

Univerzita Hradec Králové  
Přírodovědecká fakulta  
Katedra fyziky

**Využití mikropočítačů v průmyslu**

**Bakalářská práce**

Autor: David Vlk  
Studijní program: B1701 – Fyzika  
Studijní obor: Fyzikálně – technická měření a výpočetní technika

Vedoucí práce: Mgr. et Bc. Radek Němec, Ph.D.  
Odborný konzultant: Ing. Stanislav Hons

Univerzita Hradec Králové  
Přírodovědecká fakulta

**Zadání bakalářské práce**

<b>Autor:</b>	<b>David Vlk</b>
Studijní program:	B1701 – Fyzika
Studijní obor:	Fyzikálně – technická měření a výpočetní technika
Název práce:	Využití mikropočítačů v průmyslu
Název práce v AJ:	Using of the microcomputers in the industry
Cíl a metody práce:	Cílem práce je představit základní principy řízení strojů v průmyslu, popsat technologii mikropočítačů a uvést příklady úspěšného propojení obou oborů. Praktická část popisuje nízkonákladovou přestavbu obráběcího stroje Heckert FSS 230 s využitím platformy Arduino Mega jako řídicí jednotky. Zahrnuje analýzu obráběcího stroje, volbu vhodného HW, návrh nové elektroinstalace a vývoj softwaru pro chod stroje a jeho obsluhu.
Garantující pracoviště:	Katedra fyziky Přírodovědecké fakulty UHK
Vedoucí práce:	Mgr. et Bc. Radek Němec, Ph.D.
Odborný konzultant:	Ing. Stanislav Hons
Oponent:	Ing. Jiří Jelínek, Ph.D.
Datum zadání práce:	25.10.2017
Datum odevzdání práce:	29.5.2019

**Prohlášení:**

Prohlašuji, že jsem bakalářskou práci vytvořil samostatně a že jsem v seznamu použité literatury uvedl všechny prameny, ze kterých jsem vycházel.

V Hradci Králové dne: 29.5.2019

David Vlk

## **Poděkování**

Mé poděkování patří Mgr. et Bc. Radku Němcovi, Ph.D. za technickou inspiraci a cenné rady při vedení bakalářské práce. Děkuji také Ing. Stanislavu Honsovi za odborný dohled a věcné připomínky k praktické části bakalářské práce.

## **Anotace**

VLK, D. *Využití mikropočítačů v průmyslu. Hradec Králové, 2019.*

Bakalářská práce na přírodovědecké fakultě Univerzity Hradec Králové.

Vedoucí bakalářské práce Mgr. et Bc. Radek Němec, Ph.D.

Teoretická část bakalářské práce popisuje historický vývoj v řízení průmyslových strojů a uvádí základní princip mikropočítačů, jejich rozdělení a funkce. Podrobněji se práce věnuje využití mikropočítačů (mikroprocesorů) k automatizaci průmyslových strojů. Dále jsou uvedeny možnosti propojení mikropočítačů s komponenty strojů a varianty komunikace mezi nimi.

Cílem praktické části je zprovoznit konkrétní typ obráběcího stroje v rámci jeho kompletní nízkonákladové přestavby a generální opravy za pomoci mimo jiné běžně dostupného mikropočítače. To zahrnuje prvotní analýzu principu funkce obráběcího stroje, jeho funkcí a původní elektroinstalace, následnou volbu způsobu jeho ovládání, analýzu bezpečnosti a nutných opatření, následný vývojový diagram, vývoj programu v jazyce C++, tvorbu zjednodušeného modelu stroje, ověření funkčnosti řídicího systému na modelu a ladění jeho výkonnosti.

## **Klíčová slova**

Mikropočítač, mikroprocesor, automatizace, obráběcí stroj

## **Annotation**

VLK, D. Using of the microcomputers in the industry. *Hradec Králové*, 2019.

Bachelor Thesis at Faculty of Science University of Hradec Králové.

Supervisor: Mgr. et Bc. Radek Němec, Ph.D.

The theoretical part of the thesis describes the historical development in the control of industrial machines and introduces the basic principle of microcomputers, their division and function. The thesis deals in detail with the use of microcomputers (microprocessors) for automation of industrial machines. The following are possibilities of interconnection of microcomputers with machine components and variants of communication between them. The aim of the practical part is to put into operation a specific type of machine tool within its complete low-cost rebuilding and overhaul with the help of a commercially available microcomputer. This includes the initial analysis of the machine tool function, its functions and original wiring, the subsequent choice of its control method, safety analysis and necessary measures, the subsequent flowchart, the development of the C++ program, the creation of a simplified machine model, verification of the control system functionality on the model and tuning performance.

## **Keywords**

Microcomputer, microprocessor, automation, machine tool

# Obsah

Úvod .....	9
1 Teoretická část.....	10
1.1 Stroje v průmyslu.....	10
1.1.1 Obecný princip .....	10
1.1.2 Zdroje energie pro pohon průmyslových strojů .....	10
1.1.3 Výkonné prvky strojů a jejich regulace .....	10
1.2 Ovládání a řízení strojů .....	11
1.2.1 Obsluha člověkem.....	11
1.2.2 Číslicové řízení pomocí děrných štítků (NC).....	11
1.2.3 Řízení počítačem (CNC).....	12
1.2.4 Konfigurace CNC strojů.....	12
1.3 Jednočipové mikropočítače .....	14
1.3.1 Základní princip.....	14
1.3.2 Činnost mikropočítače .....	15
1.3.3 Činnost mikropočítače v praxi .....	16
1.3.4 Program .....	17
1.3.5 Způsoby programování .....	17
1.3.6 Vstupní zařízení.....	17
1.3.7 Zapojení vstupních zařízení a ochrana .....	18
1.3.8 Výstupní zařízení .....	19
1.4 Programování jednočipového mikropočítače v jazyce C++ .....	20
1.4.1 Obecný princip.....	20
1.4.2 Struktura programu .....	20
1.4.3 Programování vstupních kontaktů .....	20
1.4.4 Práce mikropočítače s časem .....	21
1.4.5 Komunikace mikropočítače s okolím během provozu .....	22

2	Praktická část .....	23
2.1	Popis obráběcího stroje Heckert FSS 230 .....	23
2.1.1	Původní funkce stroje.....	23
2.1.2	Technický stav stroje před generální opravou.....	24
2.2	Rozsah generální opravy .....	25
2.2.1	Změny základní konfigurace.....	25
2.2.2	Silová elektronika .....	25
2.2.3	Slaboproudá elektronika.....	25
2.2.4	Číslicový řídicí systém.....	26
2.2.5	Změny v pracovních režimech a jejich ovládání .....	27
2.3	Elektronické zapojení.....	28
2.3.1	Seznam základních komponent stroje.....	28
2.3.2	Návrh elektronického zapojení .....	29
2.3.3	Ochrana mikropočítače.....	30
2.4	Řídicí program .....	32
2.4.1	Základní požadavky .....	32
2.4.2	Struktura programu .....	32
2.4.3	Třídy.....	32
2.4.4	Časové intervaly .....	33
2.4.5	Čtení digitálních hodnot na vstupních kontaktech .....	34
2.4.6	Pracovní režimy (pro stroj) .....	36
2.4.7	Pracovní režimy (pro operační program) .....	38
2.4.8	Vývojový diagram (zjednodušený) .....	39
2.4.9	Ověření funkčnosti programu, ladění výkonnosti.....	41
3	Diskuse.....	42
	Závěr.....	43
	Seznam použité literatury.....	44
	Přílohy .....	46



# Úvod

Výpočetní technika, pojem, který je v současnosti převážně spojován zejména s osobními počítači. Obecně však do této kategorie spadá téměř každé nové elektronické zařízení, které obsahuje procesor (např. digitální hodinky, MP3 přehrávač, GPS navigace, multifunkční tiskárny, palubní počítače dopravních prostředků, řídicí systémy inteligentních budov, různé robotizované stroje, ale i zařízení s umělou inteligencí). Se zdokonalujícími se funkcemi, uživatelským prostředím a obsluhou těchto zařízení se však běžnému uživateli stále více vzdaluje samotná podstata principu funkce těchto zařízení, která má však stejný základ, tedy práci naprogramovaného výpočetního algoritmu v integrovaném obvodu (procesoru).

Význam výpočetní techniky si osobně nejvíce uvědomím, když sleduji porovnání nějakého procesu v reálném životě před a po využití počítačů. Významný technologický posun s příchodem výpočetní techniky zaznamenalo odvětví strojního průmyslu. Dříve neuvěřitelně fyzicky náročné, ruční obsluhou téměř nemožné úkony, či geometricky a matematicky složité procesy, jsou s využitím výpočetní techniky vypočteny ve zlomku sekundy a samotný pracovní úkon provede počítačem řízený stroj s vysokou přesností za podstatně kratší čas než člověk. Za zmínku stojí též podstatně vyšší bezpečnost pro osobu provádějící daný úkon.

Tato bakalářská práce se proto bude věnovat využití mikropočítačů v průmyslu.

Teoretická část má za cíl seznámit čtenáře se základními informacemi z oborů výpočetní techniky a strojů v průmyslu.

Praktická část je ukázkou využití běžně dostupné technologie pro řízení chodu obráběcího stroje, přičemž popisují každý krok celého procesu přestavby, s výjimkou řešení mechanických závad, úprav stroje a konstrukci nového pneumatického upínacího systému pro specifické výrobky, což se tématu této bakalářské práce vymyká.

# 1 Teoretická část

## 1.1 Stroje v průmyslu

Již od začátku průmyslové revoluce, tedy konce 18. století, kdy manuální nesystematickou výrobu nahradila manufaktura a těžkou práci lidem začaly usnadňovat různé stroje a zařízení, je stále zájem o větší produktivitu výroby, požadavky na přesnost a minimální odlišnost výrobků v sériové výrobě. [1] Zároveň se některá průmyslová odvětví zabývají výrobou součástí, jejichž velikost se vymyká rozsahu lidských možností (příliš velké, či mikroskopicky malé předměty). I zde proto mají své místo stroje, které jsou pro práci v těchto podmínkách konstruované.

### 1.1.1 Obecný princip

Základním principem a významem stroje je provádění mechanické, či jiné práce s využitím různých zdrojů energie.

Charakteristika činnosti stroje, například rozsah a rychlost pohybu jeho částí, je definována již konstrukční dispozicí, nebo je řízena obsluhou prostřednictvím ovládacích prvků, případně je řízena elektronicky.

### 1.1.2 Zdroje energie pro pohon průmyslových strojů

Pro pohon soudobých strojů slouží zejména tyto zdroje energie, často i kombinace více zdrojů. [2]

- Tepelné motory
- Elektrická energie
- Pneumatické systémy
- Hydraulické systémy

### 1.1.3 Výkonné prvky strojů a jejich regulace

Stroje se většinou skládají z různých částí, některé jsou statické, jiné pohyblivé, případně mají různé funkce. Nejčastější prvky, které zajišťují onen pohyb, nebo jiné funkce, se dělí a ovládají následovně: [3]

- Mechanické prvky
  - Lineární motory
  - Rotační motory
- Elektromagnetické prvky
- Zdroje tepla
- Zdroje světla, záření

Ovládání a regulace mechanických prvků je poměrně snadná. Pneumatické a hydraulické lineární či rotační motory se uvádějí v chod a zajišťují v dané poloze pomocí ventilů, rychlost pohybu se reguluje objemovým průtokem média, například škrťacími ventily. Maximální síla či točivý moment jsou regulovány tlakem média.

Elektrické lineární a rotační motory se uvádějí v chod pomocí spínačů či stykačů a regulují úroveň elektrického napětí a proudu. [3]

U ostatních prvků strojů je situace podobná, zásadní otázkou je obsluha funkce zapnuto/vypnuto a nastavení výkonu daného prvku nejčastěji je dáno regulací dodávané energie ze zdroje.

## **1.2 Ovládání a řízení strojů**

Obsluhou a řízením strojů se rozumí ovládání jejich činnosti a manipulace s komponenty v reálném čase. Například spouštění výkonných prvků, řízení pohybu částí strojů a dohled nad celým provozem.

Následující přehled mapuje historický vývoj způsobů ovládání a řízení strojů.

### **1.2.1 Obsluha člověkem**

Elektronická konfigurace takového stroje lze zjednodušeně popsat tak, že každý prvek konající práci (elektromotor, lampa apod.) je zapojen samostatně a ovládán přímo obsluhou. Zpravidla se jedná o jednoduché ruční nebo stolní stroje typu vrtačka, bruska, ale i rozměrnější stroje, například klasické ručně ovládané soustruhy, lisy a podobně.

Rozsah a rychlost pohybu částí stroje jsou doslova v rukou obsluhy, není nijak předcházeno haváriím (na soustruhu je například možné narazit soustružnickým nožem do rotujícího vřetena). [2] Kontrola pozic pohyblivých částí stroje probíhá nejčastěji odhadem, ručními měrkami a mechanickými stupnicemi. [4]

Některé stroje byly vybavené takzvanou „tvrdou automatizací“ prostřednictvím narážek, vaček a šablon. Například po mechanickém umístění narážek do krajní polohy tyto aktivují mikrospínač, který ukončí danou výrobní operaci. [5]

### **1.2.2 Číslíkové řízení pomocí děrných štítků (NC)**

Stroje označené jako NC, z anglického „Numerical Control“ jsou řízeny prostřednictvím pevných štítků nebo navinutých pásek, na nichž je nejčastěji pomocí děr zanesen kód, dle kterého stroj postupně provádí sled operací.

NC řízení mělo významnou roli v sériové výrobě obráběním. Zprvu stroje z děrných štítků „četly“ mechanicky a prováděly pouze jednoduché operace v jedné ose. Postupně byly schopné na základě dat z děrných štítků a pásek číst opticky a například pracovat ve více osách najednou, měnit nástroje, rychlost pohybu i podávat polotovary. [6]

Kontrola polohy pohyblivých částí stroje byla řešena například elektroopticky. S měnící se polohou odměřovací zařízení vysílá signál s úměrným počtem impulsů, které zaznamenává čítač impulsů (např. dekatronový). Pokud se informace o počtu impulsů z děrné pásky shodne s počtem impulsů zaznamenaných čítačem, dojde ke konkidenci (shodě) a daná operace je dokončena. K měření polohy sloužily i přesné potenciometry zapojené jako napěťové děliče, kde pozice pohyblivé části přímo souvisí se snímaným napětím na jezdcí potenciometru. [7]

### 1.2.3 Řízení počítačem (CNC)

CNC stroje vznikly zavedením počítače k řízení provozu strojů v průmyslu. Cílem je stále větší automatizace výrobních postupů, zvětšující se nároky na přesnost, rychlost výroby a možnost rutinní obsluhy méně kvalifikovanými pracovníky.

Stroje tohoto typu obsahují počítač s pamětí na několik programů, obrazovku a ovládací tlačítka, kterými se stroj ovládá a volí mezi uloženými programy, případně se i programy dají upravovat a vytvářet nové. Možné je též nahrávání programů z média či jiného počítače. [8]

Běžně jsou plně automatizované všechny procesy a ty stroj dokáže provádět souběžně. Pohyblivé prvky strojů umožňují plynulé řízení, mají přesně měřenou polohu a rychlost (například i přesný úhel natočení vřetena soustruhu).

S postupem času jsou CNC stroje schopné automaticky měnit nejen nástroje v otočném držáku, ale i vyměnit nový nástroj ze zásobníku za použitý. Běžně si stroje podávají polotovary z připravených míst. Současné CNC stroje jsou často vybaveny tzv. adaptivním řízením, které zohledňuje například délkové dilatace teplem, hlídá a vyřazuje použité nástroje, sleduje zatížení stroje a upravuje řezné podmínky, provádí mezioperační a výstupní měření výrobků zabudovanými sondami či lasery. [8]

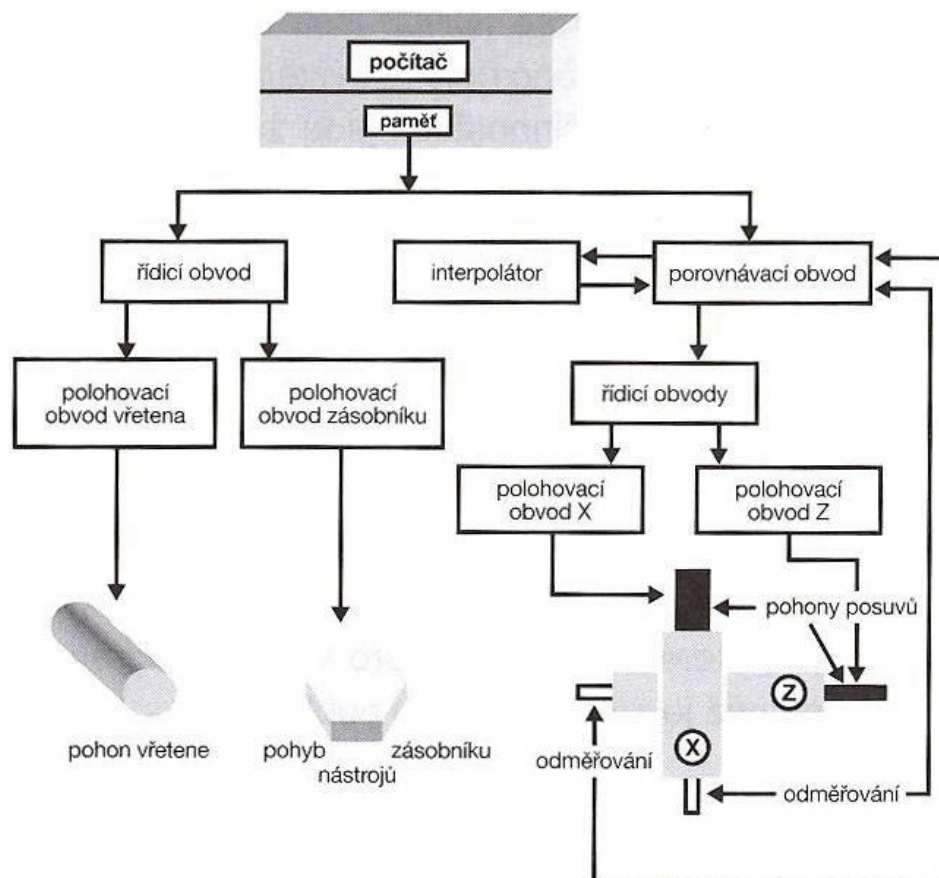
### 1.2.4 Konfigurace CNC strojů

Základní technická řešení konfigurace strojů viz Obrázek 1: Schéma CNC obráběcího stroje, lze rozdělit a popsat takto: [9]

- Druhy pohonů pohyblivých částí
  - Šroubovicí
  - Servosystémy
  - Hydraulické systémy
  - Lineární pohony
  - Krokové elektromotory
  - Elektromotory s plynulou regulací
  - Valivé vedení, kuličkové šrouby
- Způsoby odměřování poloh pohyblivých částí
  - Snímače otáček šroubovic
  - Lasery
  - Snímače úhlu natočení
  - Přímé lineární snímače polohy

## Logické prvky CNC strojů:

- Interpolátor
    - Vypočítává optimální trajektorii nástroje a pozice pohyblivých částí stroje v závislosti na geometrii nástroje a zadaných souřadnic v programu.
    - Odpovídá za geometrickou přesnost výrobku
  - Řídící obvody
    - Logické informace se zde převádějí na analogové, či přímo silnoproudé.
    - Ovládají pohyblivé části stroje – motory, posuvy, stykače, ventily apod.
  - Porovnávací obvod
    - Porovná hodnoty zadané programem s údaji o polohách z odměřovacích prvků v reálném čase.
    - Dává příkazy řídicímu obvodu pohonů pro dosažení požadovaných pozic.
- [8]



Obrázek 1: Schéma CNC obráběcího stroje [8]

## 1.3 Jednočipové mikropočítače

Laicky lze shrnout, že jednočipový mikropočítač je většinou malé, málo výkonné zařízení, které v praxi provádí rutinní operace dle jednoho předem definovaného programu stále dokola. Ovšem relativně nízký výkon mikroprocesoru (jeho hlavní součástka) je zcela dostačující, neboť při správném použití tato zařízení provádějí přiměřené množství úkonů a soustředí se pouze na ně. Hlavními výhodami je nízká spotřeba elektrické energie, postačující pasivní chlazení, malé rozměry, a nízká cena. [10], [11], [12]

### 1.3.1 Základní princip

Většina nových mikropočítačů z výroby po připojení ke zdroji elektrické energie pouze rozsvítí několik kontrolky, případně se objeví elektrické napětí na některých kontaktech (ty se například používají pro napájení připojených periférií) a umožní připojení sebe sama k počítači, jinak mikropočítač nekoná žádnou užitečnou práci, dokud není do programové paměti mikropočítače nahrán nějaký program. Jedná se totiž o zařízení postavené na takzvané harvardské architektuře, zjednodušeně lze říci, že se skládá z mikroprocesoru obsahujícího paměť ROM (Read Only Memory – paměť pro program), paměť RAM (Random Access Memory – operační paměť používaná za chodu), čítače, časovače, okolní síť a vstupní/výstupní kontakty. Program do mikropočítače tedy musí být nahrán (například z PC) a během provozu mikropočítače není možné tento program zásadním způsobem měnit. Pokud dojde během provozu k drobné změně programu (například číselné hodnoty nějakého parametru v programu), po vypnutí a opětovném spuštění se načte program s původními hodnotami.<sup>1</sup> [11]

---

<sup>1</sup> Existují však způsoby, jak během provozu provádět změny v paměti typu ROM, ale není to doporučováno kvůli omezenému počtu cyklů přehrání paměti typu ROM. [29]

### 1.3.2 Činnost mikropočítače

Mikropočítače jsou vybaveny různým počtem vstupních a výstupních kontaktů (pinů), prostřednictvím kterých mikropočítač navenek provádí požadovanou činnost. Primárně mikropočítače pracují s logickými hodnotami 0 – logická nula; 1 – logická jednička. Tyto logické hodnoty přímo souvisí s úrovní elektrického napětí na daném vstupním nebo výstupním pinu. Logická nula je ideálně napětí pinu 0 V vůči kostře mikropočítače (pinu GND – z angl. Ground), logická jednička znamená přibližně hodnotu obdobnou napětí VCC. [13] Hodnota napětí VCC se u různých mikroprocesorů může lišit, nejčastěji se používají hodnoty 1,8 V; 2,5 V; 3,3 V a 5 V. V praxi ovšem hranice napětí, kterou mikropočítač považuje za logickou nulu, může být ve skutečnosti o něco vyšší (například z intervalu 0; 0,8 V). Hodnota napětí vyhodnocená jako logická 1 může být nižší než VCC, například z intervalu (2; 5 V). Uvedené příklady platí pro mikropočítač pracující s VCC = 5 V. [14]

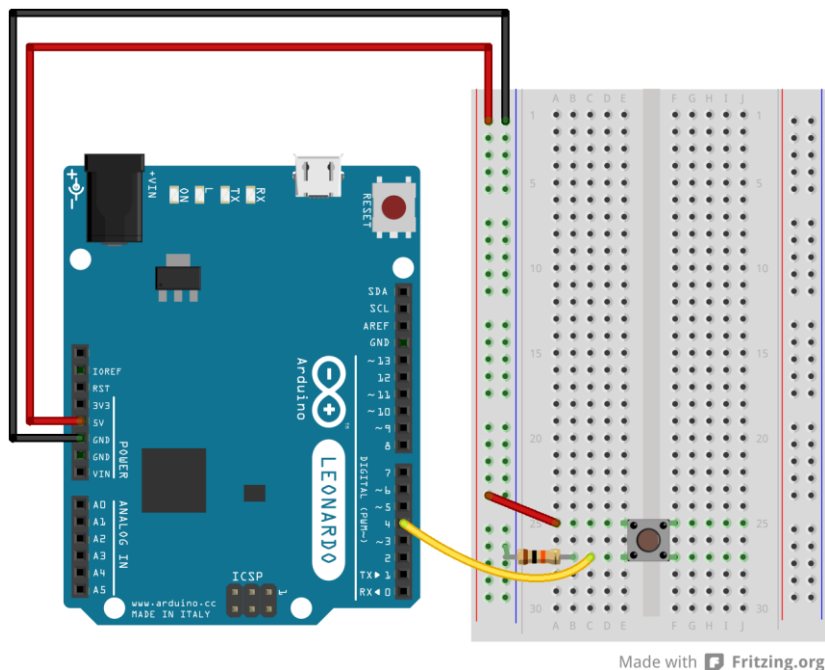
Při spuštění mikropočítače, který je již naprogramován, následuje sled procesů v tomto pořadí:

- Připojení napájení, uvedení zařízení v činnost
- Načtení programu z paměti ROM
- Provedení jednorázových úkonů definovaných v programu, uložení dat do paměti RAM
- Práce mikropočítače ve stále se opakující smyčce, to může obnášet:
  - Čtení hodnot na vstupních pinech
  - Obsluha výstupních pinů
  - Skoky do podprogramů
  - Práce s časem – čekání nebo opakování dějů po periodě
  - Úprava výchozích hodnot programu za hodnoty nové
- Odpojení napájení, vypnutí zařízení [10]

Obecně lze konstatovat, že základní činnost každého mikroprocesoru je práce ve dvojkové soustavě, tedy s logickými jedničkami a nulami na vstupních pinech (tedy měření hodnoty na piny přivedeného napětí) a přivádění napětí 0 V nebo napětí VCC na interní či výstupní piny. [13]

### 1.3.3 Činnost mikropočítače v praxi

Příklad jednoduchého programu, který může mikropočítač s připojeným tlačítkem dle Obrázek 2: Připojení tlačítka k mikropočítači provádět, je zaznamenávat stisk tlačítka a následně například rozsvěcet zabudované LED diody. [15]



Obrázek 2: Připojení tlačítka k mikropočítači

Uvedený příklad je téměř primitivní a může se zdát, že je zde mikropočítač plně nahraditelný jednoduchým elektrickým obvodem se zdrojem napětí, žárovkou a vypínačem, ovšem uvedený příklad slouží zejména jako první lekce začínajícím programátorům. V praxi takovéto primitivní zapojení, kdy je k mikropočítači připojen jeden externí prvek, který způsobuje změny logických hodnot (napětí) na vstupním pinu, umožňuje obrovské množství praktických využití, pokud takové bude naprogramováno. Mikropočítač dokáže bez problému počítat počet impulzů, ukládat do je paměti, vypočítat jejich periodu, rychlost a dále s těmi hodnotami interně kalkulovat. Navíc takové úkony dokáže dělat opakovaně a velmi rychle – řádově tisíce opakování za sekundu. [10]

Mikropočítač dokáže též velmi rychle obsluhovat (zapínat a vypínat) i výstupní piny, čímž dokáže přibližně simulovat analogové napětí v rozmezí 0 V – VCC. Případně stačí k výstupním kontaktům připojit D/A převodník, který digitální podobu úrovně napětí převede na čistě analogový signál napětí. Možnosti rychlých změn úrovní napětí na výstupních pinech se používá i pro digitální komunikaci mezi mikropočítačem a připojeným zařízením (displej, D/A převodník atd.) nebo mezi více mikropočítači vzájemně. Typickým příkladem sériové komunikace je I2C. K digitálnímu přenosu informací dochází prostřednictvím dvou vodičů. V některých případech je tato komunikace označena zkratkou TWI z anglického „Two Wire Interface“. [16]



### 1.3.4 Program

Kód programu, který je nahráván do mikropočítače, je obvykle tvořen mnoha instrukcemi, které se skládají ze dvou částí:

- Operační znak (např. sčítej, porovnej, uprav)
- Operand (s čím se instrukce provádí – číslo, odkaz na hodnotu v registru) [17]

### 1.3.5 Způsoby programování

Nezákladnějším způsobem programování mikropočítačů je programování ve strojovém kódu, tedy posloupnosti hexadecimálních čísel představujících osmibitové kódy instrukcí i hodnoty jejich operandů. Pro ruční programování se tento způsob pro náročnost téměř nepoužívá, nýbrž využívá se takzvaných vyšších programovacích jazyků v počítačových programech, které jsou více srozumitelné a zajistí následný překlad do strojového kódu, který je do mikropočítače nahráván. [18], [19]

**Základní přehled programovacích jazyků:** [18]

- Strojový kód (různé druhy mikropočítačů mají svůj strojový kód)
- Assembly (psány textové instrukce program přeložil do stroj. kódu)
- Basic (vyvinut v r. 1964)
- Pascal
- Action
- C, C++
- C#
- Python
- Wiring

### 1.3.6 Vstupní zařízení

Vstupní zařízení je takové, které se připojuje na vstupní piny mikropočítače, přičemž program mikropočítače snímá hodnotu napětí. Vstupní kontakty mikropočítačů snímají primárně digitální (logická 0, logická 1) hodnoty, ale některé piny sofistikovanějších mikropočítačů snímají i analogové hodnoty (změří hodnotu napětí v rozmezí 0 V – VCC a přidělí jí přímo úměrnou číselnou hodnotu z rozsahu například 0 – 255, což je 8-mi bitové číslo).

Typickými příklady jednoduchých vstupních zařízení jsou tlačítka, spínače, čidla. Tyto pracují pouze v režimu 0/1.

### **Mezi pokročilejší ovládací prvky patří například rotační enkoder: [20]**

- Často používaný ovládací prvek digitálních zařízení (například otočná tlačítka ovládání hlasitosti přehrávačů, otočná tlačítka přístrojů v domácnosti).
- Fyzicky se jedná o krokový snímač, se kterým je možno stále dokola otáčet. Některé typy mají například i funkci stisku či vytažení.
- K mikropočítači je enkoder běžně připojen mimo napájení několika digitálními vstupními kontakty v závislosti na všech funkcích.
- Plná otočka o 360° je skokově rozdělena na několik kroků (u otočných tlačítek běžně okolo 20-ti kroků). U některých otočných tlačítek jsou tyto kroky mechanicky patrné. Při každém přechodu mezi dvěma sousedními kroky se generují pulzy napětí na dvou kontaktech připojených na vstupní piny mikropočítače. Směr otáčení přímo souvisí s pořadím pulzů napětí na kontaktech.
- Mikropočítač tyto pro člověka časově nezaznamenatele časové rozdíly v pořadí pulzů na dvou kontaktech při správné konfiguraci zvládá, avšak jen do určité míry. Při velice hrubé a nepřiměřeně rychlé manipulaci s otočným tlačítkem, nemusí zařízení počet a směr kroků otočného pohybu vyhodnotit správně.
- Funkce stisku je obdobná principu běžného tlačítka.

Technicky složitější jsou různé snímače, například přesné analogové snímače napětí – ty měří hodnotu napětí (i střídavého a mimo rozsah 0 V – VCC) a přenáší ji do mikropočítače buď digitálně (například sériovou komunikací – jedná se o tzv. A/D převodníky), nebo analogově (generují na kontaktu do mikropočítače změřené veličině úměrnou hodnotu napětí v rozsahu v 0 V – VCC). Mikropočítač hodnotu původního (vyššího nebo nižšího) napětí dle zadaného vzorce dopočítá. Na obdobném principu, tedy převodu změřené hodnoty různých fyzikálních veličin na napětí, či na informaci, pracuje většina analogových snímačů a senzorů (měření teploty, elektrického proudu, indukce, kapacity, odporu, intenzity osvětlení, měření průtoku atd.) [21]

#### **1.3.7 Zapojení vstupních zařízení a ochrana**

Technicky je snaha snižovat spotřebu (a tím pádem i zahřívání) veškeré číslicové techniky, proto elektrické proudy používané pro logiku jsou velmi malé, respektive vnitřní odpory i vstupních kontaktů velmi velké, což v praxi může způsobovat problémy. Pokud je ke vstupnímu kontaktu připojen vodič bez ochrany, který například vede k rozepnutému tlačítku, může se prostřednictvím vodiče na digitálním pinu samovolně indukovat elektrické napětí a mikropočítač nesprávně posoudí vstupní pin jako aktivní (logická 1).

## **Zákmity digitálního vstupního signálu**

Dalším významným problémem jsou takzvané zákmity. Vyskytují se u mechanicky spínaných prvků, kdy během stisknutí nebo uvolňování tlačítka kolísá napětí na vstupním kontaktu mezi 0 V a hodnotou VCC. Takový průběh bez ošetření má za následek, že mikropočítač zaznamenává opakovaně logické 0 a 1 a opakovaně koná dle instrukcí úkony (například zaznamenává počet stisknutí tlačítka a „mylně“ dospěje k mnohonásobku fyzického stisknutí).

### **Ochrana vstupních kontaktů: [22]**

- Pull-down rezistor – udržuje napětí na úrovni 0V, zrychlí přechod z 1 na 0.
- Pull-up rezistor – udržuje napětí na úrovni VCC, zrychlí přechod z 0 na 1.
- Kondenzátor – odstraní zákmity, zamezuje ztrátě vodivosti kontaktů mechanických spínačů.
- Integrovaný obvod – odstraní zákmity a zpřesní přechod mezi logickými hodnotami napětí.
- Odporový dělič – při použití vyššího napětí logických okruhů.
- Zenerova dioda – parametrický stabilizátor.
- Optočlen – elektricky oddělí mikropočítač od připojených zařízení.

### **1.3.8 Výstupní zařízení**

Výstupní zařízení je takové, které se připojuje k výstupním kontaktům mikropočítače, který jej následně obsluhuje. Digitální výstupní kontakty mikropočítačů disponují ve stavu logické nuly napětím 0 V, respektive jsou přímo vodivě propojené s kontaktem GND. Ve stavu logické jedničky jsou vodivě propojené s kontaktem VCC a mají tak vůči kontaktu GND napětí obdobné hodnotě napětí VCC. Kontakty mikropočítače jsou tedy přizpůsobené k tomu, aby fungovaly jako zdroj elektrické energie o jmenovitém napětí, ovšem nejsou stavěné na průchod velkých elektrických proudů. V praxi proto není obvyklé, aby výstupní kontakty fungovaly přímo jako napájení externích zařízení, ale tato zařízení jsou obvykle napájena nezávisle na mikropočítači a výstupní kontakty (digitální i analogové) mikropočítačů slouží pouze jako zdroje signálů ke spínání a regulaci externích zařízení, přičemž jimi prochází velice malý elektrický proud. [23]

### **Příklady výstupních zařízení:**

- Tranzistory
- Optočleny
- Relé
- Integrované obvody
- D/A převodníky

## 1.4 Programování jednočipového mikropočítače v jazyce C++

Následující text zjednodušeně hovoří o programování jednočipového mikropočítače Arduino ATmega2560 prostřednictvím aplikace Microsoft Visual Studio s využitím nadstavby Visual Micro. Je třeba upřesnit, že se v tomto případě nejedná jen o programování v jazyce C++, ale aplikace již využívá hotové šablony, knihovny a funkce, které zjednodušují příkazy ve zdrojovém kódu. [24]

### 1.4.1 Obecný princip

Jedná se o takzvaný vyšší programovací jazyk, jehož zdrojový kód je pro člověka relativně dobře čitelný. O překlad do strojového kódu před nahráním do mikropočítače se stará takzvaný kompilér. Jazyk C++ umožňuje při programování využití „abstrakce“, kdy na sebe jednotlivé instrukce neodkazují přímo, ale prostřednictvím volitelných názvů. Programovací jazyk C++ poskytuje též základní funkce objektově orientovaného programování, což umožňuje další míru abstrakce, může šetřit paměť programu, ale hlavně zpřehledňuje program a při správném použití umožňuje jeho snazší úpravy. [25]

### 1.4.2 Struktura programu

Struktura programu, respektive zdrojového kódu je do značné míry volitelná programátorem. Obecně lze program rozdělit na tyto části:

- Deklarace základních proměnných, datových typů (převážně v úvodu programu)
- Jednorázové příkazy pro konfiguraci – „SETUP()“
- Deklarace metod, podprogramů
- Příkazy prováděné v rámci „smyčky“ rutinní činnosti – „LOOP()“
- Vytváření a používání „knihoven“ – částí programu většinou mimo hlavní zdrojový kód. V dané knihovně je kompletně naprogramovaná určitá funkce, kterou je možné následně „volat“ ze zdrojového kódu programu jednoduchými příkazy bez nutnosti pracovat se složitým kódem dané funkce.

### 1.4.3 Programování vstupních kontaktů

Aby mikropočítač správně reagoval na podněty ze vstupních zařízení, je kromě vhodné elektronické konfigurace připojených zařízení velice důležitý správný sled příkazů v kódu programu.

Možností, jak odstranit „zátky“ je známo několik:

- Časová prodleva – po prvním zaznamenání aktivní hodnoty měřit znovu po uplynutí časové prodlevy (řádově desetin sekund). Předpokládá se, že do té doby bude tlačítko pevně stisknuto
- Zohlednění aktuálního stavu – po zjištění aktivního stavu znovu nezjišťovat stav, pokud to není nutné

- Opakované měření vstupní hodnoty – pokud je naměřena shodná logická hodnota několikrát po sobě, teprve poté je vstupní kontakt považován za aktivní/neaktivní.
- Existují různé hotové knihovny „debounce“, které kombinují více nástrojů pro programové ošetření zákmitů. [26]

#### 1.4.4 Práce mikropočítače s časem

Mikropočítač již z výroby velice dobře pracuje s časem, dokáže velice přesně měřit časové intervaly v milisekundách i mikrosekundách. O aktuálním datu a čase může mít přehled až po připojení externího modulu, jež může být rádiově synchronizován.

V praxi se běžně využívají zejména časové intervaly, které se zadávají do kódu v milisekundách. Obecně to jsou příkazy typu „od teď za 500 ms proved' úlohu X“.

- Funkce delay()

Základní možností je využití funkce „delay()“, přičemž se do závorky uvádí časový interval v milisekundách. V praxi tato konfigurace způsobí, že mikropočítač bude „čekat“ až uplyne určený interval a ihned poté provede danou úlohu. Problém může být v tom, že během časového intervalu, kdy mikropočítač „čeká“, neprovádí vůbec žádné operace a nereaguje na žádné vnější podněty (s výjimkou „přerušeni“ a odpojení napájení). Výhodou však je, že mikropočítač časový interval dodrží velice přesně a požadovanou úlohu provede vždy přibližně za stejný čas. [27]

- Funkce millis()

Pokročilejší variantou je využití funkce millis(). Pomocí této funkce lze zjistit hodnota uložená ve vnitřním časovači procesoru. Zde je uchována informace o délce běhu programu od jeho spuštění v milisekundách. V okamžiku, od kterého chceme spustit nějakou úlohu, zjistíme hodnotu millis(), necháme program pokračovat dále v rutinní činnosti. V té se při každém průběhu porovnává, zda již uběhl požadovaný časový interval (jestli se o hodnotu intervalu zvýšila aktuální hodnota millis()). Až interval uběhne, program provede požadovanou úlohu. Výhodou je, že mikropočítač je během časové prodlevy zatížen pouze porovnáváním hodnot millis() a pokračuje v ostatních operacích. Nevýhodou může být zpoždění, tedy že úloha je vykonána o něco později. [27]

- Přerušeni

K mikropočítači připojené externí zařízení může dávat mikropočítači opakované impulzy, které při správném připojení a programové konfiguraci způsobí „přerušeni“ rutinní práce a vyzve k provedení dané úlohy, následně se mikropočítač vrací k rutinní činnosti.

### 1.4.5 Komunikace mikropočítače s okolím během provozu

Při práci mikropočítače je jednak užitečné vědět, jak pracuje, kde se případně stala chyba. Zároveň je v některých případech potřeba provádět změny v programu během práce bez nutnosti připojení PC a procesů programování, kompilace a přehrávání software do mikropočítače. Pro takové případy je vhodné do programu zahrnout i komunikaci z mikropočítače k obsluze a zároveň možnost obsluhy snadno zasahovat do některých částí programu. [28]

- Světelná signalizace

V praxi se běžně používá světelná signalizace na digitálních přístrojích k identifikaci stavu, ve kterém se zařízení nachází. Může k tomu být použita například LED dioda. V programu je určeno, jakým způsobem má LED dioda signalizovat daný stav (například rozsvícením, rychlostí blikání, poměrem v rozsvíceném a zhasnutém stavu). K obdobnému účelu může být použito i více diod, které poskytnou více kombinací a mohou tak zobrazit více informací.

- Zvuková signalizace

Výstupní zařízení, které vydávají zvuk, lze snadno připojit k mikropočítači. Výstupní piny mohou běžně generovat pulzní signál o slyšitelných frekvencích. Stačí k takovému kontaktu připojit piezzo reproduktor a mikropočítač může s okolím komunikovat prostřednictvím frekvence tónů, jejich délkou, či kombinací více tónů.

- LCD displej

Existují různé způsoby připojení LCD displejů, které se liší rychlostí komunikace a počtem vodičů. Existují typy displejů, jenž lze s mikropočítačem propojit sériovou komunikací I2C. Ta využívá dva komunikační vodiče, dále je displej potřeba napájet ideálně nezávisle na mikropočítači. Vzhledem k tomu, že přenos dat probíhá pouze po dvou vodičích, je tato komunikace relativně pomalá. Fyzicky taková komunikace pracuje tak, že na základě příkazu jsou na displeji ve vybraných polích zobrazeny zvolené symboly a ty jsou viditelné až do té doby, než jsou „přepsány“ jiným příkazem. Není tedy potřeba posílat povely ke zobrazení statického textu, naopak je třeba využívat komunikaci s rozvahou, neboť samotný digitální přenos textu je časově velice náročný. V praxi se volí buď aktualizace obrazu po určitých časových intervalech, nebo až ve chvíli, kdy je třeba zobrazený text změnit.

- Úprava parametrů programu během provozu

Velmi snadno lze měnit například číselné hodnoty proměnných, které jsou definovány od spuštění programu, je možné tak činit prostřednictvím tlačítek a funkčního kódu. Pokud je třeba měnit parametry více proměnných nezávisle na sobě, je třeba buď více tlačítek, nebo zvolit sofistikovanější interní program a obsluhovat volbu „který parametr upravit“ a následně „o kolik vybraný parametr upravit“. Následně je třeba dané hodnoty „uložit“ a přikázat mikropočítači zohlednit nové parametry. [28]

## 2 Praktická část

Cílem praktické části je ukázat, že je možné za pomoci nízkonákladových technologií provést přestavbu například obráběcího stroje, v tomto případě vertikální 3D frézky Heckert FSS 230, s využitím jednočipového mikropočítače Arduino Mega 2560 s procesorem ATmega 2560.

### 2.1 Popis obráběcího stroje Heckert FSS 230

Jedná se o klasickou vertikální frézku k obrábění kovů ve třech osách s rotujícím nástrojem ve vřetení, jenž je poháněno vlastním elektromotorem přes mechanickou převodovou skříň. Druhý elektromotor a převodová skříň slouží pro pojezd stolu, na kterém je upnut obrobek.

Obrábění probíhá prostřednictvím pohybu stolu a obrobku pod rotujícím nástrojem. Je třeba volit vhodné tzv. „řezné podmínky“, tedy odpovídající rychlost otáčení nástroje s ohledem na jeho průměr a vhodnou rychlost posuvu stolu.

Směr pojezdu stolu v dané ose buď na stranu A, nebo B se volí na ovládacím panelu příslušnými tlačítky. Úrovně rychlostí pojezdu stolu jsou dvě: pomalu (pro obrábění), rychle (pro pojezd mimo obrábění). Změnu mezi pomalým a rychlým pojezdem zajišťuje elektromagnety ovládaná lamelová spojka předřazená převodovce stolu. Jemná korekce pomalého pojezdu pro obrábění probíhá pomocí ručního nastavování převodu variátoru otočným kolem pod stolem stroje.

#### 2.1.1 Původní funkce stroje

K ovládní stroje po zapnutí hlavního vypínače slouží panel s tlačítky, z nichž některá se zabudovanou žárovkou signalizují stav a činnost stroje.

**Pracovní režimy:** (spuštěné 4-mi tlačítky na ovládacím panelu)

- Obrábění A (spuštění vřetena a posuvu stolu)
- Obrábění B (spuštění vřetena a posuvu stolu)
- Rychlý posuv A (spuštění rychlého posuvu stolu)
- Rychlý posuv B (spuštění rychlého posuvu stolu)
- Zastavit (zastavení výše uvedených pracovních režimů)

Chlazení a mazání emulzí pro obrábění zajišťuje čerpadlo poháněné třífázovým motorem a ovládané ručním vypínačem v zadní části stroje.

Jako bezpečnostní prvky slouží tlačítko „Total stop“ a jako ochrana proti poškození stroje celkem 6 koncových spínačů (na obou krajních polohách všech tří os pohybu stolu), které případně zastaví stroj.

### 2.1.2 Technický stav stroje před generální opravou

Původní elektrický rozvaděč byl velice poruchový a nepřehledný, neboť veškerá logika byla tvořena cca 20-ti 5-ti pólovými stykači, které byly vzájemně propojené přes své spínací, rozpínací a ovládací kontakty elektrickými vodiči. Tento zastaralý systém způsoboval například výpadky i nepříjemné rázy pro elektromotory přerušovanými či protichůdnými elektrickými impulzy, které způsobovaly též nárazové a relativně rychlé změny polohy stolu frézky, viz Obrázek 3: Původní elektrický rozvaděč.

Stroj trpěl dalšími, zejména mechanickými závadami, které byly v rámci generální opravy odstraněny.

Z uvedených důvodů nebylo možné na stroji provádět kvalitní a přesnou práci. Mimo jiné docházelo i k častým poškozením obráběcích nástrojů a hrozilo závažnější poškození různých částí stroje.



Obrázek 3: Původní elektrický rozvaděč



## **2.2 Rozsah generální opravy**

Při volbě rozsahu generální opravy je třeba zohlednit stáří a opotřebenění stroje, jeho technické možnosti, mechanické limity a základní konfiguraci. Nezanedbatelným parametrem jsou s tím související celkové náklady a technická náročnost generální opravy.

### **2.2.1 Změny základní konfigurace**

Mechanická konfigurace stroje s jednou převodovkou a jedním motorem pro pohyb stolu pro všechny tři osy však předurčuje, že i po této nízkonákladové generální opravě bude stroj schopen pohybovat stolem vždy jen v jedné ose dle ručně zvoleného směru pojezdu stolu buď na stranu A nebo B.

**Změny v základní konfiguraci budou obecně zahrnovat:**

- větší automatizaci dějů během provozu stroje, které dříve prováděla obsluha ručně
- nové pracovní režimy stroje a způsob jejich ovládnání
- intuitivní signalizace chodu stroje a případných chyb
- nový upínací mechanismus pro specifické výrobky potrubních armatur

### **2.2.2 Silová elektronika**

Veškeré původní součástky budou ze stroje odmontovány, rozvodná skříň vyčištěna a budou instalovány nové elektronické součástky silnoproudého vedení.

Zejména pro přesnější nastavení optimálních řezných podmínek a za účelem možností dalšího snížení rychlosti otáčení nástroje bude chod motoru pro otáčení nástroje nově řízen frekvenčním měničem.

Pro komfortnější obsluhu bude nyní čerpadlo pro chladicí emulgační směs spínáno pomocí nového třífázového stykače.

### **2.2.3 Slaboproudá elektronika**

Veškerá slaboproudá instalace bude odstraněna, a to nejen původní řídicí systém tvořený stykači, relátky, časovači a dalšími díly, ale budou demontovány i zdroje stejnosměrného a střídavého napětí, které již nebude potřeba. Původní systém totiž do každého spínače v ovládacím panelu, i v různých částech stroje, přiváděl napětí okolo 100V. To považuji za nebezpečné a všechny logické spínače budou připojeny pouze k napětí 5V.

## 2.2.4 Číslicový řídicí systém

V teoretické části je uvedeno, že moderní CNC stroje se skládají z několika samostatných vzájemně spolupracujících číslicových systémů, z nichž každý zvlášť provádí pouze své úkony. Není snahou postavit podobný stroj, ale demonstrovat možnosti využití jednoho jednočipového mikropočítače, který sice nedokáže provádět více úkonů v jeden okamžik, ale při vhodné konfiguraci programu může obsluhovat různé části stroje bez patrných prodlev.

Jako hlavní a zároveň jediný řídicí číslicový systém byl vybrán jednočipový mikropočítač Arduino Mega 2560 viz Obrázek 4: Mikropočítač Arduino Mega 2560.



Obrázek 4: Mikropočítač Arduino Mega 2560

### Parametry mikropočítače:

- Mikroprocesor: Atmel ATmega2560
- Taktovací frekvence: 16 MHz
- EEPROM: 4kB
- SRAM: 8 kB
- Flash Memory: 256 kB
- Provozní Napětí: 5V
- Vstupní Napětí (doporučeno): 7-12V
- Analogové Vstupní Piny: 16
- Digitální I/O Piny: 54 (z toho 14 poskytuje PWM výstup)
- DC Proud pro 3.3 V: 50 mA
- DC Proud na I/O: 40 mA
- Velikost: 100\*55 mm
- Možnost připojení například LCD displeje přes rozhraní I2C.

## 2.2.5 Změny v pracovních režimech a jejich ovládání

### • Rychlý pojezd

Původní ovládání rychlého posuvu na obě strany bylo nepraktické. Stisknutí příslušného tlačítka zapnulo rychlý posuv a ten byl aktivní, dokud nebyl stisknutím tlačítka „Zastavit“ vypnut.

Rychlý posuv se používá zejména pro příjezd před obráběním a k odjezdu po něm. Pro tyto účely je pojezd poměrně rychlý (přibližně 50 mm/s) a nutnost stisknutí dvou tlačítek během tak krátkého intervalu nepovažují za rozumné. Nově bude rychlý posuv aktivní pouze po dobu stisku tlačítka a stůl zastaví bezprostředně po jeho uvolnění.

### • Nový pracovní režim „Odjezd Y“

Obrábění většiny výrobků zpravidla ve středně velkých sériích v ose Y (svislá osa), zahrnuje tyto úkony: upnutí obrobku, rychlý příjezd před obráběním, obrábění, ukončení obrábění, odjezd, vypnutí odjezdu, vyjmutí obrobku.

Zejména v části okolo ukončení obrábění je prostor pro větší automatizaci, která nepochybně povede ke zrychlení celého procesu. Bude tedy vytvořen pracovní režim „Odjezd Y“

- Bude fungovat pouze při pohybu stolu v ose Y.
- Při stisknutí tlačítka pro rychlý odjezd směrem B při obrábění na stranu A se ukončí pomalý posuv směrem A.
- Proběhne editovatelná časová prodleva pro doběh posuvu a dokončení celého oběhu nástroje.
- Bude zapnut rychlý posuv směrem B po editovatelný časový interval, během kterého stůl odjede do požadované vzdálenosti od nástroje.
- Následně budou posuv i vřeteno zastaveny.

### • 3D obrábění

Některé druhy obrobků vyžadují obrábění ve dvou různých osách bezprostředně po sobě, přičemž nástroj je stále v kontaktu s obrobkem. V těchto situacích není vhodné nástroj zastavovat (hrozí vylomení řezné hrany).

- Na ovládací panel přidat tlačítko „Z“ pro zastavení posuvu stolu.
- Nebude mít vliv na chod vřetena.
- Umožní po stisku ruční změnu osy obrábění řadící pákou.
- Po zařazení následuje stisk tlačítka pro obrábění směrem A nebo B v dané ose.
- Nedošlo k zastavení vřetena ani toku chladicí emulgační směsi, pokračuje obrábění v jiné ose.
- Tento proces je možno opakovat neomezeně.

## 2.3 Elektronické zapojení

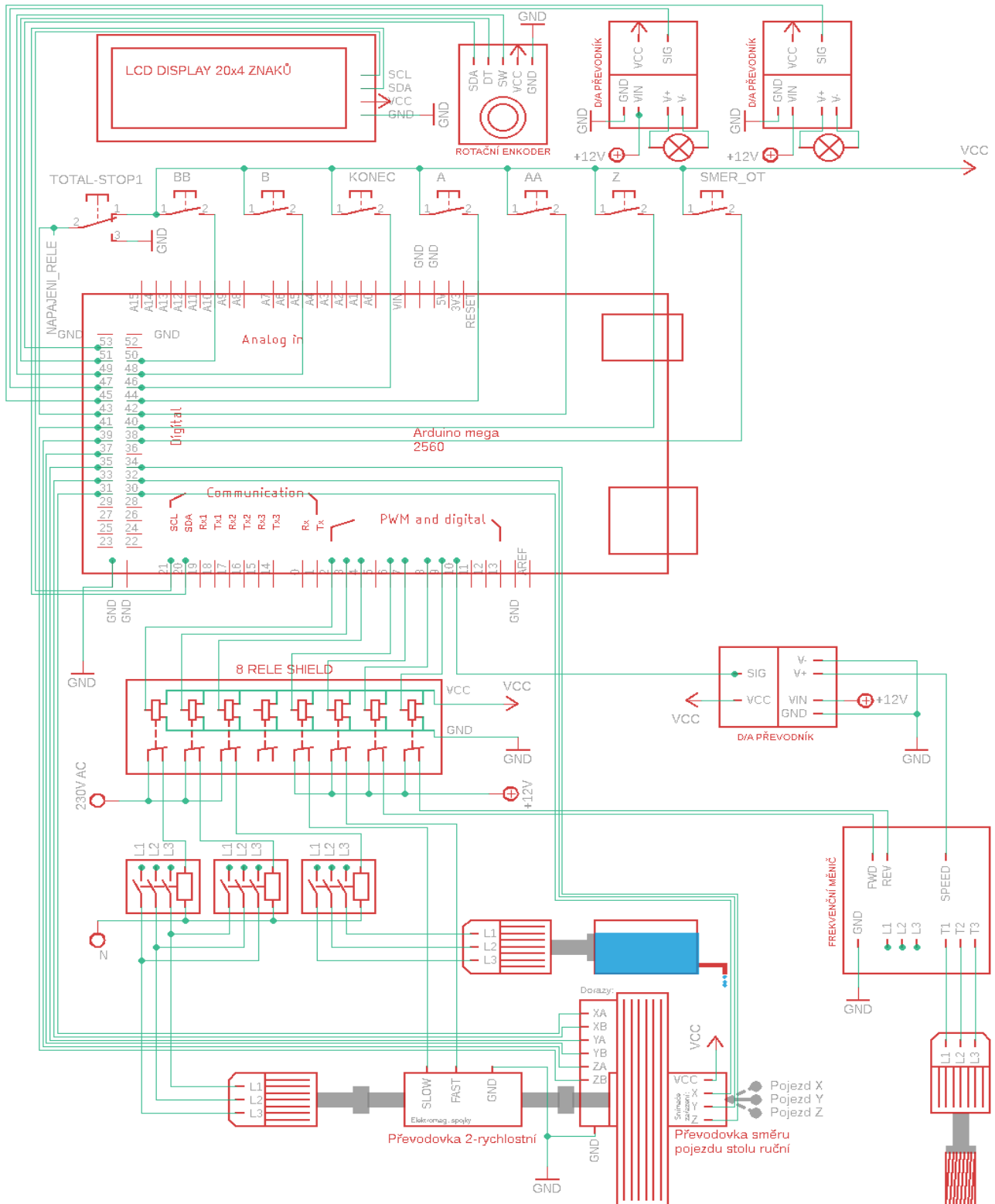
Nový elektronický systém je složen převážně z nových komponent, původní zůstaly zejména spotřebiče (elektromotory, spojky) a některé atypické snímače.

### 2.3.1 Seznam základních komponent stroje

- Řídicí systém:
  - Mikropočítač Arduino Mega 2560
  - Zdroj 12V DC (galvanicky oddělený)
  - Stabilizátor napětí nastavitelný 0-12V (pro napájení mikropočítače nastaven na 7V)
  - Prvky ochrany vstupů mikropočítače
  - 8 Relé shield (panel 8-mi relé do 250V, 10A)
  - D/A převodníky
- Ovládací prvky na panelu a jejich funkce:
  - Tlačítko AA: rychlý posuv A
  - Tlačítko BB: rychlý posuv B
  - Tlačítko A: obrábění A
  - Tlačítko B: obrábění B
  - Tlačítko Z: zastavit posuv
  - Tlačítko Konec: ukončení obrábění
  - Tlačítko TTS: total stop – zastavení stroje mikropočítačem i rozepnutí ovládacích relé
  - Tlačítko Směr ot: volba směru otáčení vřetena
  - Rotační enkoder s funkcí stisku: pro ovládání operačního programu.
- Snímače:
  - Koncový spínač XA
  - Koncový spínač XB
  - Koncový spínač YA
  - Koncový spínač YB
  - Koncový spínač ZA
  - Koncový spínač ZB
  - Snímač zařazení směru pojezdu v ose X
  - Snímač zařazení směru pojezdu v ose Y
  - Snímač zařazení směru pojezdu v ose Z
- Signalizační a komunikační prvky:
  - LED A: kontrolka při aktivním pojezdu směrem A
  - LED B: kontrolka při aktivním pojezdu směrem B
  - LED integrovaná: signalizuje stav stroje rychlostí blikání
  - LCD displej 20x4 znaků na ovládacím panelu, připojen rozhraním I2C
- Spotřebiče:
  - Elektromotor vřetena
  - Frekvenční měnič VYBO VECTOR V800 pro elektromotor vřetena
  - Elektromotor pojezdu stolu
  - Spojka pomalého pojezdu
  - Spojka rychlého pojezdu
  - Elektromotor čerpadla chladicí emulze

### 2.3.2 Návrh elektronického zapojení

Při návrhu elektronického zapojení a řídicího systému je třeba vycházet v první řadě z bezpečnostních předpisů, Viz Obrázek 5: Zjednodušené schéma elektrického zapojení. Např. tlačítko „Total stop“ musí uvést stroj do bezpečného stavu i nezávisle na mikrokontroléru.



Obrázek 5: Zjednodušené schéma elektrického zapojení

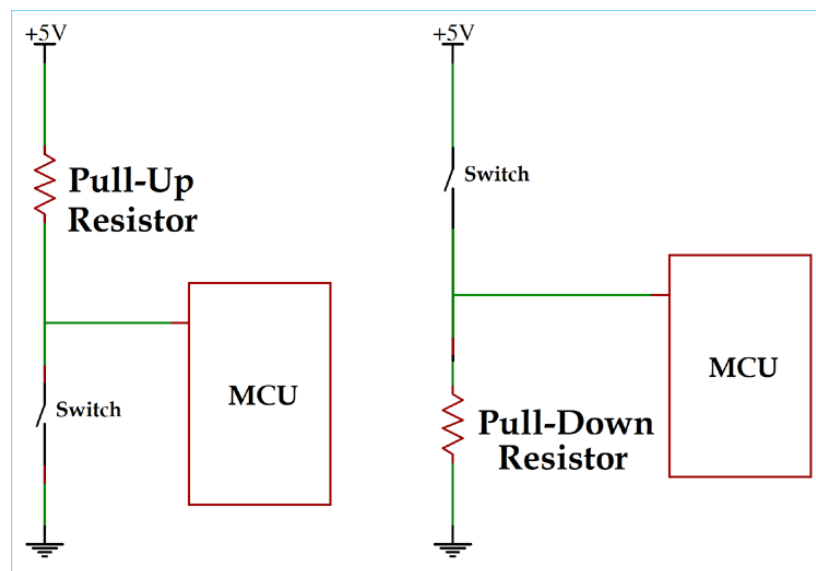
### 2.3.3 Ochrana mikropočítače

V odborných publikacích se považuje za takřka nereálné a vysoce neprofesionální připojit kontakty mikropočítače bez ochrany, a to zejména v tomto případě, kdy celý řídicí systém je v okolí vodičů a spotřebičů připojených k 400V AC.

Z variant ochrany mikropočítače, které se věnuje teoretická část, byly vybrány ty s nejlepším poměrem ceny, jednoduchosti, spolehlivosti a bezpečnosti.

- **Zapojení digitálních vstupních pinů**

Vstupní piny jsou zapojeny stíněným vedením přes tzv. PULL-DOWN a PULL-UP rezistory. Ty udržují hodnotu napětí na pinu v rozmezí 0 až 5 V a chrání proti indukci napětí, které může být mikropočítačem mylně vyhodnoceno jako logická 1, a snižují riziko indukce nebezpečně vysokého napětí, které by mohlo poškodit mikropočítač. Současně považuji za důležité provádět důmyslným způsobem „čtení“ hodnot napětí na vstupním pinu mikropočítačem tak, aby případný jeden nebo dva chybné odečty napětí nezpůsobily samovolné spuštění chodu stroje.



Obrázek 6: Schéma zapojení Pull-Up / Pull-Down rezistoru

- **Zapojení digitálních výstupních pinů**

Malé, zpravidla amatérské projekty umožňují napájet menší spotřebiče (např. LED diody) přímo výstupním digitálním kontaktem mikropočítače, nikoli však spotřebiče s větším odběrem, jakými bezpochyby jsou například elektromagnetické spojky.

Mezi výstupními digitálními kontakty mikropočítače je panel s osmi relé, napájený přímo z usměrňovače napětí na 7V.

- **Zapojení analogových výstupních pinů**

Za výstupními analogovými kontakty jsou D/A převodníky, které upravují výstupní pulzní signál z mikropočítače na stejnosměrný elektrický proud úměrný charakteristice pulzů.

Pro spojky D/A převodníky pracují s napětím 12V / 3A. Pro frekvenční měnič pracují v rozsahu 0 – 10 V. Frekvenční měnič dle této hodnoty nastaví frekvenci střídavého proudu pro elektromotor vřetena a tím mění jeho otáčky.

- **Uzemnění**

Pro elektroniku, obzvláště tu logickou, je velice důležité uzemnění a stínění, respektive pečlivé propojení všech zemnicích kontaktů všech aktivních prvků řídicího systému. Jedině tak se zamezí vzniku nežádoucího napětí mezi prvky a docílí správné úrovně nulového napětí. Celý řídicí systém je sestaven na uzemněném ocelovém plechu, navíc všechny zemnicí kontakty jsou propojeny do zemnicí svorkovnice určené pro slaboproudý obvod.

## 2.4 Řídicí program

### 2.4.1 Základní požadavky

- Sledování stavu tlačítek ovládacího panelu
- Kontrola krajních poloh posuvu stolu
- Nepřetržitá kontrola hodnot na odpovídajících vstupech (alespoň 10x/s)
- Spuštění režimů po stisku tlačítka
- Spuštění podprogramů po uplynutí časových intervalů
- Analogová komunikace s frekvenčním měničem
- Okamžitá reakce na stisk tlačítka „Total stop“
- Interní operační program ovládaný rotačním enkodérem
  - Nastavení otáček hlavního motoru (vřetena)
  - Zapnutí/vypnutí chlazení
  - Nastavení časového intervalu pro náběh
  - Nastavení časových intervalů pro doběhy

### 2.4.2 Struktura programu

Program využívá objektově orientované programování (OOP) v jazyce C++. Objekty zde jsou třídy, podtřídy a jejich instance, třídy mají své proměnné i metody. Důvodem volby této struktury je celková přehlednost programu, kratší kód a snazší úpravy a ladění programu.

Důraz struktury celého programu je zejména na přehlednost a jednoduchost hlavní smyčky programu „LOOP“, ve které program neustále pracuje a ze které se odkazuje na různé metody. Tyto jsou tvořené s ohledem na rychlost spolehlivost.

Hlavní myšlenkou je, aby mikropočítač sledoval v danou chvíli vždy jen ty hodnoty, které jsou aktuálně důležité. Ty, které důležité nejsou, v danou chvíli ignoruje. Příklad: Když stroj pracuje v ose X, pak ignoruje koncové snímače polohy stolu v ose Y a Z.

Zároveň tím předchází zbytečným a nebezpečným projevům například chyb obsluhy, kdy například při v režimu obrábění na stranu A ignoruje stav tlačítka pro rychlý posuv směrem B.

### 2.4.3 Třídy a podtřídy

- Pin\_contact (Třída všech kontaktů mikropočítače)
  - Vystup
  - Vstup
    - Vstup\_pullup
- Menu (Třída editovatelných položek v menu)
- Rezim (Třída pracovních režimů)



#### 2.4.4 Časové intervaly

Časový interval „Náběh“ slouží k rozběhu hlavního vřetena před spuštěním posuvu při obrábění. Doběh slouží k zastavení posuvu a dokončení odběru třísky, až poté se vypne hlavní motor (vřeteno).

Během časových intervalů je však potřeba, aby mikropočítač stále reagoval na podněty (například změna směru nebo rychlosti posuvu při obrábění, ukončení obrábění, Total stop, ...)

Použití jednoduché funkce „delay()“ je nevhodné, neboť ta v podstatě paralyzuje celý systém po celou dobu časového intervalu a mikropočítač nereaguje na žádné podněty.

Aplikace časových intervalů je proto řešena metodou opakovaného sledování času v milisekundách prostřednictvím funkce „if“ a „millis()“. Průběh této varianty v programu lze popsat takto:

*„Dokud neuplyne požadovaný časový interval, mikropočítač pokračuje ve své práci dále. Teprve když časový interval uplyne, spustí zvolený podprogram“.*

#### Ukázka části kódu ve smyčce „LOOP“ v režimu doběhu po obrábění:

```
if (millis() > dobeh_obr.cas + dobeh.hodnota) {  
    konec_obr.aktivuj();  
}
```

Popis:

*dobeh\_obr.cas*            číselná proměnná - uložen čas povelu „Doběh“ funkcí *millis()*

*dobeh.hodnota*        číselná proměnná - výchozí interval, editovatelný

*konec\_obr.aktivuj()* podprogram uvádějící stroj do klidu

### 2.4.5 Čtení digitálních hodnot na vstupních kontaktech

V teoretické části jsou uvedeny důvody, proč není vhodné nechat mikroprocesor číst hodnoty na kontaktních pinech připojených k tlačítkům přímo a okamžitě provádět dané příkazy.

Zde je zvolen systém čtení digitálních hodnot na vstupních kontaktech tímto způsobem:

- Každý pin má díky OOP mimo jiné tyto veličiny:
  - Def – očekávaná defaultní logická hodnota
  - Act – očekávaná logická hodnota v aktivním stavu
  - Last – poslední číselná hodnota
- Dále má každý pin díky OOP tyto metody:
- Změna: Porovnává pomocí funkce `digitalRead()` aktuální číselnou hodnotu s proměnnou Last – pokud se liší, vrátí „true“
- Změna\_2: Pomocí cyklu měří digitální hodnotu n-krát po sobě. Pokud změní n-krát hodnotu rozdílnou od Last, vrátí „true“. Další takové vyhodnocení je možné až po uplynutí intervalu 100 ms.
- Stisk: tato metoda volá nejprve funkci „Změna“. Když se vrátí „true“, volá funkci „Změna\_2“. Když se vrátí „true“ a naměřená hodnota je aktivní, považuje se tlačítko za stisknuté.
- Hodnota – měří digitální hodnotu na pinu dokud nenaměří n-násobek hodnoty buď v aktivním nebo defaultním stavu. Při naměření aktivní hodnoty vrátí „true“. Při naměření defaultní hodnoty vrátí „false“.

### Ukázka části kódu s metodami vstupích pinů :

```
boolean vstup::zmena() {
    if (digitalRead(pin) != last) return true;
    else return false;
}
```

```
boolean vstup::zmena_2() {
    if (millis() > time + 100) {
        celkem = 0;
        for (i = 0; i < n; i++)
        {
            celkem += (digitalRead(pin));
        }
        if (celkem == n * act) {
            stat = true;
            last = act;
            time = millis();
            return true;
        }
        else if (celkem == n * def) {
            stat = false;
            last = def;
            time = millis();
            return true;
        }
        else return false;
    }
    else return false;
}

boolean vstup::stisk() {
    if (zmena() && zmena_2() && stat) return true;
    else return false;
}

boolean vstup::hodnota() {
    while (0 < celkem < n) {
        celkem = 0;
        for (i = 0; i < n; i++)
        {
            celkem += (digitalRead(pin));
        }
        if (celkem == n * act) {
            stat = true;
            last = act;
            return true;
        }
        else if (celkem == n * def) {
            stat = false;
            last = def;
            return false;
        }
    }
}
```

## 2.4.6 Pracovní režimy (pro stroj)

Stroj během provozu pracuje v různých „režimech“. V každém takovém režimu obsluhuje různá výstupní zařízení a odečítá logické hodnoty ze vstupních zařízení souvisejících právě s daným režimem.

Aktuální pracovní režim stroje je vždy zobrazen na LCD displeji (první řádek)

- Total stop
  - Aktivován stiskem tlačítka TTS (monitorováno nepřetržitě)
  - Zastaví stroj a vypne všechna výstupní zařízení
  - Signalizuje stav rychle blikající kontrolkou na mikropočítači
  - Po uvolnění tlačítka TTS aktivuje režim „Připraven“
  - Tento režim má přednost před všemi ostatními
- Připraven
  - Výchozí stav, kdy není aktivní žádný jiný režim ani stisknuté žádné tlačítko.
  - Při aktivaci tohoto režimu zkontroluje, že jsou výstupní zařízení vypnutá. Pokud některé vypnuté není, vypne ho a hlásí chybu na displeji.
  - Monitoruje stav koncových snímačů polohy stolu, hlásí na displeji.
  - Snímá stav tlačítek A, AA, B, BB.
- Rychlý posuv směrem A/B
  - Aktivován tlačítkem AA/VB
  - Rozsvítí kontrolku A/B
  - Sepne spojku pro rychlý posuv
  - Zapne motor směrem A/B
  - Snímá koncové snímače krajních poloh posuvu stolu
  - Po uvolnění tlačítka AA/BB aktivuje režim „Konec obrábění“
  - Rychlý posuv opačným směrem je možný až po uplynutí časové prodlevy 1500 ms
- Náběh obrábění A/B
  - Aktivován stiskem tlačítka „Obrábění A/B“
  - Zapne hlavní motor (vřeteno) podle polohy přepínače směru otáček a aktuálně nastavené hodnoty rychlosti.
  - Vypne pojezd stolu
  - Aktivuje spojku pro pomalý pojezd stolu
  - Zapne kontrolku pro směr A/B
  - Při stisku tlačítka „Z“ aktivuje režim ručního obrábění a stornuje náběh.
  - Při stisku tlačítka „Konec“ ukončí pohyb vřetena.
  - Po uplynutí časového intervalu aktivuje režim „Obrábění A/B“
- Obrábění A/B
  - Zapne posuv směrem A nebo B
  - Snímá stav tlačítka pro opačný směr obrábění (A/B) – po stisku aktivuje „Náběh obrábění A/B“
  - Snímá koncové snímače krajních poloh posuvu stolu

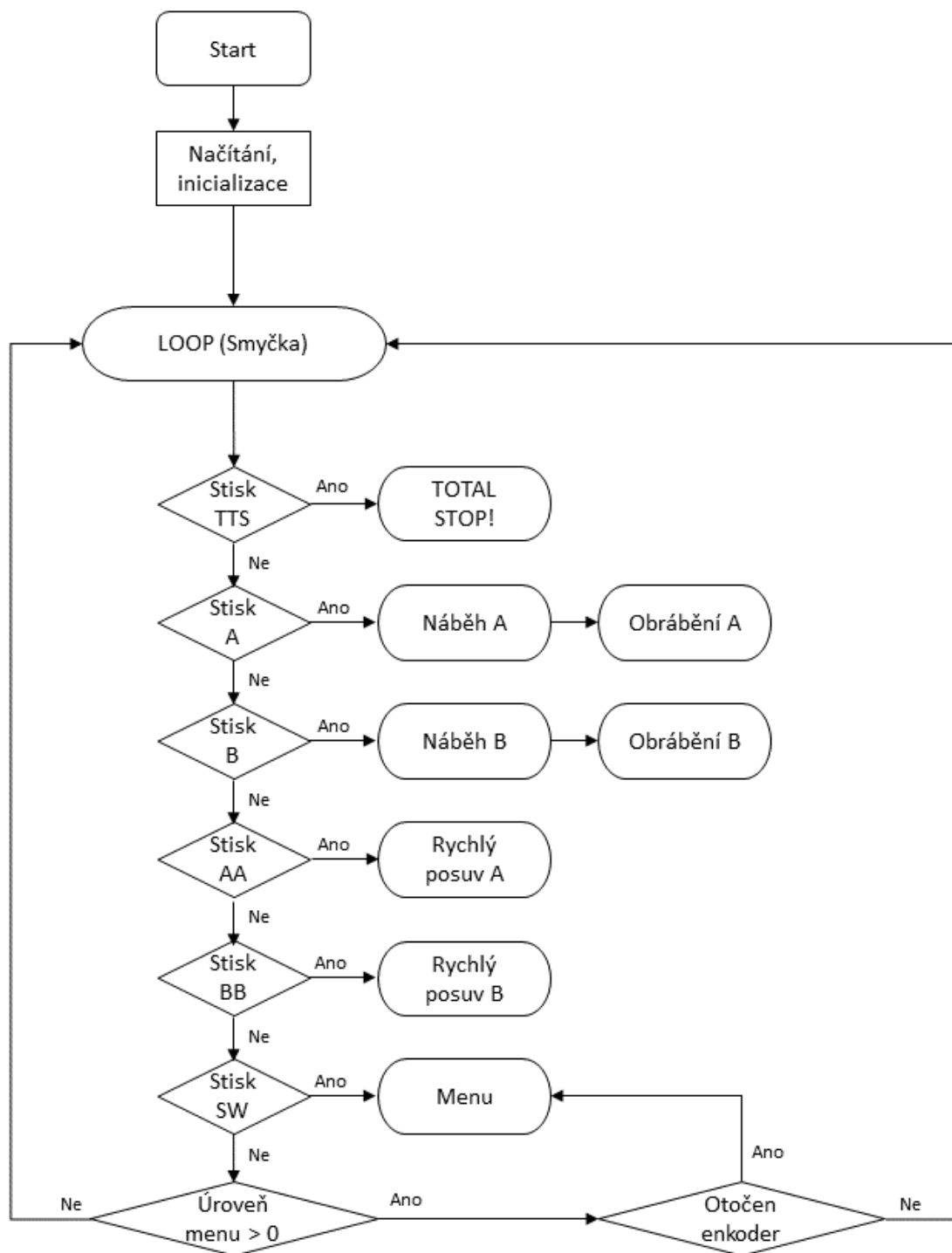
- Snímá stav tlačítka pro rychlý posuv totožným směrem (AA/BB) – po stisku aktivuje režim rychlého obrábění tímtéž směrem
- Při obrábění A v ose Y stisk tlačítka BB aktivuje „Odjezd Y“
- Dále snímá stav tlačítek Konec, Z
- Rychlé obrábění A/B
  - Vypne spojku pro pomalý posuv
  - Sepne spojku pro rychlý posuv
  - Zapne motor posuvu stejným směrem
  - Snímá koncové snímače krajních poloh posuvu stolu
  - Dále snímá stav tlačítek Konec, Z
  - Po uvolnění tlačítka AA/BB aktivuje „Náběh obrábění A/B“
- Doběh
  - Vypne motor pro posuv stolu
  - Zhasne kontrolky A, B
  - Snímá stav tlačítek A, B
  - Po uplynutí časového intervalu aktivuje režim „Konec obrábění“
- Konec obrábění
  - Ukončí pohyb stolu
  - Zastaví vřeteno
  - Zhasne kontrolky A, B
  - Vypne spojky pro pomalý i rychlý posuv
  - Aktivuje režim připraven
- Odjezd Y
  - Ukončí pohyb stolu
  - Zhasne kontrolku A
  - Zapne kontrolku B
  - Po časovém intervalu „doběh“ zapne rychlý posuv směrem B
  - Po časovém intervalu „Odjezd Y“ aktivuje „Konec obrábění“
- Ruční obrábění
  - Aktivován tlačítkem Z při obrábění
  - Vypne posuv stolu
  - Vypne spojky pro pomalý i rychlý posuv
  - Zhasne kontrolky A, B
  - Vřeteno ponechá v pohybu
  - Snímá stav tlačítek A, B, Konec.

### 2.4.7 Pracovní režimy (pro operační program)

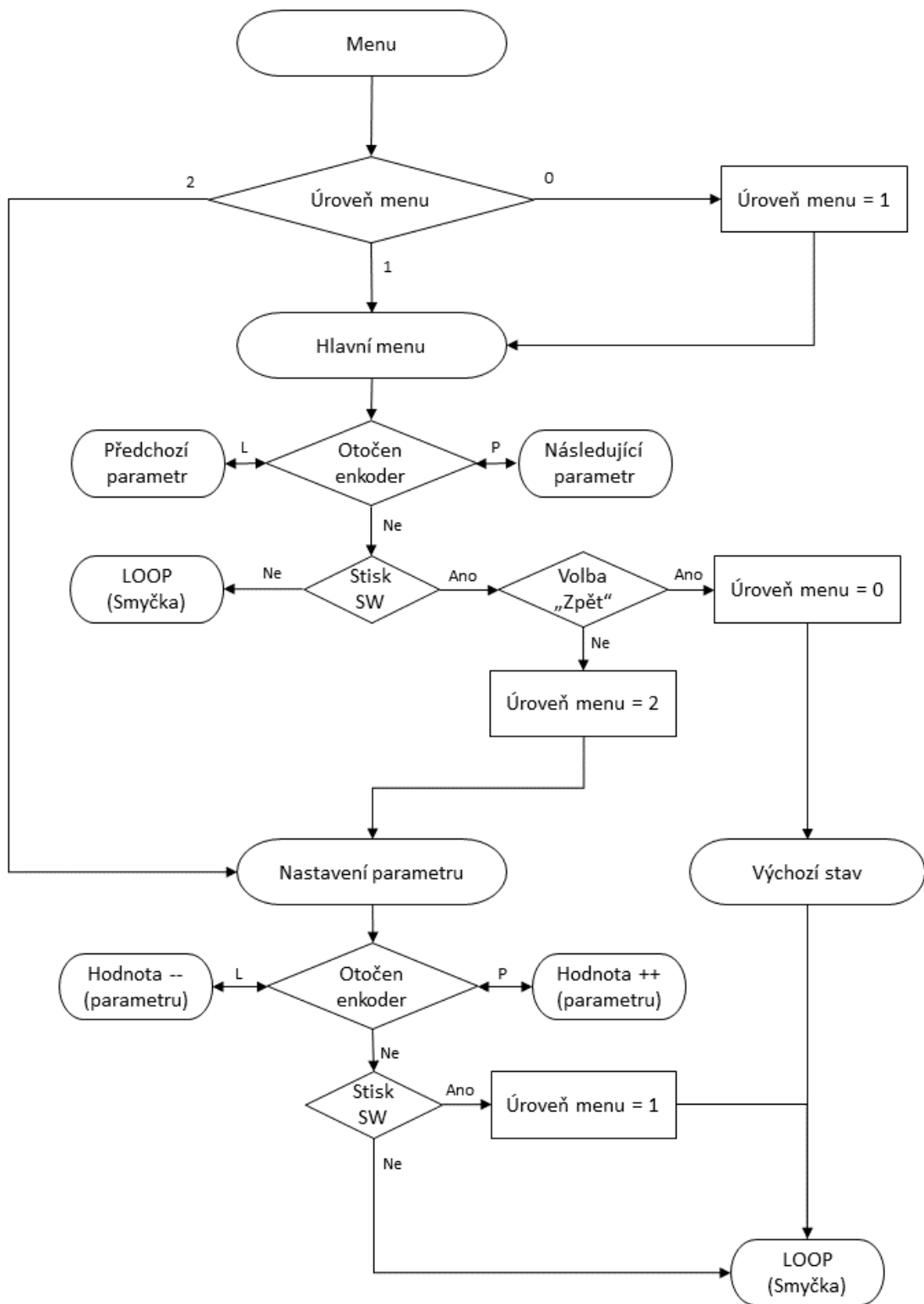
Operačnímu programu slouží 3. a 4. řádek LCD displeje, kde je zobrazena úroveň operačního programu a níže editovaná veličina a její hodnota včetně jednotek.

- Výchozí stav
  - Základní stav
  - Stisk tlačítka enkoderu vyvolá hlavní menu
- Hlavní menu
  - Zobrazeno hlavní menu a položky k editaci
  - Otočení enkoderu přepíná mezi položkami
  - Stisk tlačítka enkoderu vybere položku k nastavení její hodnoty
  - Výběr položky zpět aktivuje výchozí stav
- Nastavení
  - Otočení enkoderu mění hodnotu položky o její přírůstek
  - Stisk enkoderu ukončí nastavení a aktivuje hlavní menu

## 2.4.8 Vývojový diagram (zjednodušený)



Obrázek 7: Vývojový diagram základního pracovního režimu



Obrázek 8: Vývojový diagram interního menu pro úpravy parametrů



#### **2.4.9 Ověření funkčnosti programu, ladění výkonnosti**

Před nasazením nového řídicího systému přímo do stroje je třeba odstranit možné chyby. Je vhodné sestavit zjednodušený model celého stroje a na něm program otestovat. Při té příležitosti lze i ladit výkonnost programu ve všech režimech, tedy rychlost vykonání jednorázových, ale i rutinních operací.

K měření časového úseku během vykonání jednorázové úlohy může posloužit funkce `millis()`, kterou je zaznamenán strojní čas před instrukcí a po ní. Rozdíl naměřených časů lze následně zobrazit na displeji nebo prostřednictvím sériové komunikace na PC.

K měření rychlosti rutinních operací lze využít jeden digitální výstup a do hlavní smyčky programu LOOP přidat příkazy ke změně hodnoty daného digitálního výstupu. V praxi tak bude mikropočítač při každé smyčce opakovaně střídat logické nuly a jedničky. Výsledná výkonnost přímo souvisí s generovanou frekvencí pulzů, kterou lze měřit například osciloskopem, nebo digitálním multimetrem s funkcí měření frekvence.

### 3 Diskuse

Hodnocení výsledků daného řešení je možné jednak na základě změřené výkonnosti nového řídicího systému a dále porovnáním spolehlivosti řídicího systému celého stroje před a po generální opravou.

Měření výkonnosti metodou časových rozdílů při konání jednorázových úloh bylo zjištěno, že ty nejnáročnější úlohy trvají mikropočítači maximálně 200 ms. Měření frekvence na testovacím výstupu ukazuje ve všech stavech hodnoty mezi 1,9 kHz a 9 kHz. Znamená to tedy, že mikropočítač provede všechny požadované rutinní operace minimálně 3800-krát každou sekundu.

Z technického hlediska je prodleva při jednorázových operacích způsobena zejména malou rychlostí sériové komunikace I2C při přenosu instrukcí LCD displeji. K radikálnímu snížení prodlevy během ladění programu přispělo omezení počtu přenášených příkazů LCD displeji na minimum. Tedy dávat příkazy ke zobrazení pouze těch obrazových políček a jen tehdy, když je třeba dané políčko změnit. Prakticky je jednorázová prodleva maximálně 200 ms při přechodu mezi pracovními režimy přijatelná a při práci stroje není patrná. Počet opakování všech rutinních úkonů za sekundu, které stroj během jednotlivých pracovních režimů provádí, je více než uspokojivý. Vysoká frekvence snímání digitálních vstupních kontaktů, a to v každém pracovním režimu, zaručuje například bezchybné snímání otáček rotačního enkoderu i při rychlé manipulaci s ním.

Zhodnocení spolehlivosti nového řídicího systému je možné na základě již více než 18-ti měsíčního provozu stroje po generální opravě. Během té se vyskytly pouze mechanické závady na původních částech (například zaseknutý koncový spínač polohy stolu vlivem nečistot nebo opotřebovaná matice svislého pojezdu).

## Závěr

Záměrem teoretické části byl obecný popis průmyslových strojů a historický vývoj způsobů jejich ovládání. Dále měl být laicky představen obecný princip funkce mikropočítačů a uvedeny praktické příklady využití jejich funkcí k řízení průmyslových strojů.

V praktické části je třeba objektivně uznat, že při dnešní konkurenci na trhu s CNC obráběcími stroji tento obráběcí stroj, respektive výroba nového stroje s těmito parametry a funkcemi, nemůže obstát. Svůj účel však tato nízkonákladová generální oprava letitého stroje v rodinné firmě splnila. Byly obnoveny původní funkce stroje, bylo přizpůsobeno ovládání stroje k dosažení větší přesnosti a rychlosti práce. Navíc přibýly funkce nové, které zefektivňují práci při malosériové výrobě potrubních dílů (nastavení časových intervalů pro náběh a doběh, regulace otáček frekvenčním měničem, automatické spínání toku chladicí směsi a nový pracovní režim „Odjezd Y“).

V úvahu připadalo i zkonstruování snímání polohy a řízení pojezdu stroje v jedné ose. Do programu v mikropočítači by jistě bylo možné tyto funkce vnést díky přehledné struktuře objektově orientovaného programování, kdy by stačilo na základě nastavitelného programu na LCD displeji pouze „volat“ již hotové instance, tedy podprogramy představující jednotlivé pracovní režimy a obsluhu výstupních zařízení. Snímání polohy by zřejmě též bylo možné díky dostatečné rezervě ve výkonnosti aktuálního řešení. Avšak byla by nutná rozsáhlá konstrukční úprava mechanismu pohonu a regulace pojezdu a značné investice do lineárních snímačů pohybu. Automatizace pojezdu na daném stroji proto zatím nebyla realizována.

Jsem si též vědom, že využití mikropočítačů v průmyslu běžně spočívá buď v koordinaci více zařízení, kdy každé koná svou úlohu s maximální „soustředěností“, nebo v použití jednoho mikropočítače, který však provádí vždy jen jednu činnost. Proto vlastní technické řešení, které využívá jediný mikropočítač k obsluze požadovaných funkcí souběžně, považuji za úspěšné, byť lehce atypické.

## Seznam použité literatury

- [1] J. Mareš, Industrializace Československa – její klady a zápory, Sborník Československé geografické společnosti, 1988.
- [2] J. Řasa a V. Gabriel, Strojírenská technologie 3 - metody, stroje a nástroje pro obrábění 1. díl, Praha: Scientia, 2000.
- [3] „Porovnání pneumatických, hydraulických a elektrických lineárních pohonů,“ E-konstruktor, 2015. [Online]. <https://e-konstruktor.cz/novinka/porovnani-pneumatickych-hydraulickych-a-elektrickych-linearnich-pohonu>. [Přístup získán 5 5 2019].
- [4] K. Kocman a J. Prokop, Technologie obrábění, Brno: VUT, 2001.
- [5] A. Tkáčová, Programování CNC strojů se systémem Fanuc, Brno: VUT, 2009.
- [6] F. Wagner a R. Novotný, Technika a programování NC strojů, Praha: Wahlberg, 1994.
- [7] J. Tlustý a J. Zelený, Číslicově řízené obráběcí stroje, Praha: SNTL, 1962.
- [8] M. Štulpa, CNC obráběcí stroje. 1. vydání, Praha: BEN, 2006.
- [9] J. a. k. Marek, Konstrukce CNC obráběcích strojů, Praha: MM publishing, 2010.
- [10] V. Váňa, Mikrokontroléry ATMEL AVR: programování v jazyce C : popis a práce ve vývojovém prostředí CodeVisionAVR C, Praha: BEN - technická literatura, 2003.
- [11] J. Jelínek, „Principy mikropočítačů,“ UHK, [Online]. [http://lide.uhk.cz/prf/ucitel/jelinji1/course/principy\\_pocitacu\\_axqrd.pdf](http://lide.uhk.cz/prf/ucitel/jelinji1/course/principy_pocitacu_axqrd.pdf). [Přístup získán 21 03 2019].
- [12] P. Galeta, Z. Veselá, P. Pospíšil a V. Krňávek, „Základní pojmy a terminologie techniky mikropočítačů,“ [Online]. <https://eluc.kr-olomoucky.cz/verejne/lekce/847>. [Přístup získán 18 03 2019].
- [13] D. Hanousek, „Napětí a overlocking,“ 14 3 2002. [Online]. [https://www.idnes.cz/hry/magazin/napeti-a-overclocking.A020313\\_napetiaoverclock\\_bw](https://www.idnes.cz/hry/magazin/napeti-a-overclocking.A020313_napetiaoverclock_bw). [Přístup získán 31 03 2019].
- [14] J. Robenek, „SN74LV1T: posun úrovní s logickou funkcí a jedním Vcc,“ 4 4 2014. [Online]. <https://vyvoj.hw.cz/mcu/sn74lv1t-posun-urovni-s-logickou-funkci-a-jednim-vcc.html>. [Přístup získán 02 04 2019].
- [15] Arduino.cc, „Keyboard Message,“ 09 07 2015. [Online]. <https://www.arduino.cc/en/Tutorial/KeyboardMessage>. [Přístup získán 03 04 2019].
- [16] D. Hankovec, „Sběrnice I2C, o co jde a jak pracuje,“ [Online]. <http://www.dhservis.cz/iic.htm>. [Přístup získán 03 04 2019].

- [17] V. Jáneš, „Jednočipové mikropočítače,“ ČVUT, 25 4 2012. [Online]. [https://www.fd.cvut.cz/personal/janes/hwpocitacu/prednasky2012/prednaska9/HW\\_prednaska9.pdf](https://www.fd.cvut.cz/personal/janes/hwpocitacu/prednasky2012/prednaska9/HW_prednaska9.pdf). [Přístup získán 15 4 2019].
- [18] Root.cz, „Vývojové nástroje používané v dobách osmibitových mikropočítačů,“ 19 6 2015. [Online]. <https://www.root.cz/clanky/vyvojove-nastroje-pouzivane-v-dobach-osmibitovych-mikropocitacu/>. [Přístup získán 03 04 2019].
- [19] S. Brno, „Procesory a jejich strojový kód,“ 5 8 2011. [Online]. [https://moodle.sspbrno.cz/pluginfile.php/3103/mod\\_resource/content/0/Procesory\\_a\\_jejich\\_strojovy\\_kod.pdf](https://moodle.sspbrno.cz/pluginfile.php/3103/mod_resource/content/0/Procesory_a_jejich_strojovy_kod.pdf). [Přístup získán 21 04 2019].
- [20] A. Vojáček, „Rotační enkodéry pro řízení pohonů a pohybu,“ 30 04 2017. [Online]. <https://automatizace.hw.cz/rotacni-ekodery-pro-rizeni-pohonu-a-pohybu.html>. [Přístup získán 05 04 2019].
- [21] A. Vojáček, „Analogový vs. digitální přenos hodnot. Kdy ještě volit analogový výstup?,“ 7 4 2015. [Online]. <https://automatizace.hw.cz/mereni-a-regulace-prumyslove-sbernice-a-komunikace/analogovy-vs-digitalni-prenos-hodnot-kdy-jeste-volit-analogovy-vystup>. [Přístup získán 06 04 2019].
- [22] P. Doleček, „Ošetření vstupů Arduina a jiných IO,“ 29 7 2018. [Online]. <https://www.mylms.cz/osetreni-vstupu-arduina-a-jinych-io/#delic>. [Přístup získán 2 5 2019].
- [23] P. Palacký, Vysoká škola báňská –Technická univerzita Ostrava MIKROPOČÍTAČOVÉ ŘÍDICÍ SYSTÉMY I, Ostrava: VŠB, 2007.
- [24] J. Dvořák, „Programování v jazyce Arduino C++,“ [Online]. <http://dvorkaman.asp2.cz/Creation/Programming/Arduino>. [Přístup získán 5 5 2019].
- [25] D. Čápka, „Úvod do jazyka C++,“ 29 10 2018. [Online]. <https://www.itnetwork.cz/cplusplus/kurz/zaklady/uvod-do-jazyka-cplusplus>. [Přístup získán 6 5 2019].
- [26] M. Locker, „6. Tlačítko podruhé,“ 18 1 2014. [Online]. <http://robotika.vosrk.cz/guide/arduino/lesson06/cs>. [Přístup získán 05 05 2019].
- [27] Z. Voda, „Užitečné funkce,“ 15 12 2014. [Online]. <https://arduino.cz/uzitecne-funkce-3/>. [Přístup získán 05 05 2019].
- [28] J. Boháč, „Arduino v příkladech - V.díl - Menu na LCD,“ Arduinotech.cz, 25 4 2016. [Online]. <https://www.arduinotech.cz/inpage/arduino-v-prikladech-iv-dil-manu-na-lcd/>. [Přístup získán 05 05 2019].
- [29] Arduino8.cz, „Arduino a EEPROM,“ 03 05 2013. [Online]. <https://arduino8.webnode.cz/news/lekce-17-arduino-a-eprom/>. [Přístup získán 22 03 2019].

## **Přílohy**

- [1] Disk CD: Zdrojový kód programu mikropočítače ve stroji Heckert FSS 230.