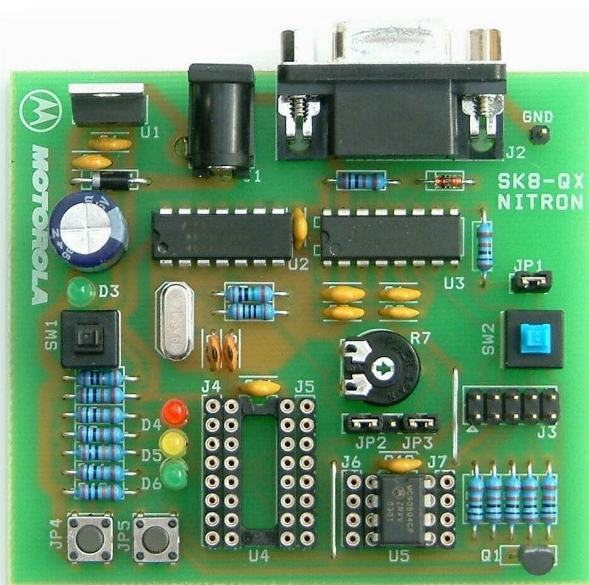


Vývojový kit JANUS

uživatelský manuál



Číslo verze dokumentu: 1
Určeno pro kit verze 2.1 a 2.1.1
V Praze dne 27.5.2003

Obsah

1	Kit JANUS	5
2	Uvedení kitu do chodu (quickstart).....	5
2.1	Krok 1: ICS08 a simulace mikrokontroleru.....	6
2.2	Krok 2: Simulace ve spolupráci s kitem (In circuit simulation).....	7
2.3	Krok 3: Programování paměti FLASH v MCU (nahrávání programu)	9
2.4	Krok 4: Trasování programu v MCU (debug).....	11
3	Hardware kitu.....	13
3.1	Blokové schéma kitu – logické části	13
3.1.1	Napájecí zdroj.....	14
3.1.2	Generátor hodin (ladicí adaptér)	14
3.1.3	Sériové rozhraní (ladicí adaptér).....	14
3.1.4	Konfigurační propojky (ladicí adaptér).....	14
3.1.5	Jednoduché periferie	14
3.1.6	MINIMON.....	14
3.1.7	Samotný mikrokontroler.....	15
4	Mikrokontrolery HC08	15
4.1	Vývody mikrokontroleru NITRON.....	16
4.2	Monitor mode – vývoj v aplikaci.....	16
4.3	Přepínání MONITOR/USER MODE.....	17
4.4	Komunikace s mikrokontrolerem v MONITOR MODE	18
4.5	Obvodové řešení v kitu JANUS	19
4.6	Tlačítka a konfigurační propojky	19
5	Vývojový software	22
5.1	P&E Micro - ICS08	22
5.2	Metrowerks – CodeWarrior for HC08.....	22
5.3	Cosmic – IDEA08.....	22
5.4	ImageCraft – ICC08	22
6	Příklady	23
6.1	QT-first	23
6.2	QT-buttons-flipflop.....	24
7	Kit ve funkci externího ladicího adaptéru.....	25
7.1	Popis signálů konektoru MINIMON.....	25
8	Přílohy	28
8.1	Nastavení nástrojové lišty ICS08	28
8.2	Programování a mazání FLASH, pomoc při ztrátě vstupního hesla.....	29
8.3	Kompatibilita s DEMO kitem Motorola M68DEMO908QT4	30
8.4	Zapojení kitu.....	31
8.4.1	Napájecí zdroj.....	31
8.4.2	Oscilátor	31
8.4.3	Interface RS232.....	32

Vývojový kit JANUS

8.4.4	Patice pro mikrokontrolery.....	33
8.4.5	Jednoduché periferie	34
8.4.6	MINIMON.....	35
8.5	Rozpis součástek na desce a schéma kitu JANUS	36

1 Kit JANUS

Kit JANUS je koncipován jako velmi jednoduchý a snadno dostupný nástroj pro seznámení se s problematikou mikrokontrolerů Motorola 68HC08. Kit umožňuje vývoj software specificky pro podskupinu mikrokontrolerů HC08 „NITRON“, tedy 68HC908QT a QY. Lze jej velmi dobře využít pro demonstraci možností mikrokontroleru, pro návrh jednoduchých aplikací nebo pro malosériové programování mikrokontrolerů. Pokud je cílová aplikace s mikrokontrolerem 68HC08 vybavená odpovídajícím servisním konektorem (MON08, MINIMON, apod.), lze kit zároveň využít i jako vývojový a programovací adaptér.

Kit je nabízen jak ve formě finálního výrobku, tak v podobě stavebnice obsahující plošný spoj a všechny potřebné elektronické součástky.

Balení kitu obsahuje:

- Vlastní kit Janus – dle typu buď hotový otestovaný, nebo stavebnice sestávající z prázdného plošného spoje a součástek
- Mikrokontroler NITRON MC68HC908QT4CP (osmivývodová verze, 4kB FLASH)
- Vývojový systém Metrowerks CodeWarrior HC08 SE
- CD s dokumentací a dalšími vývojovými prostředky ICS08 (P&E Micro) a COSMIC (IDEA08)
- Propojovací sériový kabel 9F-9M (pouze pro verzi hotového kitu)

2 Uvedení kitu do chodu (quickstart)

Předpokládané vybavení:

- Malý vývojový kit JANUS s osazeným mikrokontrolerem NITRON MC68HC908QT4CP (MC908QT4CP)
- Napájecí zdroj 8-25V, alespoň 50mA se souosým napájecím konektorem 2,1mm nebo 2,5mm, záporný pól na obalu¹ (←⊖→). Například lze použít typu „kostka do zásuvky“ pro napájení osobních přehrávačů apod.
- Sériový kabel 9F-9M (prodlužovací kabel k sériové myši, kabel k modemu apod.) nebo jiný odpovídající konektoru sériového portu použitého počítače.
- Software ICS08 pro HC08 firmy P&E Micro (na příloženém CD). Rovněž lze software po zaregistrování zdarma stáhnout z <http://www.pemicro.com>.
- Vzorový program v assembleru qt-first.asm (na příloženém CD). Další vzorové programy jsou k dispozici též na CD a dále na adrese <http://www.hc08.cz>



¹ Chybná polarizace napájení nezpůsobí poškození kitu, kit pouze nebude pracovat (nerozsvítí se D3).

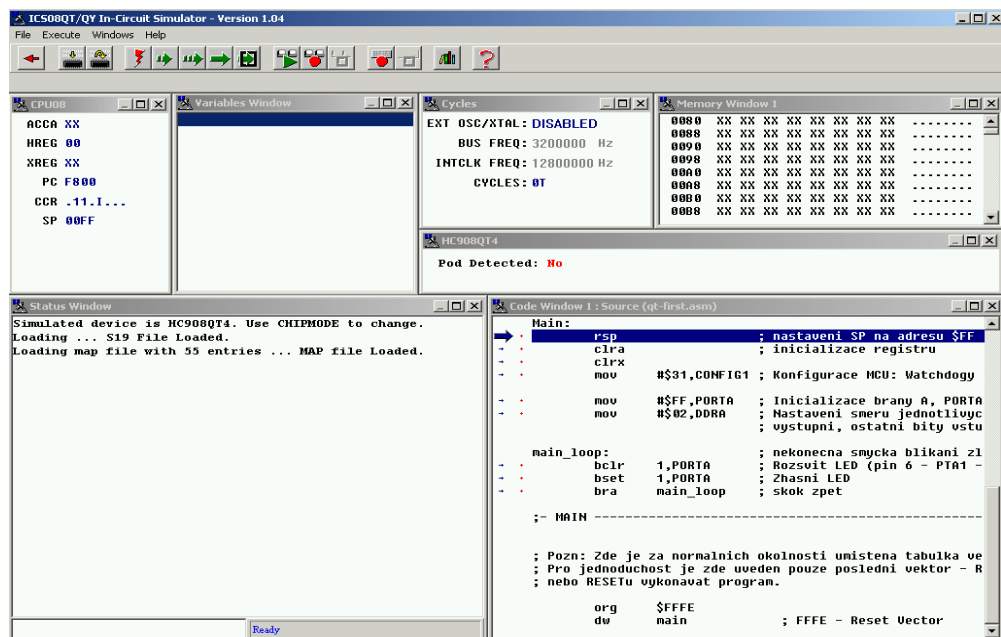
2.1 Krok 1: ICS08 a simulace mikrokontroleru

Pozn.: V tomto kroku probíhá simulace pouze na úrovni počítače, kit není v provozu a nemusí být k počítači ani připojen.

1. Nainstalujte software ICS08 (pozn.: Ve verzi 1.07 nebo nižší jsou chybně nastavena tlačítka v nástrojové liště. Podívejte se do kapitoly Pomoc při potížích - Nastavení nástrojové lišty ICS08)
2. Překopírujte obsah adresáře examples z CD do adresáře s vývojovým software ICS08 (c:\pemicro\ics08qtqyz, popřípadě Vámi zvolený)
3. Spusťte vývojové prostředí WinIDE
4. Pomocí volby File->Open otevřete vzorový program qt-first.asm².


(Pozn: Program qt-first.asm se odkazuje na soubor popisu registrů MCU qtqy_registers.inc. Tento soubor se musí nacházet ve stejném adresáři jako program sám.)

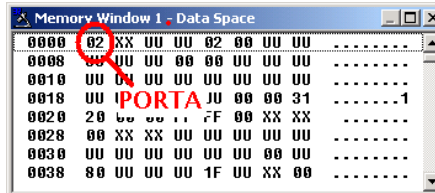
5. Pomocí ikony  v nástrojové liště přeložte program
6. Pomocí ikony  spusťte simulátor MCU





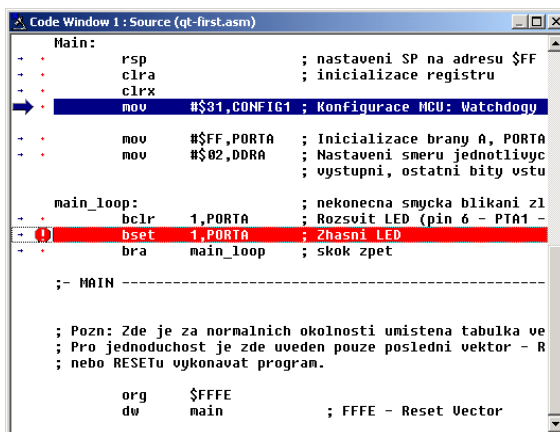
V tomto okamžiku je spuštěn úplný simulátor, který přímo s kitem nespolupracuje, funkce mikrokontroleru včetně jeho periférií jsou zcela simulovány na úrovni počítače.

² qt-first.asm je velmi jednoduchý příkladový program, který je určen především jako pomůcka pro pochopení práce s vývojovým prostředím ICS08 a pro první proniknutí do problematiky mikrokontrolerů HC08. Jeho detailní popis lze nalézt v kapitole Příklady tohoto manuálu.

Pomocí tlačítka  (step) nebo klávesy F5 vyzkoušejte trasování programu instrukcí po instrukci. V oknech CPU08 a Memory window lze pozorovat změny obsahu registrů a paměti.



Tlačítko  (multiple step) spouští simulovaný běh programu tak, že se neustále obnovují na obrazovce všechny informace. Tlačítko  (Go!) spustí simulaci maximální rychlostí bez občasného vyvolávání informací na obrazovce. Obě tyto simulace lze zastavit např. klávesou <ESC>, nebo breakpointem:




Breakpoint lze nastavit například takto:

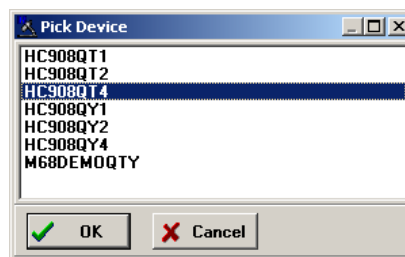
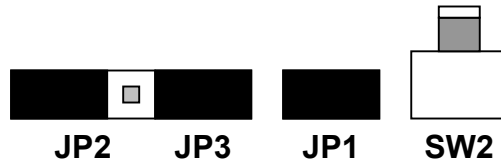
- Levým tlačítkem myši klikněte na instrukci, kde má být umístěn breakpoint
- Pravým tlačítkem vyvolejte menu a zvolte „toggle breakpoint at cursor“
- Provedení simulace pomocí mutlipe_step nebo go! se pak na breakpointu zastaví.
- Breakpoint zrušíte obdobným postupem. V simulátoru můžete mít najednou nastavené neomezené množství breakpointů.

2.2 Krok 2: Simulace ve spolupráci s kitem (In circuit simulation)

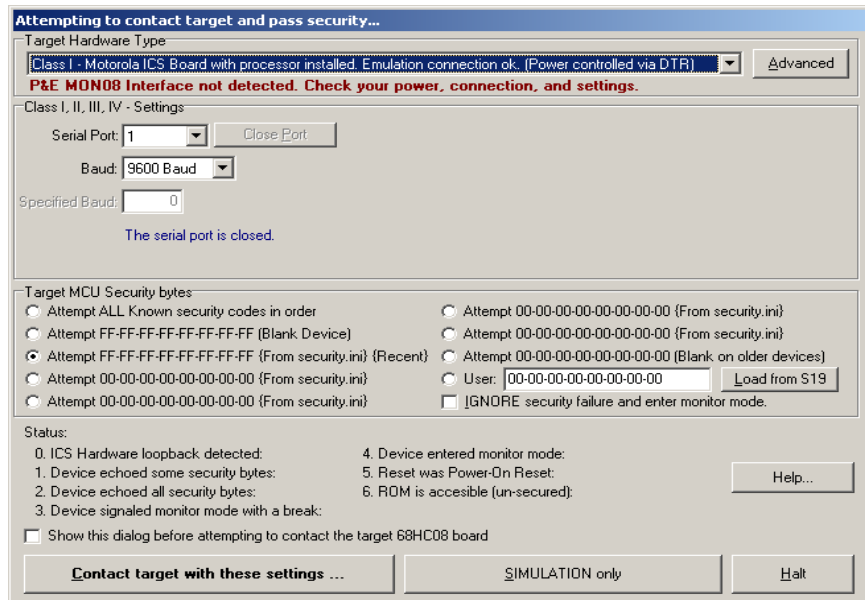
In-circuit-simulator se od standardního simulátoru liší v tom, že vstupy a výstupy mikrokontroleru nejsou simulovány, ale odpovídají přímo stavům vývodů mikrokontroleru. Vnitřní funkce mikrokontroleru jsou ale nadále simulovány v počítači, vývojový kit tak přechází do funkce specifické I/O desky. V tomto režimu se do paměti FLASH mikrokontroleru Váš software nenahrává, v paměti RAM mikrokontroleru běží speciální software ovládající na základě příkazů počítače vstupy a výstupy.

Vývojový kit JANUS

- Ukončete simulátor
- Připojte kit k počítači sériovým kabelem
- Zkontrolujte jumpery na kitu, zda jsou nastaveny správně.
- Připojte kit ke zdroji napájení. V levé horní části kitu se rozsvítí zelená LED (D3) signalizující napájení mikrokontroleru.
-  Spusťte In-circuit-simulator
- Vyberte typ obvodu, který používáte (obvykle HC908QT4)³:



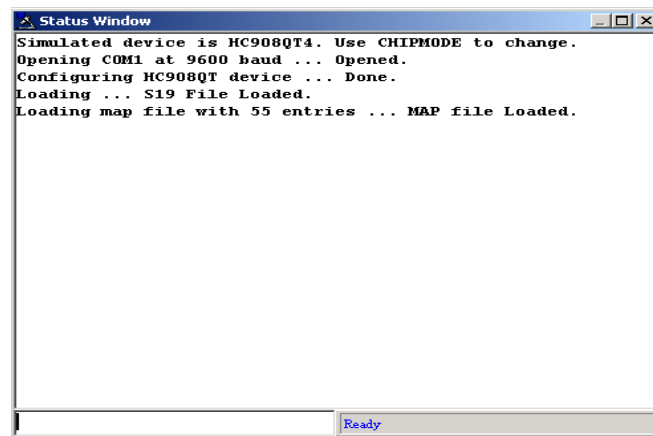
Dále zvolte Hardware type: „Class I“, nastavte port, ke kterému máte kit připojen, a přenosovou rychlost 9600bd:



³ Tento výběr se provádí pouze při prvním spuštění aplikace, při příštích už simulátor předpokládá použití stejné součástky. Pro změnu typu simulované součástky lze použít příkaz CHIPMODE.

Po zvolení „Contact target with these settings“ se ICS připojí k Vašemu kitu. Pokud se tak nestane (objeví se obdobné okno znovu), resetujte kit stisknutím vypínacího tlačítka SW1 a pokuste se o spojení znovu. Pokud problém přetrvává, vyhledejte v části řešení potíží detailní popis problematiky komunikace ICS08 s vývojovým kitem.

Po úspěšném propojení kitu s in-circuit-simulátorem (ICS) se objeví sestava oken shodná s úplným simulátorem, pouze s tím rozdílem, že příkazové okno hlásí připojení ke kitu:




13. Dále postupujte obdobně jako v případě úplné simulace (krokování, spouštění programu, breakpointy apod.). Na rozdíl od minulého případu bude na krokování programu reagovat i žlutá LED kitu, ne pouze výpis paměti na obrazovce počítače. Reakce simulátoru budou nepatrně pomalejší, neboť počítač kromě vlastní simulace komunikuje s kitem.

2.3 Krok 3: Programování paměti FLASH v MCU (nahrávání programu)

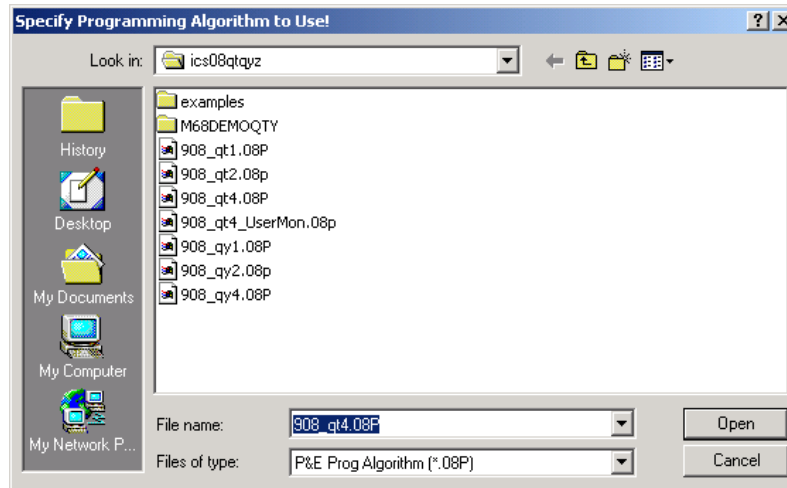
Jednou z funkcí vývojového prostředí ICS08 je též programátor FLASH mikrokontroleru. Pomocí tohoto nástroje lze paměť FLASH mikrokontroleru mazat, nahrávat nebo číst její obsah.

14. Ukončete práci s ICS

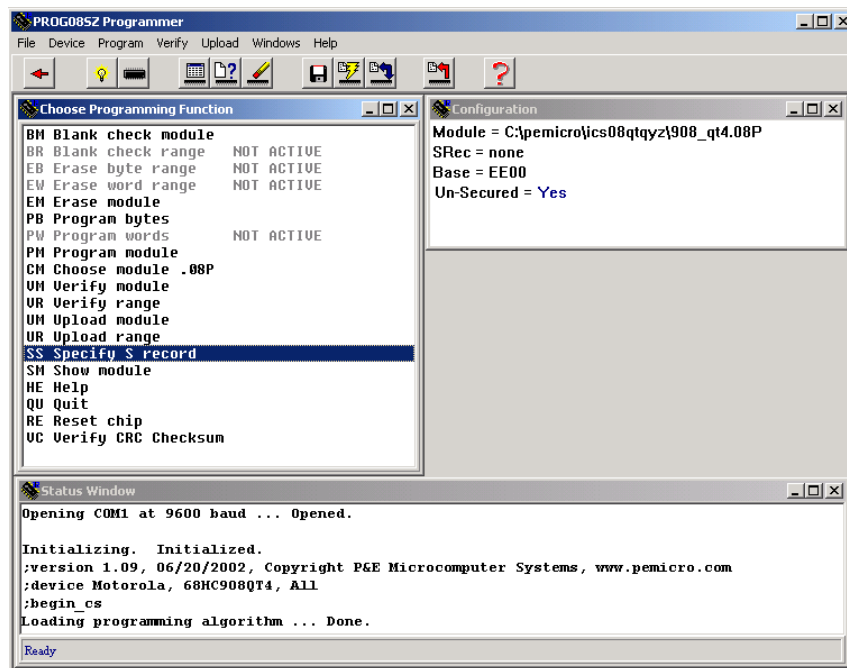


15. Pomocí ikony  v nástrojové liště spustíte programátor (ujistěte se, že neběží žádná další součást ICS08 (např ICS, nebo ICD), která by pracovala s kitem a tedy blokovala sériový port).

16. Při úspěšném navázání spojení s kitem se objeví okno s výběrem programovacího algoritmu⁴:

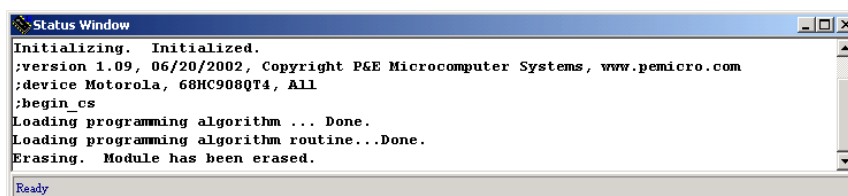


17. Zvolte algoritmus 908_qt4.08p.



⁴ V době vydání této příručky je pro NITRON funkční pouze algoritmus 908_qt4.08p a 908_qt4_UserMon.08p určený pro práci s kitem M68DEMO908QT4. Nicméně pomocí algoritmu 908_qt4.08p lze programovat jakýkoliv mikrokontroler NITRON řad QT či QY.

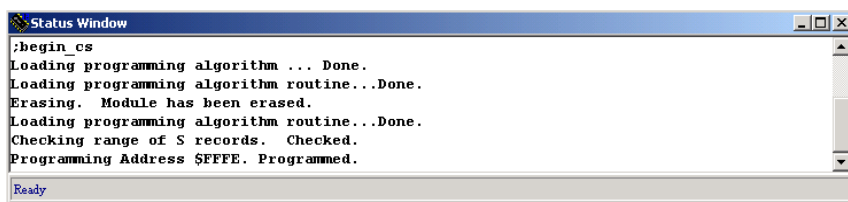
18. V levém horním pod-okně naleznete seznam příkazů, které je programátor schopen provádět. Nejprve vyberte pomocí příkazu SS „Specify S record“ soubor, jehož obsah budete do mikrokontroleru nahrávat – v adresáři examples naleznete qt-first.s19.
19. Dříve než poprvé vymažete FLASH mikrokontroleru, je pro budoucí použití vhodné poznamenat si kalibrační konstantu interního generátoru hodin⁵ (viz kapitola OSC uživatelského manuálu MC68HC908QY4/D k mikrokontroleru NITRON). Její hodnota je uložena z výroby v paměti FLASH a kompletním smazáním FLASH se vymaže i ta.
20. Zvolte příkaz SM – Show module a zadejte adresu **FFC0**. Na této adrese je z výroby změřená hodnota kalibrační konstanty. Po zaznamenání okno s výpisem paměti zavřete.
21. Před programováním je vhodné FLASH mikrokontroleru vymazat příkazem EM „Erase module“. Ve stavovém pod-okně naleznete potvrzení, že byl mikrokontroler úspěšně vymazán:



```

Status Window
-----
Initializing.  Initialized.
;version 1.09, 06/20/2002, Copyright P&E Microcomputer Systems, www.pemicro.com
;device Motorola, 68HC908Q14, All
;begin_cs
Loading programming algorithm ... Done.
Loading programming algorithm routine...Done.
Erasing.  Module has been erased.
Ready
    
```

22. Nyní zvolte funkci PM „Program module“:



```


Status Window
-----
;begin_cs
Loading programming algorithm ... Done.
Loading programming algorithm routine...Done.
Erasing.  Module has been erased.
Loading programming algorithm routine...Done.
Checking range of S records.  Checked.
Programming Address $FFFE. Programmed.
Ready
    
```

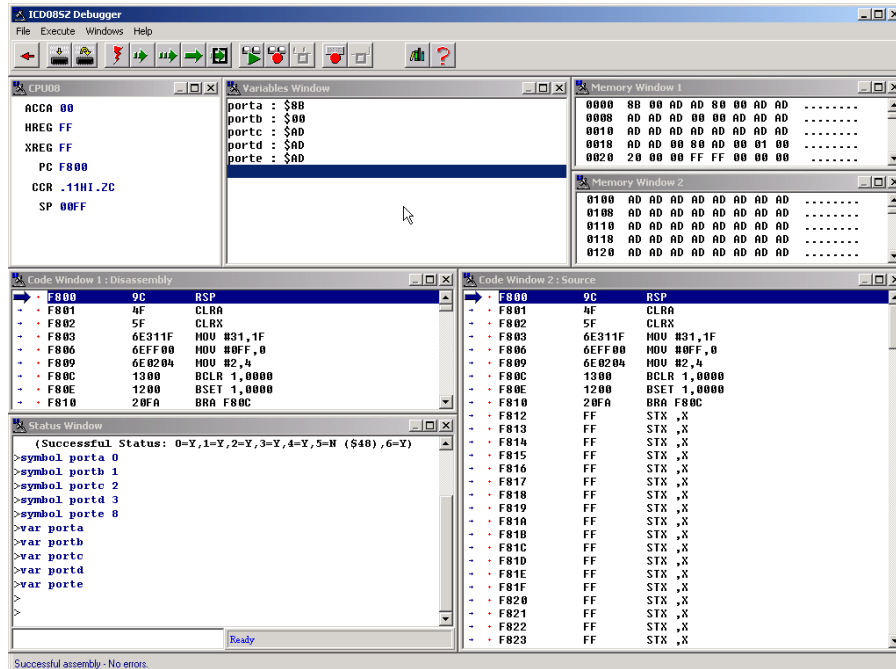
23. Do paměti mikrokontroleru byl nahrán program. Nyní lze např. kit přepnout do uživatelského režimu (SW2) a program restartem desky (SW1) spustit, anebo přikročit k dalšímu kroku – ladění programu přímo v mikrokontroleru – In-circuit-debugging.

2.4 Krok 4: Trasování programu v MCU (debug)

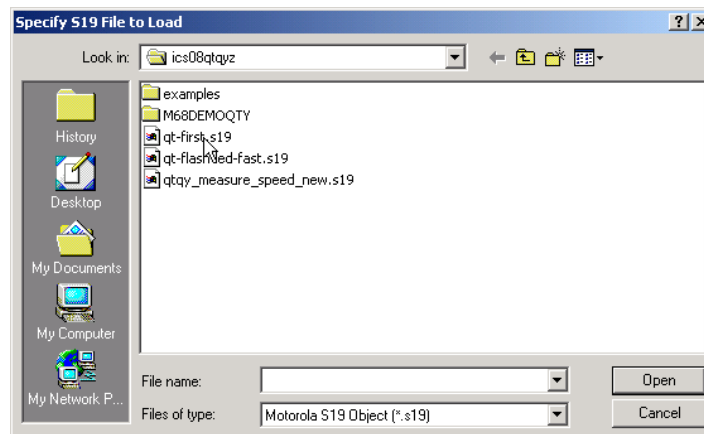
Pod pojmem debugging se v problematice mikrokontrolerů míní situace, kdy je program trasován přímo v samotném mikrokontroleru a nadřazený počítač slouží pouze jako inteligentní monitor. V režimu in-circuit-debugging (ICD) již musí být uživatelský program nahrán v paměti FLASH, neboť mikrokontroler vykonává jednotlivé instrukce, které vybírá z paměti FLASH stejně, jako kdyby program běžel plnou rychlostí.

⁵ Jak je uvedeno v popisu OSC manuálu MC68HC908QY4/D, výrobní přesnost interního generátoru hodin je +/-25%. Oscilátor lze ale jemně doladit pomocí kalibračního registru OSCTRIM. Hodnota kalibrační konstanty je hrubě změřena ve výrobě a pro přesnost pod 5% ji lze využít. K dispozici je v paměti FLASH na adrese 0xFFC0. Není však chráněna proti vymazání a při prvním smazání celé FLASH paměti se vymaže rovněž.

24. Ukončete programátor a pomocí ikony  spusťte In-circuit-debugger.




25. Na první pohled se debugger od simulátoru liší v tom, že v okně „Source“ nevidíme skutečný zdrojový kód, ale zpětně přeložený obsah paměti FLASH. Pro přehlednější trasování můžeme i zde trasovat přímo ve zdrojovém kódu. Z nabídky File -> Load S19 File nahrajeme do debuggeru S19 soubor a k němu příslušný ASM:



26. V okně Source se objeví skutečný zdrojový kód⁶.

⁶ Debugger primárně prochází kódem v mikrokontroleru, pokud zvolíme jiný S19 soubor, než který odpovídá obsahu mikrokontroleru, nebude navigace ve zdrojovém kódu fungovat správně.

Nyní máme k dispozici stejné nástroje jako v případě ICS či simulátoru. Krokování po instrukcích bude pochopitelně výrazně pomalejší, neboť s každou instrukcí musí debugger pomocí sériové linky přečíst klíčové informace z mikrokontroleru znova.

Debugger má proti ICS výhodu v tom, že kód běží přímo v mikrokontroleru, takže pokud zvolíme rychlý běh programu , poběží program skutečnou rychlostí (real-time). Lze takto testovat části programu, které jsou časově závislé nebo již odladěné a jejich běh nepotřebujeme sledovat.

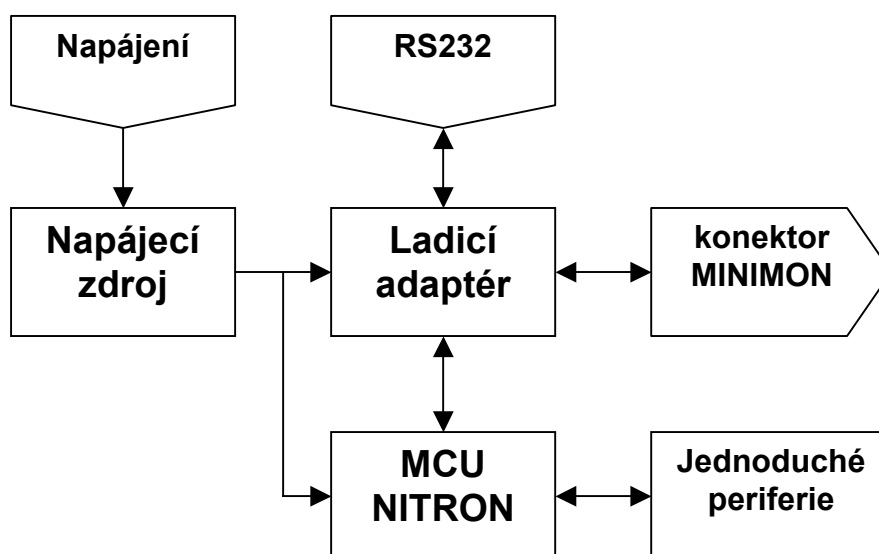
Zastavit takto běžící program lze ale pouze resetem nebo breakpointem.

27. Úspěšně jste zvládli základy práce s vývojovým prostředím ICS08.

3 Hardware kitu

Konstrukce kitu JANUS vychází z požadavku na maximální jednoduchost, možnost uživatelské stavby kitu bez laboratorního vybavení a celkově nízkou cenu. Z toho vychází i součástková základna, na které je kit postaven, všechny součástky byly vybírány s ohledem na jejich běžnou dostupnost v maloobchodní síti.

3.1 Blokové schéma kitu – logické části



Vývojový kit JANUS vznikl sloučením několika celků. Základem je samotný mikrokontroler NITRON, který spolu s jednoduchými periferiemi tvoří modelovou aplikaci. Pro účely vývoje je na kitu ladící adaptér, který zajišťuje potřebné signály pro vývoj software přímo v mikrokontroleru (In circuit development). Zahrnuje v sobě generátor hodin, sériové rozhraní a obvod pro reset z počítače. Kit je dále vybaven stabilizovaným napájecím

zdrojem s možností momentového vypnutí napájení pro simulaci zapínání mikrokontrolerového systému v reálné aplikaci.

3.1.1 Napájecí zdroj

Kit se napájí usměrněným napětím 9-25V, maximální spotřeba je 50mA. Toto napětí je v kitu upraveno standardním zdrojem se stabilizátorem 7805 na 5V a rozvedeno po desce. Proti náhodnému prepólování je kit chráněn diodou. Pro účely přechodu mezi jednotlivými pracovními režimy mikrokontroleru je zdroj kitu vybaven tlačítkem vypnutí napájení (SW1). Toto tlačítko je bez aretace a slouží ke krátkodobému vypnutí mikrokontroleru a následnému provedení „POR“ – tzv. power-on reset.

3.1.2 Generátor hodin (ladicí adaptér)

Mikrokontrolery 68HC908QY/QT jsou standardně vybaveny vnitřním generátorem hodin 12,8MHz, avšak pro běh v módu monitoru vyžadují externí generátor hodin. Kit proto obsahuje jednoduchý krystalový oscilátor s kmitočtem 9,8304MHz.

3.1.3 Sériové rozhraní (ladicí adaptér)

Pro komunikaci s mikrokontrolerem řady 68HC08 v monitoru se využívá sériové linky RS232. Kit proto obsahuje obvod sériového rozhraní, který převádí standardní sériovou linku na poloduplexní signál pro mikrokontroler.

3.1.4 Konfigurační propojky (ladicí adaptér)

Jak je již patrné z textu výše, nabízí kit i přes značnou obvodovou minimalizaci jistou úroveň flexibility. Pomocí propojek (jumper) lze přepínat pracovní módy mikrokontroleru, odpojovat hodiny a připojovat periferie. Lze tedy do jisté míry simulovat situaci mikrokontroleru v reálném hardware.

3.1.5 Jednoduché periferie

Pro první seznámení se s mikrokontrolery 68HC08 a pro vývoj jednodušších aplikací lze s výhodou využít dvou tlačítek, tří LED a jednoho trimru jako základních vstupně-výstupních obvodů. Uživatel tak nemusí před začátkem vývoje tyto periferie připravovat a zároveň může využít řadu příkladů, které jsou psány právě pro spolupráci s takto navrženými periferiemi.

3.1.6 MINIMON

V případě vývoje nebo úpravy software v cílové aplikaci lze kit JANUS použít též jako programovací/ladicí adaptér mezi aplikací a počítačem. Pro tento případ je kit vybaven konektorem MINIMON, který obsahuje všechny potřebné signály pro vývoj software v aplikaci. Pomocí vhodného kabelu lze připojit aplikaci vybavenou konektorem MON08, MINIMON, nebo uživatelem definovaným, na který jsou vyvedeny signály nezbytné pro ladění. Detailní popis konektoru MINIMON je uveden v kapitole „Kit ve funkci externího ladicího adaptéru“.

3.1.7 Samotný mikrokontroler

Mikrokontroler je na kitu umístěn v patici, pro snadné vyjmutí. Všechny jeho vývody jsou kopírovány na dvou pomocných kontaktních dutinkových lištách, kam lze v případě potřeby připojit další periferie nebo měřicí zařízení.

S ohledem na to, že mikrokontroler NITRON (HC908QT/QY) se vyrábí v několika různých pouzdrech, obsahuje kit dvě patice pro provedení v pouzdrech DIL8 a DIL16. Signály jsou pro obě patice společné, proto nelze zároveň provozovat obě verze mikrokontrolerů.

4 Mikrokontrolery HC08

Obvody HC08 tvoří ucelenou řadu výkonných 8mi bitových mikrokontrolerů se stejným jádrem CPU08. Jednotlivé typy se liší pouze skladbou periférií a velikostí pamětí. Značení mikrokontrolerů HC08 se pak skládá z těchto částí⁷:

MC	68	HC	9	08	QT	4	C	P
1	2	3	4	5	6	7	8	9

- MC** – Plně kvalifikovaná součástka. Další možné značení XC – součástka, pro niž ještě nejsou provedeny všechny kvalifikační testy
- 68** – Označení řady mikrokontrolerů
- HC** – Obvodová technologie.
- 9** – Mikrokontroler obsahuje programovou FLASH. Pokud zde není nic, mikrokontroler je vyroben s pamětí ROM programovanou maskou
- 08** – Typové značení řady mikrokontrolerů HC08
- QT** – Zástupce typu QT. Další možná značení jsou např. AB, AZ, GP, GT, MR, KX, QY, atd. Popisuje skladbu periférií a přibližně cílovou skupinu aplikací, pro niž je mikrokontroler koncipován. Až na výjimky mají všechny modely stejného typu stejné periferie.
- 4** – Přibližná velikost paměti FLASH v kB (zde 4kB). Velikost paměti FLASH se u HC08 pohybuje od 1kB až do 64kB
- C** – teplotní rozsah, ve kterém je zaručena správná funkce mikrokontroleru. Teplotní rozsahy jsou tyto:

<nic>	0°C až 70°C
I	0°C až 85°C
C	-40°C až 85°C
V	-40°C až 105°C
M	-40°C až 125°C
- P** – typ pouzdra (zde DIL). Další možná značení jsou např. DW pro SOIC, FA pro 32pin LQFP apod.

⁷ Obvody s malými pouzdry (typicky 8mi vývodový NITRON apod.) mohou mít popis z prostorových důvodů zkrácený o některé nevýznamné části – např. MC68HC908QT4CP – tedy NITRON v osmivývodovém provedení má na obvodu označení MC908Q4CP

Příklady:

MC68HC908GR8CFA – Mikrokontroler HC08 typu GR obsahující 8kB FLASH, certifikovaný pro práci v teplotách -40°C až 85°C, v pouzdru LQFP s 32 vývody.

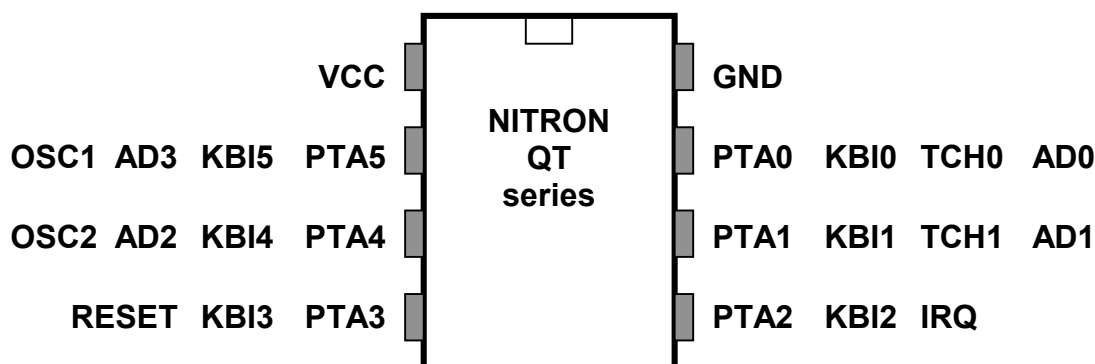
MC68HC908QY2CDW – Mikrokontroler HC08 typu QY (NITRON) s 2kB FLASH, certifikovaný pro práci v teplotách -40°C až 85°C, v pouzdru SOIC.

4.1 Vývody mikrokontroleru NITRON

Mikrokontrolery MC68HC908QT mají 8 vývodů, z čehož jsou dva napájecí a 6 je k dispozici uživateli. Na většinu vývodů jsou připojeny signály vnitřních periférií (vstupy A/D převodníku, výstupy časovače apod.). V případě, že je příslušná periferie neaktivní (signál nevyužívá), nebo na daný vývod žádná periferie připojena není, funguje vývod jako obecný signál s možností individuálního nastavení vstup-výstup.

„Větší“ verze NITRONu (68HC908QY) je v 16ti vývodovém pouzdře a oproti verzi QT nabízí uživateli dalších 8 vývodů s 8mi obecnými vstupně-výstupními signály (GPIO).

Detailní popis významu jednotlivých signálů lze najít v technické dokumentaci k mikrokontrolerům NITRON MC68HC908QT/QY.



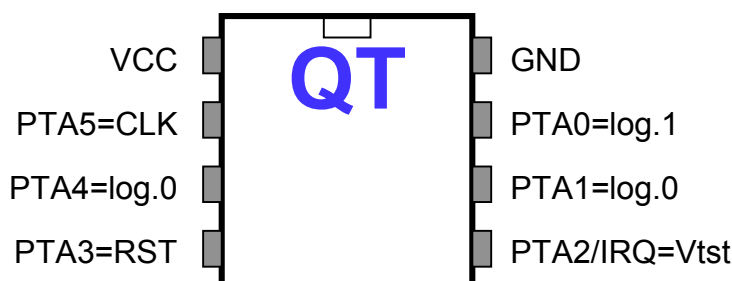
4.2 Monitor mode – vývoj v aplikaci

Jak již bylo zmíněno výše, mikrokontrolery HC08 disponují funkcí pro přímé ladění software přímo v aplikaci, tzv. ICD (in circuit development). Tato funkce umožňuje přímo v aplikaci přepnout mikrokontroler do speciálního vývojového režimu, ve kterém získá uživatel nad mikrokontrolerem plnou kontrolu. Touto cestou lze jednat vyvíjet software, diagnostikovat, programovat FLASH, atd. přímo v aplikaci. Odpadá tím potřeba obvodového emulátoru, programátoru a nutnosti vyjmát součástku z aplikace v případě diagnostiky či aktualizací firmware.

Mikrokontrolery HC08 pracují z tohoto pohledu ve dvou režimech. Za normálních okolností běží v uživatelském režimu, tzv. USER MODE, kdy se po resetu spustí uživatelský program. V případě, že je požadavek vstoupit do monitoru, např. z důvodu ladění programu nebo přeprogramování obsahu FLASH, přepne se mikrokontroler do MONITOR MODE, kdy se místo uživatelského programu spustí po RESETu speciální monitor (firmware v paměti ROM mikrokontroleru) a začne komunikovat s nadřazeným systémem, obvykle s počítačem

připojeným přes vývojový adaptér. Požadavek na přechod do MONITOR MODE se mikrokontroleru dává přivedením definovaných log. hodnot na vývody v době RESETu⁸.

4.3 Přepínání MONITOR/USER MODE



Po RESETu mikrokontroler testuje stav vývodů, zda-li jsou na ně přivedeny signály odpovídající podmínce nastartování monitoru. Pokud ano, mikrokontroler přejde do MONITOR MODE a umožní tak uživateli ladění v aplikaci (In-Circuit-Debugging – ICD). Při jakékoliv jiné konfiguraci stavů signálů nastartuje mikrokontroler v uživatelském režimu a spustí uživatelský program (USER MODE).

V případě, že mikrokontroler nastartuje v uživatelském režimu a startovací adresa uživatelského programu (adresy \$FFFE a \$FFFF) není nastavena, spustí se tzv. FORCED MONITOR MODE – monitorovací režim s mírnými odlišnostmi od standardního, popsaného v minulém odstavci. Tento speciální případ monitoru je určen spíše pro průmyslové využití, např. programování při sériové výrobě, a jeho detailní popis není předmětem tohoto textu. Lze jej najít v dokumentaci k mikrokontrolerům HC08.

Filozofie rozhodování, zda-li mikrokontroler nastartuje v uživatelském nebo monitorovacím režimu, je shodná pro všechny zástupce řady HC08, liší se pouze namapováním signálů na konkrétní vývody. Pro NITRON je konfigurace signálů pro vstup do monitoru následující:

PTA0	log.1
PTA1	log.0
PTA2/IRQ	Vtst
PTA4	log.0

Když mikrokontroler detekuje po RESETu tuto kombinaci signálů, spustí MONITOR MODE a na signály PTA3 a PTA5 připojí vstupy RESET a CLK:

PTA3	/RESET
PTA5	CLK

Mikrokontroler tedy v MONITOR MODE očekává od vnějšího prostředí ještě obsluhu signálu RESET a přivedení hodinového signálu, neboť vnitřní generátor hodin je v tomto režimu odpojený.

Několik hodinových taktů po resetu lze logické hodnoty na signálech PTA0, 1 a 4 odpojit. Vývod PTA0 nabyde významu komunikačního signálu a PTA1 a 4 mohou být k dispozici laděné aplikaci⁹.

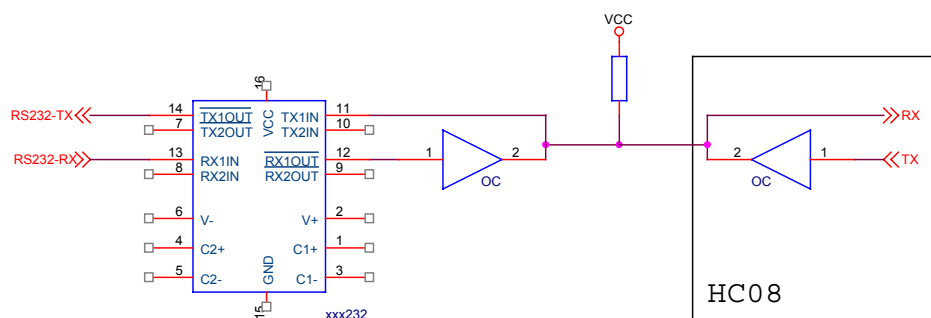
⁸ Jak bude vysvětleno dále, pokud mikrokontroler neobsahuje žádný program (startovací adresa je rovna 0xFFFF), mikrokontroler spouští monitor mode bez ohledu na stav vývodů.

Požadavek na konkrétní frekvenci hodinového signálu není, mikrokontroler akceptuje jakýkoliv kmitočet v rozsahu povolených pracovních podmínek. Uživatel ale musí zvolenému kmitočtu přizpůsobit komunikační rychlost s nadřazeným počítačem, neboť ta je pro NITRON a většinu dalších zástupců HC08 pevně stanovena na 1/1024 kmitočtu vstupních hodin. Tento poměr nelze měnit. Doporučený kmitočet je 9,8304MHz, kterému odpovídá komunikační rychlost 9600bd.

Režim	/IRQ	/RST	Reset vektor	PTA1	PTA4	ExtCLK [MHz]	BUSCLK [MHz]	COP
Vtst monitor mode	Vtst	Vdd	X	1	0	9,8304	2,4576	Vyp
Forced monitor mode	Vdd	X	0xFFFF	X	X	9,8304	2,4576	Vyp
Forced monitor mode	Vss (GND)	X	0xFFFF	X	X	X	3.2	Vyp
User mode	Vss-Vdd	X	naprogramovaný (není 0xFFFF)	X	X	X	-	Zap

4.4 Komunikace s mikrokontrolerem v MONITOR MODE

Jak již bylo naznačeno v minulém odstavci, po nastartování MONITOR MODE probíhá komunikace mezi laděným mikrokontrolerem a řídicím počítačem formou sériového přenosu po jediném signálu, tedy poloduplexně. Za předpokladu, že se řídicí počítač připojuje přes standardní sériový port RS232, musí vývojový adaptér pouze transformovat napěťové úrovně na 5V logiku a složit signály Tx a Rx sériové linky do signálu jediného obousměrného. Na převod napěťových úrovní jsou k dispozici hotové integrované obvody, složení signálů Rx a Tx do jednoho se pak provede formou složení signálů do sběrnice pomocí otevřených kolektorů a zakončovacího odporu „pull-up“, jak schematicky naznačuje následující obrázek.



Nepovinnou, avšak velmi užitečnou funkcí je možnost RESETu mikrokontroleru přímo počítačem. Zaužívané řešení je propojení signálu DTR sériové linky na RESET mikrokontroleru. Počítač tak může ve vhodných okamžicích mikrokontroler resetovat a celý proces ladění či programování více automatizovat¹⁰.

⁹ Tento problém, kdy monitor obsadí téměř všechny signály mikrokontroleru částečně řeší tzv. USER MONITOR, tedy v podstatě emulátor MONITORu, ale pracující v USER mode s vnitřním oscilátorem a bez potřeby zvýšeného napětí na Vtst. Více detailů lze nalézt v kapitole kompatibilita s kitem M68DEMO08QT4.

¹⁰ Většina vývojových nástrojů tuto funkci předpokládá, je tudíž žádoucí ji v ladicím adaptéru implementovat.

4.5 Obvodové řešení v kitu JANUS

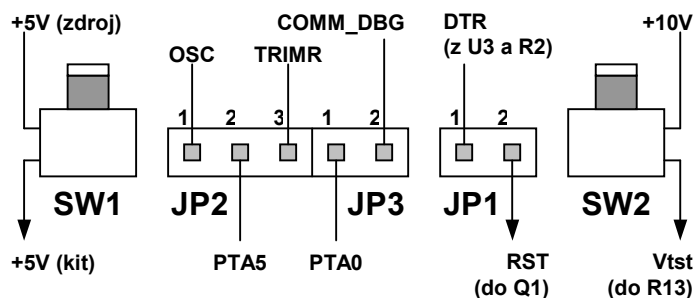
Výše popsané řešení komunikace v ladicím adaptéru pro mikrokontrolery HC08 je implementováno i v kitu Janus. Na místě OC budiče je použita shottky dioda, s níž má obvod díky nízkému úbytku v propustném směru diody dostatečné provozní parametry. Jako komunikační signál je v případě NITRONu zvolen signál PTA0 (pin 7 u typu QT, resp. Pin 12 u QY). S ohledem na to, že na signálu PTA0 je při RESETu vyžadována log.1, musí být sériový port v té době v klidu.

Filozofie přepínání mezi uživatelským a monitorovacím režimem v kitu JANUS vychází z faktu, že pokud nejsou v okamžiku startu mikrokontroleru (reset) na všech signálech PTA0,1,2 a 4 požadovaná napětí (viz výše), mikrokontroler nastartuje v USER MODE. Kit tedy trvale dodává požadované logické signály na všechny vývody vyjma jediného, vývodu PTA2/IRQ (tedy PTA0,1 a 4). Zvýšené napětí Vtst na vývodu PTA2/IRQ, jako poslední podmínka pro přechod z USER MODE do MONITOR MODE, je pak přiváděno přes přepínač SW2. Pokud je přepínač v poloze USER MODE (zamáčknut), zvýšené napětí je odpojeno a mikrokontroler po zapnutí nastartuje v uživatelském režimu. Pokud uživatel požaduje vstup do monitorovacího režimu, přepne SW2 do polohy MONITOR MODE (vysunut), na PTA2/IRQ se tím přivede Vtst a mikrokontroler po RESETu spustí monitor.

Logické hodnoty jsou na signálech PTA1 a 4 zajištěny pomocí rezistorů 100k (R10, R12), na signálu PTA0 rezistorem 1k (R1), zvýšené napětí Vtst je spínáno tlačítkem SW2. Od eventuelního dalšího hardware (vnější uživatelská aplikace) se pak v zásadě požaduje, aby v době resetu buď neovlivňoval ty signály, které rozhodují o přepnutí (PTA0, 1, 2, 4), anebo aby na nich udržoval příslušné hodnoty sám. Například lze na signál, kde je požadována log.1, připojit LED přes odpor na kladné napájecí napětí. Při RESETu, kdy je signál vstupní, působí LED jako slabý pull-up a přivede na daný signál log.1. Za běhu uživatelské aplikace pak signál funguje jako výstupní a ovládá diodu LED klasickým způsobem.

S ohledem na to, že mikrokontroler NITRON disponuje velmi malým počtem vývodů, může nastat situace kdy nebude možné sloučit požadavky monitorovacího režimu s požadavky uživatelské aplikace, jak bylo naznačeno v minulém odstavci. Pro tento případ je na kitu JANUS několik zkratovacích propojek (jumperů), které umožní některé signály zcela odpojit.

4.6 Tlačítka a konfigurační propojky

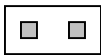


JP1 – připojuje na signál PTA3 funkci RESET (resetuje se pomocí signálu DTR na sériové lince). RESET je odpojitelný, neboť v USER MODE lze vývod 5 využít jako obecný signál vstupu/výstupu a připojení na DTR sériové linky tedy nemusí být žádoucí.

- JP2** – připojuje na signál PTA5 buď hodinový signál z krystalového oscilátoru, nebo jezdec trimru R7. S ohledem na to, že mikrokontroler vyžaduje v MONITOR MODE vnější taktování, musí být pro monitorování JP2 přepnut do polohy 1-2, čímž se na PTA5 připojí krystalový oscilátor. V uživatelském režimu vnější hodinový signál není potřeba a PTA5 lze využít pro jiné účely (např. pro připojení trimru R7)
- JP3** – připojuje komunikační linku na signál PTA0. Mikrokontroler v režimu MONITOR MODE komunikuje s nadřazeným počítačem prostřednictvím poloduplexní sériové linky emulované právě na signálu PTA0. Pro přechod do MONITOR MODE a následné spojení s nadřazeným počítačem musí být tato propojka zkratována. V uživatelském režimu lze PTA0 opět využít jako obecný signál.
- SW1** – krátkodobě (po dobu stisku) odpojuje napájecí zdroj od zbytku kitu. Tlačítko je určeno pro vyvolání „POR“ – power-on-reset. Nahrazuje ve své podstatě funkci RESET mikrokontroleru, protože jednak signál RESET nemusí být k dispozici, lze jej programově odpojit a využít vývod jinak, a za druhé je POR nezbytný po chybně zadaném heslu, kdy samotný signál RESET nestačí.
- SW2** – připojuje na signál PTA2/IRQ zvýšené napětí. Ve spolupráci s rezistory R1, R9, R10 pak rozhoduje o přepnutí do MONITOR MODE.

Příklady nastavení

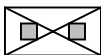
legenda:



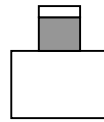
jumper rozpojen



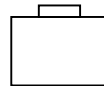
jumper zkratován



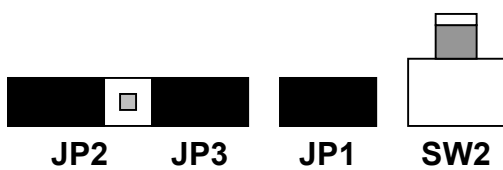
v diskutovaném případě na nastavení nezáleží



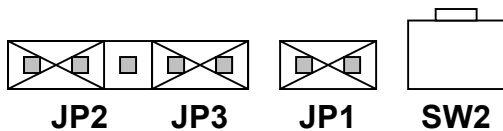
přepínač ve stavu A (vysunut)



přepínač ve stavu B (zasunut)



Výchozí nastavení pro práci v MONITOR MODE (debugging). JP2 v pozici 1-2 připojuje hodinový signál, JP3 komunikaci s PC, JP1 připojuje funkci RESET ze sériového portu a SW2 (stav A) přepíná MCU do MONITOR MODE přivedením zvýšeného napětí na IRQ.



USER MODE – Pokud je přepínač SW2 ve stavu B, mikrokontroler po startu přejde do USER MODE. Lze tedy takto snadno přepínat mezi MONITOR MODE a USER MODE pouhým přepínáním SW2.



Připojení trimru R7 na signál PTA5 (nelze za běhu MONITOR MODE, protože se tím odpojí hodinový signál).



Alternativní možnost připojení trimru R7 na signál PTA0 (nelze za běhu MONITOR MODE, protože není připojen komunikační signál).

5 Vývojový software

Kit vychází ze standardní filosofie MON08 a je tedy kompatibilní s veškerým software, který obsahuje podporu MON08. Z běžně dostupných software, které jsou volně k dispozici, je to např. WinIDE pro 68HC08QT/QY firmy PE Micro (www.pemicro.com), CodeWarrior SE HC08 verze alespoň 2.1.1 firmy Metrowerks (www.metrowerks.com) nebo Cosmic IDEA08 (www.cosmic.com).

5.1 P&E Micro - ICS08

Vývojový software ICS08 firmy P&E Micro zmiňovaný v příkladech i v této příručce obsahuje kompletní vývojové prostředí Assembleru pro všechny mikrokontrolery řady HC08. Kromě editoru a překladače obsahuje rovněž programátor, simulátor, obvodový simulátor a debugger.

Vývojové prostředí je k dispozici ke stažení na webových stránkách společnosti a je zdarma. www.pemicro.com

5.2 Metrowerks – CodeWarrior for HC08

Vývojový nástroj CodeWarrior pro HC08 je plnohodnotný zástupce vývojových prostředí CodeWarrior. Obsahuje všechny potřebné nástroje pro vývoj aplikací v jazyce C/C++ včetně programátoru a debuggeru. Obsahuje rovněž podpůrné nástroje jako MAKE, RCS apod.

CodeWarrior for HC08 lze objednat na webových stránkách společnosti. Ve verzi SE (special edition) je s omezením maximální velikosti kódu 4kB zdarma (registrace je nezbytná). Pro větší výsledný kód je již zpoplatněn. www.metrowerks.com

5.3 Cosmic – IDEA08

Vývojové prostředí IDEA08 firmy Cosmic obsahuje kompletní prostředí pro vývoj aplikací v jazyce C. Prostředí je oproti nástroji CodeWarrior výrazně jednodušší, neobsahuje řadu profesionálních funkcí, ale je snazší na první seznámení.

Prostředí IDEA08 je k dispozici na webových stránkách společnosti. Do velikosti výsledného kódu 4kB je zdarma (registrace nutná), pro větší aplikace je již zpoplatněn. www.cosmic-software.com

5.4 ImageCraft – ICC08

Vývojové prostředí ICC08 firmy ImageCraft obsahuje základní sestavu nástrojů pro vývoj aplikací v jazyce C. Prostředí obsahuje pouze nástroje pro kompilaci a její podporu (RCS apod). Vývojové nástroje jako programátor, debugger apod. neobsahuje, ale kompilátor vytváří soubory ve formátu ICS08. Pro ladění lze tedy použít část prostředí z ICS08. <http://www.imagecraft.com>

6 Příklady

Pro demonstraci možností mikrokontroleru NITRON je nezbytné mít k dispozici alespoň triviální vstupně-výstupní periferie. Kit nabízí uživateli až tři LED jako periferie výstupní a dvě tlačítka a jeden potenciometr jako periferie vstupní.

Tlačítka spínají do země (log.0) a jsou připojena k mikrokontroleru takto: JP4 (levé) spíná signál PTA4, JP5 (pravé) pak PTA2. Při čtení jejich stavu je třeba počítat s negativní logikou. Kit není osazen pull-up rezistory, pro korektní zjišťování stavu se předpokládá využití integrovaných pull-up rezistorů přímo v MCU samotném (viz registr PTAPUE).

Tři různobarevné LED v levé dolní části desky jsou zapojeny se společnou anodou, tedy svítí při log.0 na odpovídajícím výstupu a jsou připojeny na vývody MCU takto:

Červená LED (D4) je připojena na vývod PTB0, který je však pouze na 16ti vývodové verzi (pin 15). Lze ji využít jako obecný signál. Přímou připojená je však pouze k mikrokontroleru 68HC908QY (16ti vývodová verze). V případě práce s mikrokontrolerem 68HC908QT (osmivývodový) není červená LED přímo připojena a lze ji propojit s jakýmkoliv vývodem MCU pomocí zkratovacího drátku a precizních dutinek.

Žlutá LED (D5) je připojena na vývod PTA1 mikrokontroleru. Lze ji využít jakkoliv.

Zelená LED (D6) je připojena na vývod PTA3 mikrokontroleru. V USER MODE funguje jako běžná LED a lze jí ovládat signálem PTA3. V MONITOR MODE přechází PTA3 do funkce signalizace RESETu a LED svítí, když je MCU právě ve stavu RESET.

Trimr R7 je v MONITOR MODE odpojen (viz popis JP2), neboť již nezbyvá volný signál, kam by mohl být připojen. Jak bylo uvedeno výše, lze jej v USER MODE připojit buď na PTA5 pomocí JP2 (vstup AD3 A/D převodníku), nebo na PTA0 pomocí alternativní pozice jumperu mezi JP2 a JP3 (vstup AD0 A/D převodníku).

Na přiloženém CD je k dispozici několik příkladů v Assembleru a C pro seznámení se se základními funkcemi mikrokontroleru, kitu a ladicích nástrojů. Mimo jiné to jsou následující.

6.1 QT-first

Příklad qt-first.asm slouží k prvnímu seznámení s vývojovým kitem Janus a softwarovým vybavením pro ladění v ASM – ICS08.

Program příkladu nejprve provede základní inicializaci mikrokontroleru

```
Main:
    rsp                ; nastaveni SP na adresu $FF (zde konec RAM)
    clr               ; inicializace registru
    clrx
    mov                # $31, CONFIG1 ; Konfigurace MCU: Watchdogy COP a LVI vypnute

    mov                # $FF, PTA     ; Inicializace brany A, PORTA = FF
    mov                # $02, DDRA   ; Nastaveni smeru jednotlivych bitu brany A. Bit 1 je
    ; vystupni, ostatni bity vstupni
```

a následně rozblíká žlutou LED:

```
main_loop:
    bclr               1, PTA        ; Rozsvit LED (pin 6 - PTA1 - zlutá LED)
    bset               1, PTA        ; Zhasni LED
    bra                main_loop    ; skok zpět
```

6.2 QT-buttons-flipflop

Příklad qt-buttons-flipflop demonstruje základní možnosti vstupů a výstupů mikrokontroleru a filozofii jejich používání.

Program nejprve inicializuje mikrokontroler, vypíná nepotřebné funkce (Watchdogy, apod.) a nastavuje vstupně-výstupní brány tak, aby směry signálů odpovídaly použití na kitu:

```
; CPU initialization (registers, system configuration etc.)

    rsp                ; stack pointer reset
    clr               ; register init
    clrx
    mov    #$31,CONFIG1 ; MCU runs w/o LVI and COP support
    mov    #0,CONFIG2  ; IRQ off, RST off, internal OSC

; GPIO ports init. All signals are configured as INPUT except PTA1
;   being OUTPUT because of LED connected.

    lda    #$FF
    sta    PTA
    sta    PTB ; QY version has also port B
    mov    #$02,DDRA
    mov    #0,DDRB
    mov    #$14,PTAPUE ; Pushbuttons (PTA2,4) should be pulled-up
```

V první fázi jsou zinicilizovány registry, jsou nakonfigurovány základní vlastnosti mikrokontroleru¹¹ a v druhé fázi se nastavují směry I/O signálů. Jako výchozí směr je zvolen vstup, neboť tak se lze vyhnout kolizi signálů v případě, kdy je k mikrokontroleru připojena nějaká další periférie, se kterou program nepočítá¹².

Hlavní smyčka programu pak čeká na stisknutí jednoho z tlačítek simulujících vstupy R-S klopného obvodu:

```
main_loop:
    brclr  4,PTA,main_set ; wait for any button
                    ; Button PTA4 acts as SET
    brclr  2,PTA,main_reset ; Button PTA2 acts as RESET
    bra    main_loop      ; no button pressed
```

V případě, že uživatel stiskne tlačítko PTA4 (levé tl. – JP4), změní se vnitřní stav simulovaného klopného obvodu na log.1 a rozsvítí ze žlutá LED (PTA1 – D5):

```
main_set:
    lda    PTA ; Flip-flop SET = LED on
    and    #$FD ; boolean instructions bit control
    sta    PTA ; PTA1 (Y) on
    bra    main_loop ; go back to main loop
```

Pozn.: LED je rozsvěcena log. 0 na odpovídajícím výstupu. Pro demonstraci je log.0 vytvořena aritmetickou operací AND.

¹¹ Pomocí registrů CONFIG1 a CONFIG2 se nastavují základní vlastnosti mikrokontroleru (zdroj hodin, aktivace watchdogů, režimy se sníženou spotřebou apod.) Aby se zabránilo případné kolizi při chybě v programu, lze tyto registry zapsat po startu mikrokontroleru pouze jednou.

¹² S ohledem na to, že je mikrokontroler vyráběn na technologii CMOS, mají vstupy vysokou impedanci. Z tohoto důvodu se doporučuje nepoužité signály buď ošetřit (např. uzemněním přes vhodný rezistor), nebo je ve finální aplikaci nastavit jako výstupní. Zamezí se tím tak jejich eventuelnímu kmitání, které zvyšuje spotřebu a elektromag. vyzařování.

Stisknutí tlačítka PTA2 (pravé tl. – JP5) způsobí překlopení simulovaného R-S klopného obvodu do log.0 a LED je zhasnuta:

```
main_reset:                                ; Flip-flop RESET = Red LED on, Yellow LED off
    bset    1,PTA                            ; PTA1 (Y) off (bit instructions control)
    bra     main_loop                        ; go back to main loop
```

Zhasínání LED se provádí pomocí instrukce bitových operací.

7 Kit ve funkci externího ladicího adaptéru

Princip ladění v aplikaci byl u mikrokontrolerů HC08 vyvinut původně proto, aby bylo možné v mikrokontroleru HC08 již zapojeném v cílové aplikaci ladit či přehrávat firmware.

Využití této vlastnosti pro konstrukci jednoduchého vývojového kit je ve své podstatě pouze jednou z mnoha aplikací, což je patrné i z blokového schematu kitu. Aby mohl uživatel skutečně využívat funkcí ICD i ve své aplikaci, je kit vybaven konektorem MINIMON. Po uvolnění obou patič na kitu a správném nastavení propojek vznikne ladicí adaptér pro kompletní řadu mikrokontrolerů HC08.

Konektor MINIMON obsahuje všechny potřebné signály pro ladění aplikací postavených na libovolném zástupci řady HC08 v 5V provedení. Na straně aplikace samotné lze použít buď konektor obdobný a propojovat aplikaci s adaptérem jednoduchým plochým kabelem, nebo pokud to okolnosti vyžadují, použít konektor proprietární a tomu přizpůsobit i propojovací kabel. V každém případě je však nezbytné zajistit pro laděný mikrokontroler potřebné signály pro vstup a provoz monitoru, ať už propojením z MINIMON, nebo jinou formou.

7.1 Popis signálů konektoru MINIMON

VCC	1	2	CLK
GND	3	4	IRQ
/RESET	5	6	log.1
log.0 (S/P)	7	8	log.0
log.1 (DIV)	9	10	COMM

VCC – Napájení 5V – Napájecí vodič lze chápat jak ve smyslu napájení aplikace z kitu, tak obráceně, napájení kitu z aplikace. Je nezbytné zachovat jistá pravidla: Pokud bude aplikace napájena z kitu, musí být zajištěno dostatečné chlazení stabilizátoru U1 a nesmí být překročen jeho maximální proud. Aplikace by neměla mít další zdroj pro větev 5V, neboť tlačítko SW1 pak bude působit i na vypínání aplikace a na vybíjení jejích kondenzátorů přes R5.

Obráceně, pokud bude ladicí adaptér (kit) napájen přes MINIMON z aplikace, která bude mít vlastní zdroj, mělo by být další napájení kitu přes konektor J1 odpojeno. V takové

konfiguraci se nesmí používat tlačítko SW1 pro cyklování napájení, neboť by opět došlo ke zkratování zdroje aplikace přes rezistor R5.

GND – signálová a napájecí zem

/RESET – resetovací signál z počítače odvozený z RS232. Signál má negativní logiku (aktivní v log.0) a chová se jako otevřený kolektor. S ohledem na to, že je na něj též připojena LED D5, je do jisté míry vybaven i obvodem pull-up. Využití tohoto signálu v proprietárním řešení není nezbytné, ale značně zrychlí proces ladění.

S/P (log.0) – Signál přepínání sériového/paralelního nahrávání hesla při startu monitoru. S ohledem na to, že paralelní nahrávání hesla se využívá čistě pro testovací účely při výrobě obvodu, je na tomto signálu pouze log.0 odpovídající sériovému nahrávání. U řady zástupců HC08 není tento signál implementován a je předvolen sériový způsob nahrávání. Pokud tento signál na mikrokontroleru vyveden je, lze jej v aplikaci nahradit vhodným obvodovým řešením, které (nejméně) v okamžiku resetu zajistí přivedení log.0 na odpovídající vývod mikrokontroleru. Tento signál je potřebný pouze po dobu trvání signálu RESET, po startu monitoru již není vyžadován. Řešení v kitu zajišťuje log.0 na signálu připojením přes rezistor R16 (1k) na zem.

DIV – Signál přepínání dělicího poměru CLK/komunikační rychlost mezi 1/1024 (log.1), nebo 1/512 (log.0). Řada zástupců HC08 (včetně NITRONu samotného) tento signál nemá vyveden a dělicí poměr je pevně stanoven na 1/1024. U mikrokontrolerů, které tento signál požadují, jej lze místo propojení s monitorovacím konektorem nahradit obvodovým řešením, které v okamžiku resetu přivede na odpovídající vývod log.1, resp. log.0 pro dělicí poměr 1/1024, resp. 1/512. Tento signál je potřebný pouze po dobu trvání signálu RESET, po startu monitoru již není vyžadován. Řešení v kitu zajišťuje log.1 na signálu připojením přes rezistor R15 (1k) na Vcc.

CLK – Hodinový signál 9,8304MHz pro mikrokontroler. Pokud aplikace obsahuje vlastní generátor hodin, nemusí být tento signál využit, uživatel musí pouze přizpůsobit komunikační rychlost zvolené frekvenci generátoru hodin v aplikaci. S ohledem na to, že během ladění nelze jako zdroj hodin použít ani samotný krystal, ani vnitřní oscilátor, musí aplikace buď obsahovat autonomní generátor, nebo využít tohoto signálu. Signál je natolik silný, že jej lze přímo připojit do krystalového obvodu zapojeného podle doporučení v dokumentaci k MCU bez dalších zkratovacích propojek. Kit budí tento signál přímo výstupem z invertoru U2B.

IRQ – Po tomto signálu se do mikrokontroleru v cílové aplikaci přivádí zvýšené napětí Vtst jako jeden se signálů požadavku vstupu do monitoru. Zvýšené napětí lze opět realizovat přímo na aplikační desce, ale s ohledem na to, že je nezbytné pouze pro vlastní ladění, je nejjednodušší řešení propojit vývod IRQ mikrokontroleru s tímto signálem. Pozn.: Další logika aplikace, která vývod IRQ používá, musí akceptovat přítomnost zvýšeného napětí Vtst (Vcc+2,5V až Vcc+4V). Zvýšené napětí na vývodu IRQ je požadováno po celou dobu monitorovacího režimu. Kit přivádí na tento signál napětí 10V ze zdvojevače v U3 přes rezistor R13 (1k) a přepínač SW2.

log.0, log.1 – signály bez dalšího specifického významu, které jsou vyžadovány mikrokontrolerem pro přechod do monitoru. Lze je opět nahradit z vnitřní logiky aplikace, jako v případě S/P a DIV. Tyto signály jsou potřebné pouze po dobu trvání signálu RESET, po startu monitoru již nejsou vyžadovány. Řešení v kitu zajišťuje logické úrovně na signálech připojením přes rezistory R17 (1k) na zem resp. R16 (1k) na Vcc.

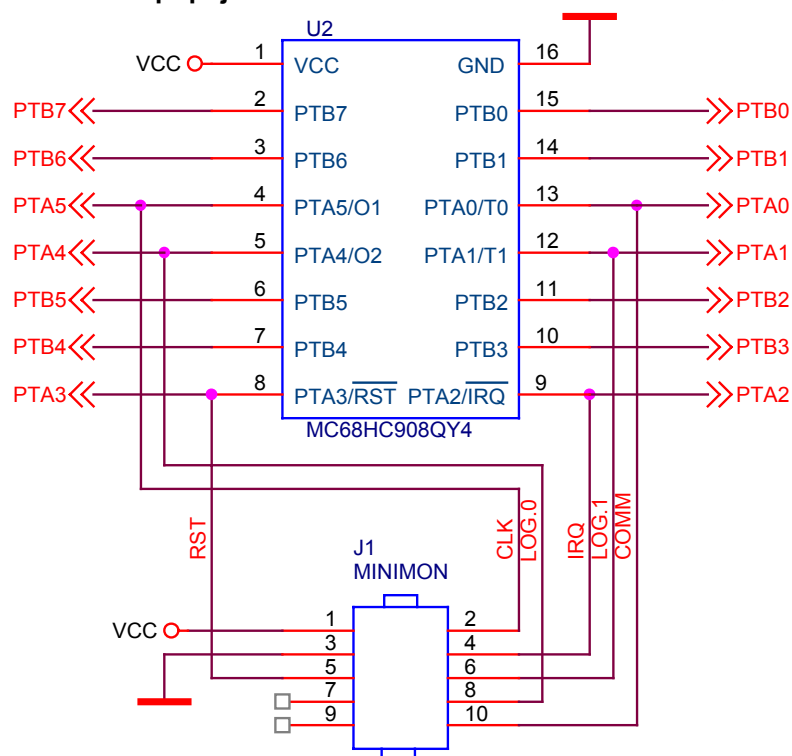
COMM – komunikační signál – pomocí tohoto signálu komunikuje počítač s mikrokontrolerem během monitorovacího režimu. Připojení tohoto signálu k mikrokontroleru je nezbytné. Tento signál je v provozu po celou dobu monitorování a

aplikace jej po dobu monitorování musí respektovat. Kit realizuje tento signál jako otevřený kolektor (D2) s pull-up rezistorem (R1 – 1k) na Vcc.

Mapování signálů na vývody některých mikrokontrolerů z řady HC08

	NITRON	KX8	GP32	GR8
/RESET	PTA3	PTB7	RST	RST
S/P	-	PTA1	PTA7	PTA1
DIV	-	-	PTC3	-
CLK	PTA5	PTB6	OSC1	OSC1
IRQ	PTA2	IRQ1	IRQ	IRQ
log.1	PTA1	PTB0	PTC0	PTB0
log.0	PTA4	PTB1	PTC1	PTB1
COMM	PTA0	PTA0	PTA0	PTA0

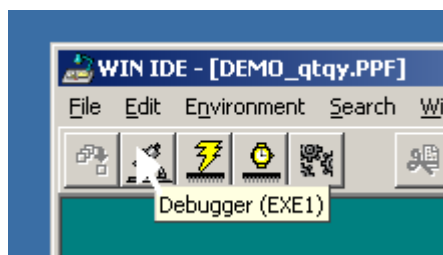
Příklad připojení MINIMON k NITRONu MC68HC908QY




8 Přílohy

8.1 Nastavení nástrojové lišty ICS08

Vývojové prostředí pro mikrokontrolery NITRON má ve verzi 1.05 a nižší chybně nakonfigurovaná tlačítka pro spouštění ICS, programátoru, debuggeru a simulátoru.



Pokud naleznete pod tlačítkem  funkci debugger (viz on-line nápověda), máte verzi, kde jsou tlačítka nastavena špatně. Pro správné nastavení proveďte následující úpravy. V menu Environment zvolte Setup environment a v záložkách EXE1-EXE3 upravte nastavení takto:

EXE1: Type: In circuit simulator

EXE path: C:\pemicro\ics08qtqyz\ICS08qtqyZ.EXE
(cesta dle umístění instalace)
Options: %FILE.S19%

EXE2: Type: Flash Programmer

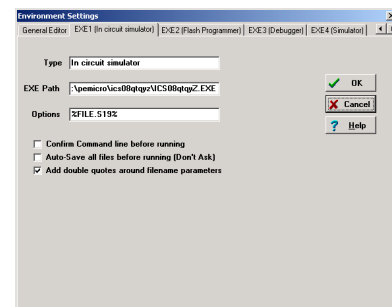
EXE path: C:\pemicro\ics08qtqyz\PROG08SZ.EXE
Options:

EXE3: Type: Debugger

EXE path: C:\pemicro\ics08qtqyz\ICD08SZ.EXE
Options: %FILE.S19%

EXE4: Type: Simulator

EXE path: C:\pemicro\ics08qtqyz\ICS08qtqyZ.EXE
Options: %FILE.S19% /S



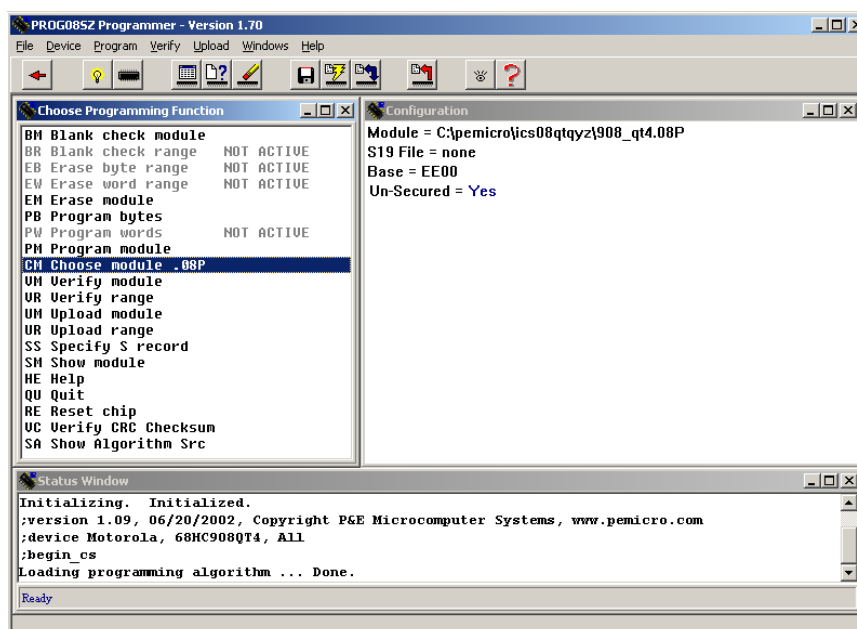
Po uložení této úpravy budou již tlačítka nastavena správně a popis bude odpovídat.

8.2 Programování a mazání FLASH, pomoc při ztrátě vstupního hesla

Jak již bylo zmíněno výše, mikrokontrolery HC08 chrání obsah paměti FLASH před zneužitím pomocí 64 bitového hesla. Vývojový systém ICS08 si udržuje databázi hesel včetně hesla aktuálního, avšak i přes to může dojít ke ztrátě hesla pro vstup do daného MCU. Jedinou možností, jak toto vyřešit, je smazání celé paměti FLASH včetně hesla a pochopitelně i chráněného programu.



Pro vymazání a/nebo naprogramování paměti FLASH zvolte v hlavním okně ICS08 tlačítko „Programmer“, popřípadě spustit přímo PROG08SZ (pozn.: Před spuštěním programátoru musíte uvolnit sériový port, ke kterému je kit připojen, jinak programátor vyhlásí chybu, že kit/port je obsazen).



Pokud není aktuální vstupní heslo do mikrokontroleru známé, objeví se po startu programátoru okno „Attempting to contact target and pass security...“, ve kterém je třeba zaškrtnout „Ignore security failure“, aby programátor dovolil pracovat s mikrokontrolerem i přes neznámé heslo (aby bylo možné vymazat celou paměť FLASH).



Poté se již programátor nastartuje s tím, že některé funkce nejsou dostupné.

Pro vymazání obsahu FLASH je v okně programovacích funkcí k dispozici povel „EM Erase module“. Stavové okno informuje o průběhu mazání.

Nyní, pokud nebylo přístupové heslo známé (v konfiguračním okně je Un-secured = No), nabídne programátor možnost restartu mikrokontroleru s novým (již známým – prázdným) heslem. V tomto momentě je nutné provést na mikrokontroleru POR, tedy mikrokontroler vypnout a zapnout (na kitu je k tomuto určené tlačítko SW1) a následně restart potvrdit. Programátor zaznamená do konfiguračního souboru nové (prázdné) heslo a připojí se k mikrokontroleru.

Pro naprogramování paměti FLASH se musí nejprve zvolit soubor, jehož obsah se bude do MCU nahrávat pomocí volby „SS Specify S record“. S-record je soubor s příponou S19 obsahující binární data přeloženého programu ve formátu Motorola HEX.

Pozn.: Programátor nepřebírá z vývojového prostředí jméno aktuálně otevřeného souboru, takže je jednak nutné jej zde specifikovat a dále, pokud uživatel začne pracovat na jiném software, musí zde jméno příslušného souboru změnit.

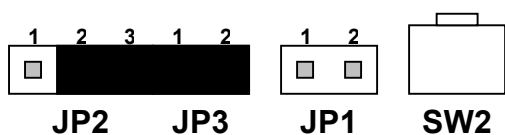
Volba „PM Program Module“ spustí naprogramování paměti FLASH. Následně lze zkontrolovat, zda-li proběhl zápis v pořádku, volbou „VM Verify Module“.

8.3 Kompatibilita s DEMO kitem Motorola M68DEMO908QT4

Z důvodů možnosti využití příkladů a materiálů souvisejících s demonstračním kitem Motoroly pro NITRON je JANUS konstruován tak, aby na něm bylo možné DEMO kit plně emulovat.

Z hlediska software obsahuje „nenaprogramovaný“ mikrokontroler NITRON v DEMO kitu krátký program, který umožní přechod do MONITOR MODE bez potřeby dříve zmiňovaných podmínek (požadovaných log. hodnot na většině signálů) a bez potřeby externího zdroje hodinového signálu. Pro „emulaci“ DEMO kitu je třeba nejprve přepnout mikrokontroler do MONITOR MODE standardním způsobem (viz nastavení jumperů) a tento tzv. USER-MONITOR do mikrokontroleru nahrát jako běžný program. Zdrojový kód tohoto USER-MONITORu lze najít v aplikační zprávě AN2305 na CD či webu Motoroly.

Po nahrání USER-MONITORu do MCU se musí jumpery nastavit takto:



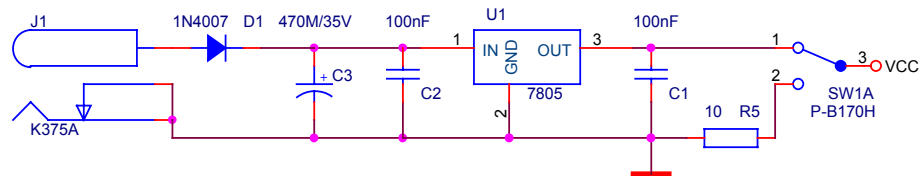
JP2 je přepnut do polohy 2-3, propojuje tedy trimr R7 na signál PTA5. JP3 je zkratován – připojuje komunikaci na signál PTA0. SW2 je v poloze B (USER MODE), neboť v tomto stavu nemá zvýšené napětí na PTA2/IRQ smysl. Pro 100% kompatibilitu s DEMO

kitem by měl být rozpojen i JP1 – funkce RESETu pomocí signálu DTR není v DEMO kitu implementována. Zkratování JP1 v podstatě obsadí signál PTA3 jako vstup signálu DTR z RS232.

8.4 Zapojení kitu

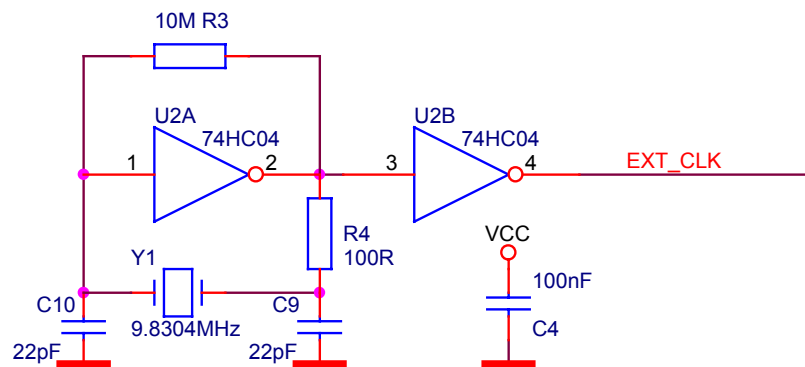
Zapojení kitu sestává z několika logických celků: napájecí zdroj, oscilátor, interface RS232, patice pro mikrokontroler(y) a jednoduché periferie.

8.4.1 Napájecí zdroj



Kit je vnitřně napájen napětím 5V, které vytváří stabilizátor U5 (7805) ve standardním zapojení. Vstup napájení je řešen obvyklým souosým konektorem J1 a je chráněn diodou D1 proti přepólování. Výstup 5V ze stabilizátoru je veden přes momentové tlačítko SW1 (tlačítko bez aretace), které umožňuje krátkodobě vypnout napájení pro funkci resetu a POR¹³ mikrokontroleru. Rezistor R5 slouží pro úplné vybití blokovacích kondenzátorů na kitu, stisknutím SW1 se odpojí napájení kitu a přes R5 se zkratuje Vcc a GND. Na desce kitu je zdroj umístěn v levém horním rohu. Jedná se o součástky: napájecí konektor U4, ochranná dioda proti přepólování D6, blokovací kondenzátory C9, C10 a C11, stabilizátor U5, vybijecí odpor R14 a signalizační dioda napájení D3 s R6.

8.4.2 Oscilátor

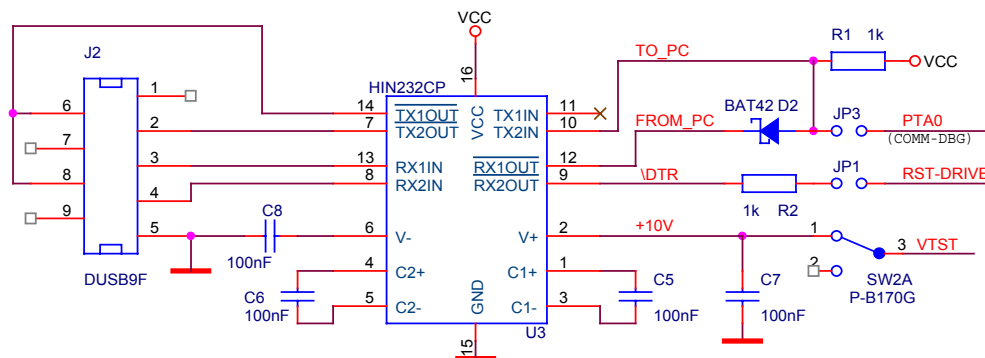


Generování kmitočtu 9,8304MHz je z důvodu ceny řešeno krystalovým oscilátorem s 74HC04 (U2) v klasickém zapojení s invertorem U2A (Y1, R3, R4, C9 a C10). Hodinový signál je z oscilátoru veden na druhý inverter (U2B), kde se dotvaruje, a dále přes zkratovací propojku JP2 (pin1) na odpovídající vstup mikrokontroleru. Propojka JP2 umožňuje v případě potřeby hodinový signál od mikrokontroleru odpojit. Hodinový signál je

¹³ POR – Power On Reset – start mikrokontroleru po zapnutí napájení. Některé vnitřní funkce mikrokontroleru (např. nový pokus zadání hesla) se nulují pouze při POR, „běžný“ RESET nestačí.

z výstupu U2B veden rovněž na vývod CLK konektoru MINIMON. Na kitu se oscilátor nachází v horní části desky vpravo od zdroje.

8.4.3 Interface RS232

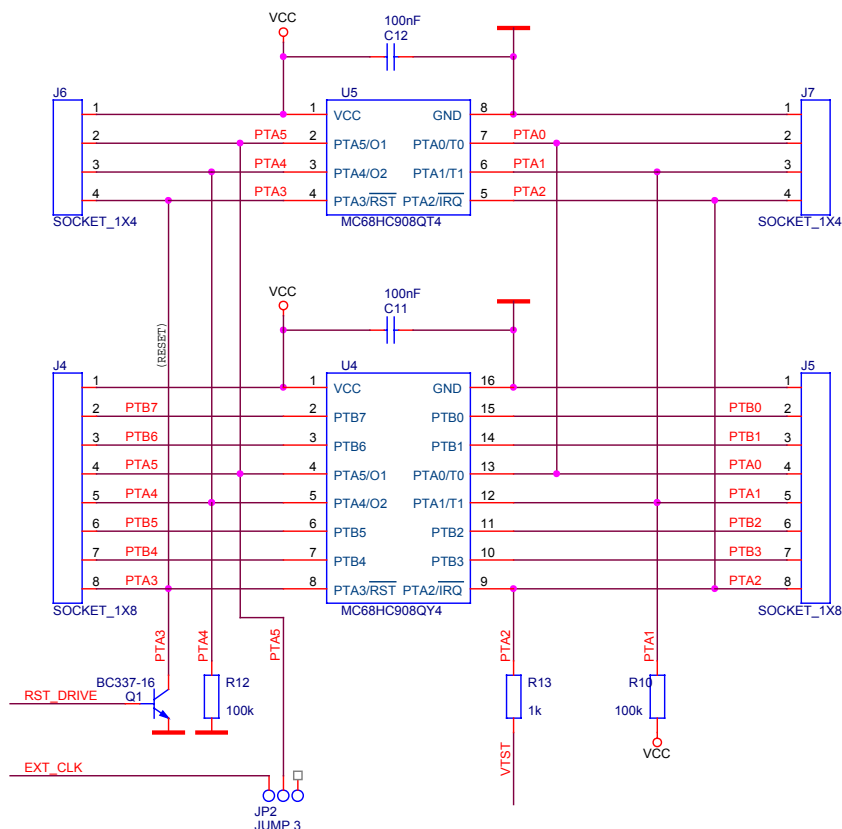


Pro spojení s PC slouží interface RS232 vycházející ze standardního zapojení obvodu MAX202 (U1) nebo obdobných (J1 – konektor DSUB9F pro připojení sériového kabelu, C2, C3, C4, C5 – kapacity pro napěťový měnič). Sériový datový signál (FROM_PC) je veden do obvodu drátového AND (R1 a D1), kde se vytváří poloduplexní sériový signál COMM_DBG. Ten je jednak připojen na vývod mikrokontroleru (obvykle PTA0) a zároveň je veden přes vysílač TX1 obvodu U3 zpět do PC. Toto zapojení umožní obsadit obousměrnou komunikací pouze jediný vývod mikrokontroleru. Vedlejším efektem je v podstatě generátor echa, neboť do PC se vrací veškerá data z COMM_DBG, ať už pochází z mikrokontroleru nebo z PC samotného.

Ze spojení s PC se dále využívá signál DTR pro resetování mikrokontroleru z počítače. Aktivací DTR (otevřením kom. linky) se RESET mikrokontroleru uvolňuje, deaktivací DTR se mikrokontroler RESETuje. Signál je veden přes R2 a Q1 tvořící invertor s otevřeným kolektorem na RST mikrokontroleru a na vývod RST konektoru MINIMON. V době ladění je na signálu DTR log.0 (užívá se negativní logiky).

Kromě napěťových uprav signálů TxD, RxD a DTR se obvodu U3 využívá i jako generátoru napětí 10V. Z tohoto napětí se pak vytváří Vtst (pomocí R13) – zvýšené napětí sloužící jako jeden ze signálů pro přepnutí MCU do MONITOR MODE (SW2, R13). S ohledem na to, že toto zvýšené napětí musí být k dispozici prakticky zároveň s napájením, je nezbytné, aby náběh zdvojovače v U3 byl co nejrychlejší. Prakticky vyhoví pouze verze RS232 budiče s 100nF kondenzátory. „Klasický“ typ s kondenzátory 1uF nabíhá příliš pomalu a napětí Vtst není na mikrokontroler přivedeno včas.

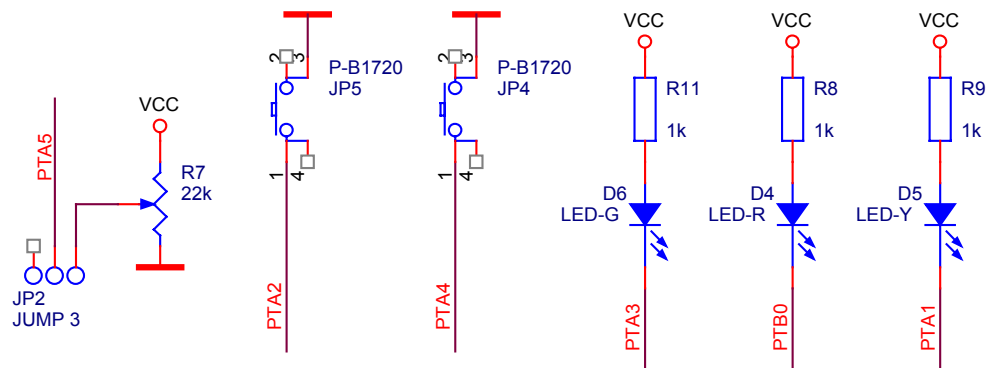
8.4.4 Patice pro mikrokontrolery



Dvě patice pro mikrokontrolery NITRON na kitu nabízí možnost ladění jak v 8mi vývodové (U5 – 908QT), tak i v 16ti vývodové (U4 – 908QY) verzi. Signály však nejsou nijak odděleny, takže lze zároveň používat pouze jednu patici a jeden mikrokontroler.

K paticím přísluší několik součástek, které zajišťují přepínání USER a MONITOR MODE a správné resetování. Jedná se o rezistory R10 a R12, které zajišťují správné log. hodnoty pro start mikrokontroleru v monitorovacím režimu, a dále o resetovací obvod R2 a Q1 zmíněný už v části rozhraní RS232. Signalizační LED D4 je využívána sdíleně buď jako signalizace RESET, nebo jako jednoduchá periferie.

8.4.5 Jednoduché periferie



Pro snazší pochopení problematiky HC08 nabízí kit několik jednoduchých periferií: tři LED¹⁴ (D2, D3, D4), dvě tlačítka JP5 a JP6 a potenciometr R8. LED jsou zapojeny se společnou anodou a jejich buzení odpovídá negativní logice, LED tedy svítí při log.0 na příslušném výstupu mikrokontroleru.

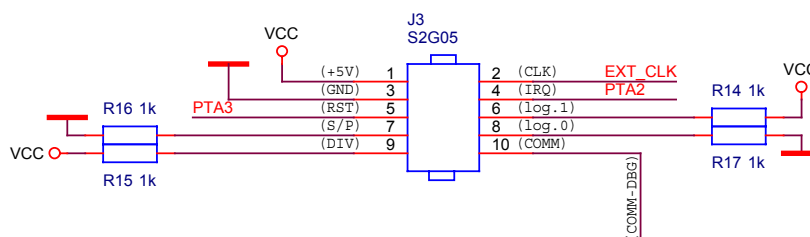
Tlačítka jsou zapojena „proti zemi“, stisknuté tlačítko se tedy projeví log.0 na příslušném vstupu. S ohledem na to, že jsou příslušné signály vybaveny vnitřními pull-up rezistory (viz registr PTAPUE), nejsou ze externí potřeba.

Připojení periferií

Červená LED	PTB0 (pouze na 16ti vývodové verzi MCU)
Žlutá LED	PTA1
Zelená LED	PTA3/RESET
Tl. JP4 (levé)	PTA4
Tl. JP5 (pravé)	PTA2
Trimr R7	PTA5/ADC3 (Pokud je JP2 v poloze 2-3)

¹⁴ Všechny tři LED jsou přímo připojeny pouze k 16ti vývodové verzi mikrokontroleru (QY). Při použití osmivývodové verze (QT) lze červenou LED připojit na libovolný vývod mikrokontroleru zkratovacím drátkem zasunutým do pinu 15 patice U4 a do dutinky odpovídající požadovanému signálu.

8.4.6 MINIMON



Pro použití kitu Janus jako ladicího/programovacího adaptéru je k dispozici konektor MINIMON (J3), který obsahuje všechny nezbytné signály pro ladění v cílové aplikaci.

Skladba signálů na konektoru MINIMON je navržena tak, aby bylo konstrukční řešení aplikace ovlivněno minimálně (detailní popis signálů viz kapitola „Kit ve funkci externího ladicího adaptéru“).

Jak je popsáno v uživatelském manuálu, přítomnost všech signálů je vyžadována pouze po dobu RESETu a většinu z nich lze krátce po uvolnění RESETu odpojit. Oproti tomuto úplnému obvodovému řešení adaptéru MINIMON je zde vynechána část odpojování nepotřebných signálů, všechny signály jsou na vývody přivedeny trvale.

Jelikož je kit Janus ve své podstatě sloučením ladicího adaptéru a jednoduché aplikace sestávající z mikrokontroleru NITRON a několika periférií (LED, tlačítka, trimr), jsou všechny signály potřebné pro provoz mikrokontroleru HC08 v MONITOR MODE k dispozici. MINIMON konektor je pouze v ucelené podobě zpřístupňuje uživateli.

8.5 Rozpis součástek na desce a schéma kitu JANUS

