Vyšší odborná škola, Střední škola,
Centrum odborné přípravy, Sezimovo Ústí

Tvorba software pro mikrokontrolér PIC v přijímači dálkového ovládání

Zdeněk Přech

student 4.ročníku oboru Elektrotechnika – počítačové systémy

ŽÁKOVSKÝ PROJEKT - PRAKTICKÁ ČÁST MATURITNÍ ZKOUŠKY

Sezimovo Ústí 2010

VYŠŠÍ ODBORNÁ ŠKOLA
STŘEDNÍ ŠKOLA
CENTRUM ODBORNÉ PŘÍPRAVY
SEZIMOVO ÚSTÍ

Dne:

Schvalují:

Výtisk č.:

Počet listů:

ZADÁNÍ ŽÁKOVSKÉHO PROJEKTU - praktické části maturitní zkoušky pro školní rok 2009-2010

1. Číslo zadání (třída, číslo žáka – příklad SG4-02(0910)	2. Název a charakter práce (obor)
ET4B-12/0910	**Kapitola 1 Tvorba software pro mikrokontrolér PIC v přijímači**

3a. Odborný konzultant (firma)	Ing. Vladimír Čebiš	4. Zadavatel a vedoucí projektu
3b. Jazyková a stylistická úprava	Mgr. Jitka Hajšmanová	Ing. Vladimír Čebiš
3c. Cizojazyčné texty a anotace:	Mgr. Marie Carvová	

5. Řešitel (jméno, příjmení, třída, datum nar.)

Zdeněk Přech ***nar. 23.8.1990***

6. Charakteristika zadávané práce – projektu (definice problému, cíl a očekávané výsledky řešení, definice výchozích a omezujících podmínek, vstupní podklady pro řešení, uzlové body realizace)

1) Vytvořit software pro přijímač dálkového ovládání, který bude umožňovat následující funkce:
   - konfiguraci a volbu režimu pomocí sériové linky
   - autonomní funkce pro využití v robotice

2) Navrhnout a prakticky odzkoušet různé způsoby použití přijímače dálkového ovládání s různými funkcemi softwaru

(Pokračování na druhé straně)

7. Požadovaný termín odevzdání	8. Předpokládaný termín obhajoby projektu u maturitní zkoušky	9. Další formy vyhodnocení
6. 04. 2010	**Květen 2010**	1. *Odborný předmět* 2. *Obhajoba u maturity* 3. *Český jazyk a estetika* 4. *Cizí jazyk*

(Při nedostatku místa pokračovat na volných listech)

10. Požaduje se předložit

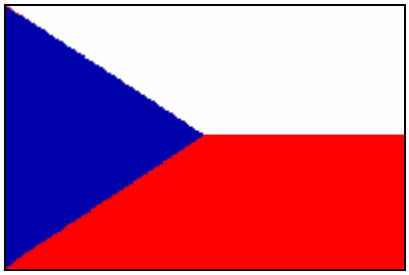
1. Zpracovaný projekt podle zadání COP v tištěné podobě **včetně české a cizojazyčné anotace** v rozsahu 1 strany A4 (25 – 35 řádek)
2. Zpracovaný reálný výstup z projektu (výrobek, materiál apod.), pokud byl součástí ŽP
3. Zpracovaný projekt podle zadání COP na CD (povinně v PDF, nepovinně další)
4. Zpracovanou prezentaci projektu na CD (povinně Power Point, nepovinně další)
5. Grafické ztvárnění projektu formou plakátu velikosti A3 (v PDF a tištěné podobě)
6. Přílohy dle uvážení autora nebo pokynů SOČ (pokud práce postupuje do SOČ)

11. Materiální zajištění (předpokládané náklady a podíl úhrady)

- Součástky v hodnotě do 200Kč hradí škola
- K dispozici je veškeré potřebné technické vybavení, včetně software školy
- Tisk potřebné dokumentace si bude student zajišťovat sám nebo v COP za obvyklou úhradu
- Práce bude zpracována převážně v rámci výuky předmětu PX PR dle tématického plánu v laboratoři

12. Vedoucí projektu :	**13. Zadání převzal (žák)**
Ing. Vladimír Čebiš	Zdeněk Přech
Datum 8. 10. 2009 **Podpis**	**Datum** 8. 10. 2009 **Podpis**

Anotace



Práce se zabývá problematikou tvorby softwaru v assembleru PIC pro vytvořené hardwarové zařízení. HW vytvořil a oživil David Černý ve školním roce 2007/2008 jako svůj maturitní projekt. Jedná se o přijímač dálkového ovládání s výstupem na sériovou linku RS232C. Autor vytvořil jednoduchý SW pro demonstraci základních funkcí svého zařízení.

Hlavním úkolem práce je vytvořit plně funkční verzi SW, která bude použitelná ve stavebnicovém řešení malých mobilních robotů pro účely jejich bezdrátového ovládání. Základní funkcí SW je detekce kódu vysílaného IR dálkovým ovladačem a zjištění čísla stisknutého tlačítka. Podle tohoto čísla SW vybere z paměti EEPROM PIC povel o délce 2-8 bajtů. Délku tohoto povelu SW samostatně zjišťuje podle tzv. systémového bajtu.

Dalším úkolem je obsluha sériové linky na straně příjmu. SW umožňuje měnit obsah paměti EEPROM, tedy reakci na stisknuté tlačítko na ovladači, podle pokynů zaslaných po sériové lince z nadřízeného systému. V tomto režimu by měl přijímač mít nastavenou svou adresu a realizovat dohodnutý protokol. Může být tedy poté zařazen jako prvek robotické sítě, používané pro stavebnicové řešení robotů.

SW byl prakticky vyzkoušen pro všechny kombinace tlačítek na dálkovém ovladači, různé délky povelů v EEPROM a různé adresy. SW s HW lze využít všude tam, kde je třeba zajistit odesílání nejvýše 31 povelů sestávajících ze 2-8 bajtů po sériové lince ovládané bezdrátově na přímou viditelnost do 10m.

Annotation

Diese Arbeit beschäftigt sich mit Software-Entwicklung in Assembler für das erstellte PIC-Hardware-Gerät. HW erstellte und belebte David Černý im Schuljahr 2007/2008 als sein Abiturprojekt. Dies ist eine Ferbedienung Empfänger für die Ausgabe an die serielle Schnittstelle RS232C. Der Autor schuf eine einfache Software, um die grundlegenden Funktionen seines Gerätes zu demonstrieren.

Die Hauptaufgabe der Arbeit ist, eine voll funktionsfähige Version der Software zu schaffen, die diese Verfahren in den modularen Lösungen für kleine mobile Roboter für die Zweckte der drahtlosen Steuerung benutzt. Die Grundfunktion der Software ist, den Cod den die IR-Fernbedienung emittiert, und die Identifikation der Nummer der gedrückten Taste zu erkennen. Nach dieser Nummer wählt die SW aus dem Speicher EEPROM PIC die Befehlslänge 2-8 Bytes. Die Länge dieses Befehls bestimmt die SW selbstständig nach dem so genannten Systembyte.

Eine weitere Herausforderung besteht in Bedienung der seriellen Schnittstelle auf der Empfangseite. Die Software ermöglicht, den Inhalt des EEPROM zu ändern, eine Antwort auf den gedrückten Controller, nach den Anweisungen, die über eine serielle Schnitt stelle von dem übergeordneten Systém gesendet sind. In diesem Modus hat der Empfänger seine festgelegte Adresse und realisiertden vereinbarten Protokoll. Es kann daher als ein Element des Roboter-Netzes, das für modulare Lösung der Roboter verwendet ist, eingestuft werden.

Sw war praktisch für alle Kombinationen der Tasten auf der Fernbedienung, für die unterschiedlichen Befehlslängen und für verschiedene EEPROM-Adressen getestet. SW mit HW kann dort eingesetzt, werden wo es notwendig ist, höchstens 31 Befehle, die aus 2-8 Bytes bestehen, über die serielle drahtlos kontrollierte Schnittstelle auf die direkte Sichtbarkeit von 10m zu senden.

Žákovský projekt byl zpracován v rámci řádného ukončení 4.ročníku maturitního studia Elektrotechnika – počítačové systémy. Vedoucím práce byl Ing. Vladimír Čebiš, kterému tímto děkuji za odborné konzultace a cenné rady týkající se struktury i obsahu práce.

Zároveň děkuji Ing. Jiřímu Bumbovi, za pomoc při realizaci a odladění částí kódu v assembleru PIC.

Prohlašuji, že jsem žákovský projekt vypracoval samostatně za použití odborné literatury a dalších zdrojů, uvedených v závěru práce.

V Sezimově Ústí 20. dubna 2010

Zdeněk Přech

Obsah

Tvorba software pro mikrokontrolér PIC v přijímači dálkového ovládání

1/ ÚVOD

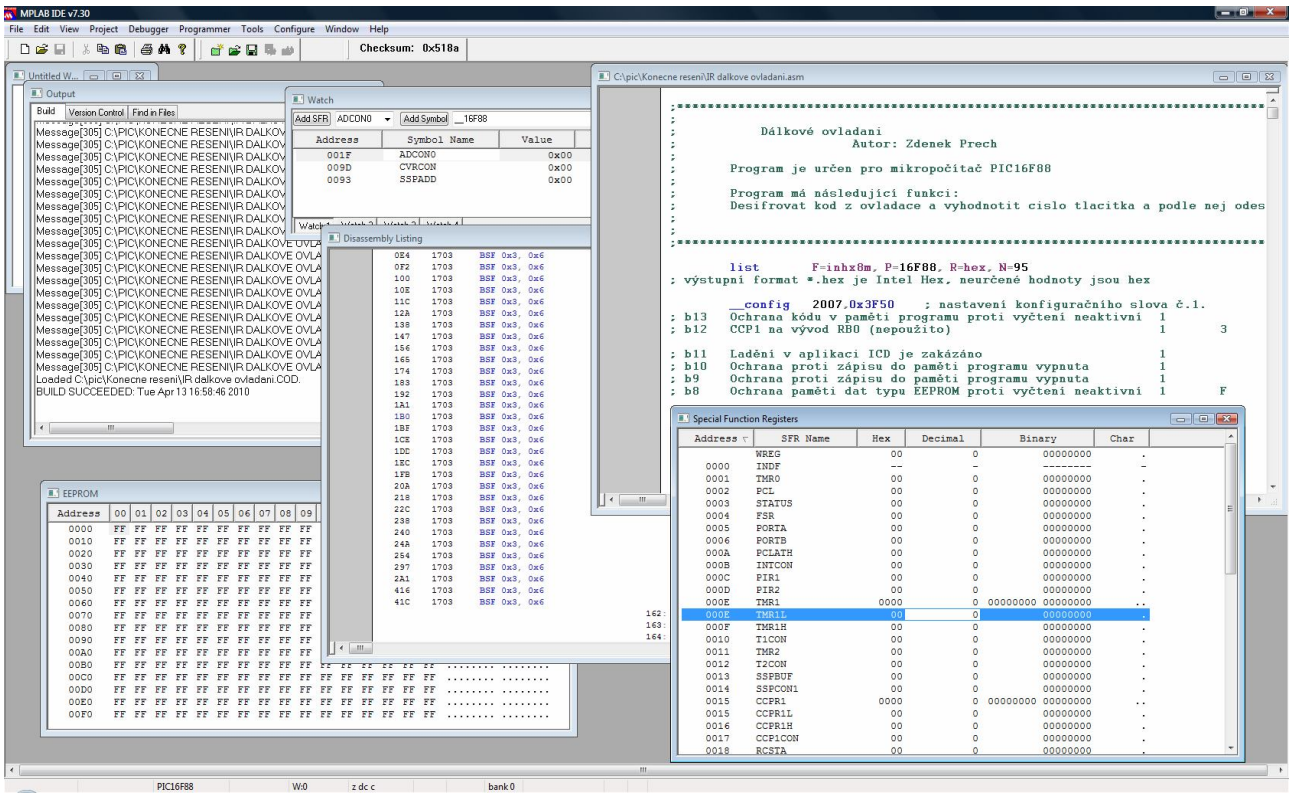
1.1. Popis vývojového prostředí MPlab

MPLab je vývojové prostředí pro MCU Microchip. Programy lze zde psát jak v klasickém ASSEMBLERU, tak neustále se rozšiřujícím jazyku C. Toto prostředí obsahuje programový editor pro vlastní psaní, simulátor pro odzkoušení programu v teoretické oblasti, a z ostatních funkcí například ICD, tj. In Circuit Debugger, kde se využívá vývojových desek Microchip.

Pro vlastní programování si vystačíme s editorem a simulátorem. Ostatní vývojové desky se dají koupit, ale není nezbytně nutné jich využívat.

Instalace

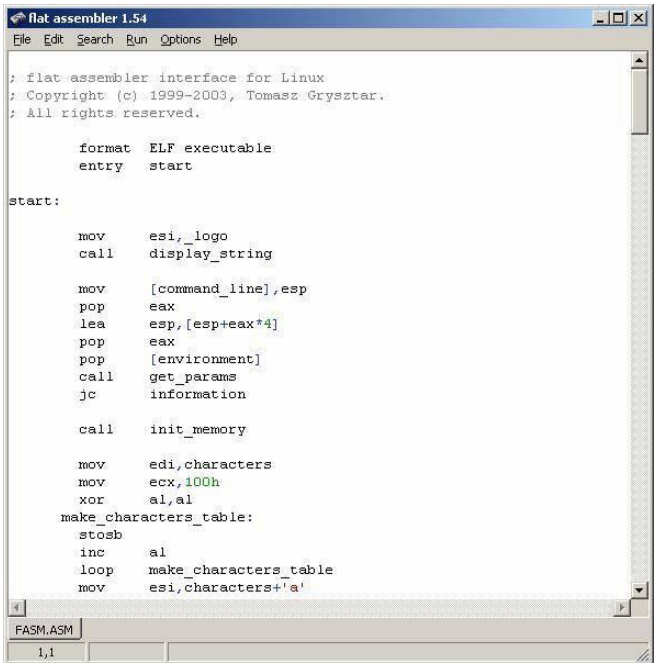
Vlastní instalace je velmi jednoduchá, při instalaci se instalátor pouze zeptá, jestli si přejete implementovat PICC, kompilátor pro jazyk C. Instalační soubor je na stránkách Microchip.

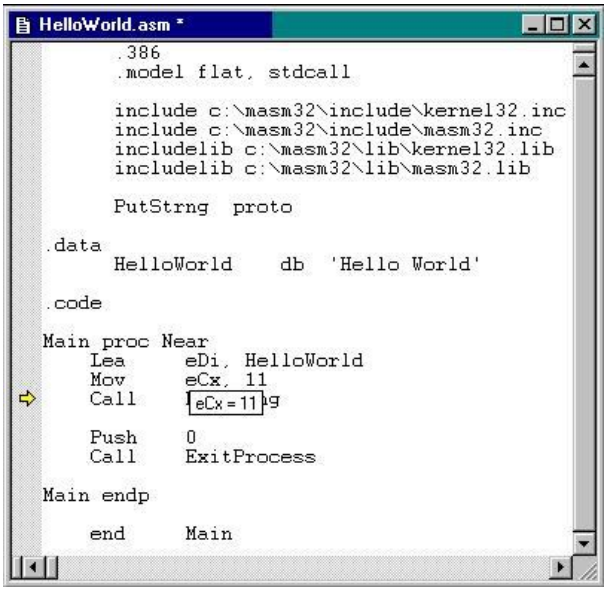


1.2. Programovací jazyk assembler

Assembler, jinak jazyk symbolických adres (zkratka JSA, anglicky assembly language) nebo také jazyk symbolických instrukcí je v informatice nízkoúrovňový programovací jazyk, který je tvořen symbolickou reprezentací jednotlivých strojových instrukcí a konstant potřebných pro vytvoření strojového kódu programu pro daný procesor. Symbolickou reprezentaci tvoří zpravidla výrobce procesoru a je založena na mnemotechnických zkratkách, které vyjadřují, co daná strojová instrukce dělá, označují symbolicky registr, slovní zkratku podmínky a podobně. JSA je proto závislý na konkrétním procesoru a zapsaný program je obtížně přenositelný na jinou platformu (na rozdíl od vysokoúrovňových programovacích jazyků).

Pro překlad JSA do strojového kódu se používá program, který nazýváme assembler (překladač). Oba názvy jsou často nesprávně zaměňovány.





Terminologie

Anglické slovo assembler znamená sestavovatel a označuje pouze překladač, tj. program, který sestavuje strojový kód. Programovací jazyk zpracovávaný takovým překladačem se označuje JSA v angličtině se jmenuje assembly language.

V praxi se ovšem velmi často (a zcela nesprávně) pro označení JSA používá termín assembler.

Charakteristika

JSA je programovací jazyk nejnižší úrovně a je závislý na strojovém kódu procesoru. Každá rodina procesorů má svůj vlastní odlišný JSA, protože ve strojových instrukcích různých rodin procesorů a možnosti rozdělování a adresování paměti bývají zásadní rozdíly. Každá firma vyrábějící procesory si definuje vlastní pravidla pro JSA svých procesorů, z kterých mohou (ale také nemusejí) vycházet nezávislí autoři a firmy.

Společným rysem drtivé většiny JSA je, že kódovou jednotkou je zde jeden řádek.

Program v JSA se skládá z:

překladových direktiv

- tyto direktivy ovlivňují způsob překladu (například pro jakou verzi procesoru se překládá, zda se ignorují velká a malá písmena, zda se generuje výpis a s jakým stránkováním, atp.). Rovněž označují začátek a konec kódových sekcí.

strojových instrukcí

-symbolicky zapsané strojové instrukce jsou při překladu nahrazeny odpovídajícím strojovým kódem

definic obsahu paměti

-můžeme inicializovat obsah paměti, nebo vyhradit v paměti místo pro proměnné

návěstí

-návěstí umožňují pojmenovat místa v paměti počítače. Návěstí umístěné před instrukcí se používá jako pro definici bodu v programu, na který můžeme skočit, návěstí umístěné před definicí obsahu paměti se používá při odkazování na tuto paměť

maker

-makra slouží pro nahrazení často používaných sekvencí instrukcí, umožňují zpřehlednit a zjednodušit kód vytvořením pseudoinstrukcí a formalizací často používaných konstrukcí

podmínkových bloků

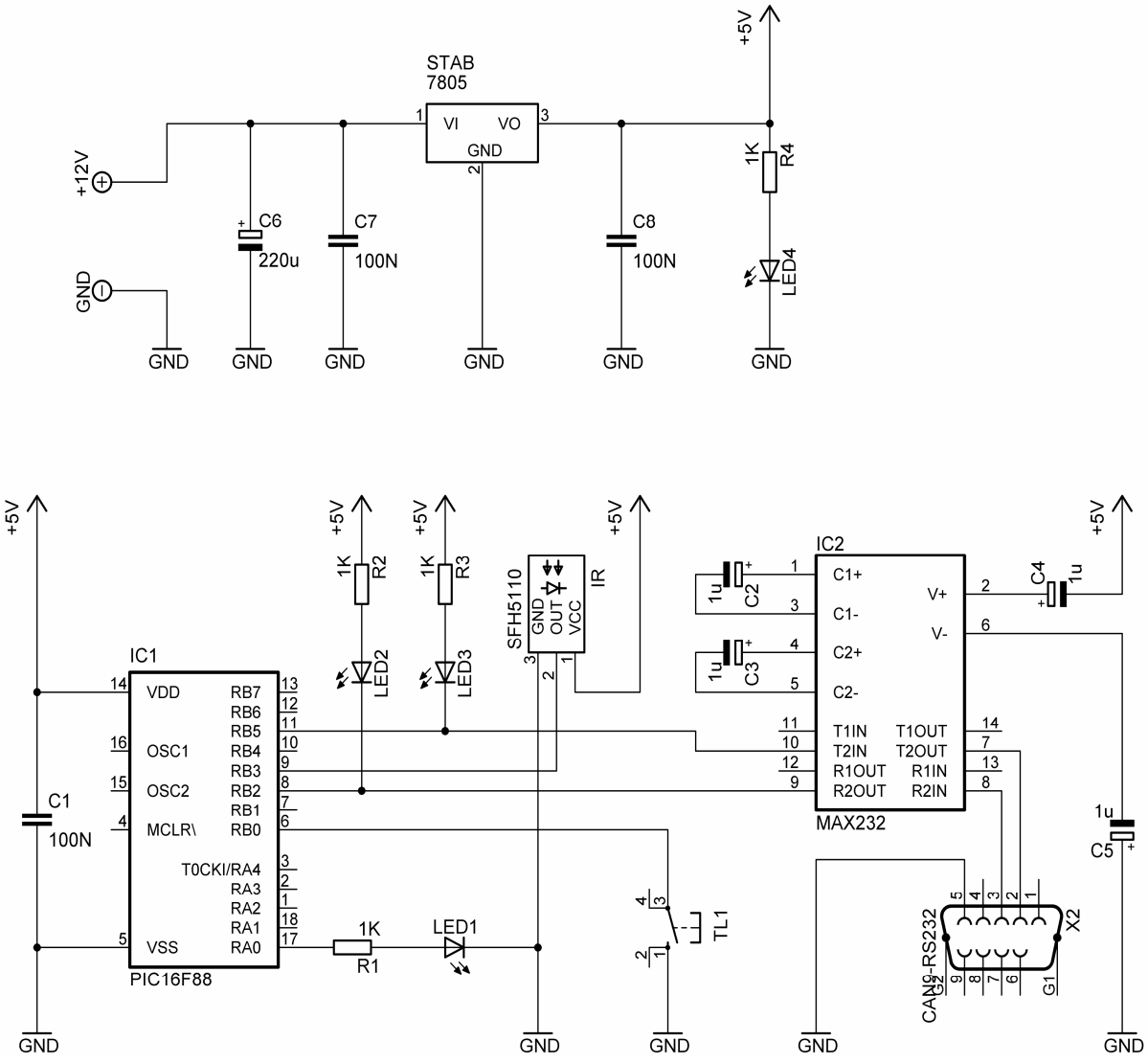
-podmínkové bloky dovolují generovat odlišný kód v závislosti na nastavení překladových symbolů, což může být užitečné například při ladění, nebo u kódu určeného pro více platforem

definic překladových symbolů

-překladové symboly pomáhají při vytváření dobře strukturovaného kódu programu

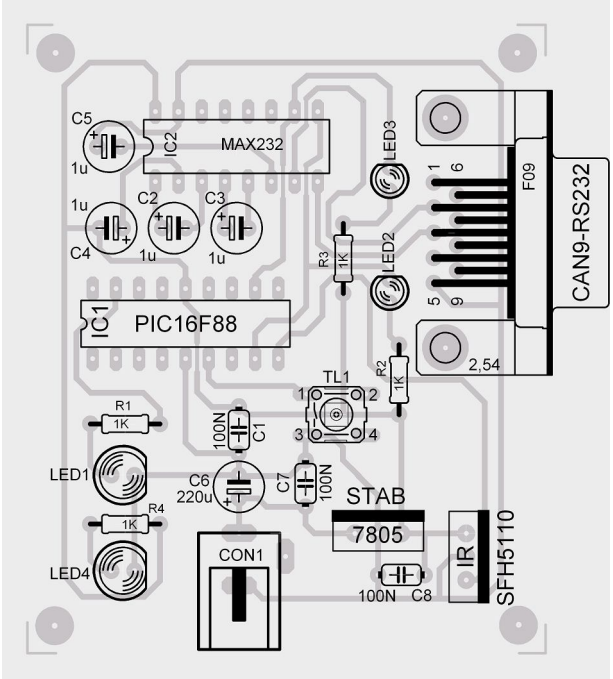
2/ Analýza informačních zdrojů

2.1. Popis HW řešení přijímače dálkového ovládání Davida Černého

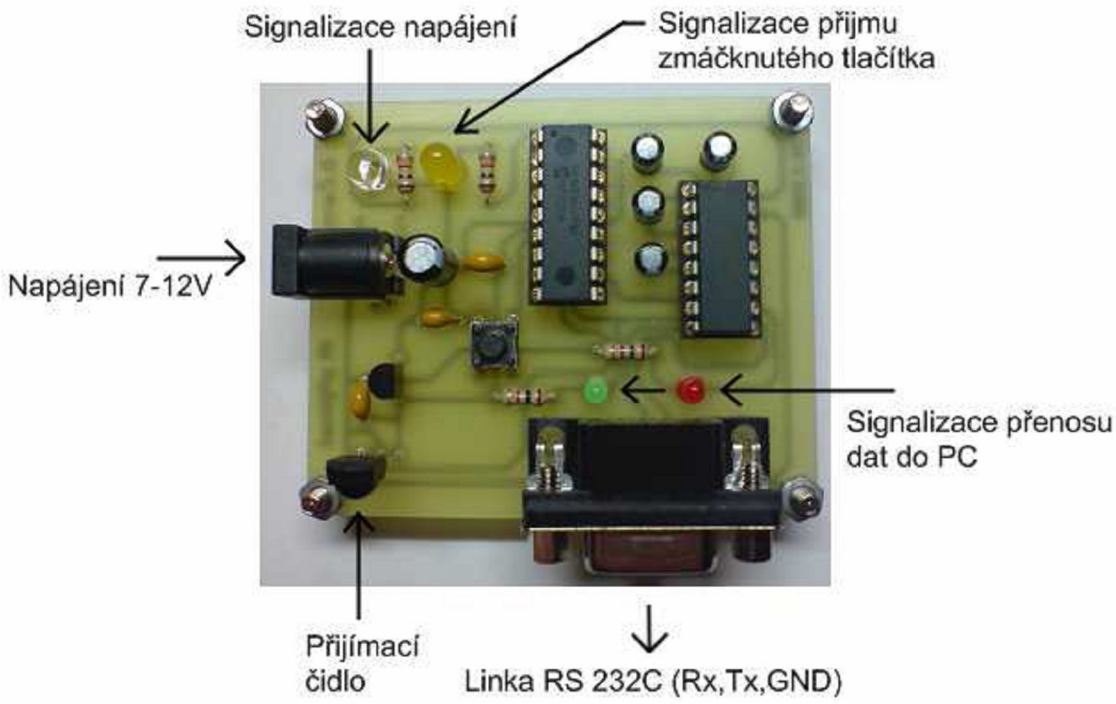


První částí je stabilizované napájení s napětím 5V. Jako vyhodnocovací jednotka je použit jednočipový mikropočítač PIC v němž je aktivován modul USART pro komunikaci s PC prostřednictvím linky RS232C. Pro převod úrovní je použit MAX232. Jako snímač IR signálu z ovladače je použito čidlo SFH5110. Pro signalizaci jsou použity LED diody různých barev. Modrá LED signalizuje napájení, žlutá LED přijatý signál z ovladače, červená LED signalizuje vyslané data po lince RS232C do PC.

Osazovací výkres



Fotografie finálního výrobku s popisem

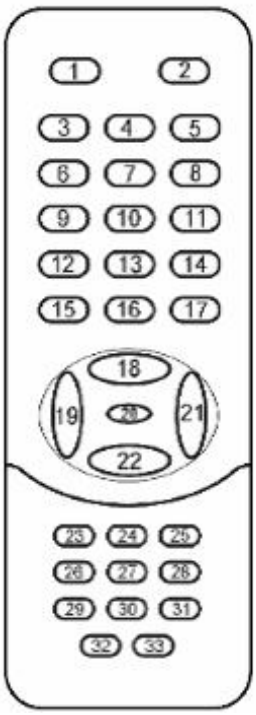


2.2. Popis oživovacího SW Davida Černého

David Černý se ve svém projektu zabývá základním oživením přijímače dálkového ovládání. Jeho SW dokáže zjistit, zda-li nastala synchronizace s dálkovým ovládáním. Dále pak sejme zákaznické číslo, které je přiděleno výrobcem a přijímač pozná jestli se opravdu jedná o dálkový ovladač k němu přidělený. Jako poslední krok proběhne číselná identifikace stisknutého tlačítka, která je následně odeslána po sériové lince.

Část tohoto programu byla využita v mém řešení daného problému. Zejména část se synchronizací a zákaznickým číslem.

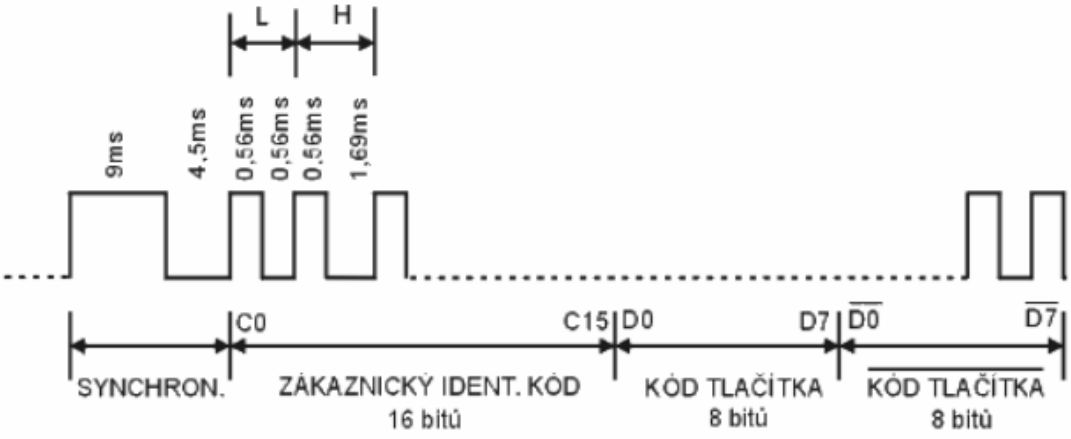
2.3. Univerzální dálkový IR ovladač IR33K1



Technické údaje – vysílač IR33K1:

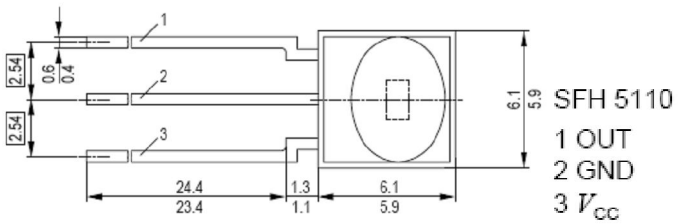
Napájení: 3V / 2 x mikrotužka AAA
Provozní napětí: 2,2V až 3,6V
Klidový odběr proudu: 20uA
Odběr proudu při vysílání: 40mA
Nosná frekvence: 38kHz
Rozměry: 144 x 47 x 20mm
Životnost tlačítek: 100 000 stisků
Pracovní teplota: -10°C až +40°C
Druh kódu: NEC 32 datových bitů
Počet tlačítek: 33

Tento vysílač odesílá kód typu NEC, který má 32 datových bitů. Tento kód je rozdělen na několik částí, jedna z nich je samotná synchronizace. Poté následuje identifikační kód, který má 16 bitů a určuje, zda se jedná o správný signál z daného vysílače. Jako poslední přijde kód tlačítka s osmi bity, který se pro kontrolu jednou zopakuje a neguje.

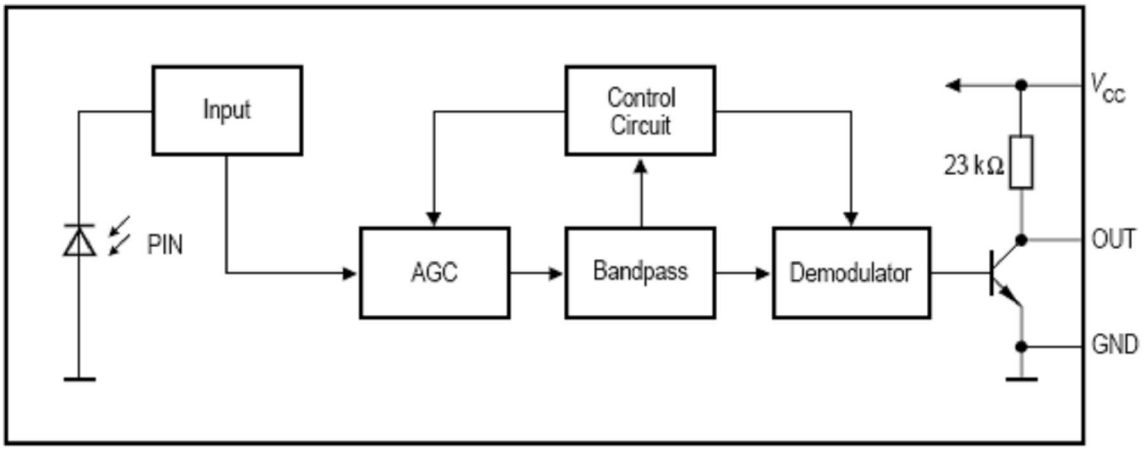


2.4. IR čidlo SFH 5110

Infračervený signál, namodulovaný na nosné frekvenci 38kHz, je zachycen a demodulován infra přijímačem. Pro detekci IR signálu jsem použil čidlo SFH 5110.
Toto čidlo je schopno převádět přijatý signál na logickou úroveň.



Blokové schéma IR čidla:



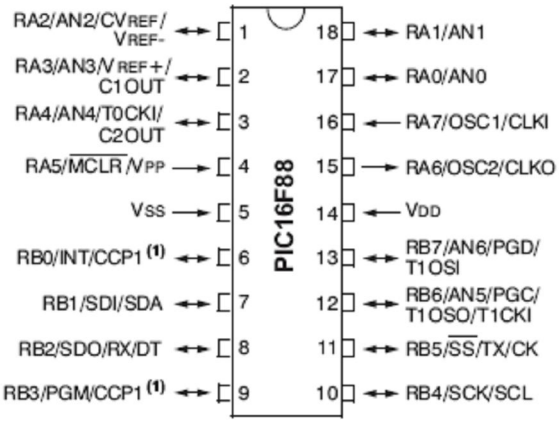
Pro demodulaci signálu je použit jednočipový mikropočítač PIC16F88. Pro tento mikropočítač jsem vytvořil a odladil funkční program pro rozeznání a dekódování přijatého signálu z vysílače.

Komunikace s PC proběhne pomocí sériové linky RS232C. Pro tuto linku je potřeba použít převodník úrovní MAX232. Jedná se o převodník TTL na RS232. Obsahuje dvě dvojice oddělovačů konvertujících napěťové úrovně. Napětí pro RS 232 se získává pomocí nábojové pumpy výstupní napětí proto značně závisí na kvalitě použitých kondenzátorů, která u elektrolytických kondenzátorů časem značně klesá.

Propojení pomocí linky RS232C jsou použity tři vodiče (Rx,Tx,GND). Přenosová rychlost je 9600Bd.

2.5. PIC 16F88

Jednočipový mikropočítač PIC:



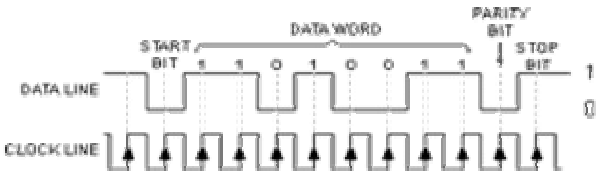
Mikropočítač	Paměť programu		Paměť dat		Vstupně výstupních vývodů (I/O)	Vstupů AD převodníku	Modulů CCP (PWM)	AUSART	Komparátorů	SSP	Čítače 8/16 bitů
	Bajtů	Instrukcí	RWM (bajtů)	EEPROM (bajtů)							
PIC16F88	7168	4096	368	256	16	1	1	ANO	2	ANO	2/1

Použitý modul USART/SCI:

Adresovatelný univerzální synchronní asynchronní přijímač vysílač AUSART je jedním ze dvou sériových komunikačních portů mikropočítače. AUSART bývá rovněž někdy označován zkratkou SCI (sériový komunikační interface). AUSART může být konfigurován jako plně obousměrný (duplexní) asynchronní komunikační port připojitelný například k portu COM osobního počítače. Druhá možnost konfigurace modulu AUSART je poloduplexní synchronní port pro připojení k periferním zařízením jako jsou externí AD převodníky, paměti nebo další integrované obvody.
Modul AUSART může být konfigurován v následujících režimech:
– Asynchronní (plně duplexní)
– Synchronní – Master (poloduplexní)
– Synchronní – Slave (poloduplexní)

2.6. Linka RS232

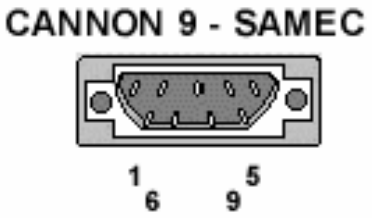
RS232 Používá asynchronní přenos informací. Každý přenesený byte konstantní rychlostí je proto třeba synchronizovat. K synchronizaci se používá sestupná hrana tzv. Start bitu. Za ní již následují posílaná data.



Synchronní přenos informací znamená, že na nějakém vodiči nebo vodičích se nastaví určitá úroveň, která přenáší informaci a validita informace se potvrdí impulzem, nebo změnou úrovně synchronizačního signálu. Synchronizačním signálem se tedy informace kvantují.

Asynchronní přenos dat probíhá v určitých sekvencích. Data jsou přenášena přesně danou rychlostí a uvozena startovací sekvencí, na kterou se synchronizují všechna přijímací zařízení. Všechny strany obsahují vlastní přesný oscilátor, díky kterému odečítají data v přesně definovaných intervalech. Po ukončení sekvence je další příjem opět synchronizován startovní sekvencí.

Popis vývodů COM:



PIN	NÁZEV	SMĚR	POPIS
1	CD	<--	Carrier Detect
2	RXD	<--	Receive Data
3	TXD	-->	Transmit Data
4	DTR	-->	Data Terminal Ready
5	GND	---	System Ground
6	DSR	<--	Data Set Ready
7	RTS	-->	Request to Send
8	CTS	<--	Clear to Send
9	RI	<--	Ring Indicator

3/ Teoretický rozbor

3.1. Přehled komerčně používaných řešení

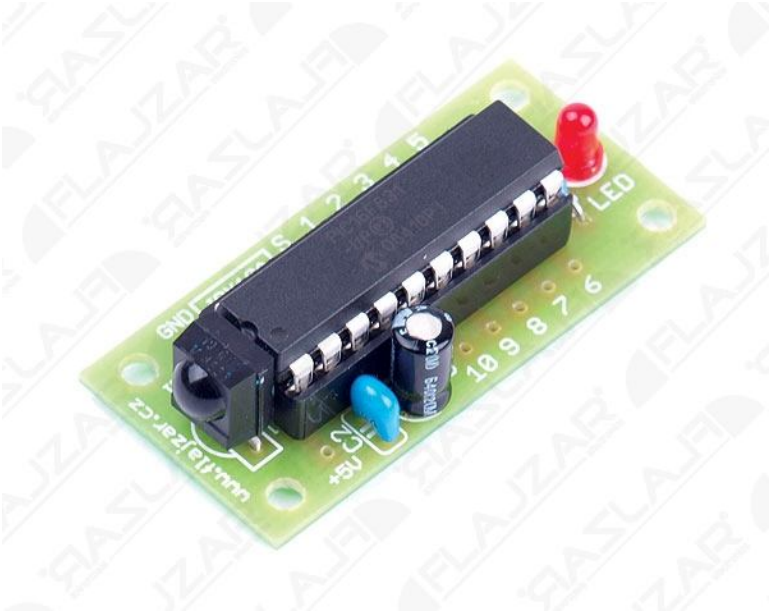
a) MK161 - PŘIJÍMAČ DÁLKOVÉHO OVLÁDÁNÍ IR

Cena s DPH: 373 Kč

2-kanálový přijímač IR dálkového ovládání (kompatibilní s MK162, K6710, K8051, K8049) s reléovým výstupem 24V/1A.
Napájení 12VDC 75mA. Rozměry 45 x 50 x 15mm.

+	-
2.kanálový přijímač	Pouze reléový výstup
	Chybí komunikace pomocí RS232C
	Bez možnosti konfigurace

b) IR mini přijímač IRX 10



Cena s DPH: 402 Kč

Malý univerzální modul přijímače IR signálu z IR vysílačů, zejména IRM10.
Spolupracuje však i s typem č.1467, IR33K1, IR18K2 atd.

Dálkový ovladač IRM10 je nabízen bez potisku a můžete jej tedy maximálně přizpůsobit vašemu použití.

Napájení přijímače 5V, klidový odběr jen 2,3mA.
Rozměry 40 x 19 x výška 12mm.
10 přímých logických výstupů, společný výstup S, signalizace LED.

Při použití s dalšími alternativními vysílači je možné využít 12 výstupů, nebo BCD výstup.
Dodáváno ve formě stavebnice nebo sestaveného a oživeného modulu.

+	-
Malé rozměry	Chybí komunikace pomocí RS232C
10 přímých logických výstupů	Konfigurace pomocí propojek
BCD výstup	

c) Univerzální stavebnicový IR přijímač firmy Flajzar

Cena s DPH: 353 Kč

Stavebnice přijímače infračerveného záření, který je možné ovládat libovolným infra ovládáním, např. od televize, videa, věže apod. Základem je integrovaný infra přijímací modul a dekódovací procesor PIC. Na výstupu relé se zatížením až 230V / 2A.

Rozměry desky 70 x 55mm.

Propojkou lze nastavit dva pracovní režimy:
1) při neosazené propojce J1 relé drží jen po dobu držení tlačítka
2) při osazení propojky J1 relé střídavě zapíná a vypíná. Tedy prvním stiskem na ovladači sepne, druhým rozepne.

+	-
Nízká cena	Chybí komunikace pomocí RS232C
Komunikace s libovolným IR ovládáním	Konfigurace pomocí propojek
Již naprogramovaná PIC	

3.2. Požadované funkce SW

1) Základní funkcí SW je detekce kódu vysílaného IR dálkovým ovladačem a zjištění čísla stisknutého tlačítka. Podle tohoto čísla SW vybere z paměti EEPROM PIC povel o délce 2-8 bajtů. Délku tohoto povelu SW samostatně zjišťuje podle tzv. systémového bajtu. Celkový počet povelů je 31. Použitá tlačítka jsou 2-32. Tlačítko s číslem 1 zůstává volné, pro možnost dalšího využití např. spuštění urč. režimu přijímače dálkového ovládání.

2) Obsluha sériové linky na straně příjmu. SW umožňuje měnit obsah paměti EEPROM, tedy reakci na stisknuté tlačítko na ovladači, podle pokynů zaslaných po sériové lince z nadřízeného systému. V tomto režimu by měl přijímač mít nastavenou svou adresu a realizuje dohodnutý protokol. Může být tedy poté zařazen jako prvek robotické sítě, používané pro stavebnicové řešení robotů.

3.3. Možnosti realizace funkcí SW

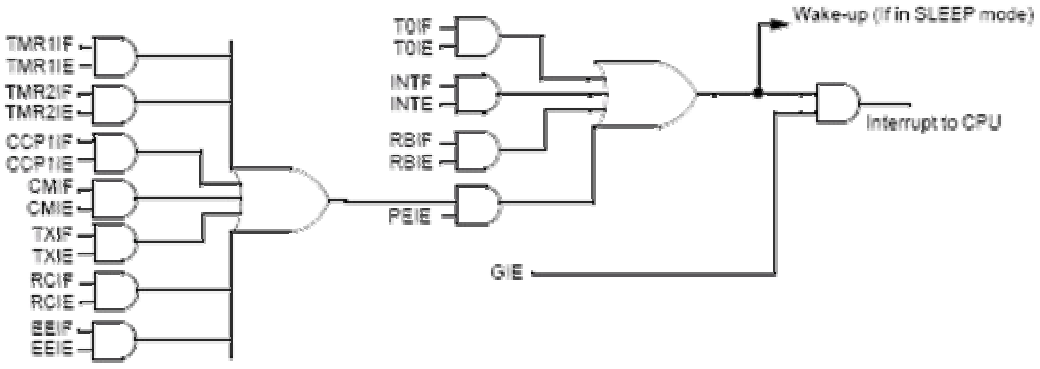
Je známé že každý problém se dá řešit mnoha způsoby, a proto je každý SW jedinečný. Zde je další možnost realizace funkcí SW, která se v jazyce assembler používá velmi často. Lze například využít pro detekci stisknutí tlačítka.

Přerušení

Přerušení je funkce, která po předem nastavené akci přeruší aktuální činnost procesoru a vykoná danou práci. Poté se procesor vrátí zpět a pokračuje kde skončil. Jinými slovy, procesor dokončí právě rozdělanou instrukci a skočí na 4 řádek programu! Zde se nachází rutina, která do pomocných registrů uloží pracovní registr W a další tři důležité registry - STATUS, PCLATH a FSR a skočí na podprogram obsluhy přerušení. Zde se obvykle zjistí jaký typ přerušení proběhl (pokud nepoužíváte jen jedno) a provede se příslušná akce. Při návratu z přerušení se opět obnoví uložené registry a příkazem RETFIE se procesor vrátí přesně do místa kde byl před přerušením a pokračuje zde dál.

Zdrojů přerušení je samozřejmě víc. Zapnou se jen ty, co jsou zrovna potřeba. Takže přerušení lze provést od:
- změny na pinu RB0/INT
- změny na pinech RB4 - RB7
- přetečení TMR0
- TMR1
- TMR2
- změny stavu komparátoru
- USART
- CCP

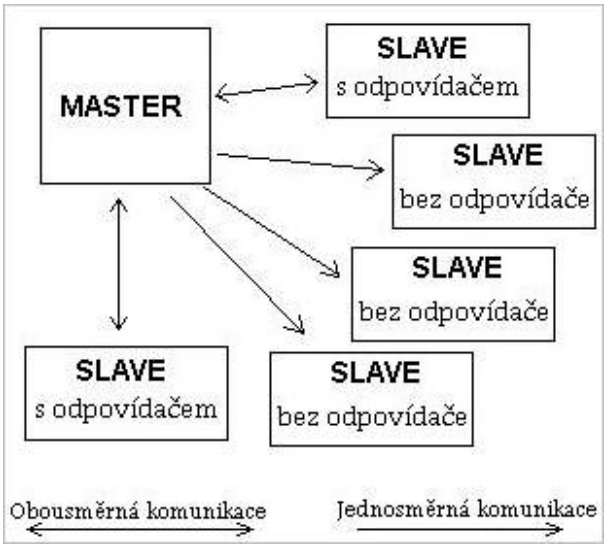
A vlastní sytém přerušení vypadá takto:



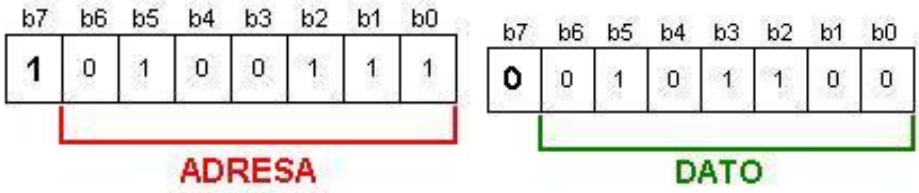
A k čemu je to dobré? Nemusíte neustále hlídat stav těchto zdrojů přerušení. Například máte na pinech RB4 až RB7 tlačítka, nemusíte se tedy neustále dotazovat na jejich stav. Jednoduše nastavíte příslušné přerušení a vše proběhne automaticky.
Přerušení od přetečení některého z časovačů se používá na tvorbu přesných časů, atd.

3.4. Protokol pro stavebnicová řešení robotů

Architektura sítě je založena na spojení master – slave. To znamená, že master, jako řídící zařízení, ovládá ostatní zařízení typu slave, která jsou mu podřízena. Standardně je v síti jedno zařízení typu master a určitý počet zařízení typu slave, který závisí na technických možnostech sítě. Typickým zařízením master je PC, slave může reprezentovat například sedmi-segmentový displej.



Uživatelký povel sestává z 2-8 bajtů. První bajt je vždy adresový, jeho nejvyšší bit b7 je „1“ bit b6 je „0“. Druhý bajt je systémový a určuje délku povelu v rozsahu 2-8 bajtů. Případné další bajty jsou vždy datové. Adresový a datový bajt odlišuje nejvyšší bit, jak znázorňuje následující obrázek.



Povel: Adresový bajt + Systémový bajt + Datový bajt (0-6)

1. bajt – adresový:	**1**Taa Sasa	T=1	systémový povel (na systémovou adresu)
		T=0	uživatelský povel (na uživatelskou adresu)
		aaaaaa	adresa prvku v rozsahu 0-63 dec
2. bajt – systémový:	**0**ppp Gbbb	bbb	počet bajtů povelu kromě adresového (1-7)
		G=1	programovací režim
		G=0	uživatelský režim
		ppp	rezerva
3. – 8. bajt – datový:	**0**ddd dddd	ddddddd	přenášená data

Přehled uživatelských povelů všech možných délek:

10aa aaaa **0**ppp 0001

10aa aaaa **0**ppp 0010 **0**ddd dddd

10aa aaaa **0**ppp 0011 **0**ddd dddd **0**ddd dddd

10aa aaaa **0**ppp 0100 **0**ddd dddd **0**ddd dddd **0**ddd dddd

10aa aaaa **0**ppp 0101 **0**ddd dddd **0**ddd dddd **0**ddd dddd **0**ddd dddd

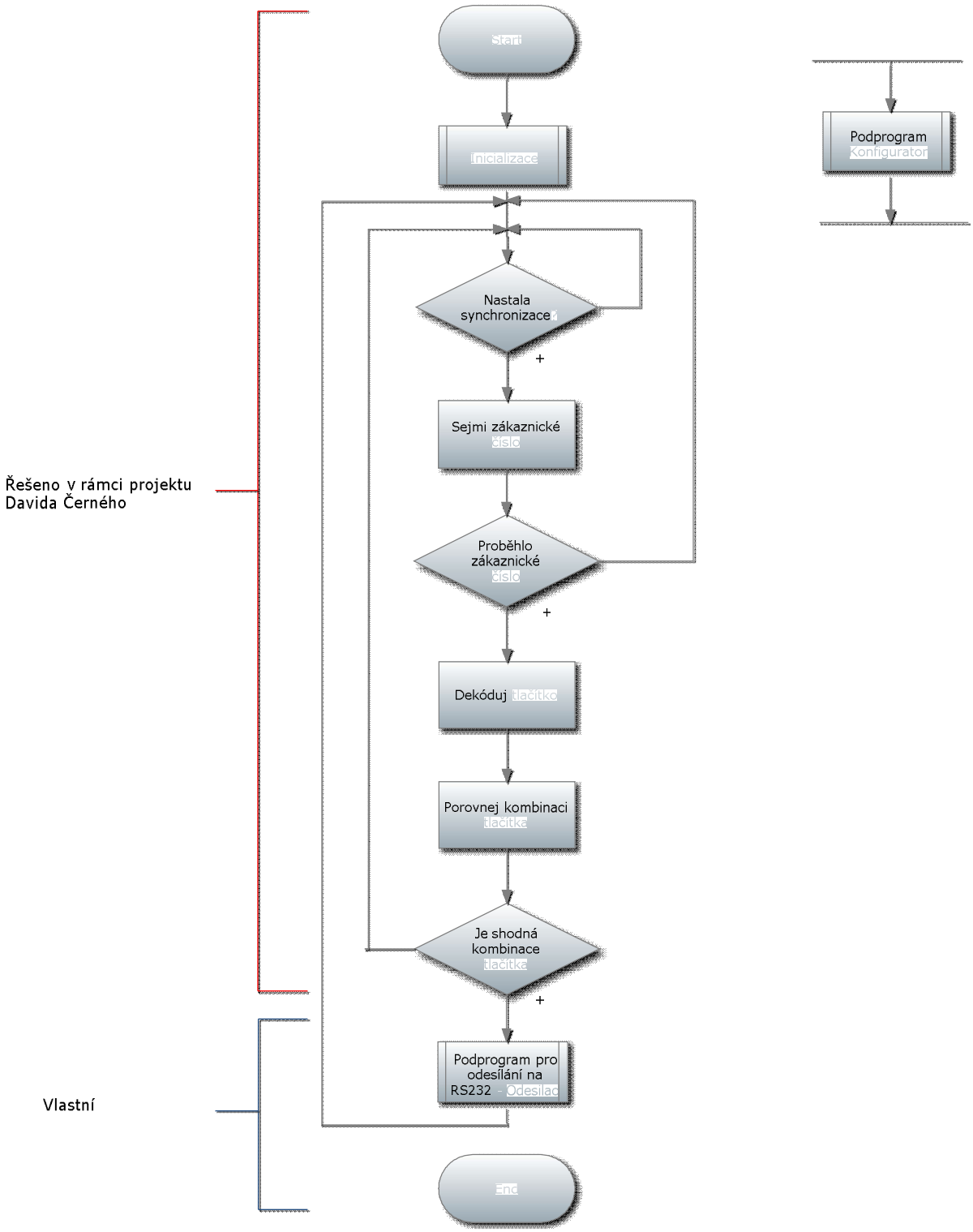
10aa aaaa **0**ppp 0110 **0**ddd dddd **0**ddd dddd **0**ddd dddd **0**ddd dddd **0**ddd dddd

10aa aaaa **0**ppp 0111 **0**ddd dddd **0**ddd dddd **0**ddd dddd **0**ddd dddd **0**ddd dddd **0**ddd dddd

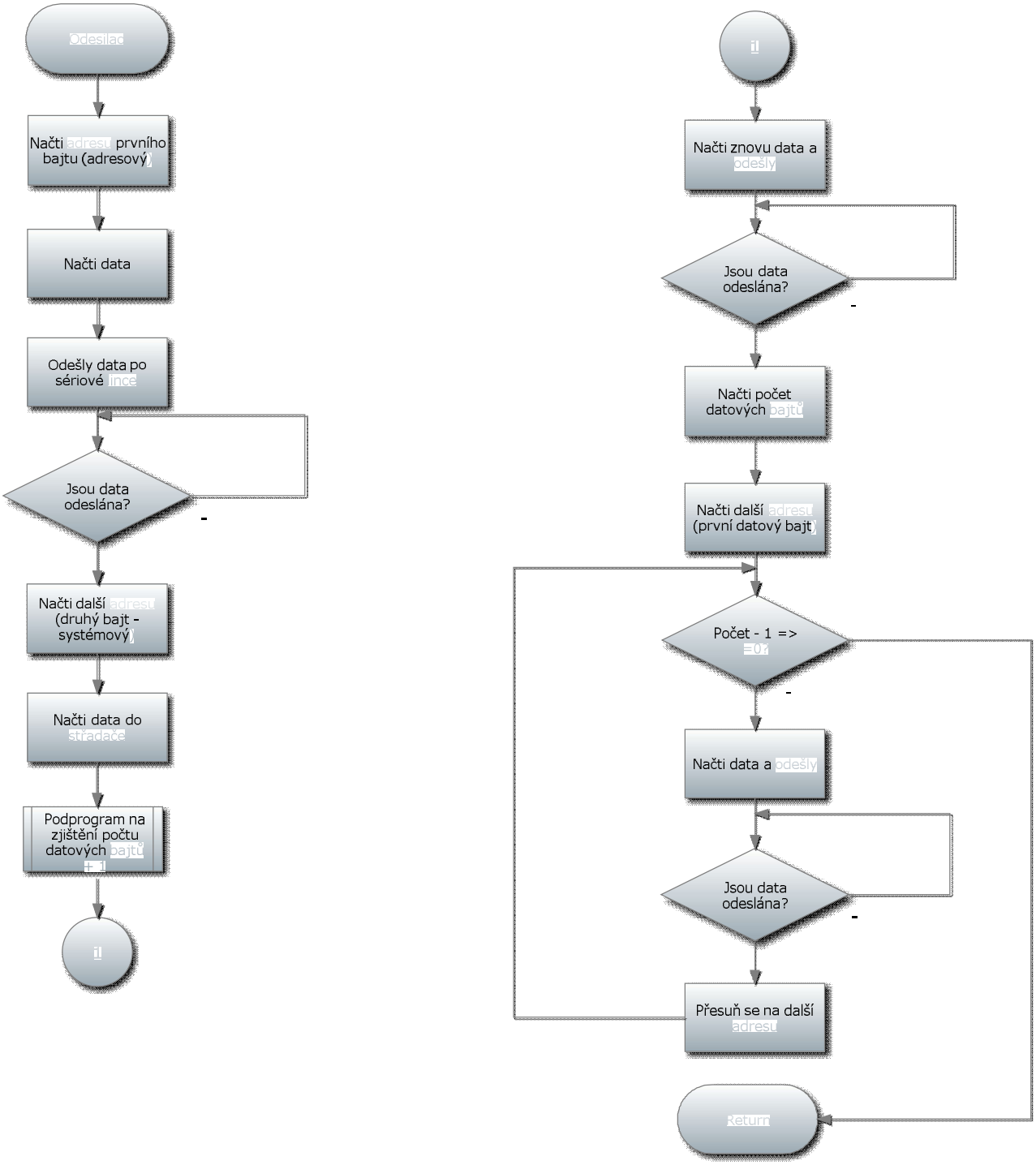
4/ Popisná část

4.1. Vývojové diagramy SW

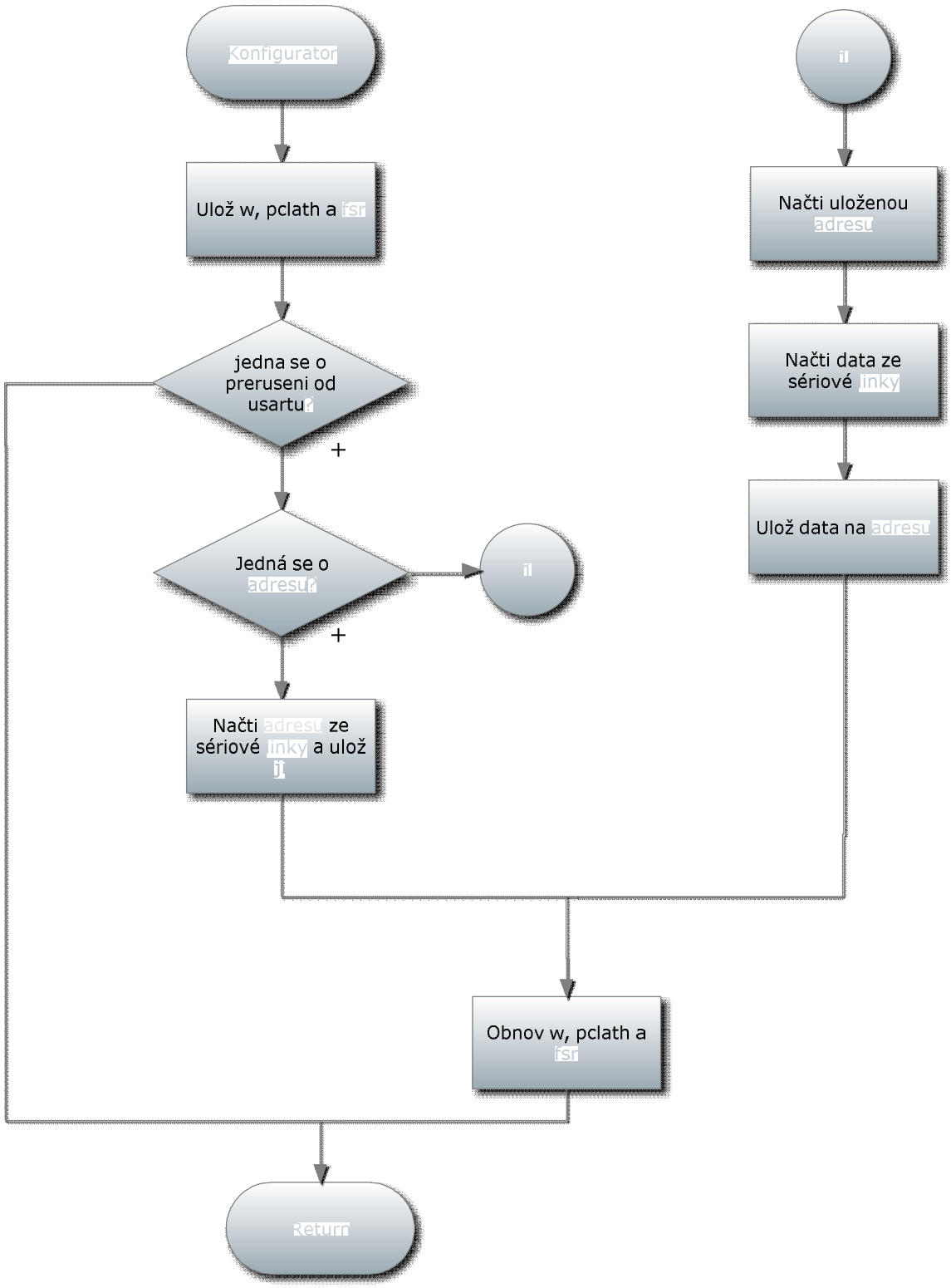
a) Vývojový diagram hlavní programové smyčky



b) Vývojový diagram podprogramu pro čtení a odesílání dat z paměti EEPROM - Odesilac



c) Vývojový diagram podprogramu pro ukládání dat do paměti EEPROM přes sériovou linku RS232 – Konfigurator



4.2. Seznam proměnných

Proměnná	Adresa v paměti RWM	Význam
pomm	0x20	proměnná pro ošetření protokolu
W_save	0x21	záloha w při přerušení
Sta_sav	0x22	záloha status při přerušení
PCL_sav	0x23	záloha pcl při přerušení
FSR_sav	0x24	záloha fsr při přerušení
auloz	0x25	proměnná určující přijaté dato či adresu
avyloz	0x26	proměnná pro zapsání dat do EEPROM
pom	0x70	vlastní proměnná
pom2	0x71	vlastní proměnná
pocms	0x72	vlastní proměnná
zak1	0x73	vlastní proměnná
zak2	0x74	vlastní proměnná
zak3	0x75	vlastní proměnná
zak4	0x76	vlastní proměnná
zak5	0x7A	vlastní proměnná
zak6	0x7B	vlastní proměnná
kam	0x77	vlastní proměnná
pocreg	0x78	vlastní proměnná
konec	0x79	vlastní proměnná
kolo	0x7F	proměnná pro zjištění délky odesílaného povelu

Proměnná	Rozsah hodnot	lokální/globální	Projekt David Černý/Zdeněk Přech
pomm	0-255	lokální	Zdeněk Přech
W_save	0-255	lokální	Zdeněk Přech
Sta_sav	0-255	lokální	Zdeněk Přech
PCL_sav	0-31	lokální	Zdeněk Přech
FSR_sav	0-255	lokální	Zdeněk Přech
auloz	1-2	lokální	Zdeněk Přech
avyloz	0-255	lokální	Zdeněk Přech
pom	0-33	globální	David Černý
pom2	0-231	globální	David Černý
pocms	0-81	globální	David Černý
zak1	0-15	globální	David Černý
zak2	0-15	globální	David Černý
zak3	0-15	globální	David Černý
zak4	0-15	globální	David Černý
zak5	0-15	globální	David Černý
zak6	0-15	globální	David Černý
kam	0-4	globální	David Černý
pocreg	0-9	globální	David Černý
konec	0-2	globální	David Černý
kolo	0-7	globální	Zdeněk Přech

4.3. Mapa paměti EEPROM

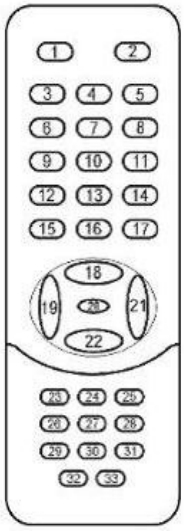
Číslo tlačítka		Adresa v paměti
2	Adresový bajt	00h
	Systémový bajt	01h
	6x Datový bajt	02h … 07h
3	Adresový bajt	08h
	Systémový bajt	09h
	6x Datový bajt	0Ah … 0Fh
4	Adresový bajt	10h
	Systémový bajt	11h
	6x Datový bajt	12h … 17h
5	Adresový bajt	18h
	Systémový bajt	19h
	6x Datový bajt	1Ah … 1Fh
6	Adresový bajt	20h
	Systémový bajt	21h
	6x Datový bajt	22h … 27h
7	Adresový bajt	28h
	Systémový bajt	29h
	6x Datový bajt	2Ah … 2Fh
8	Adresový bajt	30h
	Systémový bajt	31h
	6x Datový bajt	32h … 37h
9	Adresový bajt	38h
	Systémový bajt	39h
	6x Datový bajt	3Ah … 3Fh

Číslo tlačítka		Adresa v paměti
10	Adresový bajt	40h
	Systémový bajt	41h
	6x Datový bajt	42h … 47h
11	Adresový bajt	48h
	Systémový bajt	49h
	6x Datový bajt	4Ah … 4Fh
12	Adresový bajt	50h
	Systémový bajt	51h
	6x Datový bajt	52h … 57h
13	Adresový bajt	58h
	Systémový bajt	59h
	6x Datový bajt	5Ah … 5Fh
14	Adresový bajt	60h
	Systémový bajt	61h
	6x Datový bajt	62h … 67h
15	Adresový bajt	68h
	Systémový bajt	69h
	6x Datový bajt	6Ah … 6Fh
16	Adresový bajt	70h
	Systémový bajt	71h
	6x Datový bajt	72h … 77h
17	Adresový bajt	78h
	Systémový bajt	79h
	6x Datový bajt	7Ah … 7Fh

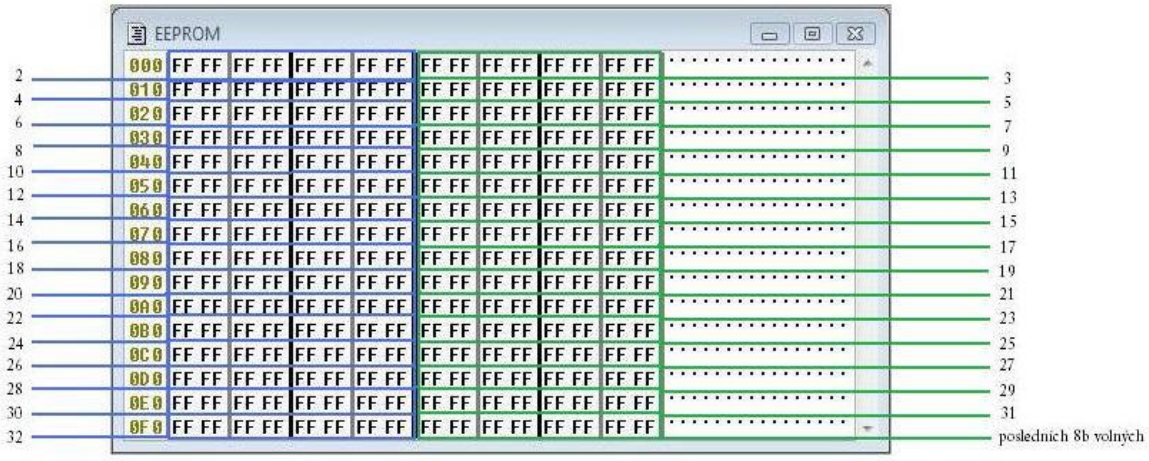
Číslo tlačítka		Adresa v paměti
18	Adresový bajt	80h
	Systémový bajt	81h
	6x Datový bajt	82h … 87h
19	Adresový bajt	88h
	Systémový bajt	89h
	6x Datový bajt	8Ah … 8Fh
20	Adresový bajt	90h
	Systémový bajt	91h
	6x Datový bajt	92h … 97h
21	Adresový bajt	98h
	Systémový bajt	99h
	6x Datový bajt	9Ah … 9Fh
22	Adresový bajt	A0h
	Systémový bajt	A1h
	6x Datový bajt	A2h … A7h
23	Adresový bajt	A8h
	Systémový bajt	A9h
	6x Datový bajt	AAh … AFh
24	Adresový bajt	B0h
	Systémový bajt	B1h
	6x Datový bajt	B2h … B7h
25	Adresový bajt	B8h
	Systémový bajt	B9h
	6x Datový bajt	BAh … BFh

Číslo tlačítka		Adresa v paměti
26	Adresový bajt	C0h
	Systémový bajt	C1h
	6x Datový bajt	C2h … C7h
27	Adresový bajt	C8h
	Systémový bajt	C9h
	6x Datový bajt	CAh … CFh
28	Adresový bajt	D0h
	Systémový bajt	D1h
	6x Datový bajt	D2h … D7h
29	Adresový bajt	D8h
	Systémový bajt	D9h
	6x Datový bajt	DAh … DFh
30	Adresový bajt	E0h
	Systémový bajt	E1h
	6x Datový bajt	E2h … E7h
31	Adresový bajt	E8h
	Systémový bajt	E9h
	6x Datový bajt	EAh … EFh
32	Adresový bajt	F0h
	Systémový bajt	F1h
	6x Datový bajt	F2h … F7h
	8 bajtů volné paměti	F8h … FFh

4.4. Realizované povely



Jednotlivé povely v paměti EEPROM přiřazené urč. tlačítku.



4.5. Zdrojový kód hlavní programové smyčky

```
                org         0

                goto        kam24
;-------------------------------------------------------------------------------------
                org         4           ;preruseni
                bcf         intcon,7    ;zakazani preruseni GIE
                bsf         porta,0
                call        zp14        ;rozsviceni LED 1
                call        prijem
                retfie


;-------------------------------------------------------------------------------------

kam24           call        init        ;podprogram (RA0-vystup,PULL)
                call        det9        ;synchronizace
                bsf         porta,0     ;pri spravne kombinaci rozsvit LED
t34             call        start       ;dekodovaní siugnálu
                call        zakaznicke  ;kontrola zakaznickeho cisla (3444h)
                call        kody        ;porovnávaní a vyhodnocení tlacitek
                goto        kam24
```

4.6. Zdrojové kódy klíčových podprogramů

a) Podprogram pro čtení a odesílání dat z paměti EEPROM

```
                                        ;odeslani prvniho bajtu

cti             bank3
                bsf             eecon1,0        ;povoleni cteni
                bank2
                movf            eedata,w        ;nacteni dat do stradace
                bank0
                movwf           pomm            ;ulozit data do osetrujiciho registru
                bsf             pomm,7          ;osetreni dat podle protokolu
                bcf             pomm,6          ;osetreni dat podle protokolu
                movf            pomm,w          ;nacteni osetrenych dat
                movwf           txreg           ;vyslani dat na seriovou linku
cekej           btfss           txif            ;jsou data odeslana, pokud ne cekej
                goto            cekej
                bank2
                incf            eeadr,f         ;jdi na dalsi adresu
                call            cti2
                return

;-----------------------------------------------------------------------------------------

                                        ;odeslani druheho bajtu

cti2            bank3
                bsf             eecon1,0        ;povoleni cteni
                bank2
                movf            eedata,w        ;nacteni dat do stradace
                bank0
                movwf           pomm
                call            pocet1
                bank3
                bsf             eecon1,0        ;povoleni cteni
                bank2
                movf            eedata,w        ;nacteni dat do stradace
                bank0
                movwf           pomm            ;ulozit data do osetrujiciho registru
                bcf             pomm,7          ;osetreni dat podle protokolu
                movf            pomm,w          ;nacteni osetrenych dat
                movwf           txreg           ;vyslani dat na seriovou linku
                call            zp14
                bank2
                incf            eeadr,f         ;jdi na dalsi adresu
                call            cti3
                return
```

```
;-------------------------------------------------------------------------------------

                                        ;podprogram pro urceni delky povelu

pocet1      bcf         pomm,7          ;osetreni rezervnich bitu dat
            bcf         pomm,6          ;osetreni rezervnich bitu dat
            bcf         pomm,5          ;osetreni rezervnich bitu dat
            bcf         pomm,4          ;osetreni rezervnich bitu dat
            bcf         pomm,3          ;osetreni rezervnich bitu dat
            movlw       d'1'
            subwf       pomm,w          ;odecti nactenou hodnotu od stradace
            btfss       zero            ;je 0?
            goto        pocet2          ;ne jdi na dalsi cislo
            movlw       d'1'            ;ano...nacti pocet opakovani
            movwf       kolo
            return

pocet2      movlw       d'2'
            subwf       pomm,w          ;odecti nactenou hodnotu od stradace
            btfss       zero            ;je 0?
            goto        pocet3          ;ne jdi na dalsi cislo
            movlw       d'2'            ;ano...nacti pocet opakovani
            movwf       kolo
            return

pocet3      movlw       d'3'
            subwf       pomm,w          ;odecti nactenou hodnotu od stradace
            btfss       zero            ;je 0?
            goto        pocet4          ;ne jdi na dalsi cislo
            movlw       d'3'            ;ano...nacti pocet opakovani
            movwf       kolo
            return

pocet4      movlw       d'4'
            subwf       pomm,w          ;odecti nactenou hodnotu od stradace
            btfss       zero            ;je 0?
            goto        pocet5          ;ne jdi na dalsi cislo
            movlw       d'4'            ;ano...nacti pocet opakovani
            movwf       kolo
            return

pocet5      movlw       d'5'
            subwf       pomm,w          ;odecti nactenou hodnotu od stradace
            btfss       zero            ;je 0?
            goto        pocet6          ;ne jdi na dalsi cislo
            movlw       d'5'            ;ano...nacti pocet opakovani
            movwf       kolo
            return
```

```
pocet6      movlw     d'6'
            subwf     pomm,w      ;odecti nactenou hodnotu od stradace
            btfss     zero        ;je 0?
            goto      pocet7      ;ne jdi na dalsi cislo
            movlw     d'6'        ;ano...nacti pocet opakovani
            movwf     kolo
            return

pocet7      movlw     d'7'
            subwf     pomm,w      ;odecti nactenou hodnotu od stradace
            btfss     zero        ;je 0?
            goto      koko        ;ne ukonci
            movlw     d'7'        ;ano...nacti pocet opakovani
            movwf     kolo
koko        return

;-------------------------------------------------------------------------------------

                                  ;podprogram pro odeslani daneho poctu datovych bajtu

cti3        bank0
            decfsz    kolo,f      ;odecti 1 od poctu opakování, pokud 0 vrat se
            goto      ctii        ;pokracuj ve cteni z pameti EEPROM
            return

ctii        bank3
            bsf       eecon1,0    ;povoleni cteni
            bank2
            movf      eedata,w    ;nacteni dat do stradace
            bank0
            movwf     pomm
            bcf       pomm,7
            movf      pomm,w
            movwf     txreg       ;vyslani dat na seriovou linku
            call      zp14
            bank2
            incf      eeadr,f     ;dalsi adresa
            goto      cti3
zpet        return
```

b) Podprogram pro ukládání dat do paměti EEPROM přes sériovou linku RS232

```
prijem                                  ;ulozeni w, PCL, FSR, status
            movwf       W_save          ; w
            movf        status,w
            clrf        status          ; banka 0
            movwf       Sta_sav         ; status
            movf        pclath,w
            movwf       PCL_sav         ; PCL
            movf        fsr,w
            movwf       FSR_sav         ; FSR

;-------------------------------------------------------------------------------------
                                        ;test od ceho bylo prerusenie
            btfsc       pir1,5
            goto        prer0           ; od usartu
            goto        konec_pre       ; nezname = nic
;-------------------------------------------------------------------------------------

                                        ;program pro zapis do EEPROM
prer0       bank0                       ;jedna se o adresu ci dato?
            movlw       d'1'
            subwf       auloz,w         ;odecti nactenou hodnotu od stradace
            btfss       zero            ;je 0? = je adresa
            goto        cislo2          ;ne je dato
            bank0
            movf        rcreg,w         ;nacti adresu a uloz ji do registru
            movwf       avyloz
            incf        auloz           ;zvedni obsah auloz¨
            goto        konec_pre       ;vrat se


cislo2      movlw       d'2'            ;jedna se opravdu o dato?
            subwf       auloz,w
            btfss       zero            ;je 0? = zahaj zapis do EEPROM
            goto        konec_pre       ;neni? = ukonci preruseni
pisee       bank3
            btfsc       eecon1,1        ;cekani na dokonceni zapisu
            goto        pisee
            bank0
            movf        avyloz,w
            bank2
            movwf       eeadr           ;nastaveni adresy do ukazatele
            bank0
            movf        rcreg,w
            bank2
            movwf       eedata          ;priprava dat
            bank3
            bsf         eecon1,2        ;povoleni zapisu do EEPROM
            bcf         eecon1,7        ;aktivace pameti dat EEPROM
```

```
                bcf     intcon,7        ;zakaz preruseni
                movlw   h'55'
                movwf   eecon2          ;zapis 55h
                movlw   h'AA'
                movwf   eecon2          ;zápis AAh
                bsf     eecon1,1        ;zahájení zápisu
                bsf     intcon,7        ;povolení přerušení
                bcf     eecon1,2        ;zákaz dalších zápisů
                bank0
                movlw   b'00000001'
                movwf   auloz
                goto    konec_pre

;-------------------------------------------------------------------------------------
                                        ;obnovenie w, PCL, FSR, status
konec_pre
                bank1
                movlw   b'11000000'     ; povoleni zpet pozadovanych preruseni
                movwf   intcon
                movlw   b'00100000'
                movwf   pie1
                bank0                   ; banka 0
                movf    FSR_sav,w       ; FSR
                movwf   fsr
                movf    PCL_sav,w       ; PCL
                movwf   pclath
                movf    Sta_sav,w       ; status
                movwf   status
                swapf   W_save,f        ; w
                swapf   W_save,w
                return
```

5/ ZÁVĚR

Výsledkem je funkční zařízení ovládané mikropočítačem PIC, které je schopno přijímat a vyhodnocovat stisknuté tlačítko na ovladači, jemuž je přiřazen daný povel, který si volí uživatel a poté se odešle po sériové lince RS232C danému prvku. Dále zařízení podporuje možnost programovat povely do PIC přímo z PC pomocí odpovídače sítě Robo-COP.

Zařízení lze používat jako převodník z IR na sériovou linku i pro stavebnicová řešení robotů. Nelze jej zapojit jako plnohodnotný prvek této sítě. Důvodem je nekompatibilita protokolu na straně příjmu. Do budoucna by toto mělo být odstraněno použitím operačního systému v PIC.

6/ SEZNAM POUŽITÉ LITERATURY

H.Häberle a kol.: Průmyslová elektrotechnika a informační technologie, europa–Sobotáles cz. Praha 2003, ISBN : 80–86706–04–4

MAXIM MAX232 datasheet [online]. San Gabriel Drive : 2003.
Dostupné na WWW:
<http://www.datasheetcatalog.org/datasheet/maxim/MAX220-MAX249.pdf>.

Infineon *SFH5110–38 datasheet* [online]. Leden 01, 2000.
Dostupné na WWW:
<http://www.datasheetcatalog.org/datasheet/infineon/1–sfh5110.pdf>

Manuál PIC16F88-český překlad
Dostupné na WWW:
<www.copsu.cz/mikrop>

Robo-COP
Dostupné na WWW:
<http://www.copsu.cz/mikrop/zak_projekty/Robocop/index.html>

MICROCHIP USART : Using the USART in Asynchronous Mode [online].
Microchip, October 22, 2002 [cit. 5. prosince 2007].
Dostupné na WWW:
<http://ww1.microchip.com/downloads/en/DeviceDoc/usart.pdf>.

IR přijímače FLAJZAR
Dostupné na WWW:
<www.flajzar.cz>

Teorie IR datového přenosu
Dostupné na WWW:
<www.hw.cz>

Škola programování PIC
Dostupné na WWW:
<www.pandatron.cz>

7/ PŘÍLOHY

Obsah CD nosiče

-Dokumentace v pdf. a doc. formátu
-Zdrojové soubory a programy v asm, lst, hex
-Datasheeaty, manuály
-Fotografie
-Projektový plakát
-Programy pro přijem signálu ze seriové linky (SCONTROL, RoboCOP)