Step by Step Guide* [online]. 2015 [cit. 2022-05-20]. Dostupné z: <https://www.plcademy.com/structured-text-tutorial/>
- [26] JACK, Hugh. *Automating Manufacturing Systems with PLCs* [online]. Lulu.com, 2010 [cit. 2022-05-20]. ISBN 978-0557344253. Dostupné z: <https://freecomputerbooks.com/Automating-Manufacturing-Systems-with-PLCs.html>
- [27] *Function Block / Structured Text Introduction guide* [online]. 2013 [cit. 2022-05-20]. Dostupné z: https://www.fa.omron.com.cn/data_pdf/mnu/r144-e1-04_cx-programmer.pdf?id=1605
- [28] ZPS MCV1260i. In: *Tajmac-zps.cz* [online]. [cit. 2022-05-20]. Dostupné z: <https://www.tajmac-zps.cz/mcfv-1260>

7 SEZNAM OBRÁZKŮ

Obr. 1: Sinumerik 808: Panel řízení [18].....	18
Obr. 2: Sinumerik 808: Příklad možné konfigurace [2].....	18
Obr. 3: Sinumerik 828: Panel řízení [3].....	19
Obr. 4: Sinumerik 828: Příklad možné konfigurace [3].....	19
Obr. 5: Sinumerik 840: Varianty řízení [4].....	20
Obr. 6: Sinumerik ONE: Příklad hardwarové konfigurace [5]	21
Obr. 7: TNC 128: Panel řízení [21].....	22
Obr. 8: TNC 320: Panel řízení [22].....	23
Obr. 9: TNC 620: Panel řízení [23].....	23
Obr. 10: TNC 620: Základní hardwarová konfigurace [9]	24
Obr. 11: TNC 640: Panel a standartní rozložení klávesnice [10]	25
Obr. 12: TNC 640: Schéma zapojení panelu a řídicího PC [10].....	25
Obr. 13: Instrukční list: Ukázka struktury jazyka	28
Obr. 14: Jazyk strukturovaného textu: detekce pozice vision senzorem [27].....	29
Obr. 15: Funkční blok	30
Obr. 16: Funkční blokové schéma	30
Obr. 17: Ladder diagram: struktura jazyka	32
Obr. 18: Vývojové prostředí HEIDENHAIN: PLCdesign.....	34
Obr. 19: virtualTNC: Hlavní obrazovka panelu TNC64.....	35
Obr. 20: Vertikální obráběcí centrum: MCV1260i [28]	37
Obr. 21: GTK demo: ukázka grafických prvků	38
Obr. 22: HEIDENHAIN PLC: Časové rozložení cyklu	39
Obr. 23: TNC 640: Tabulka přehledu binárních hodnot vstupů	41
Obr. 24: První koncept: Půdorys obráběcího centra	42
Obr. 25: NC Control: rozvržení využitelných oken.....	43
Obr. 26: SELinux: projev blokace programu výrobce stroje	44
Obr. 27: PLCdesign: Vytvoření markeru pro novou obrazovku.....	45
Obr. 28: PLCdesign: umístění a kód první obrazovky.....	45
Obr. 29: PLCdesign: Hlavní program obrazovek	46
Obr. 30: TNC 640: První obrazovka v systému.....	47
Obr. 31: Diagnostická obrazovka: Půdorys pro vyobraze	49
Obr. 32: Diagnostická obrazovka: Použití vytvořeného widgetu	50
Obr. 33: Diagnostická obrazovka: Přidání tabulky detailu signálu	52
Obr. 34: Diagnostická obrazovka: Záložka pro výměníky palet	55
Obr. 35: Diagnostická obrazovka: Příklad finálního rozhraní pro stroj.....	56
Obr. 36: TNCscope: import záznamu z vytvořeného softwaru	58
Obr. 37: Obrazovka pro parametrizaci řízení SIMOTION skrze PROFINET ...	59
Obr. 38: TNC 640: Aktuální editování parametrů stroje	61
Obr. 39: Koncept obrazovky pro zobrazení vybraných parametrů.....	62

8 SEZNAM TABULEK

Tab. 1:	Přehled modelových řad Sinumerik [2, 3, 4, 5, 6].....	17
Tab. 2:	Programování PLC: Rozdělení programovacích standardů [13, 20].	27
Tab. 3:	Instruction list: Přehled vybraných instrukcí [17]	28
Tab. 4:	Seznam softwaru dodávaného k řídicím systémům TNC [17].....	33
Tab. 5:	Zvolené parametry obrazovky	48
Tab. 6:	Rozměry předdefinovaných oken (TNC s rozlišením 1280x1024)...	48
Tab. 7:	Volitelné parametry třídy Signal_icon	50

9 SEZNAM PŘÍLOH

Archiv obsahující zdrojové kódy obrazovek, dostupné také z:
https://github.com/NavryCZ/TNC_tools