PICkit™ 3
Programátor a debugger
Manuál

Dejte si pozor na následující detaily kódové ochrany na zařízeních Microchip:

- Výrobky Microchip splňují specifikace obsažené ve svém datasheetu.

- Microchip věří, že série jeho produktů je jednou z nejzabezpečenějších svého druhu na současném trhu, pokud jsou tyto výrobky používány zamýšleným způsobem a za normálních podmínek.

- Existují nepoctivé a nejspíš také nelegální metody pro prolomení ochranného kódu. Všechny tyto metody, jak víme, vyžadují použití výrobků Microchip způsobem, který neodpovídá jejich specifikacím v datasheetu. Osoba, která se věnuje takové činnosti, se nejspíše dopouští krádeže duševního vlastnictví.

- Microchip je ochoten spolupracovat se zákazníkem, který má starost o integritu svého kódu.

- Ani Microchip ani žádný výrobce polovodičů nemůže zajistit bezpečnost svého kódu. Kódová ochrana neznamená, že je výrobek "neprolomitelný".

Kódová ochrana se neustále vyvíjí. Microchip neustále vylepšuje kódovou ochranu na svých produktech. Pokusy o prolomení kódové ochrany Microchip mohou být porušením Digital Millennium Copyright Act. Pokud takové činy dovolují neautorizovaný přístup k vašemu softwaru nebo jiné práci chráněné autorskými právy, máte právo na žalobu.

Předmluva

Poznámka pro zákazníky

Všechna dokumentace stárne a tento manuál není výjimkou. Nástroje Microchip a jejich dokumentace se neustále vyvíjí, aby uspokojily potřeby zákazníků, takže některé popisy a postupy se mohou lišit od těch obsažených v tomto dokumentu. Navštivte stránky www.microchip.com pro nejnovější možnou dokumentaci.

Pro nejnovější informace o vývojových prostředcích se podívejte na MPLAB® X IDE online pomoc. Vyberte Help menu a poté Topics pro otevření seznamu možných souborů s online pomocí.

Úvod

Tato kapitola obsahuje základní informace, které je užitečné znát před použitím startovací sady PICkit™. Obsahuje tato témata:
- Rozvržení dokumentu
- Vyjadřování používané v tomto manuálu
- Registraci záruky
- Doporučená četba
- Stránky Microchip
- Servis upozornění zákazníka na změny ve vývojových systémech
- Zákaznická podpora

Rozvržení dokumentu

Tento dokument popisuje, jak použít PICkit 3 jako vývojový nástroj pro emulaci a odstranění chyb firmwaru na cílové desce. Dokument má následující rozvržení:

Část 1 - Začínáme

- Kapitola 1. Přehled
Popíše programátor/debugger PICkit 3.
- Kapitola 2. Teorie operace
Zjednodušený popis práce programátoru/debuggeru PICkit 3.
- Kapitola 3. Instalace
Jak nainstalovat programátor/debugger PICkit 3.
- Kapitola 4. Základní nastavení
Poskytuje instrukce, jak začít používat programátor/debugger PICkit 3 pro programování podporovaných zařízení.
- Kapitola 5. PICkit 3 Debug Express
Poskytuje základní informace o použití PICkit 3™ Debug Express.
- Kapitola 6. Řešení problémů, první kroky
První věci, které byste měli zkusit, když máte problémy s debuggerem.
- Kapitola 7. Často kladené dotazy (FAQ)
Seznam častých dotazů, užitečné pro řešení problémů.
- Kapitola 8. Chybové hlášky
Seznam chybových hlášek a doporučená řešení.
- Kapitola 9. Seznam funkcí debuggeru
Shrnuje funkce debuggeru, které jsou k dispozici.
- Kapitola 10. Hardwarové specifikace
Detaily hardwaru a elektrické specifikace ůro PICkit 3.
- Dodatek A. Schémata PICkit 3
Poskytuje diagramy schémat hardwaru programátoru/debuggeru PICkit 3.
- Dodatek B. Rady k operaci
Pojednává o problémech s operací, které musíte zvážit, když navrhujete aplikaci.

Vyjadřování používané v tomto manuálu

V tomto manuálu je použito následující vyjadřování:

Popis	**Význam**	**Příklad**
Font Arial:		
Kurzíva	Citované knihy	*MPLAB® IDE User's Guide*
	Zdůraznění	...je to *jediný* kompiler...
Velká písmena	Okno	okno Output
	Dialog	dialog Settings
	Výběr menu	vyberte Enable Programmer
Uvozovky	Název pole v okně dialogu	"Save project before build"
Podtržená kurzíva se závorkou >	Cesta v menu	*File>Save*
Tučné písmo	Dialogové tlačítko	Klikněte na **OK**
	Záložka	Klikněte na záložku **Power**
Text v špičatých závorkách < >	Klávesa na klávesnici	Stiskněte <Enter>, <F1>
Font Courier New:		
Obyčejný font Courier New	Vzorek zdrojového kódu	`#define START`
	Názvy souborů	`autoexec.bat`
	Umístění souboru	`c:\mcc18\h`
	Klíčová slova	`_asm, _endasm, static`
	Příkazy	`-Opa+, -Opa-`
	Hodnoty bitu	`0,1`
	Konstanty	`0xFF, 'A'`
Kurzíva	Proměnná	`file.o`, kde `file` může být jakýkoliv platný název souboru
Hranaté závorky [ ]	Volby	`mcc18 [volba] file [volba]`
Složené závorky a svislé čárky { | }	Výběr vylučujících se možností nebo výběr NEBO	`errorlevel {0|1}`
Trojtečky...	Nahrazují opakující se text	`var_name [, var_name...]`

	Reprezentují kód posytnutý uživatelem	`void main (void)`
		`{ ...`
		`}`

Registrace záruky

Vyplňte, prosím registrační kartu (Warranty Registration Card) a zašlete ji poštou. Tímto způsobem získáte možnost obdržet nová vylepšení výrobku. Interim software uvolňuje nové verze na stránkách Microchip.

Doporučená četba

Tento manuál popisuje, jak používat PICkit 3. Níže je seznam dalších užitečných dokumentů. Následující dokumenty Microchip jsou k dispozici a doporučené jako zdroje referencí.

44-Pin Demo Board User's GUide (DS41296)
Nahlédněte do tohoto dokumentu pro instrukce k použití 44-Pin demo board jako vývojového nástroje k emulaci a debugování firmwaru na cílové desce.

Low Pin Count Demo Board User's Guide (DS51556)
Nahlédněte do tohoto dokumentu pro instrukce k použití low pin count device (8-pin, 14-pin a 20-pin). Tento dokument obsahuje sérii cvičení.

MPLAB® IDE User's Guide/Help (DS51519)
Nahlédněte do tohoto dokumentu pro více informací o instalaci a funkcích softwaru MPLAB Integrated Development Environment (IDE). K dispozici je též online Help verze.

In-Circuit Serial Programmer™ (ICSP™) Guide (DS30277)
Tento dokument obsahuje průvodce pro úspěšné ICSP programování. Jsou v něm aplikační poznámky ohledně hardwaru a specifikací ICSP programování.

MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian User's Guide (DS33014)
Popisuje, jak použít Microchip PIC® MCU Assembler (MPASM assembler), linker (MPLINK linker) a knihovníka (MPLIB librarian).

README pro PICkit™ 3 Debug Express
Pro nejnovější informace ohledně používání PICkit 3 Debug Express si přečtěte soubor "`Readme for PICkit 3.htm`" (HTML soubor) v podsložce Readmes v instalační složce MPLAB IDE. Soubor Readme obsahuje nové informace a známé problémy, které se nemusí nacházet v tomto manuálu.

PICkit™ 3 Debug Express C18 Lessons
Tato cvičení vás provedou použitím PICkit 3 Debug Express s MPLAB C kompilerem pro PIC18 MCU. Jsou k dispozici na MPLAB ICE CDROm a na stránkách Microchip.

Readme Files
Pro nejnovější informace o použití jiných nástrojů si přečtěte Readme soubory pro dané nástroje s podsložce Readmes v instalační složce MPLAB IDE. Readme soubory obsahují nové informace a známé problémy, které nemusí být obsaženy v manuálu.

Stránky Microchip

Microchip poskytuje online podporu skrze svoje stránky www.microchip.com. Tyto stránky se používají jako způsob tvorby souborů a a informací snadno dostupných zákazníkům. Tento web obsahuje následující:

- Product support (podpora výrobků) - datasheety a opravné informace, aplikační poznámky a vzorky programů, zdroje, manuály a dokumenty hardwarové podpory, nejnovější vydání softwareu a archivovaný software.
- General Technical Support (základní technická podpora) - často kladené dotazy (FAQ), požadavky na technickou podporu, online diskuzní skupiny, soupis konzultantů Microchip.
- Business of Microchip (obchodování Microchip) - výběr výrobků a průvodce objednávkou, poslední tiskové zprávy Microchip, seznam seminářů a akcí, seznam prodejen Microchip, distributorů a zástupců společnosti.

Servis upozornění zákazníka na změny ve vývojových systémech

Tato služba pomáhá udržovat zákazníky v obraze o aktuálních výrobcích Microchip. Zaregistrovaní uživatelé budou dostávat e-maily s upozorněním, kdykoliv dojde ke změnám, aktualizacím nebo opravám souvisejícím s danou skupinou výrobků nebo vývojářským nástrojem.

Pro registraci vstupte na stránky Microchip www.microchip.com, klikněte na Customer Change Notification a pokračujte podle instrukcí k registraci.

Kategorie skupiny výrobků Development Systems jsou:

- Compilers - nejnovější informace Microchip C kompilerech, assemblerech, linkerech a dalších jazykových nástrojích. Ty zahrnují všechny MPLAB C kompilery; všechny MPLAB assemblery (včetně assembleru MPASM™); všechny MPLAB linkery (včetně objektového linkeru MPLINK™); a všech MPLAB knihovníků (včetně objektového knihovníka MPLIB™).

- Emulators - nejnovější informace o in-circuit emulátorech Microchip. Ty zahrnují MPLAB REAL ICE™, MPLAB ICE 2000 in-circuit emulátory.

- In-Circuit Debuggers - nejnovější informace o Microchip in-circuit debuggerech. Ty zahrnují MPLAB ICD 2, ICD 3, PICkit™ 2 a PICkit™ 3.

- MPLAB® IDE - nejnovější informace o MPLAB IDE, Windows® Integrated Development Environment (integrované vývojářské prostředí) pro vývojářské nástroje. Tento seznam je zaměřen na MPLAB IDE, MPLAB IDE Project Manager, MPLAB Editor a MPLAB SIM simulator, stejně jako na základní funkce editace a debugování.

- Programmers - nejnovější informace o programátorech Microchip. Ty zahrnují programovací zařízení MPLAB PM3 a PICSTART® Plus, PICkit 2 a PICkit 3 vývojářské programátory.

Zákaznická podpora

Uživatelé výrobků Microchip mohou obdržet pomoc několika způsoby:

- Distributor nebo zástupce
- Místní kancelář společnosti
- Prodejní technik (Field Application Engineer - FAE)
- Technická podpora

Zákazníci by měli kontaktovat svého distributora, reprezentanta nebo prodejního technika (FAE) kvůli podpoře. Pomoc mohou poskytnout také místní kanceláře společnosti. Pro kompletní a aktuální seznam kanceláří nahlédněte na stránky Microchip.

Technickou podporu lze získat také přes stránky: http://support.microchip.com.

Část 1 - Začínáme

Kapitola 1. Přehled

1.1 Úvod

Zde budete seznámeni se systémem programátoru a debuggeru PICkit 3™.
- Definice PICkit3.
- S čím vám může PICkit 3 pomoci.
- Části PICkit 3.
- Podpora zařízení a funkcí.

1.2 Definice PICkit 3.

Programátor a debugger PICkit 3 (viz obrázek 1-1) je jednoduchý a levný in-circuit debugger, který je ovládán pomocí PC se softwarem MPLAB IDE (v8.20 nebo novější) na platformě Windows®. PICkit 3 je nedílnou součástí sady nástrojů vývojového inženýra. Použití aplikace se může lišit od vývoje softwaru po integraci hardwaru.

PICkit 3 je debugovací systém používaný pro vývoj hardwaru a softwaru s Microchip PIC® mikrokontroléry (MCU) a dsPIC® Digital Signal kontroléry (DSC), které jsou založeny na In-Circuit Serial Programming™ (ICSP™) a Enhanced In-circuit Serial Programming dvojvodičovými sériovými rozhraními.

Kromě funkcí debuggeru lze PICkit 3 použít také použít jako vývojový programátor.

Debugovací systém spustí kód jako skutečné zařízení, protože pro emulaci využívá zařízení s vestavěným emulačním obvodem, namísto speciálního debugovacího čipu. Všechny dostupné funkce daného zařízení jsou přístupné interaktivně a lze je nastavit a modifikovat pomocí rozhraní MPLAB IDE.

PICkit 3 debugger byl vyvinut pro programování a debugování zabudovaných procesorů s debugovací funkcí. Mezi vlastnosti PICkit 3 patří:
- Rychlá USB podpora využívající standardní Windows ovladače.
- Spuštění v reálném čase.
- Procesory pracující s maximální rychlostí.
- Vestavěný monitor přepětí a zkratu.
- Nízké napětí do 5V (rozsah 1.8-5V).
- Diagnostické LED (zapnuto, aktivní, status).
- Čtení/psaní programu a data paměti mikrokontroléru.
- Mazání všech typů paměti (EEPROM, ID, konfigurace a programování) s ověřením.

- Periferní zmrazení na bodu přerušení programu.

Poznámka: PICkit 3 je zamýšlen pro vývojové programování. Pro produkční programování zvažte MPLAB PM3 device programme nebo jiný programátor od třetích stran, navrženy pro produkční prostředí.

Obrázek 1-1: PICkit™ 3 MCU programátor a debugger

Legenda:
1) Poutko
2) USB Port
3) Značka pinu 1
4) Programovací konektor
5) Indikační LED
6) Tlačítko (budoucí použití)

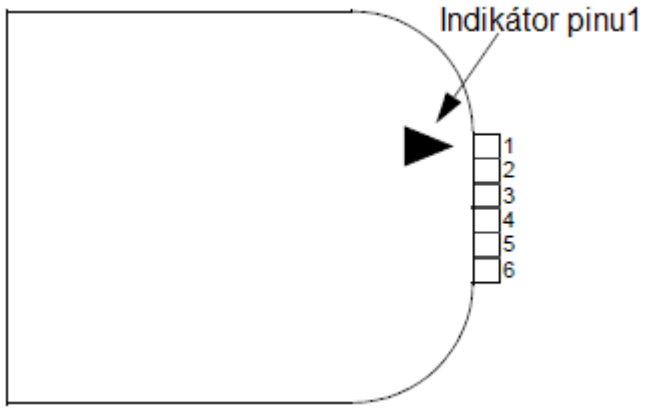
1.2.1 Poutko
Poutko poskytuje možnost uchycení, díky kterému můžete PICkit 3 zavěsit.

1.2.2 USB Port
Tento USB Port je USB mini-B konektor. Připojte PICkit 3 k PC pomocí přiloženého USB kabelu.

1.2.3 Značka pinu 1
Tato značka ukazuje umístění pinu 1 pro správné připojení konektoru.

1.2.4 Programovací konektor
Programovací konektor je 6-pinový header (rozteč 0.100"), který se připojuje k cílovému zařízení. Viz specifikace na obrázku 1-2.

Obrázek 1-2: Piny konektoru programátoru PICkit™ 3



Popis pinů*
1 = VPP/MCLR
2 = VDD cíle
3 = VSS (uzemnění)
4 = ICSPDAT/PGD
5 = ICSPCLK/PGC
6 = LVP

* 6-pinový header (rozteč 0.100") přijímá 0.025" čtvercové piny.

Poznámka: Funkce pinů programovacího konektoru jsou rozdílné pro programování sériových EEPROMS a HCS zařízení. Pro toto rozložení pinů viz ReadMe soubor pro PICkit 3 *(Help>Readme)* přiložený k softwaru MPLAB IDE.

1.2.5 Stavové LED

Stavové LED indikují stav PICkit 3.
1. **Power**-zapnuto (zelená) - PICkit 3 je napájen přes USB port.
2. **Active**-aktivní (modrá) - PICkit 3 je připojen k PC pomocí USB portu a je aktivní komunikace.
3. **Status**
   Busy-zaneprázdněný (žlutá) - právě probíhá proces programování
   Error-chyba (červená) - došlo k chybě

1.3 S čím vám může PICkit 3 pomoci

Programátor a debugger PICkit 3 vám umožní:

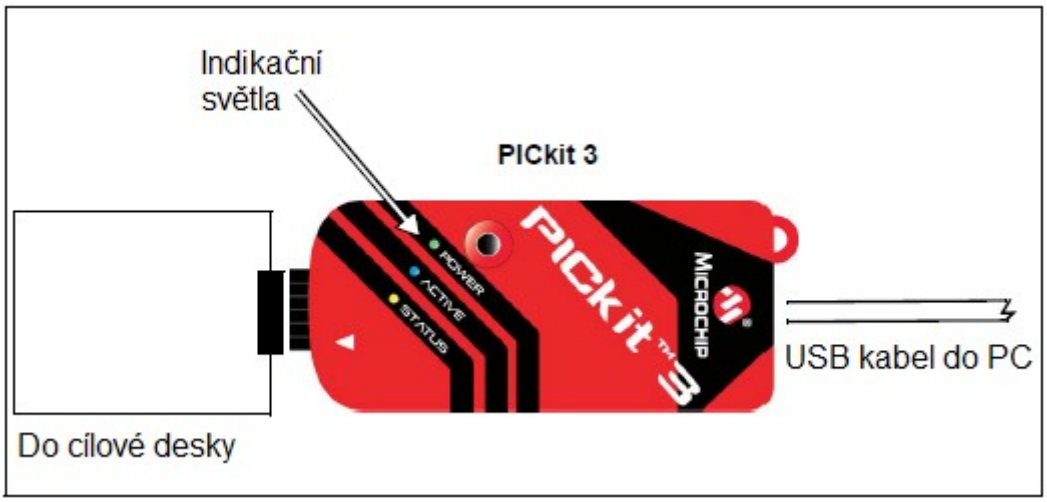
- debugovat aplikace na vašem hardwaru v reálném čase
- debugovat s hardwarovými body přerušení
- nastavit body přerušení na základě interních událostí
- monitorovat registry vnitřních souborů
- emulovat plnou rychlostí
- programovat vaše zařízení

1.4 Části PICkit 3

Programátor/debugger PICkit 3 se skládá z následujících částí:

1. PICkit 3 s indikačními světly pro napájení, aktivitu a status.
2. USB kabel poskytující komunikaci mezi debuggerem a PC a napájení.
3. CD-ROM se softwarem MPLAB IDE a online dokumentací.

Obrázek 1-3: Základní systém debuggeru



Další vybavení, které si můžete objednat samostatně:
- PICkit 3 Debug Express Kit, který obsahuje:
    - 44-pinovou demo desku s PIC18F45K20 MCU
    - volnou verzi MPLAB C kompileru pro PIC 18 MCU
    - snadno pochopitelné lekce a cvičení
    - další softwarové nástroje, příklady se zdrojovým kódem a plnou dokumentaci.
- Přechodový socket
- ICD headery
- Rozšiřující sady MPLAB X IDE procesoru

1.5 Podpora zařízení a funkcí

Tabulky 1-1 a 1-2 zobrazují podporu zařízení a jejich funkce.

Tabulka 1-1 16-bitová (data memory) zařízení

Funkce	dsPIC33F, PIC24F/H	dsPIC30F **SMPS**(1)	dsPIC30F
Reset application	C	C	C
Run, Halt	C	C	C
Single step	C	C	C
Animate	C	C	C
Full-speed simulation	C	C	C
Hardware breakpoints	C	C	C
Peripheral freeze(2)	C	C	C
Break on data fetch or write	C	C	C
Break on stack overflow	N	N	N
Stopwatch	C	C	N
Pass counter	C	C	C
WDT overflow	C	C	N
Standard speed communication	C	C	C
Processor Pak	F	F	N

Legenda:
C = aktuální podpora
D = podpora záleží na zařízení
F = nyní bez podpory, ale do budoucna plánovaná
N = podpora nedostupná.

Poznámky:
1: Current Switch Mode Power Supply (SMPS) zařízení: dsPIC30F1010/2020/2023.
2: Tato funkce pracuje rozdílně v závislosti na vybraném zařízení.

Tabulka 1-2 8-bitová (data memory) zařízení

Funkce	PIC18FXXJ	PIC18F, PIC18F Enh, PIC18FXXK	PIC12F, PIC16F
Reset application	C	C	C
Run, Halt	C	C	C
Single step	C	C	C
Animate	C	C	C
Full-speed simulation	C	C	C
Hardware breakpoints	C	C	C
Peripheral freeze(1)	C	C	C
Break on data fetch or write	C	C	N
Break on stack overflow	C	C	N
Stopwatch	C	N	N
Pass counter	C	C	N
WDT overflow	C	N	N
Standard speed communication	C	C	C
Processor Pak	F	F	F

Legenda:
C = aktuální podpora
F = nyní bez podpory, ale do budoucna plánovaná
N = podpora nedostupná.

Poznámka:
1: Tato funkce pracuje rozdílně v závislosti na vybraném zařízení.

Kapitola 2. Teorie operace

2.1 Úvod

Zde se nachází zjednodušený popis toho, jak PICkit 3 pracuje. Cílem je poskytnout dostatek informací, aby bylo možné navrhnout cílovou desku, která bude kompatibilní s debuggerem jak pro debugování, tak programování. Základní teorie debugování v obvodu a programování je popsána, takže problémy, pokud se s nimi setkáte, můžete vyřešit rychle.

- PICkit 3 vs. PICkit 2
- Komunikace debuggeru s cílem
- Komunikační spojení
- Debugování
- Požadavky pro debugování
- Programování
- Zdroje používané debuggerem

2.2 PICkit 3 vs. PICkit 2

Programátor a debugger PICkit 3 má podobné funkce jako debugger PICkit 2.

Podobnosti obou debuggerů jsou následující:
- Napájení z PC skrze USB kabel
- Poskytuje programovatelné napájecí napětí

PICkit 3 se liší od PICkit 2 v následujícím:
- Zvětšený prostor pro EE programové obrazy (512Kbytů)
- True voltage reference
- Rozšířený rozsah napětí (1.8-5V VDD; 1.8-14V VPP)

2.3 Komunikace debuggeru s cílem

V následujících sekcích jsou rozebrány konfigurace systému debuggeru.

UPOZORNĚNÍ: Neměňte hardwarová spojení, zatímco je PICkit 3 nebo cíl napájený.

Standardní ICSP komunikace se zařízením

Systém debuggeru lze konfigurovat pro použití standardní ICSP komunikace jak pro programování, tak pro funkce debugování. Toto 6-pinové spojení je stejné, jako u staršího PICkit 2.

Modulární kabel lze zapojit do:

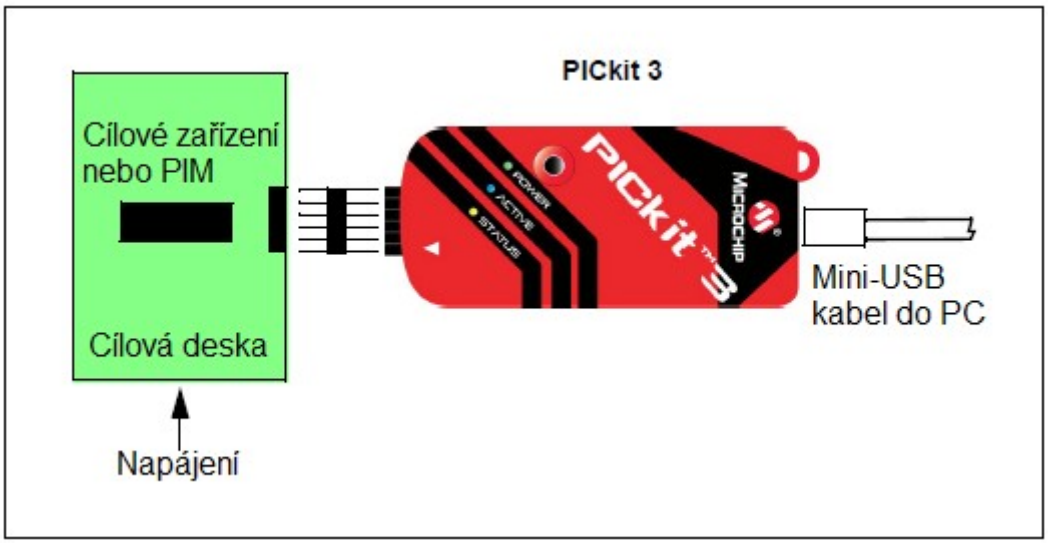
- odpovídajícího socketu na cíli, kde je cílové zařízení na cílové desce (obrázek 2-1) nebo
- standardního adaptér/header deskového komba (k dispozici jako Processor Pak), které je poté zapojeno do cílové desky (obrázek 2-2).

Poznámka: starší header desky používaly 6-pinový (RJ-11) modulární konektor namísto

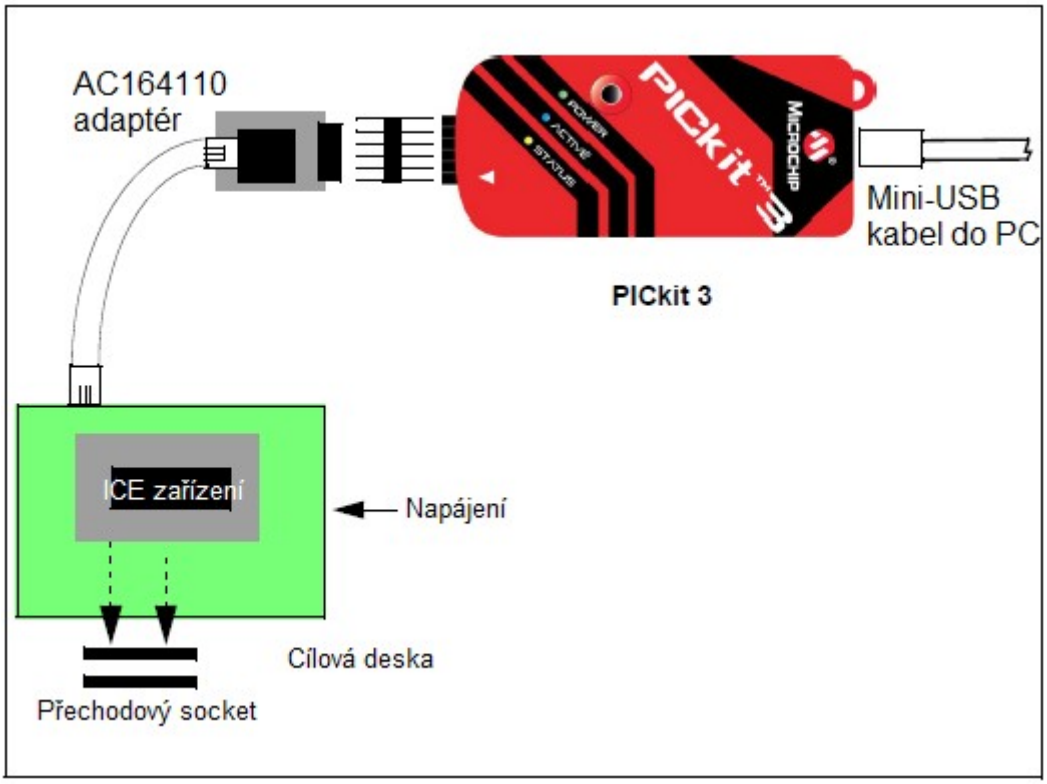
8-pinového konektoru, takže tyto headery lze napojit do debuggeru pomocí AC164110 ICSP adaptéru.

Pro více informací o standardní komunikaci, viz **Kapitola 10. Specifikace hardwaru.**

Obrázek 2-1: Standardní systém debuggeru - zařízení s ICE obvodem na desce



Obrázek 2-2: Standardní systém debuggeru - ICE zařízení



2.4 Komunikační spojení

2.4.1 Standardní komunikační cílová spojení

2.4.1.1 Pomocí konektoru Single In-Line

Použijte 6-pinový in-line konektor mezi PICkit3 a konektor cílové desky. Viz obrázek 2-1. Také viz Tabulka 2-1 a **Sekce 10.6 "Standardní komunikační hardware".**

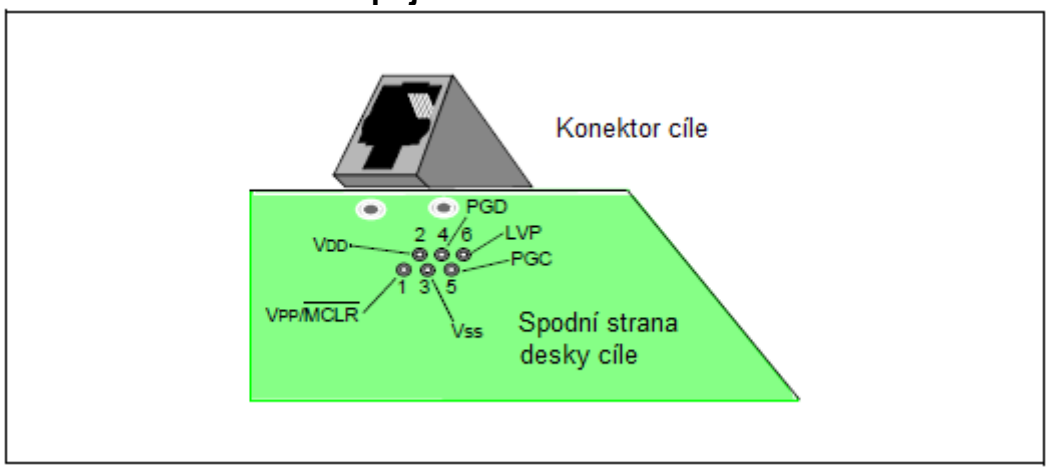
Tabulka 2-1: Rozložení pinů cílového konektoru

Pin konektoru	Pin mikrokontroléru
1	MCLR/VPP
2	VDD
3	Uzemnění
4	PDG (ICSPDAT)
5	PGC (ICSPCLK)
6	LVP

2.4.1.2 Pomocí adaptéru

Použijte adaptér AC164110 mezi PICkit 3 a modulární rozhraní (šestivodič) kabelu. Číslování pinů pro konektor je zobrazeno na spodní straně cílové desky plošných spojů na obrázku 2-3.
Poznámka: Kabelová spojení na debuggeru a cíli jsou k sobě vzájemně jako zrcadlové obrazy, jinými slovy pin 1 ja jednom konci kabelu je zapojen do pinu 6 na druhém konci kabelu. Viz **Sekce 10.6.2.3 "Specifikace modulárního kabelu".**

Obrázek 2-3: Standardní RJ-11 spojení s cílem

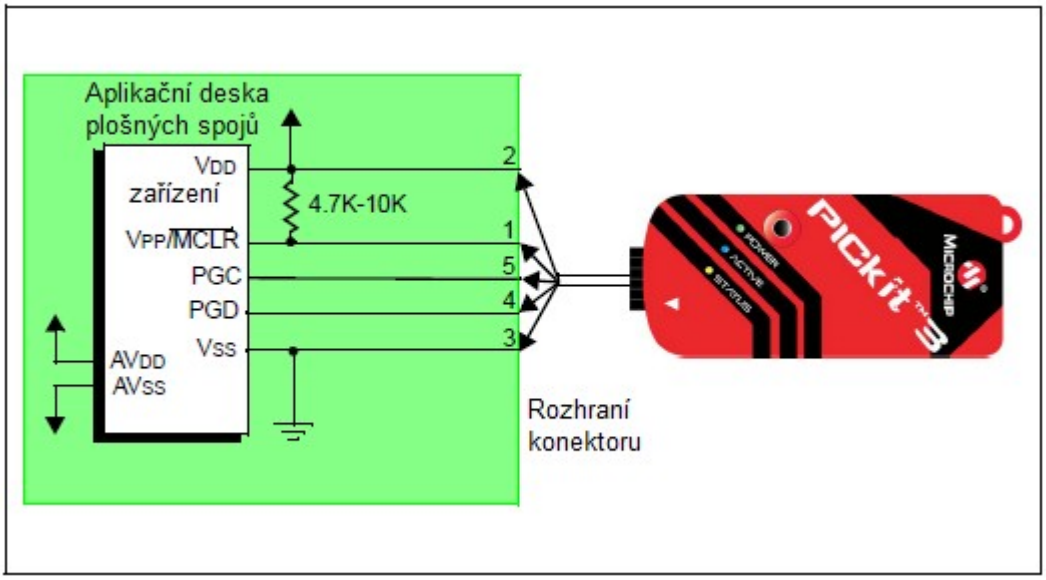


2.4.2 Obvod cílového spojení

Obrázek 2-4 ukazuje spojení PICkit 3 ke konektoru na cílové desce. Diagram také ukazuje zapojení z konektoru do zařízení na cílové desce plošných spojů. Doporučuje se připojit k VPP/MCLR lince vytažitelný rezistor (obvykle kolem 10kΩ), aby mohla být linka

synchronizována pro reset zařízení.

Obrázek 2-4: Obvod standardního spojení cíle



2.4.3 Napájený cíl

V následujících popisech jsou pouze tři linky aktivní a relevantní pro jádro operace debuggeru: piny 1 (VPP/MCLR), 5 (PGC) a 4 (PGD). Piny 2 (VDD) a 3 (VSS) jsou na obrázku 2-4 zobrazeny pouze pro úplnost. PICkit 3 má dvě konfigurace pro napájení cílového zařízení: interní debugger a externí napájení cíle.

Doporučený zdroj napájení je externí a odvozený od použítí cíle. V této konfiguraci je VDD cíle zjištěno debuggerem, aby dovolilo překlad úrovně pro operaci cíle s nízkým napětím. Pokud debugger nezjistí napětí na této VDD lince (pin 2 konektoru rozhraní), nebude pracovat.

2.4.4 Napájený debugger

Vnitřní energie debuggeru je omezena na 30mA. To se může hodit u velmi malých aplikací, které mají VDD zařízení oddělené od zbytku aplikace obvodem pro nezávislé programování. Nicméně se to nedoporučuje pro obecné použití, protože to vyžaduje více proudu od systému USB napájení z PC.

Ne všechna zařízení mají AVDD a AVSS linky, ale pokud jsou na cílovém zařízení přítomna, všechna musí být připojena k odpovídajcím úrovním v pořadí pro operaci debuggeru. Nelze je nechat nezapojená.

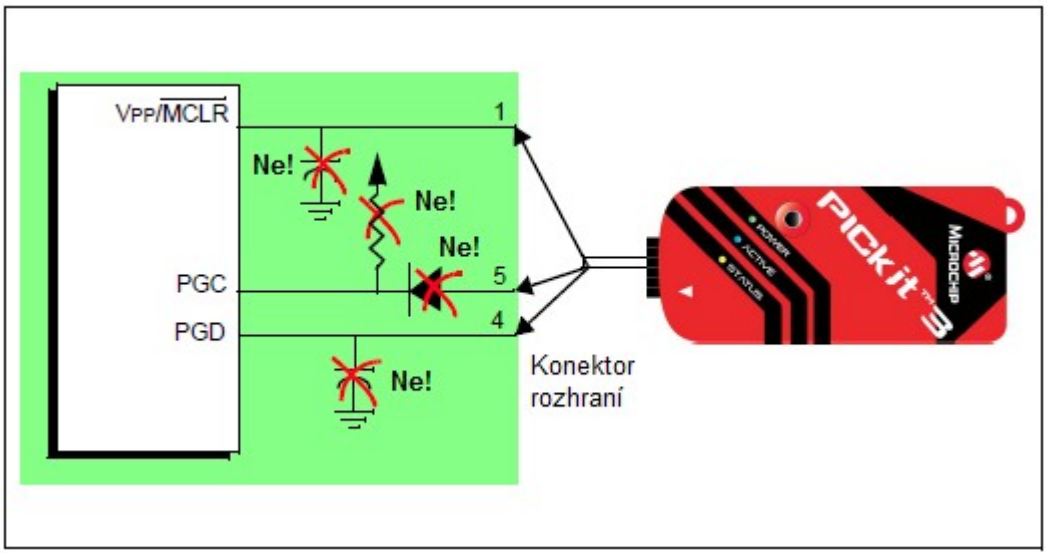
Obecně se doporučuje, aby všechny VDD/AVDD a VSS/AVSS linky byly zapojeny k odpovídajícím úrovním. Navíc zařízení s VCAP linkou (například PIC18FXXJ MCU) by měla být připojena k odpovídajícímu kapacitoru nebo levelu.

Poznámka: Propojení je velmi jednoduché. Jakékoliv problémy, které nastanou, jsou obvykle způsobeny jinými spojeními nebo komponenty na těchto kritických liniích, které narušují práci PICkit 3, jak uvidíte v následujících sekcích.

2.4.5 Obvody, které zabrání debuggeru ve funkci

Obrázek 2-5 ukazuje aktivní linku debuggeru s některými komponenty, které zabrání debugger systému PICkit 3 ve funkci.

Obrázek 2-5: Nesprávné komponenty obvodu



Zvláště je třeba řídit se následujícím:
- Nepoužívejte vytažitelné (pull-up) díly na PGC/PGD - naruší úrovně napětí, protože tyto naruší napěťové úrovně, jelikož tyto linky mají 4.7 kΩ pull-down rezistory v debuggeru.
- Nepoužívejte kapacitory na PGC/PGD - zabrání rychlým přechodpm na datových a hodinových linkách během programovacích a bedugovacích komunikacích.
- Nepoužívejte kapacitory na MCLR - zabrání rychlým přechodům VPP. Jednoduchý vytažitelný rezistor obvykle postačuje.
- Nepoužívejte diody na PGC/PGD - zabrání obousměrné komunikaci mezi debuggerem a cílovým zařízením.

2.5 Debugování

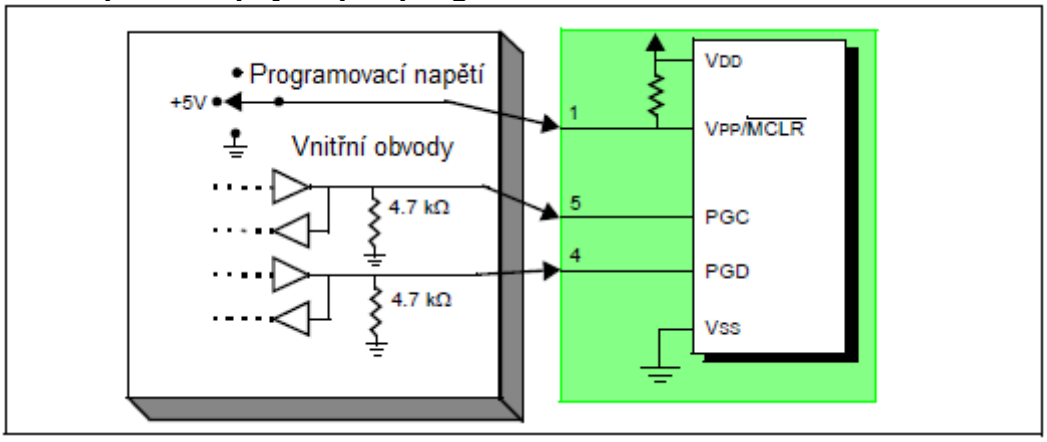
Při použití PICkit 3 jako debuggeru jsou dva kroky. První krok vyžaduje, aby aplikace byla programována do cílového zařízení (obvykle přímo pomocí samotného PICkit 3). Druhý krok používá interní debugovací hardware cílového Flash zařízení pro spuštění a otestování programu aplikace. Tyto dva kroky jsou přímo spojené s MPLAB IDE operacemi:

1. Programování kódu do cíle a aktivace speciálních debugovacích funkcí (pro detaily viz další sekce).
2. Použití debuggeru pro nastavení bodů přerušení a spuštění.

Pokud nelze zařízení naprogramovat správně, PICkit 3 nebude schopen provést debugování.

Obrázek 2-6 ukazuje základní spojení vyžadovaná pro programování. Povšimněte si, že je stejné jako na obrázku 2-4, ale pro přehlednost nejsou zobrazeny linky VDD a VSS z debuggeru.

Obrázek 2-6: Správné spojení pro programování



Vidíte zobrazen zjednodušený diagram některých vnitřních obvodů PICkit 3. Pro programování není třeba na cílovém zařízení mít hodiny, ale je třeba, aby bylo zajištěno napájení. Při programování debugger dává programovací úrovně na VPP-MCLR, zasílá pulsy hodin na PGC a sériová data skrze PGD. Pro potvrzení, že část byla naprogramována správně jsou hodiny zaslány do PGC a data jsou přečtena z PGD. Tím se potvrdí ICSP protokolu zařízení, že probíhá tvorba.

2.6 Požadavky pro debugování

Pro debugování (nastavení bodů přerušení, prohlížení registrů, atd.) pomocí systému PICkit 3 je kritických několik elementů, které musí pracovat správně:
- Debugger musí být připojen k PC. Musí být být z PC napájen pomocí USB kabelu a musí komunikovat se softwarem MPLAB X IDE pomocí USB kabelu. Pro více detailů viz **kapitola 3. "Použití debuggeru".**
- Debugger musí být připojen, jak vidíte na obrázku 2-6 k VPP, PGC a PGD pinům na cílovém zařízení pomocí modulárního kabelu rozhraní (nebo ekvivalentního). VSS a VDD jsou také vyžadovány k propojení mezi debuggerem a cílovým rozhraním.
- Cílové zařízení musí být napájené a mít funkční a zapnutý oscilátor. Pokud cílové zařízení není z jakéhokoliv důvodu spuštěné, PICkit 3 nemůže debugovat.
- Cílové zařízení musí mít správně naprogramovná svá konfigurační slova:
    - Konfigurační bity oscilátoru by měly odpovídat RC, XT, atd. v závislosti na designu cíle.
    - U některých zařízení je základně umožněn hlídací časovač (Watchdog Timer), který je třeba vypnout.
    - Cílové zařízení nesmí mít umožněnou ochranu kódem.
    - Cílové zařízení nesmí mít umožněnou ochranu čtení.
- LVP by mělo být vypnuté.

Když splníte podmínky vypsané výše, můžete pokračovat v následujícím:
- Sekvence operací vedoucí k debugování
- Detaily debugování

2.6.1 Sekvence operací vedoucí k debugování

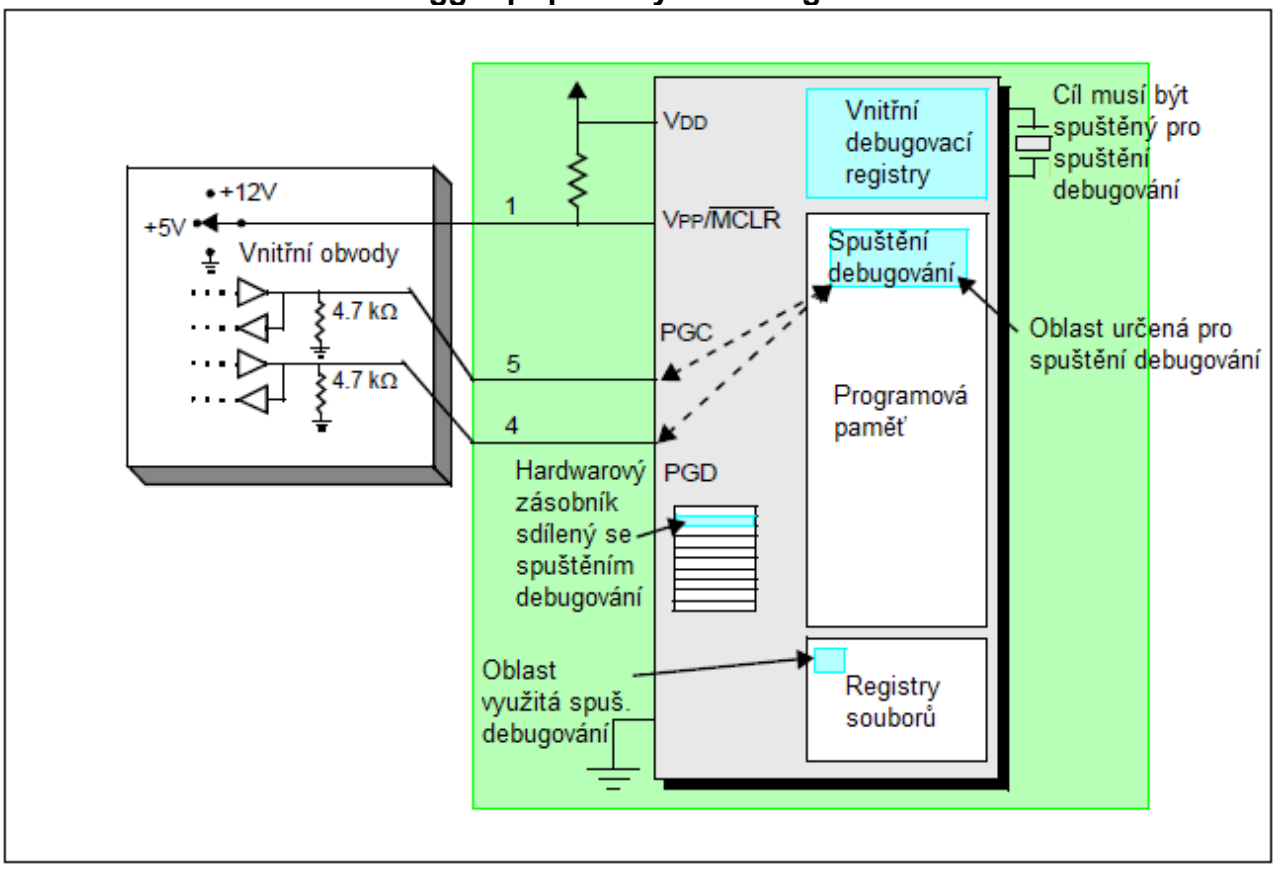
Pokud jsou splněny požadavky pro debugování, lze vykonat následující akce, když je PICkit 3 nastaven v MPLAB IDE menu jako proudový nástroj *(Debugger>Select Tool>PICkit3):*

- Aplikační kód je kompilován/sestaven výběrem *Project>Build Configuration>Debug.*
- Když vyberete Debugger>Program, aplikační kód je naprogramován do paměti zařízení skrze ICSP protokol, jak je popsáno výše.
- Malý program o provedení debugování se načte do vyšší oblasti programové paměti cílového zařízení. Jelikož provedení debugování musí zůstat v programové paměti, program aplikace nesmí využívat tento rezervovaný prostor. Některá zařízení mají speciální oblasti paměti určené pro provedené debugování. Pro detaily se podívejte na datasheet svého zařízení.
- Speciální "in-circuit debug" registry v cílovém zařízení jsou umožněny. Ty dovolují, aby bylo debuggerem aktivováno spuštění debugování.
- Cílové zařízení je udržováno v resetu udržováním VPP-MCLR linky nízko.

2.6.2 Detaily debugování

Obrázek 2-6 ukazuje systém PICkit 3, když je připraven spustit debugování.

Obrázek 2-6: PICkit™ 3 Debugger připravený na debugování



Obvykle je, kvůli zjištění, zda program aplikace funguje správně, bod přerušení vložen do kódu programu ze začátku. Pokud je bod přerušení nastaven z uživatelského rozhraní MPLAB IDE, adresa bodu přerušení je uložená ve speciálním vnitřním debugovacím registru. Příkazy na PGC a PGD komunikují přímo s těmito registry pro nastavení adresy

bodu přerušení.

Dále je v MPLAB IDE vybrána funkce *Debugger>Run* nebo ikona Run (šipka dopředu). Debugger dá příkaz ke spuštění debugování. Cíl začne od Reset vektoru a spustí se, dokud počítadlo programu nedosáhne adresy bodu přerušení, která byla předtím uložena ve vnitřním registru debugování.

Poté, co jsou spuštěny instrukce na adrese bodu přerušení, debugovací mechanismus cílového zařízení "vystřelí" a přenese programové počítadlo zařízení na spuštění debugování (podobně jako přerušení) a aplikace uživatele je zastavena. Debugger komunikuje se spuštěním debugování pomocí PGC a PGD, získá informace o stavu bodu přerušení a zašle je zpět do MPLAB IDE. MPLAB IDE poté vyšle sérii příkazů do debuggeru, aby získal informace o cílovém zařízení, jako je obsah registru souborů a stav CPU. Tyto příkazy jsou povedeny pomocí spuštění debugování.

Spuštění debugování běží stejně jako aplikace v programové paměti. Využívá některé oblasti zásobníku pro své dočasné proměnné. Pokud zařízení není z jakéhokoliv důvodu spuštěné, například nemá oscilátor, má chybné spojení napájení, zkrat na cílové desce, atd., pak spuštění debugování nemůže komunikovat s PICkit 3 a MPLAB IDE zobrazí chybovou hlášku.

Dalším způsobem, jak získat bod přerušení, je stisknout v MPLAB IDE tlačítko Halt (symbol "pauzy" napravo od symbolu šipky Run). To vymění linky PGC a PGD, takže debugovací mechanismus cílového zařízení přepne programové počítadlo z uživatelského kódu v programové paměti na spuštění debugování. Opět je cílová aplikace programu přerušena a MPLAB IDE použije komunikaci debuggeru se spuštěním debugování pro zjištění stavu cílového zařízení.

2.7 Programování

Použití PICkit 3 jako programátoru ro naprogramování skutečného (ne -ICE/-ICD) zařízení, například zařízení ne na header desce. Vyberte PICkit 3 z *Programmer>Select Programmer* a kompilujte/sestavte váš aplikační kód pomocí "Build Configuration" na liště nástrojů MPLAB IDE, nastaveným na "Release". Toto lze také nastavit výběrem *Project>Build Configuration>Release*.

Všechny funkce debugování jsou vypnuty nebo odstraněny, když je debugger použit jako programátor. Když používáte výběr Programmer>Program pro programování zařízení, MPLAB IDE zablokuje všechny in-circuit debug registry, aby programátor/debugger PICkit 3 programoval pouze cílový aplikační kód a konfigurační bity (a EEPROM data, pokud jsou s dispozici a vybraná) do cílového zařízení. Spuštění debugování se nenačte. Jako programátor může debugger pouze přepínat MCLR linku mezti resetem a startem cíle. Nelze nastavit bod přerušení a nelze zobrazit ani upravovat obsah registrů.

Systém PICkit 3 programuje cíl pomocí ICSP. VPP, PGC a PGD linky by měly být připojeny podle popisu výše. Během programování nejsou potřeba hodiny a lze programovat všechny módy procesoru, včetně ochrany kódu, Watchdog Timer a ochranu čtení.

2.9 Zdroje použité debuggerem

Pro kompletní seznam zdrojů použitých debuggerem pro vaše zařízení se podívejte na online help soubor v MPLAB IDE pto PICkit 3.

Kapitola 3. Instalace

3.1 Úvod

Zde se dozvíte, jak nainstalovat a použít PICkit 3.

- Instalace softwaru
- Připojení cíle
- Příprava cílové desky
- Příprava MPLAB X IDE

3.2 Instalace softwaru

Pro instalaci softwaru MPLAB IDE si nejprve sežeňte nejnovější instalační soubor MPLAB IDE (`MPxxxxx.exe`, kde `xxxxx` reprezentuje verzi MPLAB IDE) buď ze stránek Microchip (www.microchip.com) nebo z MPLAB IDE CD-ROM (DS51123). Poté spusťte tento soubor a postupujte instalačním dialogem.

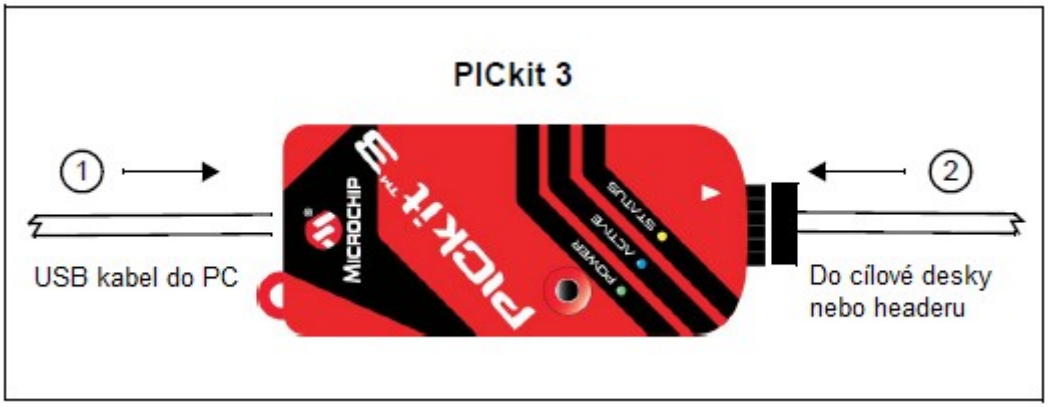
Poznámka: Pro použití programátoru/debuggeru PICkit 3 je třeba MPLAB IDE v8.20 nebo novější.

3.3 Připojení cíle

V přístroji je zabudované spojení pro výběr druhu komunikace s cílem. Viz **Sekce 2.3 "Komunikace debuggeru s cílem"** pro více detailů a diagram.

1. Pokud není zapojený, zapojte USB/napájecí kabel.
2. Připojte komunikační kabel(y) mezi debugger a cíl, pokud používáte zástrčku RJ11, nebo připojte přímo k 6-pinovému in-line headeru.

Obrázek 3-1: Zapojení komunikace a USB/napájecích kabelů



3.4 Příprava cílové desky

3.4.1 Použití produkčních zařízení

Pro produkční zařízení lze debugger zapojit přímo do cílové desky. Zařízení na cílové desce musí mít vestavený debugovací obvod, aby mohlo proběhnout debugování pomocí PICkit 3. Poraďte se s datasheetem, abyste zjistili, zda má zařízení potřebný debugovací obvod. Mělo by mít konfigurační bit "Background Debugger Enable".

Poznámka: V budoucnu bude možné použít zařízení s obvody, které podporují ICD.

Cílová deska musí mít konektor pro přizpůsobení se komunikacím vybraným pro debugger. Pro informace o spojení viz **Sekce 2.3 "Komunikace debuggeru s cílem"**, "Standardní komunikace ICSP zařízení".

3.4.2 Použití ICE zařízení

Pro ICE zařízení je vyžadována ICE header deska. Header deska obsahuje hardware, který je třeba pro emulaci specifického zařízení nebo skupiny zařízení. Pro více informací o ICE headerech viz *"Header Board Specification"* (DS51292).

Poznámka: V budoucnu bude možné použít ICD header desky s ICD zařízeními (*Device*-ICD).

Přechodový socket se používá s ICE headerem pro propojení headeru k cílové desce. Přechodové sockety jsou v dispozici v různých typech dovolujících běžnému headeru, aby byl propojen k jednomu z podporovaných stylů zapojení. Pro více informací o těchto socketech viz *"Transition Socket Specification"* (DS51194).

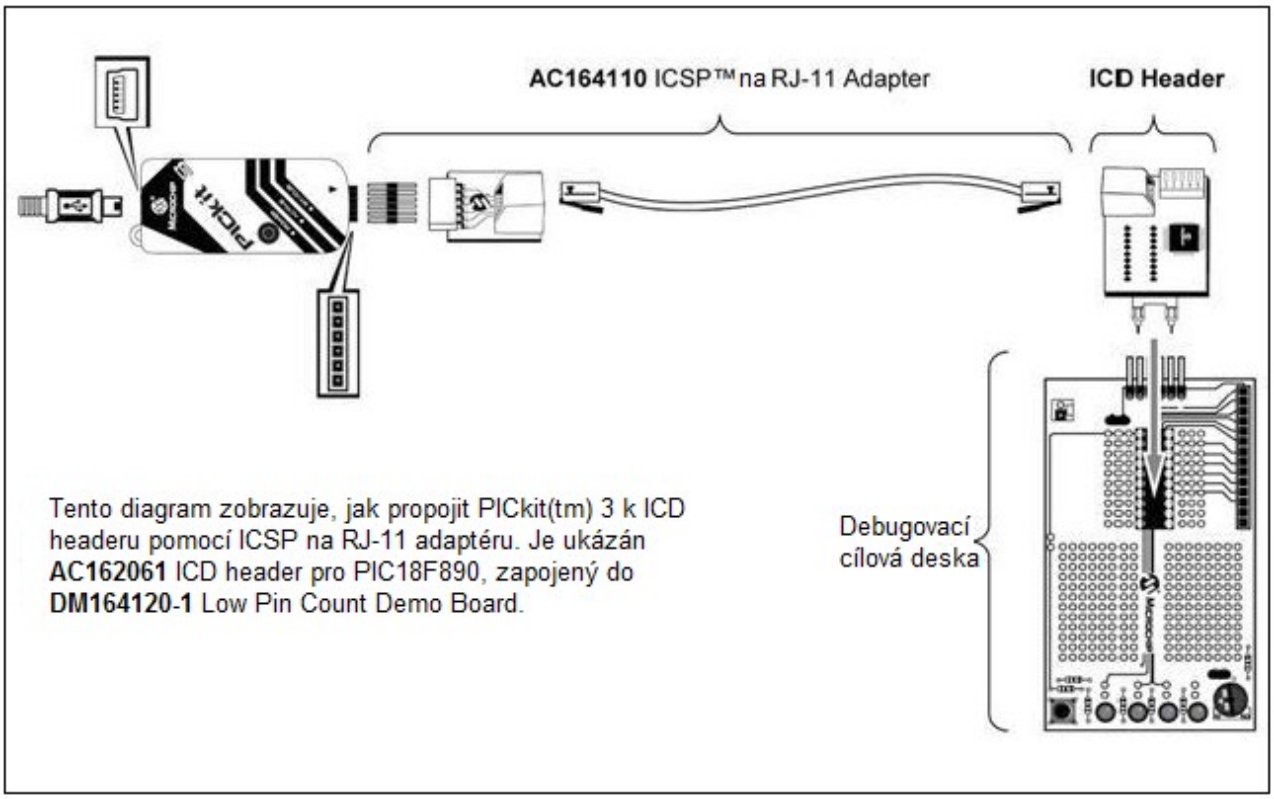
Rozložení header desky bude rozdílné pro headery nebo rozšiřující sady procesorů. Pro informace o spojení viz **Sekce 2.3 "Komunikace debuggeru s cílem"**, "Standardní komunikace ICSP zařízení".

3.4.3 Použití ICD Headeru

Všechny Baseline a některé Mid-Range PIC mikrokontroléry vyžadují speciální -ICD zařízení připojené k debugovacímu obvodu header desky pro umožnění funkce debugování. Pro seznam těchto zařízení a požadovaných čísel částí ICD header desky se podívejte do *"Processor Extension Pak and Header Specification"* (DS51292). Tato publikace je k dispozici online na stránkách www.microchip.com.

Každá ICD header deska má nezbytné -ICD zařízení, a to je použito v cílové desce namísto produkčního mikdokontroléru. Nicméně většina header desek má RJ-11 debugovací konektor, který vyžaduje AC164110 RJ-11 na ICSP adaptérovou sadu pro připojení k PICkit 3. Obrázek 3-2 ukazuje použití AC162061 ICD headeru pro PIC18F45K20 s AC164110 adaptérovou sadou a Low-Pin-Count Demo Board.

Obrázek 3-2: Použití ICD header desky



Mnoho Mid-Range PIC mikrokontrolérů a všechna PIC18 a 16-bitová PIC mikrokontrolérová zařízení nevyžadují ICD header a mohou být debugována přímo pomocí ICSP programovacích spojení.

3.4.4 Napájení cíle

Toto jsou nezbytnosti pro konfiguraci:

- Když používáte USB spojení, PICkit 3 lze napájet přes PC, ale tak lze poskytnout pouze omezený proud (do 30mA) při VDD od 1.8-5V do malé cílové desky.

- Požadovaná metoda je pro cíl, aby poskytl VDD, jelikož tak lze poskytnout vyšší proud. Další výhodou je, že je tak funkční detekce cíle plug-and-play, např. MPLAB X IDE vám dá vědět v okně Output, když nalezne cíl a detekuje zařízení.

Poznámka: Napětí cíle je použito pouze pro napájení ovladačů pro ICSP rozhraní; napětí cíle nepohání PICkit 3. Energie PICkit 3 je odebírána pouze z USB portu.

Pokud jste tak již neučinili, propojte PICkit 3 s cílem pomocí odpovídajících kabelů (viz **Sekce 3.3 "Připojení cíle"**). Poté napájejte cíl. Můžete také napájet cíl z PICkit 3, viz **Sekce 9.5.8 "Dialog nastavení, záložka Power"** pro instrukce.

3.6 Příprava MPLAB IDE

Jakmile je hardware připojený a napájený, MPLAB IDE lze připravit pro použití s debuggerem PICkit 3.

U některých zařízení musíte vybrat komunikační kanál v Configuration bits, např.

PGC1/EMUC1 a PGD1/EMUD1. Ujistěte se, že zde vybrané piny jsou stejné jako ty fyzicky připojené k zařízení.

Pro více o přípravě projektu a začátku s PICkit 3 viz **Kapitola 4. "Základní nastavení"**.

Kapitola 4. Základní nastavení

4.1 Úvod

Zde se dozvíte, jak začít s používáním programátoru/debuggeru PICkit 3.
- Spuštění softwaru MPLAB IDE
- Vytvoření projektu
- Zobrazení projektu
- Sestavení projektu
- Nastavení konfiguračních bitů
- Nastavení debuggeru nebo programátoru
- Omezení debuggeru/programátoru

4.2 Spuštění softwaru MPLAB IDE

Po instalace softwaru MPLAB IDE (**Sekce 3.2 "Instalace softwaru"**), jej můžete vyvolat následujícími způsoby:
- Vyberte *Start>Programs>Microchip>MPLAB IDE vx.xx>MPLAB IDE*, přičemž vx.xx je číslo verze.
- Dvakrát klikněte na ploše na ikonu MPLAB IDE.
- Spusťte soubor `mplab.exe` v podsložce `mplab ide\core` v instalační složce MPLAB IDE.

Pro více informací o softwaru viz:
- *"MPLAB IDE User's Guide"* (DS51519) - průvodce použitím MPLAB IDE.
- Soubory online pomoci - nejaktuálnější informace o MPLAB IDE a programátoru/debuggeru PICkit 3.
- Soubory Readme - informace o každém vydání jsou obsaženy v `Readme for MPLAB IDE.txt` a `Readme for PICkit 3 debugger.txt`. Oba soubory se dají najít v podsložce `Readmes` v instalační složce MPLAB IDE.

4.3 Tvorba projektu

Nejjednodušším způsobem, jak vytvořit nový projekt je vybrat *Project>Project Wizard*. S pomocí Project Wizard lze vytvořit nový projekt a jazykový nástroj pro budování projektu, který chcete vytvořit. Wizard vás provede procesem přidání zdrojových souborů, knihoven, atd. do různých "uzlů" v projektovém okně. Viz dokumentace MPLAB IDE pro více detailů o použití Project Wizard. Základní kroky jsou následující:

- Vyberte vaše zařízení (např. PIC18F45K20)
- Vyberte sadu jazykových nástrojů (např. Microchip C Compiler Toolsuite)
- Pojmenujte projekt
- Přidejte aplikační soubory (např. `program.c, support.s, counter.asm`)

Poznámka: Pokud nemáte základní linker skript ve ssvém projektu, Project Manager pro vás vybere odpovídající linker skript.

4.4 Zobrazení projektu

Poté, co Project Wizard vytvoří projekt, projekt a s ním spojené soubory jsou viditelné v okně Project. Klikněte pravým tlačítkem myši na okno projektu, abyste zobrazili menu s dalšími volbami pro přidání nebo odstranění souborů.

Viz dokumentace MPLAB IDE pro více detailů ohledně okna Project.

4.5 Sestavení projektu

Poté, co je projekt vytvořen, je třeba sestavit aplikaci. Tím vytvoříte objektový (hex) kód pro aplikaci, kterou lze programovat do cíle pomocí programátoru/debuggeru PICkit 3.

Pro volby sestavení vyberte *Project>Build Options>Project*.

Poznámka: V liště nástrojů Project Manager (*View>Toolbars>Project Manager*) vyberte "Debug" z roletové nabídky, když používáte PICkit 3 jako debugger, nebo vyberte "Release", když jej používáte jako programátor.

Když máte hotovo, vyberte *Project>Build All* pro sestavení projektu.

4.6 Nastavení konfiguračních bitů

Ačkoliv mohou být konfigurační bity zařízení nastaveny v kódu, mohou být také nastaveny v MPLAB IDE okně Configuration. Vyberte *Configure>Configuration Bits*. Kliknutím na text ve sloupci "Settings" je můžete změnit.

Mezi důležité konfigurační bity patří:

- Watchdog Timer Enable - u většiny zařízení je toto nastavení umožněno od začátku. je dobré tento bit zablokovat.
- Comm Channel Select - u některých zařízení budete muset vybrat komunikační kanál pro zařízení, např. PGC1/EMUC1 a PGD1/EMUD1. Ujistěte se, že zde vybrané piny jsou stejné, jako ty fyzicky napojené na zařízení.
- Oscillator - vyberte nastavení konfigurace, které odpovídá oscilátoru cíle.

4.7 Nastavení debuggeru nebo programátoru

Vyberte *Debugger>Select Tool>PICkit 3* pro výběr programátoru/debuggeru PICkit 3 jako debugovacího nástroje. Menu debuggeru a lišta nástrojů MPLAB IDE se změní na zobrazení voleb debugování, jakmile je vybraný nástroj. Také se otevře okno Output a zprávy související se stavem a komunikací PICkit 3 se zobrazí na záložce **PICkit 3**. Pro více informací viz **Sekce 9.2 "Funkce debugování"** a **"Sekce 9.3 "Debugovací dialogy/okna"**.

Vyberte *Programmer>Select Programmer>PICkit 3* pro výběr programátoru/debuggeru PICkit 3 jako programovacího nástroje. Menu programátoru a MPLAB IDE lišta nástrojů se změní na zobrazení voleb programátoru, jakmile je vybraný nástroj. Také se otevře okno Output a zprávy ohledně stavu ICE a komunikace budou zobrazeny na záložce **PICkit 3**. Pro více informací viz **Sekce 9.4 "Funkce programování"**.

Vyberte *Debugger>Settings* nebo *Programmer>Settings* pro otevření dialogu Settings (**Sekce 9.5 "Dialog Settings"**) a nastavte volby, které potřebujete.

Pokud dojde k chybě, viz:
- Kapitola 8. "Chybové hlášky"
- Kapitola 7. "Často kladené dotazy (FAQ)"

4.8 Omezení debuggeru/programátoru

Pro kompletní seznam omezení pro vaše zařízení nahlédněte do online pomoci PICkit 3 v MPLAB IDE vybráním *Help>Topics>PICkit 3* a klikněte na **OK**.

Kapitola 5. PICkit 3 Debug Express

5.1 Úvod

PICkit 3 Debug Express kit pracuje ve spojení s aplikací MPLAB IDE a může spustit, zastavit a procházet po krocích skrze programy. Lze nastavit jeden či více bodů přerušení a resetovat procesor. Jakmile je procesor zastavený, je možné prověřit a modifikovat obsah registru.

Pro více informací o použití MPLAB X IDE nahlédněte do následující dokumentace:
- MPLAB® IDE User's Guide (DS51519)
- MPLAB® IDE Online Help

5.2 Obsah PICkit 3 Debug Express Kit

PICkit 3 Degug Express kit (DV164131) obsahuje následující:
1. Vývojový programátor a debugger PICkit 3
2. USB kabel
3. 44-pinovou Demo desku se zařízením*
4. MPLAB IDE CD-ROM
5. PICkit 3 Debug Express C18 lekce (cvičení) na CD-ROM

* Pro debugování lze použít také desku Explorer 16.

5.3 Instalace hardwaru a softwaru

Nainstalujte PICkit 3 hardware a software podle **Kapitoly 3. "Instalace"**.

Poznámka: PICkit 3 Debug Express vyžaduje MPLAB IDE verzi 8.20 nebo novější.

5.3.1 Rezervované zdroje

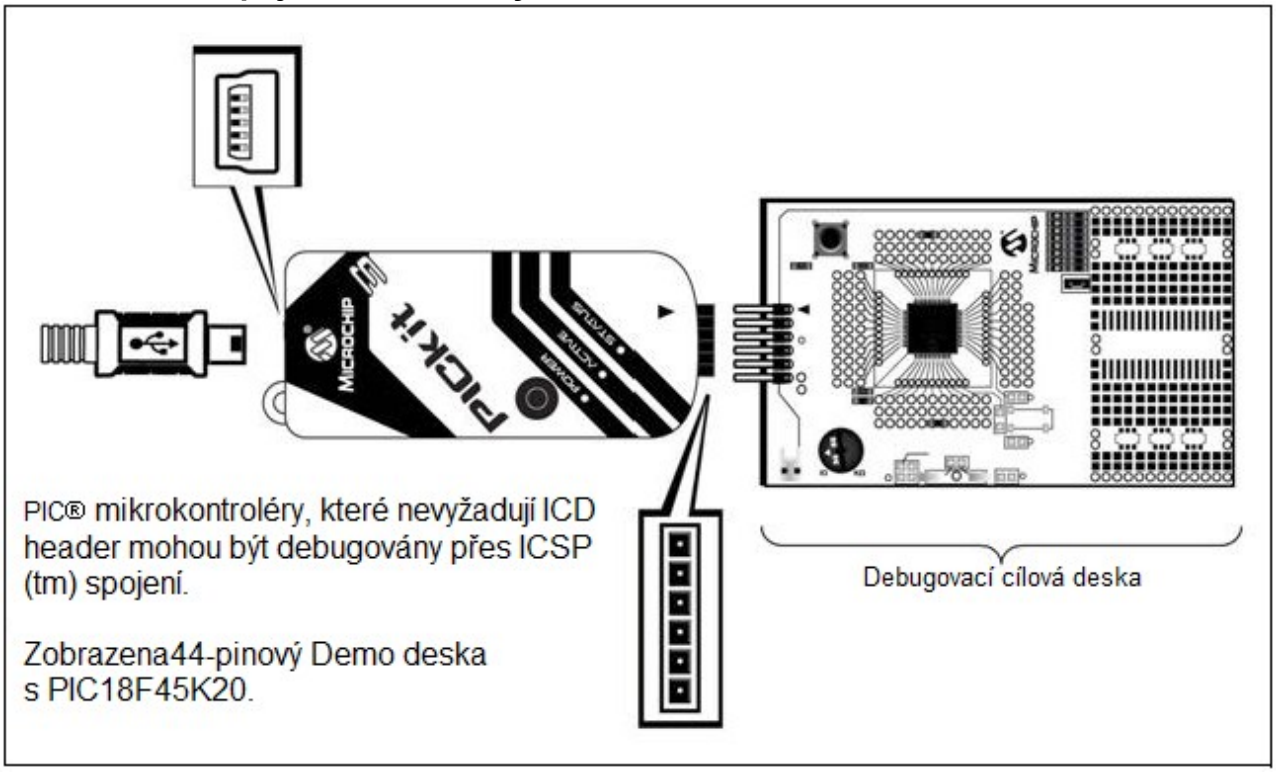
Kvůli vestavěné debugovací schopnosti ICD zařízení a ICSP funkci poskytované debuggerem používá PICkit 3 Debug Express při debugování některé on-chip zdroje.

Pro informace o zdrojích zařízení, které jsou potřeba pro in-circuit debugování nahlédněte do MPLAB PICkit 3 Help, kterou naleznete v MPLAB IDE pod *Help>Topics*. Informace o rezervovaných zdrojích pro zařízení najdete pod "Resources Used by MPLAB PICkit 3" a je stejná jako pro PICkit 3 Debug Express.

5.3.2 Připojení Demo desky

PIC18F45K20 obsažený na 44-pinové Demo desce lze debugovat jednoduše pomocí zapojení demo desky k PICkit 3, jak vidíte na obrázku 5-1.

Obrázek 5-1: Připojení Demo desky k PICkit™ 3



5.3.3 Konfigurační bity a Debug Express

PIC mikrokontrolérová zařízení, která nevyžadují ICD header a mohou být debugována přímo, obsahují DEBUG bit v konfiguračním slově (slovech), který aktivuje a deaktivuje Debug mód na PIC mikrokontroléru.

Tento bit je automaticky nastaven pomocí MPLAB IDE, když používáte PICkit 3 Debug Express a neměl by být specifikován v konfiguračních nastaveních zdrojového kódu.

Upozornění: Hodnota konfiguračního bitu DEBUG by za normálních podmínek neměla být specifikovaná ve zdrojovém kódu konfiguračních nastavení. Pokud tak učiníte, tento bit může být uplatněn při programování zařízení mimo debugger. Tím způsobíte nesprávnou nebo vůbec žádnou funkci zařízení v aplikačním obvodu.

Mnoho 16-bitových PIC mikrokontrolérových zařízení jako PIC24 a série dsPIC33 má vícenásobné ICSP programování a piny debugovacích portů označené PGC1/EMUC1 a PGD1/EMUD1. PGC2/UMUC2 a PGD2/EMUD2, atd. Zatímco jakýkoliv ICSP port může být použit pro programování, při debugování je aktivní pouze jeden port. Aktivní EMU port je nastaven na konfiguračních bitech zařízení. Pokud nastavení aktivního portu nesedí k EMU portu, ke kterému je PICkit 3 připojen, zařízení nebude moct vstoupit do módu Debug. V dialogu Configuration Bits MPLAB X IDE jsou tyto bity obvykle označovány jako "Comm Channel Select".

Část 2 - Řešení problémů

Kapitola 6. První kroky při řešení problémů

6.1 Úvod

Pokud máte problémy s PICkit 3, začněte zde:
- Pět otázek, na které byste si měli nejprve odpovědět
- Nejčastějších 10 důvodů, proč nelze debugovat
- Další věci na zvážení

6.2 Pět otázek, na které byste si měli nejprve odpovědět

1. S jakým zařízením pracujete?
Často je pro novější zařízení vyžadována novější verze MPLAB IDE. Žluté světlo = neotestovaná podpora.

2. Používáte Microchip demo desku nebo desku vlastní výroby?
Následovali jste rady ohledně rezistorů a kapacitorů pro komunikační spojení? Viz **Kapitola 2. "Teorie operace"**.

3. Napájeli jste cíl?
Debugger nemůže napájet cíl, pokud ten potřebuje víc než 30 mA.

4. Používáte USB rozbočovač? Je napájený?
Pokud máte i nadále problémy, zkuste použít debugger bez rozbočovače (zapojte jej přímo do PC).

5. Používáte standardní komunikační kabel (RJ-11) přiložený k debuggeru?
Pokud jste použili delší kabel, mohli jste způsobit chyby v komunikaci.

6.3 Nejčastějších 10 důvodů, proč nelze debugovat

1. Oscilátor nepracuje.
Zkontrolujte nastavení konfiguračních bitů pro oscilátor.

2. Cílová deska není napájená.
Zkontrolujte připojení napájecího kabelu.

3. Debugger se fyzicky odpojil od PC a/nebo cílové desky.
Zkontrolujte spojení komunikačních kabelů.

4. Zařízení je chráněné kódem.
Zkontrolujte vaše nastavení konfiguračních bitů pro kódovou ochranu.

5. Pokoušíte se znovu sestavit projekt, zatímco jste v módu Release.
Vyberte **Debug** v roletové nabídce na liště nástrojů projektu položku Build Configuration a poté přepracujte projekt.

6. Debugger je v MPLAB IDE vybrán jako programátor, ne jako debugger.

7. Byla přerušena komunikace mezi debuggerem a PC.
Obnovte spojení debuggeru v MPLAB IDE.

8. Cílová aplkace byla nějak pokažena nebo obsahuje chyby.
Zkuste předělat a přeprogramovat cílovou aplikaci. Poté započněte s Power-On-Reset cíle.

9. Jiná konfigurační nastavení narušují debugování.
Jakékoliv nastavení konfigurace, které by bránilo cíli ve spuštění kódu také zabrání debugger ve vložení kódu do módu debugování.

10. Debugger nemůže vždy vykonat požadovanou akci.
Například, debugger nemůže nastavit bod přerušení, pokud je cílová aplikace právě spuštěná.

6.4 Další věci na zvážení

1. Je možné, že chyba je pouze ojedinělou poruchou.
Zkuste operaci znovu.

2. Může se jednat o obecný problém s programováním.
K otestování přepněte na mód Programmer a naprogramujte cíl nejjednodušší možnou aplikací (např. program pro blikání LED). Pokud se program nespustí, pak víte, že je něco špatně se spuštěním cíle.

3. Je možné, že cílové zařízení bylo nějakým způsobem poškozeno (např. příliš vysokým proudem).
Vývojová prostředí jsou často nepřátelská vůči dílům. Zvažte použití jiného cílového zařízení.

4. Microchip Technology Inc. nabízí demonstrativní desku pro podporu většiny svých mikrokontrolérů.
Zvažte použití jedné z těchto desek, o kterých se ví, že fungují, abyste ověřili správnou funkci PICkit 3.

5. Prověřte operaci debuggeru, abyste zajistili správné spuštění aplikace.
Pro více informací, viz **Kapitola 2. "Teorie operace"**.

6. Pokud problémy přetrvávají, kontaktujte servis.

Kapitola 7. Často kladené dotazy (FAQ)

7.1 Úvod

Zde naleznete odpovědi na často kladené dotazy ohledně PICkit 3.
- Jak to funguje
- Co je špatně

7.2 Jak to funguje

- Co je v křemíku, který umožňuje komunikaci s PICkit 3?
PICkit 3 komunikuje s Flash křemíkem pomocí ICSP rozhraní. Využívá program spuštění debugování stažený do programu nebo testovací paměti.

- Jak je průchodnost procesoru ovlivněna spuštěním debugování?
Spuštění debugování nepracuje, když je v módu Run, takže když je kód spuštěný, nedochází k žádné redukci průchodnosti. Debugger tak "nekrade" žádné cykly z cílového zařízení.

- Jaký je PICkit 3 ve srovnání s jinými programátory/debuggery?
Viz **Sekce 2.2 "PICkit 3 vs. PICkit 2"**.

- Na debuggerech MPLAB ICE 2000/4000 musí data vycházet na sběrnici v pořadí, aby uskutečnila komplexní spustění na těchto datech. Je toto vyžadováno i pro programátor/debugger PICkit 3?
Debuggery MPLAB ICE 2000/4000 používají speciální debugovací čip (-ME) pro monitorování. Na programátoru/debuggeru PICkit 3 není -ME, takže zde nejsou žádné sběrnice pro externí monitorování. S programátorem/debuggerem PICkit 3 spíš než pomocí externích bodů přerušení je použit vestavěný obvod pro bod přerušení z debugovacím enginu - sběrnice a logika bodů přerušení jsou monitorovány uvnitř části.

- Má PICkit 3 komplexní body přerušení jako MPLAB ICE 2000/4000?
Ne. Můžete však přerušit na základě hodnoty v umístění datové paměti nebo programové adresy. Viz **Sekce 9.3.1 "Dialog bodu přerušení"** pro více informací.

- Je PICkit 3 opticky nebo elektricky izolovaný?
Ne. Nemůžete do proudového systému zavést nestálé nebo vysoké napětí (120V).

- Jaká omezení má standardní kabel?
Standardní kabel ICSP RJ-11 nedovoluje rychlosti větší než 15 Mbps.

- Nezpomalí PICkit 3 rychlost běhu programu?
Ne, zařízení poběží jakoukoliv rychlostí specifikovanou v datasheetu zařízení.

- Je možné debugovat dsPIC DSC běžící jakoukoliv rychlostí?
PICkit 3 je schopen debugování jakoukoliv rychlostí zařízení, jak je specifikovaná v datasheetu.

- Jaká je funkce pinu 6, LVP pinu?
Pin 6 je rezervován pro LVP (Low-Voltage Programming) spojení.

7.3 Co je špatně

- Můj počítač se přepnul do spánkového módu a debugger nyní nefunguje. Co se stalo?
Když používáte debugger po delší časový úsek, a zvláště jako debugger, rozhodně na svém počítači zamezte přepínání do spánkového módu. Najděte odpovídající nastavení a volbu vstupu do spánkového módu zrušte. Tím zajistíte, že veškerá komunikace v komponentech USB subsystému zůstane zachována.

- Nastavil jsem své vybavení, aby NEzamrzalo při pozastavení, ale náhle zamrzlo. Co se děje?
Pro zařízení dsPIC30F/33F a PIC24F/H je rezervován bit v ovládacím registru (obvykle buď 14 nebo 5) používaný debuggerem jako Freeze bit. Pokud jste provedli psaní v celém registru, je možné, že jste tento bit přepsali. (Bit je v módu Debug uživateli přístupný).
Abyste zabránili tomuto problému, pište pouze na bity, které si ve své aplikaci přejete změnit (`BTS, BTC`) namísto celého registru (`MOV`).

- Při použití 16-bitového zařízení došlo k náhlému resetu. Jak zjistím, co jej způsobilo?
Zvažte následující věci:
- Pro zjištění zdroje resetu zkontrolujte registr RCON.
- Hledejte v Interrupt Service Routine (ISR). Měli byste zahrnout styl kódu `trap.c`, např.,

```
void __attribute__((__interrupt__)) _OscillatorFail(void);
     :
void __attribute__((__interrupt__)) _AltOscillatorFail(void);
     :
void __attribute__((__interrupt__)) _OscillatorFail(void)
{
     INTCON1bits.OSCFAIL = 0;         //Clear the trap flag
     while (1);
}
     :
void __attribute__((__interrupt__)) _AltOscillatorFail(void)
{
     INTCON1bits.OSCFAIL = 0;
     while (1);
}
     :
```

- Použijte ASSERTy.

- Dokončil jsem debugování svého kódu. Nyní jsem naprogramoval svou část, ale ta neběží. Co je špatně?
Zvažte následující:
- Vybrali jste debugger jako programátor a pak jste zkusili programovat header desku? Header deska obsahuje -ICE/-ICD verzi zařízení a nemusí fungovat jako skutečné zařízení. Programujte pouze tato "běžná" zařízení s debuggerem jako programátorem. Běžná zařízení obsahují zařízení, která v sobě mají ICE/ICD obvody; ale nejedná se o speciální -ICE/-ICD zařízení, která jsou na header deskách.
- Vybrali jste debugger jako debugger a poté jste zkusili naprogramovat produkční

zařízení? Programování zařízení, když je debugger debuggerem naprogramuje spuštění debugování do programové paměti a nastaví jiné funkce zařízení pro debugování (viz **Sekce 2.6.1 "Sekvence operací vedoucí k debugování"**). Pro naprogramování finálního (k uveřejnění-release) kódu vyberte debugger jako programátor.
- Vybrali jste "Release" z roletové nabídky Build Configuration nebo Project? Toto musíte udělat pro finální kód. Přepracujte svůj projekt, přeprogramujte zařízení a zkuste znovu spustit svůj kód.

- Můj problém není na tomto seznamu. Co mám dělat?
Zkuste následující zdroje:
Sekce 2.8 "Zdroje použité debuggerem"
Sekce 8.3 "Obecné nápravné úkony"
Sekce 8.2 "Specifické chybové hlášky"
Kapitola 9. "Omezení"

Kapitola 8. Chybové hlášky

8.1 Úvod

PICkit 3 zobrazuje různé chybové hlášky. Některé jsou specifické a jiné lze vyřešit obecnými nápravnými úkony.
- Specifické chybobé hlášky
- Základní nápravné úkony

8.2 Specifické chybové hlášky

Chybové hlášky programátoru/debuggeru PICkit 3 jsou níže seřazeny v numerickém pořadí.

Poznámka: Čísla se ještě nemusí ukázat v zobrazených zkrávách. Použijte záložku Search v Help pro nalezení vaší zprávy a vyznačte ji níže.

Text v chybových hláškách níže je ve formě %x (proměnná) se zobrazí jako text relevantní k vaší situaci v skutečné chybové hlášce.

PK3Err0001: Chyba během psaní programování paměti.
PK3Err0002: Chyba během psaní na EEPROM.
PK3Err0003: Chyba během psaní na konfigurační paměť.
Viz **Sekce 8.3.1 "Úkony při chybách čtení/psaní".**
PK3Err0005: PICkit 3 je nyní zaneprázdněný a nelze jej nyní odpojit.
Pokud se objeví tato chyba, když se pokoušíte přepnout debugger jako debugger nebo programátor:
1. Počkejte - dejte debuggeru čas dokončit jakýkoliv úkol s aplikací. Poté to zkuste znovu.
2. Vyberte Halt pro pozastavení jakékoliv běžící aplikace. Poté zkuste opětovné přepnutí.
3. Odpojte debugger od PC. Poté jej zkuste znovu přepnout.
4. Vypněte MPLAB IDE.

PK3Err0006: Chyba během psaní user ID paměti.
PK3Err0007: Chyba během čtení programové paměti.
PK3Err0008: Chyba během čtení EEPROM.
PK3Err0009: Chyba během čtení konfigurační paměti.
PK3Err0010: Chyba během čtení user ID paměti.
Viz **Sekce 8.3.1 "Úkony při chybách čtení/psaní".**
PK3Err0011: Chyba blokového výmazu
Viz **Sekce 8.3.1 "Úkony při chybách čtení/psaní".**
Pokud to nefunguje, zkuste jiné zařízení.
PK3Err0012: Chyba stažení spuštění debugování.
Pokud se objeví tato chyba, když se pokoušíte programovat z menu Debugger:
1. Zrušte výběr debuggeru jako debugovacího nástroje.
2. Zavřete projekt a poté zavřete MPLAB IDE.
3. Restartujte debugger jako váš debugovací nástroj a pokuste se znovu naprogramovat vaše cílové zařízení.
4. Zrušte výběr debuggeru jako debugovacího nástroje a pokuste se opět naprogramovat vaše cílové zařízení.

Pokud to nefunguje, viz **Sekce 8.3.4 "Úkony při vadné instalaci".**

PK3Err0013: Chyba psaní NMMR registru.
PK3Err0014: Chyba psaní registru souborů.
Viz **Sekce 8.3.2 "Úkony při chybách komunikace debuggeru s cílem".**
PK3Err0015: Neúspěšný přesun dat. % bytů očekáváno, % bytů přesunuto.
Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**
PK3Err0016: Nelze přesunout. PICkit 3 nenalezen.
Debugger není připojen k PC.
PK3Err0017: Chyba čtení registru souborů.
PK3Err0018: Chyba čtení NMMR registru.
PK3Err0019: Chyba při čtení emulačních registrů.
PK3Err0020: Chyba při psaní emulačních registrů.
Viz **Sekce 8.3.2 "Úkony při chybách komunikace debuggeru s cílem".**
PK3Err0021: Nesprávná odezva příkazu. Odesláno x%, obdrženo x%.
PK3Err0022: Chyba získání informací o verzi PICkit 3.
PK3Err0024: Chyba stažení RS.
PK3Err0025: Chyba stažení AP.
Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**
PK3Err0026: Chyba stažení spuštění programu.
Pokud se objeví tato zpráva během pokusu o programování z menu DEbugger:
1. Zrušte výběr debuggeru jako debugovacího nástroje.
2. Zavřete projekt a poté zavřete MPLAB IDE.
3. Restartujte MPLAB IDE a opět otevřete váš projekt.
4. Zrušte výběr debuggeru jako debugovacího nástroje a pokuste se opět naprogramovat vaše cílové zařízení.
Pokud to nefunguje, viz **Sekce 8.3.4 "Úkony při vadné instalaci".**
PK3Err0027: Chyba blokového přesunu kvůli nesprávné checksum.
Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**
Také se ujistěte, že jste použili kabely o správné délce.
PK3Err0028: Chyba stažení databáze zařízení.
Pokud se objeví tato chyba:
1. Pokuste se opět o stažení. Může se jednat o ojedinělou chybu.
2. Zkuste ruční stažení. Vyberte *Debugger>Settings*, záložku **Configuration** a klikněte na **Manual Download.** Vyberte `.jam` soubor s nejvyšším číslem a klikněte na **Open**.
PK3Err0029: Chyba komunikace. Neočekávaná odezva příkazu x% z PICkit 3.
Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**
PK3Err0030: Nelze přečíst/najít soubor firmwaru %.
Pokud Hex soubor existuje:
- Opět se připojte a zkuste to znovu.
- Pokud to nefunguje, soubor může být vadný. Přeinstalujte MPLAB IDE.
Pokud Hex soubor neexistuje:
-Přeinstalujte MPLAB IDE.
PK3Err0031: Chyba připojení k PC.
PK3Err0032: Chyba nastavení PC.
Viz **Sekce 8.3.2 "Úkony při chybách komunikace debuggeru s cílem".**
PK3Err0033: % bytů očekáváno, % bytů obdrženo.
Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**
PK3Err0034: Tato verze MPLAB IDE nepodporuje revizi hardwaru %06x. Vylepšete před pokračováním na nejnovější verzi MPLAB IDE.
Najděte nejnovější MPLAB IDE na www.microchip.com.
PK3Err0035: Chyba při získání Device ID.
Viz **Sekce 8.3.1 "Úkony při chybách čtení/psaní".**

PK3Err0036: MPLAB IDE ztratil komunikaci s PICkit 3.

Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**

PK3Err0037: Vypršel čas při čekání na odezvu z PICkit 3.

PK3Err0038: Chyba inicializace PICkit 3.

PK3Err0039: Chyba autotestu PICkit 3.

Při této chybě debugger neodpovídá:

1. Odpojte a zapojte debugger.
2. Znovu připojte debugger v MPLAB IDE.
3. Pokud problém přetrvává, kontaktujte Microchip.

PK3Err0040: Cílové zařízení není připraveno na debugování. Zkontrolujte nastavení vašeho konfiguračního bitu a naprogramujte zařízení, než budete pokračovat.

Tato zpráva se objeví, když jste nenaprogramovali vaše zařízení poprvé a zkusili jste Run. Když poté obdržíte tuto zprávu nebo ihned po programování zařízení, nehlédněte do **Sekce 8.3.6 "Úkony při selhání debugování".**

PK3Err0043: Chyba stažení RS.

PK3Err0044: Chyba stažení AP.

PK3Err0045: Musíte připojit k cílovému zařízení, abyste mohli použít PICkit 3.

Nenalezen zdroj napájení.

1. Ujistěte se, že VDD a GND jsou připojeny mezi debugger a cíl.
2. Ujistěte se, že se cíl napájený.
3. Ujistěte se, že napájení cíle je postačující, aby bylo detekováno debuggerem (viz **Kapitola 10. "Specifikace hardwaru"**.)

PK3Err0046: Nastala chyba, když jste se pokoušeli přečíst hodnotu na stopkách. Stopky nemusí počítat přesně.

Viz **Sekce 8.3.2 "Úkony při chybách komunikace debuggeru s cílem".**

PK3Err0047: Chyba stažení bootloaderu.

Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**

PK3Err0052: Tato verze PICkit 3 hardwaru %x je zastaralá. Tyto verze MPLAB IDE bude podporovat pouze verze %x nebo vyšší.

Klikli jste na **Cancel**, když jste byli otázáni na stažení nejnovějšího firmwaru? Pokud ano, budete jej muset stáhnout nyní. Vyberte D*ebugger>Settings*, záložku **Configuration** a klikněte na **Manual Download**. Vyberte `.jam` soubor s nejvyšším číslem a klikněte na **Open**.

Pokud nemůžete najít žádné soubory ke stažení nebo pokud tento nefunguje (vadný soubor), budete muset získat nejnovější verzi MPLAB IDE a nainstalovat ji. Najděte na www.microchip.com nejnovější verzi MPLAB IDE.

PK3Err0053: Nemožné získat verze protokolu PICkit 3.

Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**

PK3Err0054: Definice protokolů MPLAB IDE PICkit 3 jsou zastaralé. před pokračováním musíte vylepšit MPLAB IDE.

Najděte nejnovější MPLAB IDE na www.microchip.com.

PK3Err0055: Nemožno nastavit verzi firmwaru.

PK3Err0056: Nemožno získat napětí z PICkit 3.

Viz **Sekce 8.3.3 "Úkony při chybách komunikace debuggeru s počítačem".**

PK3Err0068: Chyba při psaní do boot FLASH paměti.

PK3Err0069: Chyba při čtení boot FLASH paměti.

PK3Err0070: Chyba při psaní periferní paměti.

PK3Err0071: Chyba při čtení periferní paměti.

Viz **Sekce 8.3.1 "Úkony při chybách čtení/psaní".**

PK3Err0073: Zařízení je chráněné kódem.

Zařízení, na kterém se chystáte operovat (číst, programovat, blank check nebo potvrzení) je chráněné kódem, tj. kód nelze přečíst ani modifikovat. Zkontrolujte

nastavení konfiguračních bitů pro kódovou ochranu.
Pro zrušení kódové ochrany nastavte nebo vyčistěte odpovídající konfigurační bity v kódu nebo v okně Configuration Bits (*Configure>Configuration Bits*) podle datasheetu zařízení. Poté vymažte a přeprogramujte *celé* zařízení.

PK3Err0075: Nemožno nastavit napájení.

PICkit 3 nemůže napájet cíl.

PK3Err0080: Chyba nastavení stínového bitu/bitů.

8.3 Základní nápravné úkony

Tyto základní nápravné akce mohou vyřešit váš problém:

- Úkony při chybách čtení/psaní
- Úkony při chybách komunikace debuggeru s cílem
- Úkony při chybách komunikace debuggeru s počítačem
- Úkony při vadné instalaci
- Úkony při chybách komunikace USB portu
- Úkony při selhání debugování
- Úkony při vnitřních chybách

8.3.1 Úkony při chybách čtení/psaní

Pokud obdržíte chybu čtení nebo psaní:
1. Stiskli jste Abort? To může způsobit chyby čtení/psaní.
2. Zkuste úkon znovu. Může se jednat o jednorázovou chybu.
3. Ujistěte se, že je cíl napájen a na správné úrovni napětí pro toto zařízení. Pro požadované úrovně napětí nahlédněte do datasheetu zařízení.
4. Ujistěte se, že je spojení debuggeru k cíli správné (PGC a PGD jsou připojeny).
5. U chyb psaní se ujistěte, že je vybráno "Erase all before Program" na záložce Program Memory v dialogu Settings.
6. Ujistěte se, že použité kabely mají správnou délku.

8.3.2 Úkony při chybách komunikace debuggeru s cílem

PICkit 3 a cílové zařízení nejsou vzájemně synchronizované.
1. Vyberte **Reset** a poté zkuste úkon znovu.
2. Ujistěte se, že použité kabely mají správnou délku.

8.3.3 Úkony při chybách komunikace debuggeru s počítačem

PICkit 3 a MPLAB X IDE nejsou vzájemně synchronizované.
1. Odpojte a opět připojte debugger.
2. Připojte se k debuggeru.
3. Zkuste úkon znovu. Je možné, že chyba byla pouze jednorázová.
4. Vaše nainstalovaná verze MPLAB X IDE může být neprávná pro firmware načtený na PICkit 3. Následujte kroky popsané v **Sekci 8.3.4 "Úkony při vadné instalaci"**.

8.3.4 Úkony při vadné instalaci

Problém je nejspíše způsoben neúplnou nebo narušenou instalací MPLAB X IDE.

1. Odinstalujte všechny verze MPLAB X IDE z vašeho počítače.
2. Opět nainstalujte požadovanou verzi MPLAB X IDE.
3. Pokud problém přetrvává, kontaktujte Microchip.

8.3.5 Úkony při chybách komunikace USB portu

Problém je nejspíše způsoben vadným nebo neexistujícím komunikačním portem.

1. Připojte PICkit 3.
2. Ujistěte se, že je debugger fyzicky připojen k počítači na odpovídajících USB portu.
3. Ujistěte se, že byl odpovídající USB port vybrán v nastaveních (Settings) debuggeru.
4. Ujistěte se, že USB port není používán jiným zařízením.
5. Pokud používáte USB rozbočovat, ujistěte se, že je napájený.
6. Ujistěte se, že jsou načtené USB ovladače.

8.3.6 Úkony při selhání debugování

PICkit 3 byl neschopen učinit operaci debugování. Pro to může existovat více důvodů. Viz **Sekce 6.3 "Nejčastějších 10 důvodů, proč nelze debugovat"** a **Sekce 6.4 "Další věci na zvážení"**.

8.3.7 Úkony při vnitřních chybách

Vnitřní chyby se neočekávají a nemělo by k nim docházet. Jsou primárně pro vnitřní vývoj Microchip.

Nejjistější příčinou je vadná instalace (**Sekce 8.3.4 Úkony při vadné instalaci**).

Další možnou příčinou je vyčerpání systémových zdrojů.

1. Zkuste provést reboot vašeho systému pro uvolnění paměti.
2. Ujistěte se, že máte rozumné množství volného místa na pevném disku (a že není příliš fragmentované.)

Pokud problém přetrvává, kontaktujte Microchip.

Část 3 - Reference

Kapitola 9. Shrnutí funkcí debuggeru

9.1 Úvod
Shrnutí funkcí v menu, oknech a dialozích programátoru a debuggeru PICkit 3:

- Funkce debugování
- Debugovací dialogy/okna
- Funkce programování
- Dialog nastavení (Settings)

9.2 Funkce debugování
Když vyberete PICkit 3 z menu Debugger, mezi funkce MPLAB IDE se zařadí následující položky:

- Debugger Menu - další možnosti jsou přidány do roletového menu.
- Debugger Menu po kliknutím pravým tlačítkem myši - do tohoto menu jsou přidány další možnosti.
- Lišty nástrojů, statová lišta - pod lištou menu se objeví lišta nástrojů, ve stavové liště se objeví další informace.

9.2.1 Debugger Menu

Run F9
Spustí programový kód, dokud nenarazí na bod přerušení (breakpoint) nebo není vybrán Halt.
Spuštění začne na aktuálním programovém počítadle (Program Counter) (jak je zobrazeno ve stavové liště). Aktuální umístění programového počítadla také reprezentuje ukazatel v okně Program Memory. Zatímco program běží, je zablokováno několik dalších funkcí.

Animate
Tato volba způsobí, aby debugger doopravdy spustil při chodu jednotlivé kroky a aktualizoval hodnoty registrů během chodu.
Animate běží pomaleji, než funkce Run, ale dovoluje vám zobrazit změny hodnot registru v okně Special Function Register nebo v okně Watch.
Abyste provedli na této funkci Halt, použijte volbu menu *Debugger>Halt*, Halt na liště nástrojů nebo <F5>.

Halt F5
Halt zastaví spuštění programového kódu. Když kliknete na Halt, aktualizuje se stavová

informace.

Step Into F7
Jednotlivé kroky programem.
Pro kód assembly tento příkaz spustí jednu instrukci (jeden nebo více cyklů instrukcí) a poté se zastaví. Po spuštění jedné instrukce jsou všechna okna aktualizována.
U C kódu tento příkaz spustí jednu linii C kódu, což může znamenat spuštění jedné nebo více instrukcí sestavení, a poté se zastaví. Po spuštění se všechna okna aktualizují.
Poznámka: Nevstupte do instrukce `SLEEP`.

Step Over F8
Spuštění instrukce na aktuálním umístění programového počítadla. Při instrukci `CALL` Step Over spustí vyvolaný podprogram a zastaví na adrese následující `CALL`. Pokud je Step Over příliš dlouhá nebo "zdržuje", klikněte na Halt.

Step Out
Nedostupné.

Reset > Processor Reset F6
Vytvoří sekvenci Reset na cílovém procesoru. Díky tomu MCLR resetuje programové počítadlo na Reset vektor.

Breakpoints
Otevře dialog Breakpoint (viz **Sekce 9.3.1 "Dialog bodů přerušení"**). Použijte hardwarové body přerušení pro zastavení (halt) spuštění kódu na specifikovaných řádcích kódu. Nastavte více bodů přerušení v tomto dialogu.
Poznámka: Můžete také kliknout pravým tlačítkem myši nebo dvakrát kliknout levým na řádek kódu pro nastavení jednoduchého bodu přerušení.

Program
Stáhne váš kód do cílového zařízení.

Read
Přečte paměť cíle. Informace načte do MPLAB IDE.

Erase Flash Device
Vymaže veškerou Flash paměť.

Abort Operation
Přeruší jakoukoliv programovací operaci (např. programování, čtení, atd.) Přerušení operace nechá zařízení v neznámém stavu.

Reconnect
Pokusí znovu ustanovit komunikaci mezi PC a programátorem/debuggerem PICkit 3. Postup tohoto spojení je ukázán na záložce PICkit 3 dialogu Output.

Settings
Otevřete dialog PICkit 3 Settings (viz **Sekce 9.5 "Dialog nastavení (Settings)"**). Nastavte možnosti programu a firmwaru.

9.2.2 Debugger Menu po kliknutím pravým tlačítkem myši

Tyto možnosti menu debuggeru se objeví v menu po kliknutí pravým tlačítkem myši v kódu, například programová paměť a soubory zdrojového kódu. Popisy v jiýnch volbách menu, které tu nejsou vypsány, se dají najít v MPLAB IDE Help nebo MPLAB Editor Help.

Set Breakpoint
Nastaví nebo odstraní bod přerušení na aktuálně vybraném řádku.

Breakpoints
Odstraní, umožní nebo znemožní všechny body přerušení.

Run To Cursor
Spustí program na aktuálním umístění kurzoru. Původně Run to Here.

Set PC at Cursor
Nastaví programové počítadlo (Program Counter - PC) na umístění kurzoru.

Center Debug Location
Dá na střed aktuální řádek PC v okně.

9.2.3 Lišty nástrojů, statová lišta

Když je programátor/debugger PICkit 3 vybrán jako debugger, tyto lišty nástrojů se zobrazí v MPLAB IDE:
- Základní debugovací lišta nástrojů (Run, Halt, Animate, Step Into, Step Over, Step Out, Reset, Breakpoints).
- Jednoduchá programovací lišta nástrojů (Program, Read, Erase Flash Device).

Vybraný debugovací nástroj (PICkit 3), stejně jako další vývojářské informace, jsou zobrazeny ve stavové liště na spodku plochy MPLAB IDE. Pro informace o obsahu stavové lišty nahlédněte do MPLAB IDE online pomoci.

9.3 Debugovací dialogy/okna

Otevřete následující debugovací dialogy a okna pomocí položek menu zmíněných v **Sekci 9.2 "Funkce debugování"**.

Dialog Breakpoints
- Nastaví dialog Breakpoint
- Dialog Stopwatch
- Dialog Event Breakpoints

9.3.1 Dialog Breakpoints

Pro nastavení bodů přerušení vyberte *Debugger>Breakpoints*.
V tomto dialogu nastavte různé typy bodů přerušení. Klikněte na **Add Breakpoint** pro přidání bosů přerušení do dialogového okna. V závislosti na vašem vybraném zařízení zde mohou být další tlačítka pro pokročilejší volby bodů přerušení.

9.3.1.1 Okno dialogu Breakpoint

Okno o každém bodu přerušení vidíte v tomto okně.

Tabulka 9-1 Okno dialogu Breakpoint

Volba	Funkce
Breakpoint Type	Typ bodu přerušení - program nebo data
Address	Hex adresa umístění bodu přerušení
File Line #/Symbol Name	Název souboru a číslo řádku umístění bodu přerušení
Enabled	Vyberte pro umožnění bodu přerušení

Jakmile byl přidán do okna bod přerušení, můžete na něj kliknout pravým tlačítkem myši pro otevření menu s volbami:
- Delete - smaže vybraný bod přerušení.
- Edit/View - otevře dialog Set Breakpoint
- Delete All - smaže všechny vypsané body přerušení
- Disable All - zablokuje všechny vypsané body přerušení

9.3.1.2 Tlačítka dialogu Breakpoint
Použijte tlačítka pro přidání bodu přerušení a nastavení přídavných podmínek přerušení. Také jsou k dispozici stopky pro použití s body přerušení a spouštěči.
Poznámka: Zobrazená tlačítka záleží na vybraném zařízení.

Tabulka 9-2 Tlačítka dialogu Breakpoint

Volba	Funkce	Související dialog
Add Breakpoint	Přidá bod přerušení	**Sekce 9.3.2 "Dialog Set Breakpoint"**
Stopwatch	Nastaví stopky	**Sekce 9.3.3 "Dialog Stopwatch"**
Event Breakpoints	Nastaví přerušení na událost	**Sekce 9.3.4 "Dialog Event Breakpoints"**

9.3.2 Dialog Set Breakpoint

Klikněte na **Add Breakpoint** v dialogu Breakpoints pro zobrazení tohoto dialogu.
Zde vyberte bod přerušení pro dialog Breakpoints.

9.3.2.1 Záložka Program Memory

Zde nastavíte bod přerušení programové paměti.

Tabulka 9-3 Bod přerušení programové paměti

Volba	Funkce
Address	Umístění bodu přerušení v hexu.
Breakpoint Type	Typ bodu přerušení programové paměti. Viz datasheet zařízení pro více informací o čtení/psaní. *Program Memory Execution* - přerušení spuštění adresy výše. *TBLRD Program Memory* - přeruší čtení adresy výše. *TBLWT Program Memory* - přeruší psaní na adresu výše.
Pass Count	Přeruší na podmínce pass count. *Always break* - vždy přeruší, jak je specifikováno v "Breakpoint type". *Break occurs Count instructions after Event* - čeká na instrukci Count (0-255) před přerušením po události specifikované v "Breakpoint type" *Event must occur Count times* - přeruší pouze po události specifikované v "Breakpoint type", která se udá podle počtu Count (0-255).

9.3.2.2 Záložka Data Memory

Zde nastavíte bod přerušení datové paměti.

Tabulka 9-4 Bod přerušení Data Memory

Volba	Funkce
Address	Umístění bodu přerušení v hexu.
Breakpoint Type	Typ bodu přerušení datové paměti. Viz datasheet zařízení pro více informací o X Bus čtení/psaní. *X Bus Read* - přerušení na X bus čtení adresy výše. *X Bus Read Specific Byte* - přerušení na X bus čtení adresy výše pro specifickou hodnotu bytu v "Specific Value". *X Bus Read Specific Word* - přeruší X bus čtení na adrese výše pro specifickou hodnotu slova v "Specific Value". *X Bux Write* - přeruší na X bus psaní adresy výše. *X Bus Write Specific Byte* - přerušení na X bus psaní adresy výše pro specifickou hodnotu bytu v "Specific Value". X Bus *Write Specific Word* - přeruší X bus psaní na adrese výše pro specifickou hodnotu slova v "Specific Value".
Pass Count	Přeruší na podmínce pass count. *Always break* - vždy přeruší, jak je specifikováno v "Breakpoint type". *Break occurs Count instructions after Event* - čeká na instrukci Count (0-255) před přerušením po události specifikované v "Breakpoint type" *Event must occur Count times* - přeruší pouze po události specifikované v "Breakpoint type", která se udá podle počtu Count (0-255).

9.3.3 Dialog Stopwatch

Klikněte v dialogu Breakpoints na **Stopwatch** pro zobrazení dialogu Stopwatch Setup. Použijte stopky s body přerušení pro časování spuštění kódu. Stopky umožňují časování od jednoho stavu bodu přerušení/spouštěče k dalšímu. Hodnota stopek je decimální.

Tabulka 9-5 Nastavení Stopwatch

Volba	Funkce
Start Condition	Vybere bod přerušení, který je k dispozici nebo podmínku spouštěče pro start stopek. Body přerušení/souštěče jsou ty přidané dříve v dialogu breakpoint. Pro vymazání podmínky startu vyberte **None**. Pro pozastavení (halt) programu běžícího za tohoto stavu vyberte políčko vedle "Start condition will cause the target device to halt".
Stop Condition	Vybere bod přerušení, který je k dispozici nebo podmínku spouštěče pro zastavení stopek. Body přerušení/souštěče jsou ty přidané dříve v dialogu breakpoint. Pro vymazání podmínky zastavení vyberte **None**. Pro pozastavení (halt) programu běžícího za tohoto stavu vyberte políčko vedle "Stop condition will cause the target device to halt".
Reset stopwatch on run	Resetuje hodnotu stopek na nulu pokaždé, když běží program.

Pro nastavení bodu přerušení v kódu učiňte jednu z následujících věcí:

- Dvakrát klikněte nebo klikněte pravým tlačítkem myši pro nastavení jednotlivého bodu přerušení.
- Vyberte Debugger>Breakpoints pro otevření dialogu Breakpoints a nastavte více bodů přerušení a podmínek bodů přerušení. Viz **Sekce 9.3.1" Dialog Breakpoints"** pro více informací.

Pro rozhodnutí času mezi body přerušení použijte stopky:
1. Otevřete dialog Breakpoints (*Debugger>Breakpoints*).
2. Klikněte na **Stopwatch** pro otevření dialogu Stopwatch.
3. Pod "Start Condition" vyberte bod přerušení z roletové nabídky. Také rozhodněte. zda "Start condition will cause the target device to halt".
4. Pod "Stop Condition" vyberte bod přerušení z roletové nabídky. Také rozhodněte. zda "Stop condition will cause the target device to halt".
5. Rozhodněte, zda budete potřebovat "Reset stopwatch on run".
6. Klikněte na **OK**.

9.3.4 Dialog Event Breakpoints

Klikněte na **Event Breakpoints** v dialogu Breakpoints pro zobrazení tohoto dialogu.

Vyberte podmínku, v níž se program vždy přeruší:
- Break on Watchdog Timer - přeruší se pokaždé, když vyprší čas Watchdog Timer. Ujistěte se, že je Watchdog Timer povolen v konfiguračních bitech.
- Break on `SLEEP` instruction - přeruší, když narazí v programu na instrukci `SLEEP`.

9.4 Funkce programování

Když vyberete programátor/debugger z menu Programmer, budou do následujících funkcí MPLAB IDE přidány programovací položky:
- Menu Programmer
- Nástrojové lišty, stavová lišta

9.4.1 Menu Programmer

Program
Programuje specifikované oblasti paměti: programovou paměť, konfigurační bity, umístění ID a/nebo EEPROM data. Viz dialog Settings pro volby programování.

Verify
Ověří programování specifikovaných oblastí paměti: programovou paměť, konfigurační bity, umístění ID a/nebo EEPROM data.

Read
Přečte specifikované oblasti paměti: programovou paměť, konfigurační bity, umístění ID a/nebo EEPROM data. Viz dialog Settings pro volby čtení.

Blank Check All
Zkontroluje, zda je celá paměť zařízení vymazaná/prázdná.

Erase Flash Device
Vymaže celou Flash paměť.

Abort Operation
Přeruší jakoukoliv operaci programování (např. programování, čtení, atd.). Přerušení operace ponechá zařízení v neznámém stavu.

Reconnect
Pokus o znovuobnovení komunikace mezo PC a programátorem/debuggerem PICkit3. Postup tohoto spojení je zobrazen na záložce PICkit 3 dialodu Output.

Settings
Otevřete dialog Programmer (viz **Sekce 9.5 "Dialog Settings"**). Nastavte možnosti programu a firmwaru.

9.4.2 Nástrojové lišty/stavová lišta
Když je vybrán programátor/debugger PICkit 3 jako programátor, zobrazí se v MPLAB IDE tyto nástrojové lišty:

- Základní programovací lišta (Program, Read, Verify, Erase Flash Device, Blank Check All).

Vybraný programátor (PICkit 3), stejně jako další programovací informace jsou zobrazeny ve stavové liště na spodku plochy MPLAB IDE. Viz MPLAB IDE online help pro informace o stavovém řádku.

9.5 Dialog Settings

Vyberte buď *Debugger>Settings* nebo *Programmer>Settings* pro otevření dialogu Settings a nastavte programátor/debugger PICkit 3.

Poznámka: zobrazené záložky budou záviset na vybraném zařízení.

- Dialog Settings, záložka Program Memory
- Dialog Settings, záložka Configuration
- Dialog Settings, záložka Freeze on Halt
- Dialog Settings, záložka Status
- Dialog Settings, záložka Clock
- Dialog Settings, záložka Secure Segment
- Dialog Settings, záložka Warnings
- Dialog Settings, záložka Power
- Dialog Settings, záložka Limitations

9.5.1 Dialog Settings, záložka Program Memory

Tato záložka vám umožňuje nastavit volby debugování/programování.

Tabulka 9-6 Volby Program memory

Allow PICkit 3 to select memories and ranges	Pokud toto vyberete, PICkit™ 3 použije vaše vybrané zařízení a základní nastavení pro rozhodnutí, co programovat.
Manually Select memories and ranges	Pokud toto vyberete, musíte sami vybrat paměti, rozsah programové paměti a programové volby pro vybrané zařízení.
Memories (paměti):	
Program	Pro naprogramování Program Memory do cíle.
Configuration	Pro naprogramování Configuration bits do cíle. **Poznámka**: tato paměť je vždy naprogramována, když je debugger nastaven jako debugger.
EEPROM	Pro vymazání a následně naprogramování EEPROM paměti na cíl, pokud je to možné. Odoznačte pro vymazání EEPROM paměti z cíle.
ID	Pro naprogramování ID paměti do cíle.
Boot Flash	Pokud podporováno, označte pro naprogramování boot memory do cíle.
Program Memory Range (rozsah programovací paměti):	
Start, End	Rozsah startovní a koncové hex adresy v programové paměti pro programování, čtení nebo potvrzení. Pokud se objeví chyba programování kvůli nesprávné koncové adrese, musíte provést opětovné připojení, opravit koncovou adresu a opět programovat. **Poznámka:** rozsah adresy se nepočítá na funkci Erase. Funkce Erase smaže všechna data na zařízení.

Program Options (programové volby):	
Erase all before Program	Označte pro vymazání veškeré paměti před začátkem programování. Pokud neprogramujete nová nebo již vymazaná zařízení, je důležité mít toto políčko označené. Pokud není, zařízení nebude vymazáno a kód programu se spojí s kódem, který už je v zařízení.
Preserve Program Memory Range	Označte pro zachování rozsahu paměti. Start, End - rozsah startovní a konečné hex adresy v programové paměti pro zachování.
Automatically (automaticky)	
Program after successful build	Po úspěšném vybudování kódu aplikace program toto zakóduje do zařízení.
Run after successful program	Poté, co je kód aplikace úspěšně naprogramován do cílového zařízení, spustí kód.

9.5.2 Dialog Settings, záložka Configuration

V této záložce konfigurujte operaci debuggeru.

Tabulka 9-7 Položky konfigurace

Download Firmware	Nastavte volby stažení firmwaru.
Auto Download Latest Firmware	Označte pro dovolení automatického stažení nejnovější verze firmwaru do cílového zařízení (**doporučeno**).
Manual Download	Ruční výběr souboru firmwaru pro stažení do cílového zařízení.

9.5.3 Dialog Settings, záložka Freeze on Halt

Na této záložce vyberte periferie, aby se při halt zastavily (zmrzly).

Zařízení PIC18 MCU

Pro zmrazení/rozmrazení všech periferních zařízení pri halt označte/odoznačte okénko "Freeze on Halt". Pokud to nezmrazí vámi chtěné periferní zařízení, dávejte si pozor, protože ne všechny periferie mají možnost zmrazení při halt a nelze je debuggerem ovládat.

Zařízení dsPIC30F/33F, PIC24F/H a PIC32MX

Pro periferie v seznamu "Peripherals to Freeze on Halt" označte, že se mají při halt zmrazit. Odoznačte je, abyste je nechali běžet, zatímco program byl pozastaven. Pokud nevidíte zařízení na seznamu , označte "All Other Peripherals". Pokud to nezmrazí vámi chtěné periferie, uvědomte si, že některé periferie nemají možnost zmrazení při halt a nelze je ovládat debuggerem.

Pro výběr všech periferií, včetně "All Other Peripherals" klikněte na **Check All**. Pro odoznačení všech pefiferií, včetně "All Other Peripherals" klikněte na **Uncheck All**.

9.5.4 Dialog Settings, záložka Status

V této záložce se zobrazí stav vašeho PICkit 3 systému.

Tabulka 9-8 Položky Status

Versions (verze)	
Firmware Suite Version	Verze firmwaru debuggeru. Sestává z tří položek specifikovaných níže.
Algorithm Plug-in Version	Verze plug-inu algoritmu debuggeru. Pro vaše vybrané zařízení je použit algoritmus pro podporu zařízení zapojeného do cíle.
OS Version	Verze operačního systému debuggeru.
Voltages (napětí)	
PICkit™ 3 VPP	VPP debuggeru
PICkit 3 VDD	VDD debuggeru
Refresh Voltages	Když je aktivována tato záložka, dojde ke kontrole položek záložky Status. Pro zobrazení aktualizací klikněte na toto tlačítko.

9.5.5 Dialog Settings, záložka Clock

Na této záložce vyberte mód hodin debuggeru. Označte políčko pro "FRC in debug mode" pro použití rychlých RC hodin během debugování.

Emulace zařízení se odehraje rychlostí procesoru. Pokud běží na nízké frekvenci (např. 32kHz), všechny operace jako krokování, obnovení okna sledování, atd. jsou pomalé. Nicméně, některé procesory mají zabudovaný rychlý RC oscilátor. Pokud je označena volba "FRC in debug mode", poté co aplikace běžící normální rychlostí provede halt, přepne se na rychlejší oscilátor, což umožní emulaci a periferiím (pokud nejsou zmrazené), aby běžely rychleji.

9.5.6 Dialog Settings, záložka Secure Segment

Na této záložce můžete provést nastavení bezpečnostních segmentů zařízení CodeGuard™ Security.

Pro více detailů o fungování CodeGuard Security nahlédněte do manuálu CodeGuard Security pro 16-bitová zařízení (DS70180) a dsPIC33F/PIC24H a dsPIC30F specifikace programování zařízení k nalezení na webu Microchip.

Tabulka 9-9 Volby Secure Segment

Full Chip Programming	Klikněte pro výběr programování všech segmentů programové paměti.
Segment Programming	Klikněte pro výběr segmentu programování. Vyberte z: - Segmentů boot, secure a general - Segmentů secure a general - Pouze general segmentů

9.5.7 Dialog Settings, záložka Warnings

Na této záložce je zobrazen seznam všech varování programátoru/debuggeru PICkit 3.
- Označte varování pro jeho umožnění. Varování bude zobrazeno v okně Output.
- Odoznačte varování pro jeho zablokování.

Varování nejsou chyby a nezastaví váš projekt před sestavením. Pokud se objeví chybová hláška, viz **Kapitola 8. "Chybové hlášky"**.

9.5.8 Dialog Settings, záložka Power

V této záložce nastavte volby napájení.
Klikněte na okénko pro umožnění/zablokování "Power target circuit from PICkit 3" (napájení cílového obvodu z PICkit 3).

9.5.9 Dialog Settings, záložka Limitations

Tato záložka zobrazuje základní omezení pro vybrané zařízení. Pro informace o specifických omezeních klikněte na **Details**.

Kapitola 10. Specifikace hardwaru

10.1 Úvod
Zde naleznete detailní hardwarové a elektrické specifikace programátoru a debuggeru PICkit 3.

10.2 Obsah
Tato část obsahuje:
- Prohlášení o shodě
- USB port/napájení
- Debugger a programátor PICkit 3
- Standardní komunikační hardware
- Zvážení správné cílové desky

10.3 Prohlášení o shodě

Microchip Technology, Inc.
2355 W. Chandler Blvd.
Chandler, Arizona 85224-6199
USA

prohlašuje, že výrobek:

PICkit 3 In-Circuit Debugger/Programmer

splňuje následující standardy, dle nichž dodržuje daná omezení:

Standardy: EN61010-1 Laboratorní vybavení
Microchip Technology, Inc.
Datum: Leden 2009

<u>Důležité informace týkající se použití PICkit 3 In-circuit Debugger/Programmer</u>

Kvůli zvláštní povaze PICkit 3 In-Circuit Debugger/Programmer je uživatel tímto upozorněn, že přístroj může generovat vyšší než běžné úrovně elektromagnetického záření, které může narušovat práci všech druhů rádiového a jiného vybavení.

Aby byly splněny Evropské standardy, je třeba dodržet následující omezení:
1. Vývojový systém smí být použit pouze v průmyslovém (nebo podobném) prostředí.
2. Systém nesmí být používán do vzdálenosti do 20 metrů od jakéhokoliv vybavení, které může být ovlivněno jeho zářením (rádiové přijímače, televizory, atd.).

10.4 USB port/napájení

PICkit 3 je připojen n počítači pomocí mini Universal Serial Bus (USB) portu, verze kompatibilní s 2.0. USB konektor je umístěn na spodní straně.

Systém je schopen přes USB rozhraní načtení firmwaru.

Systém je přes USB rozhraní také napájen. Debugger je klasifikován podle USB specifikace jako high power system a vyžaduje trochu více než 100 mA, aby fungoval ve všech operačních módech (debugger/programátor).

Poznámka: PICkit 3 je napájen pomocí USB spojení. Cílová deska je napájena z vlastního zdroje. Také ji může napájen pouze PICkit 3, pouze však, pokud cíl spotřebovává méně než 30 mA.

Délka kabelu: Kabel pro spojení počítače a debuggeru o správné délce je přiložen k debuggeru.
Napájené rozbočovače: Pokud používáte USB rozbočovač, ujistěte se, že má vlastní napájení. Navíc USB porty na počítačových klávesnicích nemají dostatek energie pro napájení debuggeru.
Spánkový mód počítače: Zamezte spánkovému módu a jiným módům úspory energie na vašem počítači, abyste zajistili správnou USB komunikaci s debuggerem.

10.5 Debugger a programátor PICkit 3

Debugger se skládá z hlavní desky uzavřené v pouzdře s USB konektorem a jedním in-line konektorem. Na pouzdru debuggeru jsou indikační světla (LED).

10.5.1 Hlavní deska

Tato část má procesor s integrovaným USB 2.0 rozhraním schopným SPI sériového EE pro programování na on-board Flash emulační zařízení a LED indikátory.

10.5.2 Indikační světla (LED)

Indikační světla mají následující významy.

LED	Barva	Popis
Power	Zelená	Rozsvítí se při prvním zavedení napájení nebo připojení cíle.
Active	Modrá	Rozsvítí se, když PICkit™ 3 ustálí komunikaci s počítačem nebo odešle/přijde příkazy.
Status	Zelená	Rozsvítí se, když debugger pracuje normálně - standby.
	Oranžová	Rozsvítí se, když probíhá operace.
	Červená	Rozsvítí se, když debugování selže.

10.6 Standardní komunikační hardware

Pro standardní komunikaci debuggeru s cílem (**Sekce 2.4 "Komunikace debuggeru s cílem"**, "Standardní ICSP komunikace se zařízením"), použijte adaptér s RJ-11 konektorem.

Pro použití tohoto typu komunikace s header deskou budete možná potřebovat pro určené zařízení specifický Processor Pak, který obsahuje header desku s 8-pinovým konektorem obsahující požadované ICE/ICD zařízení a standardní adaptérovou desku.

Poznámka: Starší header desky používají 6-pinový (RJ-11) konektor namísto 8-pinového SIL konektoru, takže tyto headery mohou být připojeny přímo ke debuggeru.

Pro více informací o header deskách viz *"Processor Extension Pak and Header Specification"* (DS51292).

10.6.1 Standardní komunikace

Standardní komunikace je hlavním rozhraním cílového procesoru. Obsahuje spojení k vysokému napětí (VPP), VDD sense linky a spojení hodin a dat vyžadovaná pro programování a spojení s cílovými zařízeními.

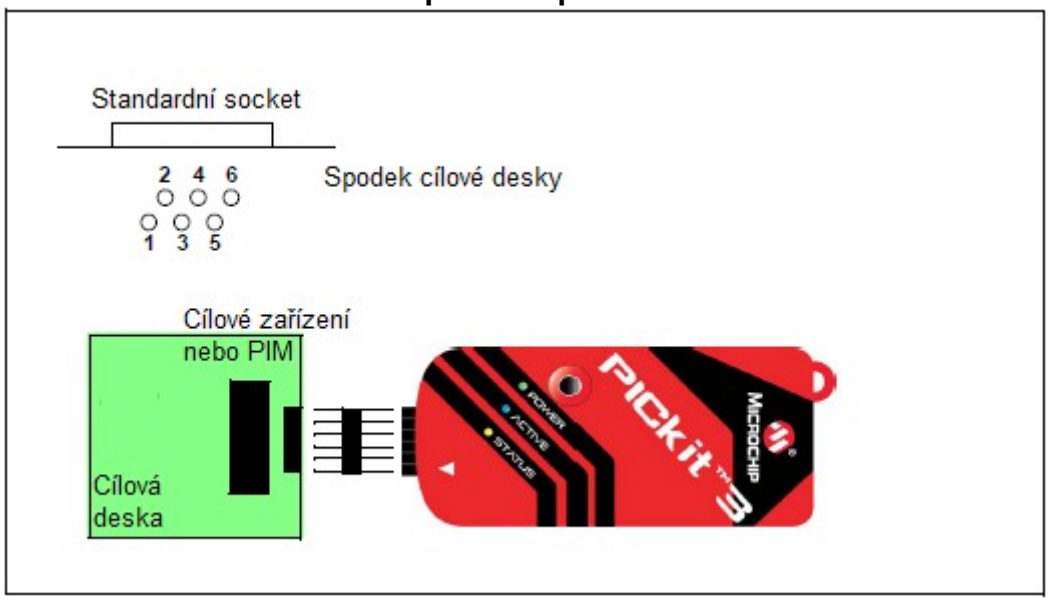
VPP vysokonapěťové linky mohou produkovat různé napětí, které se může pohybovat od 1.8 do 14 voltů, aby naplnilo požadavky na napětí pro specifický emulační procesor.

VDD sense spojení čerpá proud z procesoru cíle.

Hodinová a datová spojení jsou rozhraní s následujícími charakteristikami:

- Hodinové a datové signály jsou v módu High-Impedance (i když není zavedena žádná energie do systému PICkit 3).
- Hodinové a datové signály jsou chráněny před vysokým napětím způsobeným vadným cílovým systémem nebo nesprávným spojením.
- Hodinové a datové signály jsou chráněny před vysokým proudem způsobeným elektrickými zkraty v prototypu nebo cílových systémech.

Obrázek 10-1: Standardní rozložení pinů u 6-pin



Pin	Název	Funkce
1	MCLR/VPP	Napájení
2	VDD_TGT	Napájení cíle
3	GND	Uzemnění
4	PGD (ICSPDAT)	Standardní Com data
5	PGC (ICSPCLK)	Standardní Com hodiny
6	LVP	Programování slabým napětím

10.6.2 Modulární kabel a konektor

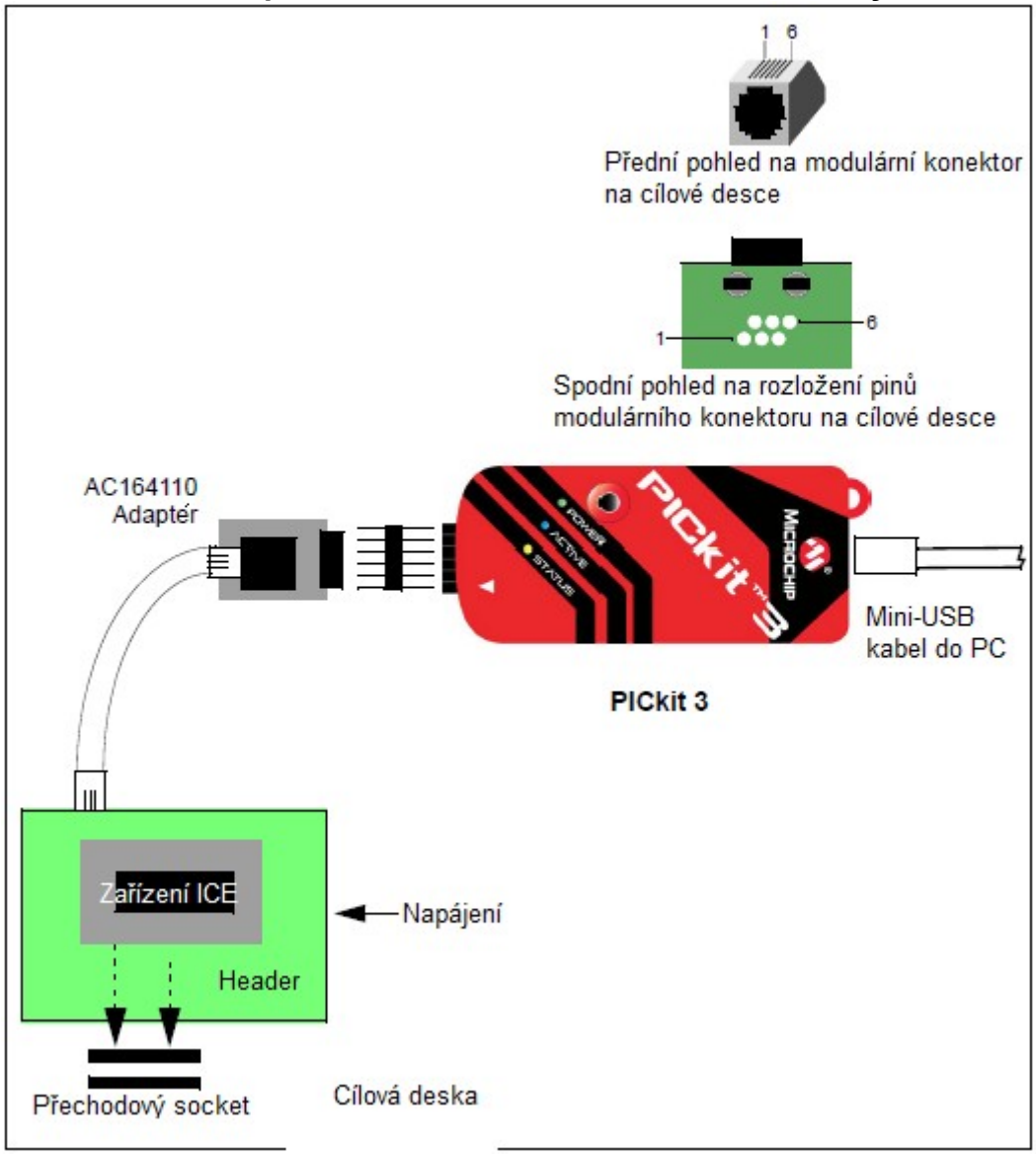
Pro standardní komunikaci propojuje modulární kabel debugger a cílovou aplikaci. Specifikace tohoto kabelu a jeho konektorů jsou vypsané níže.

10.6.2.1 Specifikace modulárního konektoru

- Výrobce, číslo dílu - AMP Incorporated, 555165-1
- Distributor, číslo dílu - Digi-Key, A9031ND

Tabulka pod obrázkem 10-2 ukazuje, jak piny modulárního konektoru odpovídají pinům mikrokontroléru. Tato konfigurace poskytuje plnou funkci ICD.

Obrázek 10-2: Rozložení pinů modulárního konektoru cílové desky



Pin modulárního konektoru	Pin mikrokontroléru
6	LVP
5	RB6
4	RB7
3	Uzemnění
2	VDD cíle
1	VPP

10.6.2.2 Specifikace modulární zástrčky

- Výrobce, číslo dílu - AMP Incorporated, 5-554710-3
- Distributor, číslo dílu - Digi-Key, A9117ND

10.6.2.3 Specifikace modulárního kabelu

- Výrobce, číslo dílu - Microchip Technology, 07-00024

10.7 Zvážení správné cílové desky

Cílová deska by měla být napájena podle požadavků vybraného zařízení (1.8V-5.0V) a jeho použití.

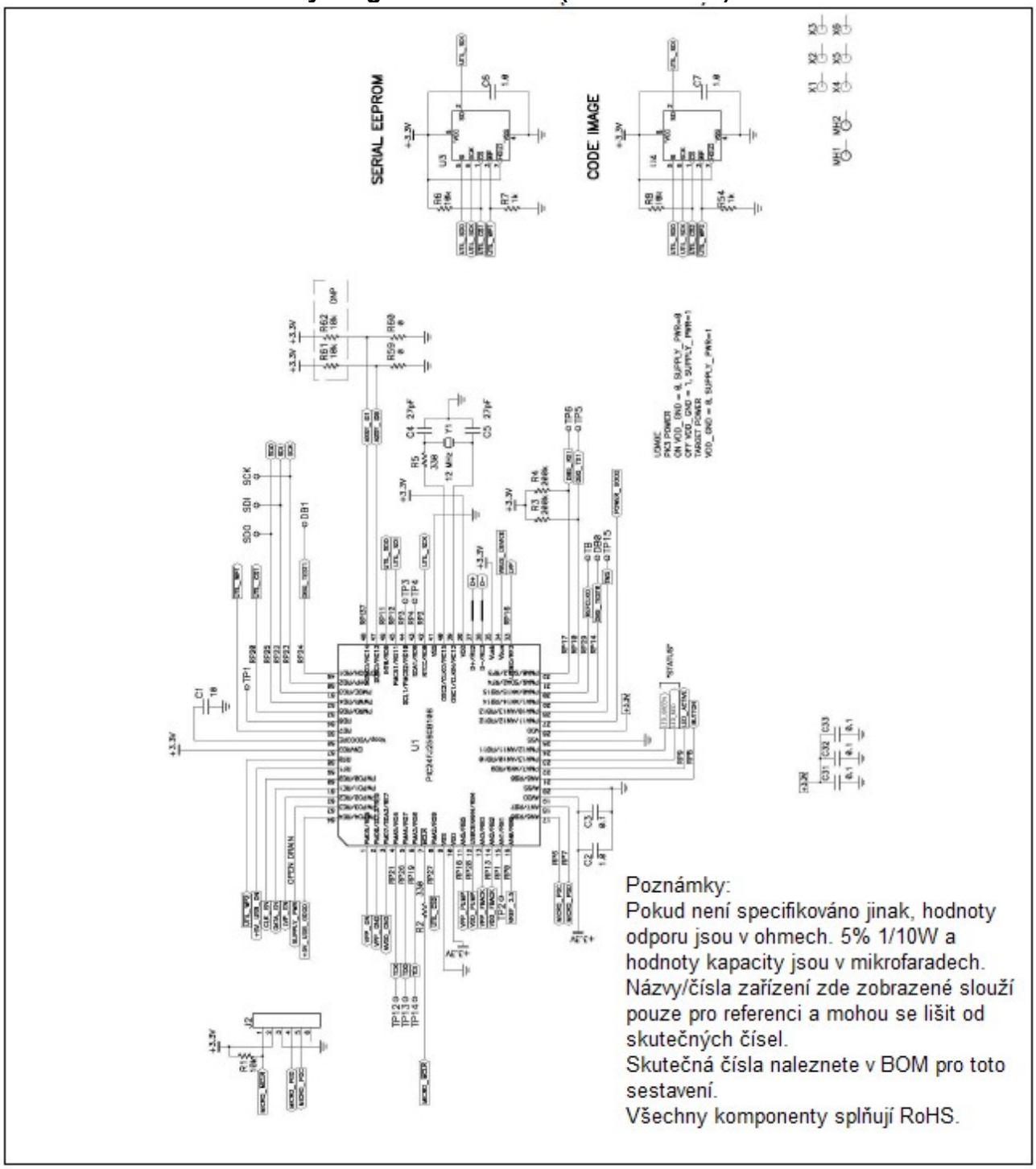
V závislosti na typu použité komunikace mezi debuggerem a cílem je třeba rozhodnout o několika věcech ohledně obvodů cílové desky:

- Sekce 2.4.2 "Obvod cílového spojení"
- Sekce 2.4.5 "Obvody, které zabrání debuggeru ve funkci"

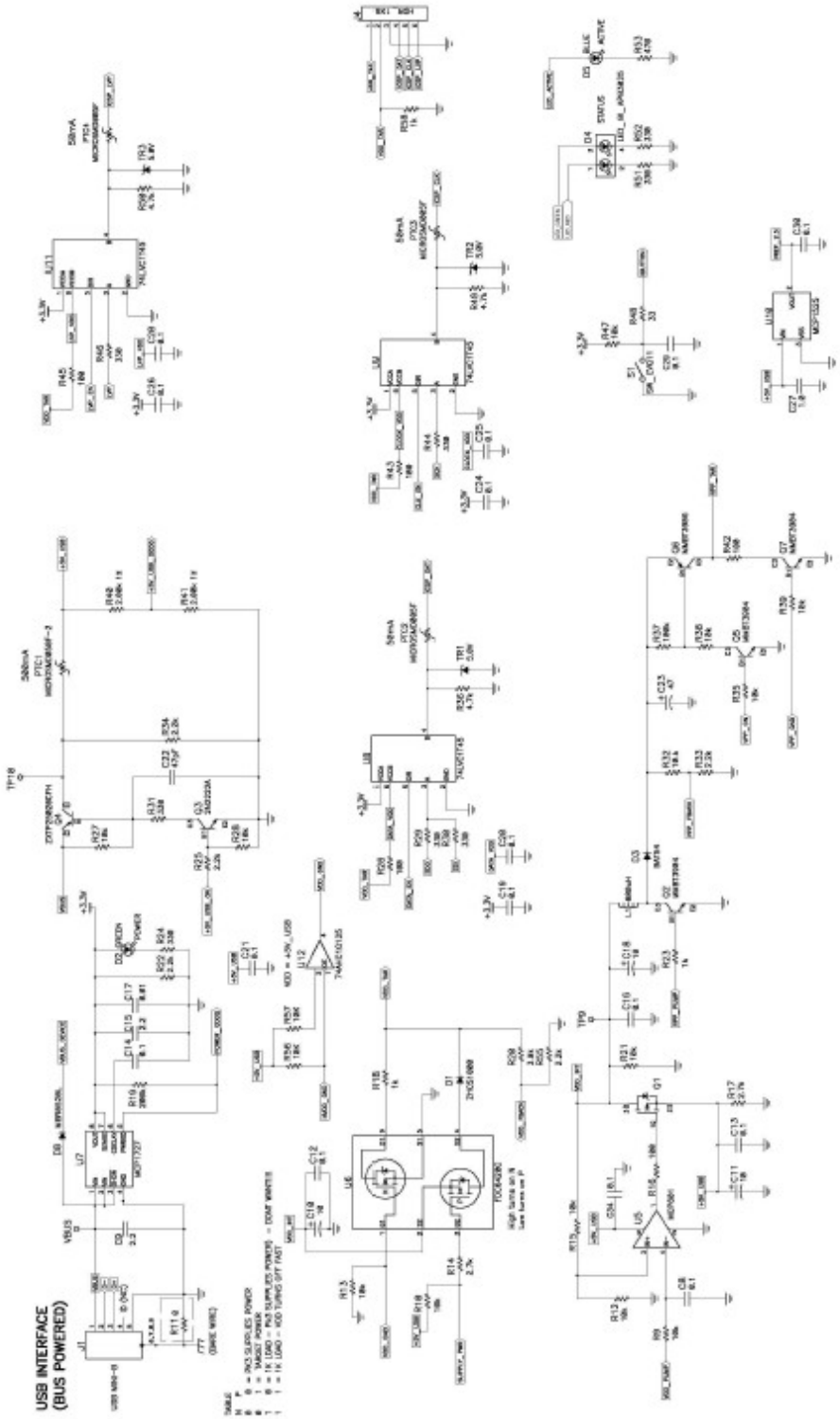
Dodatek A. Schémata PICkit 3

Zde se nachází schématické diagramy PICkit 3. Schémata Demo desek najdete v jejich manuálech.

Obrázek A-1: Schématický diagram PICkit™ 3 (stránka 1 z 2)



Obrázek A-2: Schématický diagram PICkit™ 3 (stránka 2 z 2)



Dodatek B. Doporučení k operaci

Při navrhování aplikací, které používají PICkit 3 byste měli zvážit následující:
- Nastavení obvodu oscilátoru
- Implementace a zvážení ICSP™
- Doporučená konfigurace
- Alternativní konfigurace
- Komunikační kanál
- Uzemnění a AC aplikace
- Oprava chybných signálů
- Režim spánku

Nastavení obvodu oscilátoru

Primární oscilátor
Často rozdíly mezi základními nastaveními MPLAB IDE a unikátními požadavky zařízení způsobí, že se zobrazí následujícízpráva v okně Output: "The target device is not ready for debugging. Please check your configuration bit setting and program the device before proceeding." (Cílové zařízení není připraveno na debugování. Zkontrolujte prosím vaše nastavení konfiguračních bitů a než budete pokračovat, naprogramujte zařízení.) Abyste toto opravili, nastavte konfigurační bity, aby odpovídaly nastavení oscilátoru cílové konfigurace.

Pro debugovací operace musí oscilátor aplikace (cíle) být aktivní předtím, než nastane in-circuit debugování. Zajistěte, že konfigurace oscilátoru a nastavení konfiguračního bitu MPLAB IDE jsou správné. Například, pokud vaše aplikace používá 20 MHz krystalový oscilátor, vyberte v MPLAB IDE nastavení HS (High Speed). Pro jakékoliv jiné aplikovatelné módy oscilátoru zařízení nahlédněte do datasheetu zařízení.

Krystalový oscilátor
Pokud MPLAB ICD header nebo rozšiřující processor packy použijete k připojení k cíli, může dojít k problémům s nastartováním krystalového rezonátoru. Abyste zabránili potenciálním problémům, zvažte následující:
1. Ujistěte se, že je krystal připojen poblíž základny zařízení.
2. Udržujte všechny linky v cílové aplikaci co nejkratší bez zbytečných nespojitostí jako DPS spojení nebo testovací body.
3. Minimalizujte jakékoliv kapacitivní zátěžování těchto uzlů.
4. Zabraňte použitísocketu pro umístění krystalu a kapacitoru. Připájejte zařízení přímo do DPS podložky.

Implementace a zvážení ICSP™

PICkit 3 používá sériové signalizační schéma pro programování a debugování cílového zařízení. Využité signály jsou hodinové a data. Ve většině zařízení jsou také mapována k portu (obvykle RB6 a RB7), ale jsou také definována v některých datasheetech jako PGC a PGD nebo ICSPCLK a ICSPDAT.

Navíc MCLR je také použito buď jako vysokonapěťový programovací signál nebo také jako

indikace upozornění pro zařízení.

Abyste zajistili bezproblémové in-circuit debugování, opatrně plánujte pomocí designeru, abyste zabránili jakýmkoliv problémům během vývoje aplikace nebo produkční fáze výrobku.

Doporučená konfigurace

Signály PGC a PGD jsou aktivní obousměrné signály řízené pomocí PICkit 3 a cílového emulačního zařízení. Pokud tyto signály lze udržet volné mimo jakékoliv další pasivní obvody nebo aktivní logiku v této aplikaci, zajistí bezproblémové debugování a programování. Také délka a/nebo typ kabelu mohou být při této konfiguraci zanedbatelné.

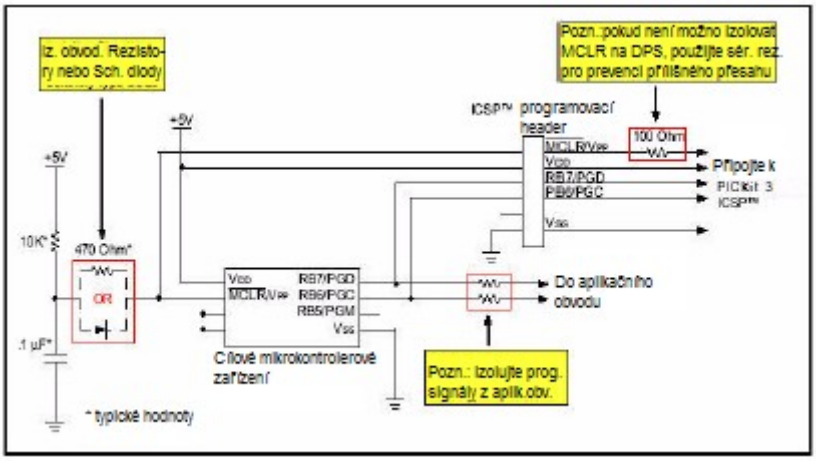
Navíc je MCLR signál použit PICkit 3 k poskytnutí napětí použitého pro programování některých zařízení nebo k upozornění na signál. V případech, kde má aplikace velký kapacitor, způsobí degradaci vzestupu a poklesu signálu. To zbrzdí schopnost PICkit 3 a zařízení, aby efektivně komunikovaly. Doporučuje se udržovat signál vytažený na VDD pomocí 10K rezistoru a uplatnit funkci power-on timer (spuštění časovačem), aby byla zajištěna správná sekvence spuštění napájení.

Alternativní konfigurace

V některých případech, zvláště u zařízení s nízkým počtem pinů můsí být piny přizpůsobeny aplikaci.

V tomto případě je minimální, resistivní izolace vyžadována mezi zařízení a aktivní uzel aplikace. Tím zajistíte, že jak aplikace, tak PICkit 3 jsou schopny ovládat PGC a PGD uzly na uzemněnní a správných VDD úrovních. Obrázek B-1 popisuje tuto konfiguraci.

Obrázek B-1: Typický ICSP™ aplikační obvod



Hodnota rezistivké izolace se bude lišit v závislosti na aplikaci a použití. Hodnoty jsou mezi 1K až 10K, jak je navrženo. V každém případě zajistě, že úrovně na PGD a PGC lze řídit k jejich odpovídajícím logickým úrovním napětí.

Komunikační kanál

Některá zařízení mají flexibilitu použití jednoho z několika komunikačních kanálů nebo pinů pro programování a debugování. Tyto kanály jsou běžně zmiňovány v datasheetech jako PGCx/PGDx, kde x je identifikátor čísla kanálu. Tyto kanály jsou často multiplexovány dalšími periferiemi (I²C™, SPI, A/D). Pokud vaše aplikace využívá tyto periferie a piny, které jsou běžné pro základní PGC/PGD piny, musíte vybrat jiný kanál a provést opatření pro vyžadovaná ICSP spojení.

Když používáte MPLAB IDE pro komunikaci s cílem a cíl používá toto nové přidělení kanálového pinu, musíte zajistit, že konfigurační bity v MPLAB IDE se shodují s kanálem spojení ve vašem cíli.

Uzemnění a AC aplikace

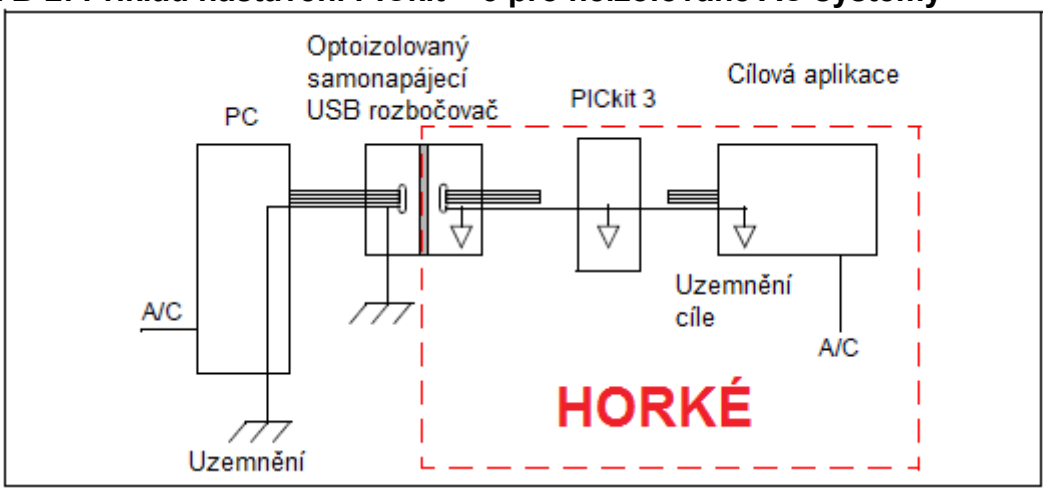
PICkit 3 se připojí k uzemnění z rozhraní USB konektoru pomocí PC.

Pro AC linkou napájené aplikace, které nejsou určené jako uzemnění toto představuje cestu, kde může rozdílné systémové napětí způsobit poškození obou systémů. V takových případech musí být PICkit 3 izolovaný. Opatrně zvažte systém uzemnění a spojení návratu signálu před připojením PICkit 3 k cíli.

Pro horké nebo uvolněné aplikace by měl být samonapájený USB rozbočovač použit mezi PC a PICkit 3, aby poskytl izolaci od uzemnění PC přes USB kabel (viz obrázek B-2). USB izolované rozbočovače, které jsou doporučené:
- High/Low-Speed 4-port USB HUB, Model UISOHUB 4, B&B Electronics Mfg. Co.
- High-Speed, 7-port USB HUB, Model HUB7P, SEALEVEL Systems, Inc.

Varování
Použití PICkit 3 bez řádné uzemňující izolace poškodí PICkit 3 nebo cílový systém, jelikož do nich bude zavedeno plné napětí z rozvodné sítě.
Tento stav je rizikový pro operátora kvůli elektrickému rázu. Proto pracujte co nejbezpečněji.

Obrázek B-2: Příklad nastavení PICkit™ 3 pro neizolované AC systémy

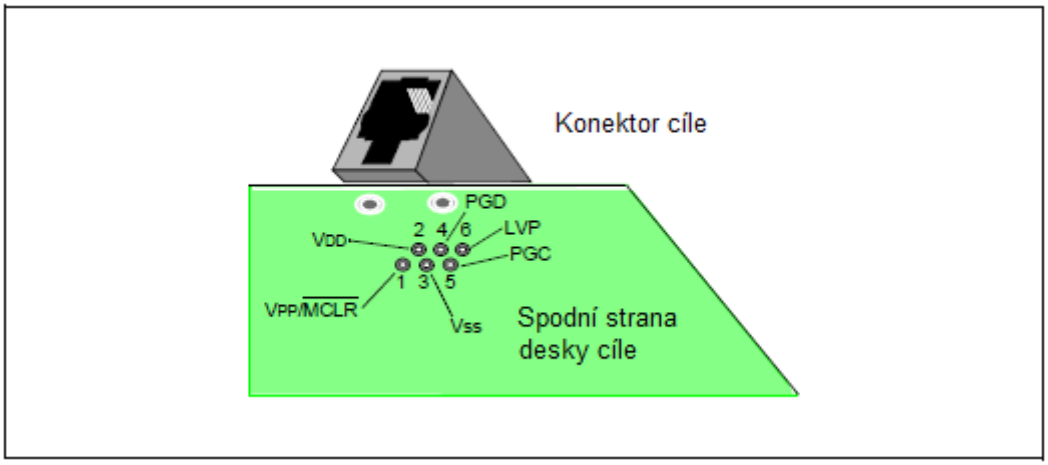


Varování
Použití optoizolovaného USB rozbočovače vytvoří horkou oblast označenou přerušovanou čárou na obrázku B-2.
Tento stav je rizikový pro operátora kvůli elektrickému rázu. Proto pracujte co nejbezpečněji.

Oprava chybných signálů
V některých případech může dojít k chybným signálům, když je zařízení programováno. Kvůli umístění PGC a PGD pinů může chyba degradovat signál a způsobit, že PICkit 3 nezvládne naprogramovat cílové zařízení. Pro nápravu zkuste následující:

Sestrojte modulární kabel RJ12. Udržujte jej co nejkratší, ideálně méně než 15 cm. Také sundejte z kabelu krytí, aby vodiče byly dál od sebe (zvláště signály PGC a PGD). Standardní modulární kabel vypadá jako na obrázku B-3. Pin 1 na jednom konci se připojuje k pinu 6 na druhém konci. To řeší problém ve skoro všech případech.

Obrázek B-3 Pinout cílového konektoru



Poznámka 1: Vybavení produkující šum (motory, stmívač, atd.) musí být na oddělených napájecích páscích od cílové aplikace a PICkit 3.

Režim spánku
Když používáte PICkit 3 po delší časové úseky, zvláště když je debugger v módu emulace, vypněte v operačním systému svého počítače režim spánku. Zajistíte tak, že zůstane zachována veškerá komunikace v USB subsystému.

Slovník

A

Absolute Section
Sekce GCC kompileru s pevnou (absolutní) adresou, kterou nelze měnit pomocí spojovacího programu.

Access Memory
Pouze PIC18 - speciální registry na zařízeních PIC18, které umožňují přístup bez ohledu na nastavení Bank Select Register (BSR).

Access Entry Points
Přístupové vstupní body poskytují způsob přenosu ovládání přes segment k funkci, která nemusí být v době spojení definovaná. Podporují oddělené spojování zaváděcích a zabezpečovacích aplikačních segmentů.

Address
Hodnota, která určuje umístění v paměti.

Alphabetic Character
Alfabetické znaky, písmena arabské abecedy (a,b, ..., z, A, B, ..., Z).

Alphanumeric
Alfanumerické znaky, alfabetické znaky a číslice (0, 1, ..., 9).

ANDed Breakpoints
Nastaví podmínku ANDed pro přerušení, tj, že bod přerušení 1 a současně bod přerušení 2 musí nastat ve stejném okamžiku, než se program zastaví. Toho lze dosáhnout pouze pokud bod přerušení dat a bod přerušení programové paměti nastanou ve stejném okamžiku.

Anonymous Structure
16-bitový C kompiler - nepojmenovaná struktura.
PIC18 C kompiler - nepojmenovaná struktura, která je součástí spojení C. Členů anonymní struktury lze dosáhnout, jako kdyby se jednalo o členy uzávřeného spojení. Například v následujícím kódu `hi` a `lo` jsou členy anonymní struktury uvnitř spojení `caster`.

```
union castaway  {
  int intval;
  struct  {
     char lo;  //accessible as caster.lo
     char hi;  //accessible as caster.hi
   };
}    caster;
```

ANSI
American National Standards Institute je organizace zodpovědná za formulaci a schvalování standardů ve Spojených Státech.

Application
Sada softwaru a hardwaru, kterou lze ovládat pomocí PIC® mikrokontroléru.

Archive
Archiv/knihovna je soubor přemístitelných modulů objektů. Je vytvořen sestavením více zdrojových souborů a poté za pomoci archiváře/knihovníka zkombinováním objektových dokumentů do jednoho souboru archivu/knihovny. Archiv/knihovnu lze spojit s objektovými moduly a jinými archivy/knihovnami pro vytvoření spustitelného kódu.

Archiver
Nástroj pro vytváření a práci s knihovnami.

ASCII
American Standard Code for Information Interchange je sada znaků využívajících 7 binárních čísel pro určení každého znaku. Zahrnuje znaky s horník a dolním indexem, číslice, symboly a kontrolní znaky.

Assembler
Jazykový nástroj, který překládá jazyk zdrojového kódu do strojového kódu.

Assembly Language
Programovací jazyk, který přepisuje binární strojový kód do symbolické formy.

Assigned Section
Sekce GCC kompileru, která byla přiřazena bloku cílové paměti ve spojovacím příkazovém souboru.

Asynchronously
Více událostí, které se udají ve stejnou dobu. To se většinou používá pro označení přerušení, která mohou nastat kdykoliv během spuštění procesoru.

Asynchronous Stimulus
Data generovaná k simulaci externích vstupů do simulovacího zařízení.

Attribute
GCC charakteristiky proměnných nebo funkcí v C programu, které se používají pro popis vlastností specifických pro daný přístroj.

Attribute, Section
GCC charakteristiky sekcí, jako "executable", "readonly" nebo "data", které mohou být specifikovány ve složce `.section` v assembler.

B

Binary
Základní dvojčíselný systém, který využívá číslice 0-1. Pravá krajní číslice počítá jedničky, druhý počítá násobky dvou, takže $2^2 = 4$, atd.

Bookmarks
Záložky slouží k snadnému nalezení určitých částí souboru.
Vyberte v liště nástrojů Editor položku Toggle Bookmarks pro přidání/odebrání záložek. Klikněte na další ikony na této liště pro pohyb k další nebo předchozí záložce.

Breakpoint
Hardware Breakpoint: událost, jejíž spuštění způsobí pozastavení.
Software Breakpoint: adresa, u níž se spuštění firmwaru pozastaví. Obvykle dosažena pomocí zvláštní instrukce pro přerušení.

Build
Zkompiluje a spojí všechny zdrojové soubory pro aplikaci.

C

C
C je základní programovací jazyk, který je úsporný ve vyjadřování, má moderní plynutí ovládání a struktur dat a bohatou sadu operátorů. C++ je verze C zaměřená na objekty.

Calibration Memory
Speciální registr funkcí nebo registrů používaný k podržení hodnot pro kalibrací PIC mikrokontrolérů on-board RC oscilátoru nebo jiných periferií.

Central Processing Unit
Část zařízení zodpovědná za získání správných instrukcí pro spuštění, dekódování těchto instrukcí a poté jejich spuštěním. Když je to třeba, funguje jako spojení a aritmetickou logickou jednotkou (ALU) pro dokončení spuštění instrukce. Ovládá adresový rozbočovač programové paměti, adresový rozbočovač datové paměti a vstupuje do zásobníku.

Clean
Odstraní všechny okamžité projektové soubory, jako objektové, kex a debug soubory, pro aktivní projekt. Tyto soubory jsou obnoveny z jicných souborů, když je projekt vytvořen.

COFF
Common Object File Format. Objektový soubor tohoto formátu obsahuje strojový kód, debugovací a jiné informace.

Command Line Interface
Způsob komunikace mezi programem a jeho uživatelem založený čistě na textovém vstupu a výstupu.

Compiler
Program, který překládá zdrojový soubor napsaný v jazyce vysoké úrovně do strojového kódu.

Conditional Assembly
Jazykový kód assembly, který je obsažen nebo vynechán v závislosti na hodnotě assembly-time nebo specifikovaného vyjádření.

Conditional Compilation
Úkon kompilace pouze fragmentu programu pro nastavení určitého konstantního vyjádření, specifikovaného preprocessorovou direktivou.

Configuration Bits
Bity se speciálním účelem, programované pro nastavení operačního módu PIC mikrokontroléru. Konfigurační bit může nebo nemusí být naprogramován.

Control Directives
Direktivy v kódovém jazyce assembly, díky nimž bude kód obsažen nebo vynechán v závislosti na hodnotě assembly-time nebo specifikovaném vyjádření.

CPU
Viz Central Processing Unit.

Cross Reference File
Soubor, který odkazuje na tabulku symbolů a seznam souborů, které odkazují na symbol. Pokud je symbol definován, první soubor v seznamu je umístěním definice. Zbývající soubory obsahují odkazy na soubor.

D

Data Directives
Data direktivy jsou ty, které ovládají umístění sestavení programu nebo datové paměti a poskytují způsob symbolického odkazování na datové položky, tj. pomocí srozumitelných názvů.

Data Memory
Na zařízeních Microchip MCU a DSC sestává datová paměť (RAM) z General Purpose Registerů (GPR) a Special Function Registrů (SFR). Některá zařízení mají také EEPROM datovou paměť.

Debugging Information
Compiler a assembler umožňují, že při výběru poskytují různý stupeň informací použitých pro debugování kódu aplikace. Pro detaily nahlédněte do dokumentace compileru nebo assembleru nebo vyberte volby debugování.

Deprecated Features
Funkce, které jsou stále podporovány z důvodů odkazu přestanou být po čase používány.

Device Programmer
Nástroj používaný pro programování elektricky programovatelných polovodičových zařízení jako jsou mikrokontroléry.

Digital Signal Controller
DSC je mikrokontrolérové zařízení s možností zpracování digitálního signálu, tj. MicrochipdsPIC DSC zařízení.

Digital Signal Processing
DSP je počítačová manipulace digitálních signálů, běžných analogových signálů (zvuk nebo obraz), které byly konvertovány z digitální formy (vzorkování).

Digital Signal Processor
Procesor digitálního signálu je mikroprocesor, který byl navržen pro použití ve zpracování digitálního signálu.

Directives
Povely ve zdrojovém kódu, které poskytují kontrolu operace jazykového nástroje.

Download
Stahování je proces zasílání dat z hostitelského zařízení do jiného, například emulátoru, programátoru nebo cílové desky.

DSC
Viz Digital Signal Controller.

DSP
Viz Digital Signal Processor

DWARF
Debug With Arbitrary Record Format. DWARF je formát debugovacích informací pro ELF soubory.

E

EEPROM
Electrically Erasable Programmable Read Only Memory. Speciální typ PROM, který lze elektricky vymazat. Data jsou psána nebo mazána po jednou bytu. EEPROM si zachováváobsah i vypnutá.

ELF
Executable and Linking Format. Objektový soubor tohoto formátu obsahuje strojový kód. Debugovací a jiné informace jsou specifikovány spolu s DWARF. ELF/DWARF poskytují lepší debugování optimalizovaného kódu než COFF.

Emulation
Proces spuštění doftwaru načteného do emulační paměti, jako by se jednalo o firmware umístěný na mikrokontrolérovém zařízení.

Emulation Memory
Programová paměť obsažená v debuggeru.

Debugger
Hardware, který provádí emulaci.

Debugger System
Systém debuggeru MPLAB ICE 2000 a MPLAB ICE 4000 sestává z pouzdra, modulu procesoru, adaptéru zařízení, cílové desky, kabelů a softwaru MPLAB IDE. Systém PICkit 3 sestává z pouzdra, ovladačové (a potenciálně přijímačové) karty, cílové desky, kabelů a softwaru MPLAB IDE.

Endianness
Řazení bytů do multi-bytových objektů.

Environment - IDE
Určité rozložení pracovní plochy pro vývoj aplikace.

Environment - MPLAB PM3
Složka obsahující soubory o tom, jak programovat zařízení. Tato složka může být přenesena na SD/MMC kartu.

Epilogue
Část kompilerem vygenerovaného kódu, zodpovědná za přerozmístění místa v zásobníku (stack), obnovení registrů a vykonání jakýchkoliv jiných požadavků pro určité zařízení, specifikovaných v runtime modelu. Tento kód se spouští po každém uživatelském kódu pro danou funkci, ihned přednostně k návratu funkce.

EPROM
Erasable Programmable Read Only Memory. Programovatelsná paměť pouze pro čtení, kterou lze obvykle vymazat vystavením ultrafialovému záření.

Error File
Chybový soubor obsahuje chybové hlášky a diagnostiky generované pomocí jazykového nástroje.

Errors
Chyba oznámí problém, kvůli kterému je nemožné pokračovat ve zpracování vašeho programu. Když je to možné, chyba identifikuje název zdrojového souboru a číslo řádku, kde je zjevně problém.

Event
Popis cyklu rozbočovače, který může obsahovat adresu, data, počet průchodů, externí vstup, typ cyklu (fetch, R/W) a časovou známku. Události (event) se používají pro popis spouštěčů (trigger), bodů přerušení (breakpoint) a přrušení (interrupt).

Executable Code
Software, který je připraven pro načtení ke spuštění.

Export
Vyšle data z MPLAB IDE/MPLAB X IDE ve standardizovaném formátu.

Expressions
Kombinace konstant a/nebo symbolů oddělených aritmetickým nebo logickým operátorem.

Extended Microcontroller Mode
V tomto módu je k dispozici jak on-chip programová paměť, tak externí paměť. Spuštění se automaticky přepne na externí, pokud je adresa programové paměti větší než prostor vnitřní paměti zařízení PIC18.

Extended Mode
V módu Extended kompiler využívá rozšířené instrukce (tj. `ADDFSR`, `ADDULNK`, `CALLW`, `MOVSF`, `MOVSS`, `PUSHL`, `SUBFSR` a `SUBULNK`) a indexuje pomocí adresování úplnou kompenzací.

External Label
Značka, která má externí spojení.

External Linkage
Funkce proměnné má externí spojení, pokud na něj lze odkázat zvenčí modulu, v němž je definována.

External Symbol
Symbol pro identifikátor, který má externí spojení. To může být reference nebo definice.

External Symbol Resolution
Proces učiněný spojovacím programem, v němž definice externích symbolů ze všech vstupních modulů jsou shromážděny při pokusu o rozhodnutí o všech externích symbolech referencí. Jakékoliv reference externích souborů, které nemají odpovídající definici způsobí nahlášení chyby spojovacího programu.

External Input Line
Externí vstupní signální logická linka (TRIGIN) pro nastavení události založené na externích signálech.

External RAM
Off-chip čtení/psací paměť.

F

Fatal Error
Chyba, která úplně zastaví kompilaci. Žádné další zprávy se neobjeví.

File Registers
On-chip datová paměť, obsahuje General Purpose Registery (GPR) a Special Function Registery (SFR).

Filter
Rozhodne výběrem, která data budou obsažena/neobsažena v zobrazení umístění nebo

Flash
Typ EEPROM, na níž jsou data zapisována nebo mazána v blocích namísto v bytech.

FNOP
Forced No Operation. Nucený NOP cyklus je druhým cyklem dvojcyklové instrukce. Jelikož je architektura PIC mikrokontroléru zřetězená, vezme předem další instrukci v prostoru fyzické adresy, zatímco provádí aktuální instrukci. Nicméně, pokud aktuální instrukce změní počítadlo programu, tato předem vzatá instrukce je explicitně ignorována a zapříčiní FNOP cyklus.

Frame Pointer
Ukazatel, který odkazuje na umístění v zásobníku, které odděluje argumenty založené na zásobníku od místních proměnných založených na zásobníku. Poskytuje užitečný základ, z něhož lze přistoupit k místním proměnným a jiným hodnotám pro aktuální funkci.

Free-Standing
Realizace, která přijímá jakýkoliv striktně souhlasný program, který nepoužívá komplexní typy a ve kterém použití funkcí specifikovaných v hlavní větě knihovny (ANSI '89 standard clause 7) je omezeno na obsah standardních záhlaví. `<float.h>`, `<iso646.h>`, `<limits.h>`, `<stdarg.h>`, `<stdbool.h>`, `<stdef.h>` a `<stdint.h>`.

G

GPR
General Purpose Register. Část datové paměti zařízení (RAM) pro obecné použití.

H

Halt
Přerušení spuštěného programu. Spuštění Halt je stejné jako zastavení na bodu přerušení.

Heap
Oblast paměti použitá pro umístění dynamické paměti, kde paměťové bloky jsou umístěny v pořadí rozhodnutém v době běhu programu (runtime).

Hex Code
Hex code je spustitelná instrukce uložená ve formátu hexadecimálního kódu. Hex code je obsažen v hex file.

Hex File
ASCII soubor obsahující hexadecimální adresy a hodnoty (hex kód) vhodné pro programování zařízení.

Hexadecimal
Základní 16-číselný systém, který využívá číslice 0-9 a písmena A-F (nebo a-f). A-F reprezentují hexadecimální číslice s hodnotami (decimálními) 10 až 15. První číslice se počítá jako jedna, další je násobkem 16, takže $16^2 = 256$, atd.

High Level Language
Jazyk pro psaní programů, který je dále odstraněn z procesoru.

I

ICD
In-Circuit Debugger. MPLAB ICD 3 je in-circuit debugger od Microchip

ICE
In-Circuit Emulator: MPLAB ICE 2000 a MPLAB ICE 4000 jsou in-circuit emulátory od Microchip.

ICSP
In-Circuit Serial Programming. Metoda programování zabudovaná v zařízeních Microchip, využívající sériové komunikace a minimálního čídla pinů zařízení.

IDE
Integrated Development Environment, jako v MPLAB IDE/MPLAB x IDE.

Identifier
Název funkce nebo proměnné.

IEEE
Institute of Electrical and Electronics Engineers.

Import
Přenesení dat do MPLAB IDE/MPLAB X IDE z vnějšího zdroje, jako třeba z hex file.

Initialized Data
Data, která jsou definována s počáteční hodnotou. V C,
`int myVar=5;`
definovaná proměnná, která je umístěna v inicializované sekci dat.

Instruction Set
Soubor instrukcí strojového kódu, kterým rozumí určitý procesor.

Instructions
Sekvence bitů, která sdělí centrální procesní jednotce (CPU), aby vykonala určitou operaci. Může obsahovat data, která mají být v operaci využita.

Internal Linkage
Funkce nebo proměnná má vnitřní spojení, pokud do ní nelze vstoupit zvenčí modulu, v němž je definována.

International Organization for Standardization
Organizace, která nastavuje standardy v mnoha technologiích a odvětvích, včetně výpočetní techniky a komunikací. Známá také jako ISO.

Interrupt
Signál do CPU, který odloží spuštění běžící aplikace a přesune ovládání na Interrupt Service Routine (ISR), aby mohla být událost zpracována. Po dokončení ISR pokračuje normální běh aplikace.

Interrupt Handler
Proces, který zpracovává speciální kód, kdy dojde k interrupt.

Interrupt Service Request (IRQ)
Událost, která způsobí, aby procesor dočasně odložil normální instrukce a spustil proces interrupt handler. Některé procesory mají takovýchto událostí, které více dovolují přerušení o různém stupni důležitosti.

Interrupt Service Routine (ISR)
Jazykové nástroje (language tools) - funkce, která se stará o přerušení (interrupt)
MPLAB IDE/MPLAB X IDE - uživatelem generovaný kód, který je vložen, když nastane interrupt. Umístění kódu v programové paměti bude obvykle záviset na typu přerušení (interrupt), který nastal.

Interrupt Vector
Adresa interrupt service routine nebo interrupt handler.

L

L-Value
Vyjádření, které odkazuje na objekt, který lze prověřit a/nebo modifikovat. L-value se používá na levé straně zadání.

Latency
Čas mezi událostí a její odezvou.

Library/Librarian
Viz Archive/Archiver.

Linker
Jazykový nástroj, který kombinuje objektové soubory k vytvoření spustitelného kódu, rozděluje reference z jednoho modulu do druhého.

Linker Script Files
Jedná se o příkazové soubory linkeru. Určují možnosti linkeru a popisují paměť, která je k dispozici na cílové platformě.

Listing Directives
Jedná se o direktivy, které ovládají formát souboru assembler listing. Dovolují specifikaci názvů, stránkování a další ovládání záznamu.

Listing File
Jedná se o textový ASCII soubor, který ukazuje strojový kód generovaný pro každý C zdroj, instrukce k sestavení, direktivy sestavení nebo makra, nacházející se ve zdrojovém souboru.

Little Endian
Schéma uspořádání dat pro multibyte data, kde poslední byte je uložen na nižší adrese.

Local Label
Jedná se o označení definované v makru s pokynem LOCAL. Tato označení jsou specifická pro daný příklad konkretizace makra. Jinými slovy, symboly a označení, která jsou prohlášena za místní nejsou již přístupná poté, co se objeví ENDM makro.

Logic Probes
K některým emulátorům Microchip lze připojit až 14 logických sond. Logické sondy poskytují externí sledovací vstupy, spouštěče výstupního signálu, +5V a uzemnění.

Loop-Back Test Board
Používá se pro test fungování MPLAB REAL ICE in-circuit emulátoru.

LVDS
Low Voltage Differential Signaling. Metoda o nízkém šumu, nízké energii a nízké amplitudě pro vysokorychlostní (gigabity za sekundu) přesun dat po měděném kabelu.
Se standardním signálem I/0 je úložiště dat podmíněné skutečnou úrovní napětí. Úroveň napětí lze ovlivnit délkou kabelu (delší kabely zvyšují odpor, který snižuje napětí). S LVDS je úložiště dat odlišeno pouze pozitivní a negativní hodnotou napětí, ne úrovní napětí. Proto mohou data putovat na větší vzdálenosti po kabelu, zatímco si udržují čistý a konzistentní datový tok.
Zdroj: http://www.webopedia.com/TERM/L/LVDS.html.

M

Machine Code
Reprezentace počítačového programu, který je skutečně čten a interpretován procesorem. Program v binárním strojovém kódu sestává ze sekvence strojových instrukcí (může být promísen s daty). Soubor všech možných instrukcí pro určitý procesor je znám jako "instruction set".

Machine Language
Sada instrukcí pro specifickou centrální procesní jednotku (CPU), navržená, aby byla použitelná procesorem bez nutnosti překladu.

Macro
Instrukce makra. Instrukce, které reprezentují sekvenci instrukcí ve zkrácené formě.

Macro Directives
Direktivy, které ovládají spuštění a umístění dat uvnitř definic těla makra.

Makefile
Exportuje do souboru instrukce pro tvorbu projektu. Použijte tento soubor pro tvorbu vašeho projektu mimo MPLAB IDE/MPLAB X IDE, např. s `make`.
Pod *Project>Build Options>Project*, záložka **Directories** musíte vybrat "Assemble/Compile/Link in the project directory" pod "Build Directory Policy", aby tato funkce fungovala.

Make Project
Příkaz, který přebuduje aplikaci, provede rekompilaci pouze těch zdrojových souborů, které se změnily od poslední kompletní kompilace.

MCU
Microcontroller Unit. Zkratka pro mikrokontrolér. Také uC.

Memory Models
Ukázka paměti k dispozici pro aplikaci.

Memory Model
Popis, který specifikuje velikost ukazatelů, které ukazjí programovou paměť.

Message
Text zobrazený, aby vás upozornil na potenciální problémy v jazyce nástroje. Zpráva nepřeruší operaci.

Microcontroller
Vysoce integrovaný čip, který obsahuje CPU, RAM, programovou paměť, I/O porty a časovače.

Microcontroller Mode
Jedna z možných konfigurací programové paměti PIC18 mikrokontrolérů. V módu Microcontroller je dovoleno pouze vnitřní spuštění. Proto jsou v tomto módu možné pouze on-chip programové paměti.

Microprocessor Mode
Jedna z možných konfigurací programovací paměti PIC18 mikrokontrolérů. V módu Microprocessor není použita programová paměť on-chip. Celá programová paměť je mapována externě.

Mnemonics
Textové instrukce, které lze přeložit přímo do strojového kódu. Také bývají označovány jako opcode.

MPASM™ Assembler
Přemístitelný macro assembler od Microchip Technology pro zařízení s PIC mikrokontroléry, zařízení KeeLoq® a paměťová zařízení Microchip.

MPLAB ASM30
Přemístitelný makro assembler Microchip pro dsPIC ovladač digitálního signálu a PIC24 PIC zařízení.

MPLAB C kompilery
Odkazuje na MPLAB C kompilery od Microchip:
- MPLAB C kompiler pro PIC18 MCU zařízení (dříve MPLAB C18)
- MPLAB C kompiler pro sdPIC Digital Signal Controller a PIC24 PIC zařízení (dříve MPLAB C30)
- MPLAB C kompiler pro PIC32MX MCU (dříve MPLAB C32)

MPLAB ICD 2
In-circuit debugger Microchip, který pracuje s MPLAB IDE. ICD podporuje Flash zařízení s vestavěným debugovacím obvodem. Hlavním dílem každého ICD je pouzdro. Kompletní systém sestává z pouzdra, ovladací (a potenciálně přijímací) karty, kabelů a MPLAB IDE softwaru.

MPLAB ICE 2000
In-circuit debugger Microchip, který pracuje s MPLAB IDE. MPLAB ICE 2000 podporuje 8-bitová PIC MCU. Hlavním dílem každého ICE je pouzdro. Kompletní systém sestává z pouzdra, procesorového modulu, kabelů a MPLAB IDE softwaru.

MPLAB ICE 4000
Nedoporučuje se pro nové designy. Viz MPLAB ICD 3 in-circuit debugger.
In-circuit debugger Microchip, který pracuje s MPLAB IDE. MPLAB ICE 4000 podporuje PIC18F a PIC24 MCU a dsPIC DSC. Hlavním dílem každého ICE je pouzdro. Kompletní systém sestává z pouzdra, procesorového modulu, kabelů a MPLAB IDE softwaru.

MPLAB IDE
Integrované vývojářské prostředí Microchip.

MPLAB LIB30
Knihovník pro použití s objektovými moduly vytvořenými buď pomocí MPLAB ASM30 nebo MPLAB C30 C kompileru.

MPLAB LINK30
MPLAB LINK 30 je objektový linker pro Microchip MPLAB ASM30 assembler a Microchip MPLAB C30 C kompiler.

MPLAB PM3
Programátor zařízení od Microchip. Programuje PIC18 mikrokontroléry a dsPIC ovladače digitálního signálu. Lze použít s MPLAB IDE/MPLAB X IDE nebo samostatně. Náhrada PRO MATE II.

MPLAB REAL ICE™ In-circuit Emulator
Příští generace in-circuit emulátorů Microchip, které pracují s MPLAB IDE/MPLAB X IDE. *Viz* ICE/ICD.

MPLAB SIM
Simulátor Microchip, který pracuje s MPLAB IDE/MPLAB X IDE s podporou PIC MCU a dsPIC DSC zařízení.

MPLIB™ Object Librarian
Knihovník Microchip, který pracuje s MPLAB IDE/MPLAB X IDE. MPLIB knihovník je objektocý knihovník pro použití s COFF objektovými moduly vytvořenými buď pomocí MPASM assembler (mpasm nebo mpasmwin v2.0) nebo MPLAB C18 C kompileru.

MPLINK™ Object Linker
MPLINK linker je objektový linker pro Microchip MPASM assembler a Microchip C18 C kompiler. MPLINK linker lze také použít s Microchip MPLIB librarian. MPLINK linker je navržen pro použití s MPLAB IDE/MPLAB X IDE, ačkoliv to není nutné.

MRU
Most Recently Used (nedávno použité). Používá se pro soubory a okna, která lze vybrat z hlavních roletových menu MPLAB IDE/MPLAB X IDE.

N

Native Data Size
Pro Native trace, velikost proměnné použité v okně Watch musí být stejné velikosti jako vybraná datová paměť zařízení: byt pro zařízení PIC18 a slova pro 16-bitová zařízení.

Nesting Depth
Maximální úroveň, do jaké mohou makra obsahovat jiná makra.

Node
Projektová komponenta MPLAB IDE/MPLAB X IDE.

Non-Extended Mode
V módu Non-Extended kompiler nevyužije rozšířené instrukce ani ty indexované doslovným adresováním.

Non Real Time
Odkazuje na procesor v bodě přerušení nebo single-step exekuční instrukce nebo MPLAB IDE/MPLAB X IDE spuštěné v módu Simulator.

Non-Volatile Storage
Úložné zařízení, jehož obsah je uchován, když je vypnuto napájení.

NOP
No Operation. Instrukce, které nemají žádný efekt, když jsou spuštěny, vyjma postupu na programovacím počítadle.

O

Object Code/Object File
Object code je strojový kód generovaný assemblerem nebo kompilerem. Objektový soubor je soubor obsahující strojový kód a možné informace o debugování. Může být ihned spustitelný nebo může být přemístitelný, vyžadující propojení s jinými objektovými soubory, např. knihovnami pro vytvoření kompletního spustitelného programu.

Object File Directives
Direktivy, které se používají pouze při vytváření objektového souboru.

Octal
Základní 8-číselný systém, který používá pouze znaky 0-7. První znak se počítá jako jedna, další se počítají jako osminásobky, takže $8^2 = 64$, atd.

Off-Chip Memory
Off-chip memory odkazuje na možnost výběru paměti pro zařízení PIC18, kde paměť může být umístěna na cílové desce, nebo kde veškerá programová paměť může být poskytnuta emulátorem. Záložka **Memory** je přístupná z *Options>Development Mode* a zobrazí dialogové okno výběru Off-Chip Memory.

One-to-One Project Workspace Model
Nejběžnější konfigurace pro vývoj aplikací v MPLAB IDE je mít one-to-one projekt v jednom pracovním místě. Vyberte *Configure>Settings*, záložku **Projects** a označte "Use one-to-one project-workspace model".

Opcodes
Operational Codes (operační kódy). *Viz* Mnemonics.

Operators
Symboly jako plus "+" a mínus "-", které se používají při tvorbě správně definovaných vyjádření. Každý operátor má přidělou prioritu, která se používá k rozhodnutí o pořadí zhodnocení.

OTP
One Time Programmable. EPROM zařízení, která nemají kryt s okénkem. Jelikož EPROM potřebují ultrafialové světlo pro vymazání paměti, mazat lze pouze zařízení s okénkem.

P

Pass Counter
Počítadlo, které se sníží pokaždé, když dojde k události (například spuštění instrukce na určité adrese). Když hodnota počítadla průchodu dosáhne nuly, událost je splněna. Můžete přidělit počítadlu průchodu, aby provedlo break a trace logic, a pro jakékoliv další události v komplexním dialogu spouštěče.

PC
Personal computer (osobní počítač) nebo Program Counter (programové počítadlo).

PC Host
Jakékoliv PC se spuštěným podporovaným operačním systémem Windows.

Persistent Data
Data, která nejsou nikdy vyčištěna nebo inicializována. Jejich určené použití je, aby aplikace mohla uchovat data přes Reset zařízení.

Phantom Byte
Neimplementovaný byt v architektuře dsPIC, který je použit, když je 24-bitové instrukční slovo bráno jako kdyby se jednalo o 32-bitové instrukční slovo. Fantomový byt se objevuje v dsPIC hex souboru (hex file).

PIC MCU
PIC mikrokontrolery (MCU) odkazují na všechny skupiny mikrokontrolerů Microchip.

PICSTART Plus
Vývojové programátory zařízení Microchip. Programují 8-, 14-, 28- a 40-pinové PIC mikrokontrolery. Musí být použity spolu s MPLAB IDE softwarem.

Plug-ins
MPLAB IDE/MPLAB X IDE mají oba zabudované komponenty a plug-in moduly pro konfiguraci systému pro rozpětí softwarových a hardwarových nástrojů. Několika plug-in nástrojů lze najít v menu Tools.

Pod
Systém MPLAB REAL ICE: krabička obsahující emulační obvody systému pro množství softwarových a hardwarových nástrojů. V menu Tools lze nají několik nástrojů.
MPLAB ICD 2/3: krabička obsahujícídebugovací obvody pro ICD zařízení na header nebo cílové desce. ICD zařízení může být produkční zařízení s vestavěným ICE obvodem nebo speciální verze ICE produkčního zařízení.
MPLAB ICE 2000/4000: rozšířená krabička debuggeru, která obsahuje emulační paměť, trace paměť, časovače událostí a cyklu a logiku trace/breakpoint.

Power-on-Reset Emulation
Proces softwarové tvorby náhodných hodnot, který tyto hodnoty zapisuje na oblasti RAM pro simulaci neinicializovaných hodnot na RAM při počáteční aplikaci energie.

Pragma
Direktiva, která má význam pro specifický kompiler. Často se pragma používá pro dopravení implementačně definované informace do kompileru. MPLAB C30 využívá atributy pro přesun této informace.

Precedence
Pravidla, která definují pořadí zhodnocení.

PRO MATE II
Již není vyráběn. Viz programátor MPLAB PM3
Programátor od Microchip. Programuje většinu PIC mikrokontrolerů stejně jako většinu paměťových a KeeLoq zařízení. Lze použít s MPLAB IDE nebo samostatně.

Profile
Seznam spouštěcích podnětů pro MPLAB SIM simulátor.

Program Counter
Umístění, které obsahuje adresu instrukce, která je právě spuštěná.

Program Counter Unit
16-bitový assembler. Konceptuální reprezentace rozvržení programové paměti. Programové počítadlo se zvedne o 2 za každé instrukční slovo. V spustitelné sekci jsou 2 jednotky počítadla ekvivalentní 3 bytům. V sekci pouze pro čtení jsou 2 jednotky počítadla ekvivalentní 2 bytům.

Program Memory
MPLAB IDE/MPLAB X IDE - Paměťová oblast v zařízení, kde jsou uloženy instrukce. Také

paměť v emulátoru nebo simulátoru obsahující stažený firmware cílové aplikace.
16-bitový assembler/kompiler - oblast paměti v zařízení, kde jsou uloženy instrukce.

Project
Projekt obsahuje soubory nutné pro sestavení aplikace (zdrojový kód, soubory skriptu linkeru, atd.) spolu s dalšími připojenými údaji pro různé nástroje pro tvorbu a možnosti tvorby.

Prologue
Část kompilerem generovaného kódu, která je zodpovědná za rozmístěbní prostoru zásobníku, uchování registrů a vykonání jakéhokoliv dalšího požadavku specifického pro daný přístroj v modelu runtime. Tento kód se spouští před jakýmkoliv uživatelským kódem pro danou funkci.

Prototype System
Pojem odkazující na cílovou aplikaci uživatele nebo cílovou desku.

PWM Signals
Pulse Width Modulation Signals. Jistá PIC MCU zařízení majíu PWM periferii.

Q

Qualifier
Adresa nebo rozsah adres použitý počítadlem průchodu (Pass Counter) nebo jako událost před další operací v komplexním spouštěči.

R

Radix
Číslo base, hex nebo decimal použité ve specifikaci adresy.

RAM
Random Access Memory (Data Memory). Paměť, v níž lze k datům přistupovat v jakémkoliv pořadí.

Raw Data
Binární reprezentace kódu nebo dat spojených se sekcí.

Read Only Memory
Paměťový hardware, který umožňuje rychlý přístup k permanentně uloženým datům, ale zabraňuje přidání nebo modifikaci dat.

Real Time
Když je in-circuit emulator nebo debugger uvolněn ze stavu halt, procesor běží v módu Real Time a chová se přesně, jak by se měl normální čip chovat. V módu Real Time je umožněn real time trace buffer emulátoru a neustále zachycuje všechny vybrané cykly, a dále jsou umožněny všechny break logic. V in-circuit emulátoru nebo debuggeru se procesor spustí v reálném čase, dokud platný bod přerušení nezpůsobí halt nebo dokud proces nepřeruší uživatel.
V simulátoru real time znamená jednoduše spuštění instrukcí mikrokontroléru tak rychle, jak je lze simulovat pomocí host CPU.

Real-Time Watch
Okno Watch, kde se mění proměnné v reálném čase při běhu aplikace. Viz individuální dokumentaci nástrojů pro rozhodnutí, jak nastavit sledování v reálném čase. Ne všechny nástroje toto podporují.

Recursive Calls
Funkce, která volá sama sobě, buď přímo nebo nepřímo.

Recursion
Koncept, že funkce makra, když byla definována, může zavolat sama sobě. Psaní rekurzivních maker byste měli věnovat velkou péči. Je snadné chytit se do nekonečné smyčky, pokud neexistuje výstup.

Reentrant
Funkce, která může mít více simultánně aktivních případu najednou. Může se to stát buď kvůli přímé nebo nepřímé rekurzi nebo pomocí spuštění během procesu přerušení (interrupt).

Relaxation
Proces konverze instrukce na identickou, ale menší instrukci. To je užitečné pro úsporu velikosti kódu. MPLAB ASM30 nyní umí dán instrukce RELAX a CALL do instrukce RCALL. Toto je učiněno, když je symbol, který je volán, mezi +/-32k instrukčními slovy z aktuální instrukce.

Relocatable
Objekt, jehož adresa nebyla přidělena k pevnému umístění v paměti.

Relocatable Section
16-bitový assembler - sekce, jejíž adresa není pevná (absolutní). Linker přidělí adresu pro tyto sekce pomocí procesu přemístění (relocation).

Relocation
Proces učiněný linkerem, ve kterém absolutní adresy jsou přiděleny k přemístěným sekcím (relocatable section) a všechny symboly těchto sekcí jsou vylepšeny na nové adresy.

ROM
Read Only Memory (Program Memory). Paměť, kterou nelze modifikovat.

Run
Příkaz, který uvolňuje emulátor z halt a dovoluje mu pouštět aplikační kódy a měnit nebo reagovat na I/O v reálném čase.

Run-time Model
Popisuje použití zdroje cílové architektury.

S

Scenario
Pro MPLAB SIM simulátor, nastavení pro ovládání podnětu.

Section
GCC ekvivalent OCG psect. Blok kódu nebo dat, který je brán linkerem jako celek.

Section Attribute
GCC charakteristika přisouzená sekci. (např. sekce `access`).

Sequenced Breakpoints
Body přerušení, které se udávají v sekvenci. Spuštění sekvence bodů přerušení je odzdola nahoru. Poslední bod přerušení v sekvenci se udá jako první.

Serialized Quick Turn Programming
Seralizace vám umožňuje programovat sériové číslo do každého mikrokontrolerového zařízení, které programátor programuje. Toto číslo lze použít jako vstupní kód, heslo nebo ID číslo.

SFR
Viz Special Function Registers.

Shell
Schránka MPASM assembleru je vstupní rozhraní makro assembleru. Existují dvě schránky MPASM assembleru: jedna pro DOS verzi a druhá pro Windows verzi.

Simulator
Software, který vytváří model operace zařízení.

Single Step
Tento příkaz postupuje kódem po jedné instrukci. Po každé instrukci MPLAB IDE/MPLAB X IDE obnoví registrační okna, sledujte proměnné a zobrazení stavu, abyste mohli analyzovat a debugovat spuštění instrukce. Můžete také procházet po jednom kroku zdrojovým kódem C kompileru, ale namísto spouštění jednotlivých instrukcí MPLAB IDE/MPLAB X IDE spustí všechny assembly level instrukce generované řádkem high level C příkazu.

Skew
Informace související se spuštěním instrukce se objeví na sběrnici procesoru v různé časy. Například spuštěné optokódy se objeví na sběrnici během spuštění předchozí instrukce, zdrojová data adresy a hodnoty a cílová adresa dat se objeví, když jsou optokódy doopravdy spuštěny. Trace buffer zachytí informaci, která je na sběrnici, najednou. Proto jeden vstup trace bufferu bude obsahovat informace o spuštění pro tři instrukce. Počet zachycených syklů z jednoho kusu informace k druhému pro spuštění jednotlivé instrukce se nazývá skew.

Skid
Když je použit bod přerušení hardwaru pro halt procesoru, je možné spustit jednu nebo více přídavných instrukcí před zastavením procesoru. Počet navíc spuštěných instrukcí po zamýšleném bodu přerušení, se nazývá skid.

Source Code
Forma, kterou je počítačový program napsán programátorem. Zdrojový kód je napsán ve formálním programovacím jazyce, který je možné přeložit do strojového kódu nebo spustit pomocí převaděče.

Source File
Textový ASCII soubor obsahující zdrojový kód.

Special Function Registers
Část datové paměti (RAM) určená pro registry, které ovládají I/O funkce procesoru, I/O status, časovače a jiné módy periferií.

SQTP
Viz Serialized Quick TUrn Programming.

Stack, Hardware
Umístění v PIC mikrokontrolérech, kde zpětná adresa je uložena, když je učiněno funkční volání (function call).

Stack, Software
Paměť použitá aplikací pro uložení zpětných adres, parametrů funkcí a místních proměnných. Tato paměť je dynamicky umístěny na runtime pomocí instrukcí v programu. Dovoluje funkční volání (function call) opětovného vstupu.

Static RAM or SRAM
Static Random Access Memory. Programová paměť, kterou můžete číst/psát na cílovou desku, která nepotřebuje být často obnovována.

Status Bar
Stavová lišta je umístěna na spodní straně okna MPLAB IDE/MPLAB X IDE a indikuje takové informace jako pozici kurzoru, vývojový mód a zařízení a aktivní nástrojovou lištu.

Step Into
Tento příkaz je stejný jako Single Step. Step Into (jako opak Step Over) následuje instrukci CALL do podprogramu.

Step Over
Step Over (překročení) vám dovoluje debugovat kód bez vstupu do podprogramů. Když překračujete instrukci `CALL`, další bod přerušení bude nastaven na instrukci po `CALL`. Pokud se z nějakého důvodu podprogram dostane do nekonečné smyčky nebo se správně nenavrátí, další bod přerušení nebude nikdy dosažen. Příkaz Step Over je stejný jako Single Step, vyjma toho, jak zachází s instrukcemi `CALL`.

Step Out
Step Out vám dovoluje vystoupit z podprogramu, který právě prostupujete. Tento příkaz spouští zbytek kódu v podprogramu a poté zastaví spuštění na návratové adrese do podprogramu.

Stimulus
Vstup do simulátoru, tj. data vygenerovaná pro nacvičení odezvy simulace na externí signály. Často jsou data dána do formy seznamu akcí v textovém soubotu. Podnět může být asynchronní, synchronní (pin), hodinový a registrový.

Stopwatch
Počítadlo pro měření cyklů spuštění.

Storage Class
Rozhoduje o životnosti paměti související s identifikovaným objektem.

Storage Qualifier
Indikuje speciální vlastnosti objektu (např. `const`).

Symbol
Symbol je obecný mechanismus pro popsání různých částí, z nichž se skládá program. Tyto části obsahují názvy funkcí, názvy proměnných, názvy sekcí, názvy souborů, tagy názvů struct/enum/union, atd. Symboly v MPLAB IDE/MPLAB X IDE odkazují hlavně na názvy proměnných, názvy funkcí a značky sestavení (assembly labels). Hodnota symbolu po propojení je její hodnota v paměti.

Symbol, Absolute
Reprezentuje okamžitou hodnotu jako definici skrze `.equ` direktivu.

System Window Control
Ovládání systémového okna je umístěno v levém horním rohu oken a některých dialogů. Kliknutím na toto ovládání obvykle vyvoláte menu, v němž jsou položky "Minimize" (minimalizovat), "Maximize" (maximalizovat) a "Close" (zavřít).

T

Target
Odkazuje na uživatelův hardware.

Target Application
Software umístěný na cílové desce.

Target Board
Obvody a programovatelná zařízení, která tvoří cílovou aplikaci.

Target Processor
Mikrokontrolerové zařízení na cílové aplikační desce.

Template
Řádky textu, které sestavíte pro zapojení do vašich souborů později. MPLAB Editor uskladňuje šablony v souborech šablon (template files).

Tool bar
Řádek nebo sloupec ikon, na kterou můžete kliknout pro spuštění funkcí MPLAB IDE.

Trace
Funkce emulátoru nebo simulátoru, která zaznamenává spuštění programu. Emulátor zaznamenává spuštění programu do svého trace bufferu, který je načten do trace okna MPLAB IDE.

Trace Memory
Sledovací paměť obsažená v emulátoru. Někdy je též nazývána trace buffer.

Trace Macro
Makro, které poskytne trace informace z dat emulátoru. jelikož se jedná o software trace,

makro musí být přidáno do kódu, kód musí být opět kompilován nebo znovu sestaven a cílové zařízení musí být naprogramováno tímto kódem, než bude trace pracovat.

Trigger Output
Trigger output odkazuje na výstupní signál emulátoru, který lze generovat na jakékoliv adrese nebo rozsahu adres, a je nezávislý na nastavení trace a breakpoint (bod přerušení). Jako trigger output lze nastavit jakékoliv číslo.

Trigraphs
Tříznakové sekvence, všechny začínající ??, které jsou definovány ISO C jako náhrady za jednotlivé znaky.

U

Unassigned Section
Sekce, která nebyla přidělena k specifickému paměťovému bloku cíle v příkazovém souboru linkeru. Linker musí najít paměťový blok cíle, do něhož umístí nepřidělenou sekci.

Uninitialized Data
Data, která jsou definována bez počáteční hodnoty. V C,
`int myVar;`
definuje proměnnou, která bude umístěna v neinicializované sekci dat.

Upload
Funkce Upload přenáší data z nástroje, například emulátoru nebo programátoru, co host PC nebo z cílové desky do emulátoru.

USB
Universal Serial Bus. Externí periferní rozhraní pro komunikaci mezi počítačem a externími periferiemi pomocí kabelu s bi-seriálním přenosem. USB 1.0/1.1 podporuje přenos dat 12 Mb za sekundu. USB 2.0 podpouje přenos dat až do 480 Mb za sekundu.

V

Vector
Umístění paměti, kam aplikace přeskočí, když dojde buď k Resetu nebo přerušení (interrupt).

W

Warning
MPLAB IDE/MPLAB X IDE - poplach, který vás upozorňuje na situaci, která by mohla způsobit fyzické poškození zařízení, softwarového souboru nebo vybavení.
16-bitový assemblre/kompiler - varování ohlašuje stavy, které mohou indikovat problém, ale nezastaví proces. V MPLAB C30 varovné zprávy ohlašují název zdrojového souboru a číslo řádku, ale obsahují text `'warning'` pro jejich odlišení od chybových hlášek.

Watch Variable
Proměnná, kterou můžete sledovat v průběhu debugování v okně Watch.

Watch Window
Okno Watch obsahuje seznam proměnných watch, které jsou obnovovány na každé bodě

přerušení.

Watchdog Timer (WDT)
Časovač na PIC mikrokontroléru, který resetuje procesor po určeném čase. WDT lze aktivovat nebo deaktivovat a nastavit pomocí konfiguračních bitů.

Workbook
Pro MPLAB SIM simulátor, nastavení pro generování SCL podnětu.

Workspace
Pracovní prosto obsahuje informace o MPLAB IDE na vybraném zařízení, vybraný debugovací nástroj a/nebo programátor, otevřená okna a jejich umístění a další nastavení IDE konfigurace.