

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

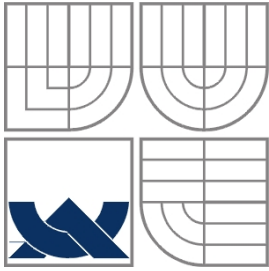
PROGRAMOVÁNÍ MIKROKONTROLERŮ V SYSTÉMU
GNU/LINUX

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

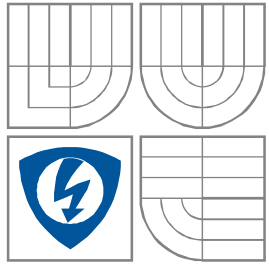
AUTOR PRÁCE
AUTHOR

Martin Jančík

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

PROGRAMOVÁNÍ MIKROKONTROLERŮ V SYSTÉMU GNU/LINUX

MICROCONTROLLER PROGRAMMING IN GNU/LINUX

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Martin Jančík

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Zbyněk Fedra, Ph.D.

BRNO, 2008

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Martin Jančík
Bytem: Podolí 336, Podolí, 664 03
Narozen/a (datum a místo): 2. dubna 1986 v Brně

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Programování mikrokontrolerů v systému GNU/Linux

Vedoucí/ školitel VŠKP: Ing. Zbyněk Fedra, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 6. června 2008

.....
Nabyvatel

.....
Autor

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Programování mikrokontrolerů v systému GNU/Linux jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 6. června 2008

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Zbyňku Fedrovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 6. června 2008

.....
podpis autora

Abstrakt

Tato práce se zabývá problematikou programování mikrokontrolérů Atmel AVR v systému GNU/LINUX. Tato práce ukazuje široké možnosti programování mikrokontrolérů systému GNU/LINUX nejen pro použití programování, ale i pro nahrávání programu do 8 bitový mikrokontroléry.

Práce se zaměřuje na software, který má grafické uživatelské rozhraní, případně software s možností ovládání pomocí konzole. Dále jsou zde popsány způsoby programování mikrokontrolérů Atmel AVR, obsahuje přehled programů pro programování mikrokontrolérů AVR v systému Linux, návrh jednoduché praktické aplikace s využitím mikrokontroléru – zobrazení systémových informací z PC na externím LCD displeji.

Klíčová slova:

Linux, AVR, C, GCC

Abstrakt

This project is dealing with programming problems of the micro-controllers Atmel AVR in the system GNU/LINUX. It demonstrates wide programming possibilities micro-controllers of the system GNU/LINUX not only for the programming tasks, but for upload program to 8-bits controllers as well.

The Project is aimed at the software, equipped with a graphic user's interface, conceivably at a software enabling its control from a console. Further more, there are programming methods for the micro-controllers Atmel AVR described here, a survey of programs for micro-controllers AVR within the system LINUX is comprised there, as well as the proposal of some simple serviceable application using the micro-controller – displaying of the system information from a PC on an external LCD screen.

Keywords:

Linux, AVR, C, GCC

JANČÍK, M. *Programování mikrokontrolerů v systému GNU/Linux*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 39 s. Vedoucí bakalářské práce Ing. Zbyněk Fedra, Ph.D.

Obsah:

| | |
|-------------------------------------------------------------------------|-----------|
| 1. Úvod: | 5 |
| 2. Mikrokontroléry AVR: | 5 |
| 3. Metody programování | 7 |
| 3.1 Paralelní:..... | 7 |
| 3.2 Sériové:..... | 7 |
| 3.2.1 ISP..... | 7 |
| 3.2.2 JTAG..... | 8 |
| 4. Přehled nástrojů dostupných pod GNU/Linux | 8 |
| 4.1 AVR Cross Development Kit..... | 8 |
| 4.1.1 Kompilátor | 8 |
| • 4.1.1.1 AVR - GCC | 9 |
| 4.1.2 Software pro nahrávání programů do mikrokontrolérů | 9 |
| • 4.1.2.1 UISP | 9 |
| • 4.1.2.2 AVRDUDE..... | 11 |
| • 4.1.2.3 Avrp..... | 13 |
| • 4.1.2.4 AVRprog | 13 |
| 4.1.3 Knihovny pro jazyk C AVR mikrokontrolérů | 14 |
| • 4.1.3.1 AVR Libc | 14 |
| • 4.1.3.2 Avr-binutils..... | 14 |
| 4.1.4 Simulátory..... | 15 |
| • 4.1.4.1 Simulavr | 15 |
| • 4.1.4.2 AVaRICE | 16 |
| • 4.1.4.3 Avr-GDB | 16 |
| 4.1.5 Assembler | 17 |
| • 4.1.5.1 Avr-avra..... | 17 |
| • 4.1.5.2 Avr-tavrsm..... | 17 |
| 4.2 Vývojové prostředí IDE | 18 |
| 4.2.1 Vývojové prostředí pracující nativně pod systémem GNU/Linux | 18 |
| 4.2.1.1 KontrollerLab | 18 |
| 4.2.1.2 PonyProg 2000 | 18 |
| 4.2.1.3 AVR-Eclipse | 20 |
| 4.2.2 Vývojové prostředí fungující pomocí programu WINE | 20 |
| 4.2.2.1 AVR studio verze 3.56:..... | 20 |
| 4.2.2.2 AVR studio verze 4.12:..... | 21 |
| 4.2.2.3 Visual Micro Lab 3.12 for AVR: | 22 |
| 4.3 Další Simulátory:..... | 22 |
| 4.3.1 Avrora | 22 |
| 5. Návrh Demo úlohy | 23 |
| 5.1 Použitý software:..... | 23 |
| 5.1.1 KontrollerLab..... | 23 |
| 5.1.2 Simulátor Visual Micro Lab 3.12 | 25 |
| 5.2 Blokové schéma navrhované demo úlohy..... | 26 |
| 5.3 Mikrokontrolér ATmega 8 | 27 |
| 5.4 Převodník úrovně RS232 / UART – MAX 232 | 28 |
| 5.5 LCD displej s řadičem HD44780 | 29 |
| 5.6 Programátor ISP | 29 |
| 5.7 Napájecí zdroj | 30 |

| | |
|------------------------------------------------------------------|-----------|
| 6. Firmware demo úlohy..... | 31 |
| 6.1 Jednotka USART a její nastavení..... | 31 |
| 6.2 Ovládání LCD displeje s řadičem HD 44780..... | 32 |
| 6.3 Programovatelné konfigurační propojky..... | 33 |
| 6.4 Program mikrokontroléru..... | 33 |
| 6.5 Skript posílající systémové informace | 33 |
| 7. Porovnání řešení se stávajícími možnostmi na UREL..... | 35 |
| 7.1 Porovnání Software..... | 35 |
| 7.2 Podpora Hardware..... | 36 |
| T8. Závěr:..... | 37 |

Rejstřík tabulek:

| | |
|-----------------------------------------------------------------------|----|
| Tab 1. Číslování a význam vývodu LCD displejů s řadičem HD44780 | 29 |
| Tab 2. Zapojení programovacího kabelu ISP | 30 |
| Tab 3. Zapojení konektoru USB typ A | 30 |
| Tab 4. Hodnoty předděliče UBRR pro krystal 8 MHz [převzato 18] | 32 |
| Tab 5. Nastavení vnitřního RC oscilátoru ATmega8 | 33 |

Rejstřík obrázků:

| | |
|-----------------------------------------------------------------------------------------------------|----|
| Obr 1. Bloková struktura mikrokontroléru AVR, ATmega 8 [převzato 18] | 6 |
| Obr 2. Simulavr v systému Mandriva 2007 | 15 |
| Obr 3. Program PonyProg 2000 v systému Mandriva 2007 – ukázka výběru zařízení | 19 |
| Obr 4. KontrollerLab v systému Mandriva 2007 | 24 |
| Obr 5. KontrollerLab (Mandriva 2008), možnost nastavení programovatelných propojek | 25 |
| Obr 6. Visual Micro Lab 3.12 v systému Mandriva 2007 pomocí Wine | 26 |
| Obr 7. Blokové schéma zobrazovače systémových informací z PC | 26 |
| Obr 8. Aplikační zapojení obvodu MAX 232 [převzato 19] | 28 |
| Obr 9. Struktura asynchronního rámce se dvěma stop bity a sudou paritou a sedmi datovými bity | 32 |
| Obr. 10 Vývojový diagram programu mikrokontroléru | 34 |

Seznam příloh:

| | |
|------------|----------------------------------------|
| Příloha A: | Schéma zapojení |
| Příloha B: | Tabulky zapojení konektorů |
| Příloha C: | Desky plošných spojů |
| Příloha D: | Osazovací plán pohled ze strany bottom |
| Příloha E: | Seznam použitých součástek |
| Příloha F: | Zdrojový kód skriptu v počítači |
| Příloha G: | Obrázek funkčního zařízení |

1. Úvod:

V dnešní době je vyvíjeno stále více systémů v kterých jsou hojně rozšířeny mikrokontroléry. Tak velice usnadňují celou realizaci a zjednodušují návrh a případné úpravy již navržených systémů. Tento vědní obor je stále více rozšířenější. Začali se také rozšiřovat jiné alternativy operačních systémů a začínají se stále více prosazovat. Mezi tyto zástupce operačních systémů také patří GNU/Linux, který již dávno není určen jen pro nasazování v serverových aplikacích ale také se začíná prosazovat na pozici desktopových operačních systémů.

Cílem této bakalářské práce je prozkoumání možností programování, simulace a nahrávání programu do mikrokontrolérů firmy ATMEL rodiny AVR. Některé z těchto nástrojů byly vyzkoušeny. Všechny tyto programy byly vyzkoušeny v systému Mandriva 2007, případně novější verze Mandriva 2008. S nástroji dostupnými pod GNU/Linux byla navržena jednoduchou ukázkovou úlohu pro mikroprocesor. Je zde popsán zvolený software a zmíněno o jeho nastavení a popisem hardware, který byl použit pro praktickou ukázkou.

Dále byl proveden praktický návrh a realizaci celé navržené demo úlohy, včetně praktické ukázky. Také byl popsán zvolený mikrokontrolér ATmega8, který je používán v praktické ukázce. Bylo porovnáno řešení se stávajícími možnostmi využívání v předmětu BMPT.

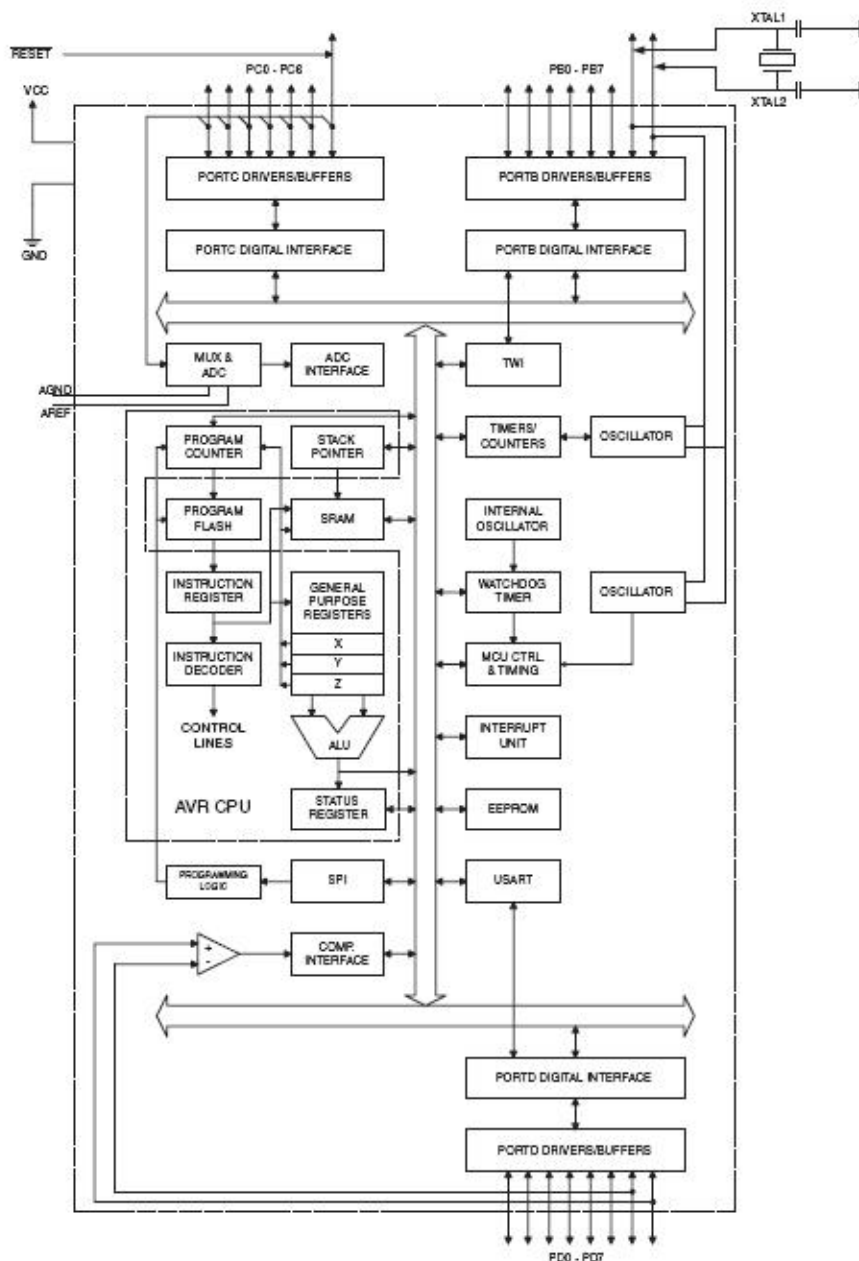
Nevýhodou této práce je, že není aktuální z důvodu rychlého vývoje aplikací v rámci projektů GNU.

2. Mikrokontroléry AVR:

Rodina mikrokontrolérů AVR s RISC architekturou je velmi zdařilým výsledkem architektury mikrokontrolérů přizpůsobených jazyku C.

AVR se ovšem se 120-ti instrukcemi u výkonnějších typů velmi přibližuje CPU s architekturou CISC. Všechny ostatní charakteristiky, jako stejná bitová šířka instrukcí a zpracování instrukcí v jednom hodinovém cyklu jsou naopak typické vlastnosti architektury RISC. Proto je možno říci, že rodina AVR využívá výhod obou architektur: výkonnosti RISC s rozsahem instrukční sady blížící se CISC. Zdá se, že to je klíčem k moderním architektuám mikrokontrolérů. Tak je možno udržet nízkou velikost programu, při větší rychlosti zpracování. Instrukční kód má šířku 16 bitů a je tedy dostatečně velký, aby mohl v jednom instrukčním slově pojmut jak instrukci, tak operand.

Jednostupňové zřetězení instrukcí (pipeline) podporuje zpracování instrukčního slova, přečtení, interpretaci a provedení v jediném hodinovém cyklu. 32 univerzálních pracovních registrů, které všechny mohou pracovat jako 8-bitový akumulátor, umožňuje vyhnout se zbytečnému přemísťování obsahu registrů u aritmetických operací.



Obr 1. Bloková struktura mikrokontroléru AVR, ATmega 8 [převzato 18]

Základní charakteristika ATMEL AVR:

- Architektura typu Harvard a RISC
- 32 identických 8-bitových registrů pro všeobecné použití, které jsou všechny použitelné jako akumulátor
- Ukazatel na zásobník (stack pointer) a tři adresní ukazatele X, Y, Z, které jsou vytvářeny z registrů R26/27=X, R28/29=Y, R30/31=Z
- Pro registry ukazatelů Y a Z je možná indexace polí pomocí 6-ti bitové relativní adresy (displacement)
- Všechny registry jsou adresovatelné prostřednictvím standardních paměťových přístupů
- Bitové adresování pro všechny registry
- S výjimkou čtyř 32bitových instrukcí mají všechny délku 16 bitů
- Lineární adresní prostor pro datovou paměť navzdory funkčnímu rozdělení

- Interní paměť: FLASH pro program
SRAM pro registry, oblast I/O a data
EEPROM pro ukládání dat
 - Externí paměť: RAM rozšiřitelná v závislosti na typu mikrokontroléru
 - 64 paměťových míst vstupu/výstupu (prostřednictvím instrukce I/O nebo adresní přístup 0x20 až 0x5F)
 - Ochrana před poklesem napájecího napětí (brown-out)
 - Žádné další dělení hodinového cyklu krystalu
 - Mnoho funkčních bloků a periférií podle typu mikrokontroléru
- [1]

3. Metody programování

Mikrokontroléry AVR firmy Atmel lze programovat buď paralelním nebo sériovým způsobem.

3.1 Paralelní:

Paralelní programování disponuje více možnostmi (lze programovat programovatelné propojky v obvodu např. SPIEN, FSTRT, BODLEVEL apod.). Oproti sériovému programování je komplikovanější. Programovaný mikrokontrolér je nutné naprogramovat v programátoru, po té vyjmout z programovací patice a vložit do zvolené aplikační desky. Tento způsob velmi komplikuje a zpomaluje celý proces vývoje aplikace, pokud se nepoužívají simulační programy. Výrobci většinou vytváří vývojové koity pro testování programů.

3.2 Sériové:

3.2.1 ISP

ISP (*In-System Programming*) – metoda, umožňuje programování obvodů přímo v zařízení pomocí vyhrazených vodičů určených pro komunikaci při programování. Obvykle není potřeba vyšší napájecí napětí než je jmenovité napájecí napětí programovaného obvodu. Tímto způsobem lze také programovat programovatelné propojky mimo propojky SPIEN, která povoluje a zakazuje sériové programování. Továrně je propojka propojená a je možné programovat mikrokontrolér pomocí sériového programování. Běžně používaná rozhraní pro ISP jsou

SPI (Serial Peripheral Interface) - rozhraní pro sériovou synchronní komunikaci,

JTAG (Joint Test Actoin Group) - rozhraní pro diagnostiku a testování integrovaných obvodů,

UART (Universal Asynchronous Receiver Transmitter) - rozhraní pro sériovou asynchronní komunikaci.

Nejčastěji se pro programování používá SPI rozhraní. K propojení AVR mikroprocesoru s programátorem slouží kromě napájení čtyři datové signály:

- **MOSI** (*Master Out – Slave In*) – sériový vstup dat do mikroprocesoru
- **MISO** (*Master In – Slave Out*) - sériový výstup dat z mikroprocesoru
- **SCK** (*Serial clock*) – sériové hodiny (synchronizace komunikace)
- **/RESET** – nulování mikroprocesoru

Metodou ISP nelze nahrávat (aktualizovat) firmware zařízení bez toho, aby nedošlo k přerušení vykonávání jeho funkce (programovaný obvod je nutné přepnout do režimu programování).

3.2.2 JTAG

JTAG(*Joint Test Action Group*) – jedná se o modifikaci ISP. Rozhraní JTAG je specifikováno normou IEEE 1149.1. Toto rozhraní umožňuje komunikovat s mikrokontrolérem během ladění programu. Umožňuje skenování vnitřních periférií jako jsou přerušení, watchdog zjišťovat obsahy registrů a jejich modifikaci apod. Výhodou této metody je možnost krokování programu přímo na čipu procesoru. Mikrokontrolér je možné přímo ovládat z AVR studia firmy Atmel. Paměť kódu programu lze zapisovat a číst přímo z aplikace (speciální instrukce mikroprocesoru). K aktualizaci firmwaru lze použít libovolného rozhraní a tak nahrání aktuální verze programu po lokální síti či z Internetu není v současné době neřešitelný problém.

[2]

4. Přehled nástrojů dostupných pod GNU/Linux

4.1 AVR Cross Development Kit

Jedná se o projekt vytvářený pro programování mikrokontrolérů v operačním systému Linux a BSD. Podporuje programování mikrokontrolérů v assembleru a také jazyku C. V tomto kitu jsou nástroje pro psaní zdrojových kódů, kompilátory i ladící programy. Používá jako rozhraní X window (X11), ale některé části kitu pracují v konzolovém režimu. Pro práci s mikrokontroléry v konzoli potřebujeme alespoň tyto balíčky:

- cdk-avr-base
- cdk-avr-binutils
- cdk-avr-gcc
- cdk-avr-libc
- cdk-avr-uisp

4.1.1 Kompilátor

Kompilátory jsou nástroje překládající algoritmus zapsaný ve vyšším programovacím jazyce do jazyka strojového, nebo spíše do strojového kódu.

• 4.1.1.1 AVR - GCC

GCC-AVR je rozšíření GCC compileru pro kompilaci do strojového kódu pro rodinu AVR procesorů od firma Atmel, novější verze zvládají s přehledem i jazyk JAVA, Fortran a další. Lze nalézt modifikace také pro procesory ARM a jiné. Umožňuje zároveň i překlad z jazyka symbolických adres. Pomocí UISP lze použít programovací kit STK500.

Podporované mikrokontroléry:

| | | | | |
|-------------|--------------|--------------|-------------|-------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461 | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |
| AT90S2333, | AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, |
| AT90S8515, | AT90C8534, | AT90S8535, | ATmega103, | ATmega128, |
| ATmega1280, | ATmega1281, | ATmega16, | ATmega161, | ATmega162, |
| ATmega163, | ATmega164P, | ATmega165, | ATmega165P, | ATmega168, |
| ATmega169, | ATmega169P, | ATmega32, | ATmega323, | ATmega324P, |
| ATmega325, | ATmega3250, | ATmega329, | ATmega3290, | ATmega406, |
| ATmega48, | ATmega603, | ATmega64, | ATmega640, | ATmega644, |
| ATmega644P, | ATmega645, | ATmega6450, | ATmega649, | ATmega6490, |
| ATmega8, | ATmega8515, | ATmega8535, | ATmega88, | AT43USB320, |
| AT43USB355, | AT90USB1286, | AT90USB1287, | AT90USB646, | AT90USB647, |
| AT90CAN128, | AT90CAN32, | AT90CAN64, | AT76C711, | AT86RF401, |
| AT94K, | AT90PWM2, | AT90PWM3. | | |

Pro práci s GCC jsou potřeba alespoň tyto balíčky:

- binutils
- avr-libc
- uisp

4.1.2 Software pro nahrávání programů do mikrokontrolérů

• 4.1.2.1 UISP

Projekt **UISP** (*Micro In-System Programmer*) má za cíl vytvořit nástroj k programování mikroprocesorů AVR firmy Atmel. Program je volně ke stažení na Internetu včetně zdrojových kódů, což dovoluje jeho snadné šíření i na další operační systémy. Ovládání programu se provádí z příkazové řádky a je ho možno snadno integrovat do libovolného vývojového prostředí, V dnešní době UISP podporuje většinu AVR mikroprocesorů, které lze programovat metodou ISP přes integrované SPI rozhraní.

Nevýhodou této programovací metody je nutnost před každým programováním mikrokontrolér vymazat, a po té znovu nahrát daný program do paměti.

Základní vlastnosti UISP :

- autodetekce připojeného mikroprocesoru,
- čtení, zápis, smazání a kontrola paměti flash a eeprom,
- programování konfiguračních a zabezpečovacích bitů,
- podpora formátu Intel-HEX a Motorola,
- ovládání z příkazové řádky (snadná integrace do libovolného vývojového systému),

- multiplatformní řešení (MS Windows, Linux, FreeBSD,...),
- podpora ISP programátorů na paralelní a sériový port.

[3]

Podporované programátory:

- **abcmini** ABCmini Board, aka Dick Smith HOTCHIP
- **alf** Nightshade ALF-PgmAVR, <http://nightshade.homeip.net> (TonyFriebel's Programmer)
- **avr910** Standard Atmel Serial Programmer (Low Cost)
(Atmel: http://www.atmel.com/dyn/resources/prod_documents/AVR910.ZIP)
(K. Leitner: <http://www.mikrocontroller-projekte.de/Mikrocontroller/AVR-Prog/AVR-Programmer.html>)
- **dapa** Direct AVR Parallel Access
- **avrisp** Atmel AVR ISP (?)
- **jtagmki** Atmel JTAG ICEP (mkI)
- **jtagmkii** Atmel JTAG ICEP (mkII)
- **bascom** Bascom SAMPLE programming cable
- **bsd** <http://www.bsdhome.com/avrprog/> (parallel)
- **butterfly** Atmel Butterfly Development Board
- **dt006** Dontronics DT006, <http://www.dontronics.com/dt006.html> (parallel)
- **pavr** Jason Kyle's pAVR Serial Programmer, <http://avr.jpk.co.nz/pavr/pavr.html> or <http://www.openh.org/projects/pavr>
- **picoweb** Picoweb Programming Cable, <http://www.picoweb.net>
- **pony-stk200** Pony Prog STK200 with LED support
- **sp12** Steve Bolt's Programmer http://www.xs4all.nl/~sbolt/e-spider_prog.html
- **stk500** Atmel STK500 Evaluation Board v1/v2 (seriell)
- **stk200** Parallel Starter Kit STK200, STK300
- **usbasp** Thomas Fischl's USB programmer (<http://www.fischl.de/usbasp/>)
- **futurlec** Futurlec.com programming cable (parallel)
- **xil** Xilinx HW-JTAG-PC download cable (parallel)
- **dasa** serial (RESET=RTS SCK=DTR MOSI=TXD MISO=CTS)
- **dasa3** serial (RESET=!DTR SCK=RTS MOSI=TXD MISO=CTS)
- **ponyser** serial (RESET=!TXD SCK=RTS MOSI=DTR MISO=CTS)
- **atisp** AT-ISP V1.1 programming cable for AVR-SDK1 (<http://micro-research.co.th/>)
- **ere-isp-avr** ISP-AVR by ERE Co., Ltd.
(<http://www.ere.co.th/download/sch050713.pdf>)

Prakticky byly vyzkoušeny programátory dapa, bsd, které bez problému fungovali. Také byl zkoušen programátor dasa, pomocí kterého se nepodařilo jednočip nadetekovat.

Podporované mikrokontroléry:

| | | | | |
|-------------|-------------|------------|------------|------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |
| AT90S2333, | AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, |
| AT90S8515, | AT90C8534, | AT90S8535, | ATmega103, | ATmega128, |
| ATmega1280, | ATmega1281, | ATmega16, | ATmega161, | ATmega162, |

| | | | | |
|-------------|--------------|--------------|-------------|-------------|
| ATmega163, | ATmega164P, | ATmega165, | ATmega168, | ATmega169, |
| ATmega32, | ATmega323, | ATmega324P, | ATmega325, | ATmega3250, |
| ATmega329, | ATmega3290, | ATmega406, | ATmega48, | ATmega64, |
| ATmega640, | ATmega644, | ATmega644P, | ATmega645, | ATmega6450, |
| ATmega649, | ATmega6490, | ATmega8, | ATmega8515, | ATmega8535, |
| ATmega88, | AT43USB1286, | AT43USB1287, | AT43USB320, | AT43USB355, |
| AT43USB646, | AT43USB647, | AT76C711, | AT86RF401, | AT90CAN32, |
| AT90CAN64, | AT90CAN128, | AT90PWM2, | AT90PWM3, | AT94K. |

• 4.1.2.2 AVRDUDE

Program původně vyvinutý pro OS FreeBSD pod označením AVRPROG, odkud se rozšířil na další operační systémy. Co do filozofie řešení je velice podobný UISP. Výhodou je, že popis programování jednotlivých podporovaných obvodů a ISP programátorů je v konfiguračních souborech. Na rozdíl od UISP lze přidat nový obvod bez toho, aby bylo nutné znovu sestavit celý program. Používá vlastní konstrukci ISP programátoru na paralelní port.

Pro tento program i vznikla grafická nadstavba AVRDUDE-GUI, která ovšem není příliš kvalitní, má několik chyb a bohužel se zastavil její vývoj. Naštěstí začal vývoj jiné grafické nadstavby AVR8 Burn-O-Mat.

Firmware se nahrává již v přeložených hexadecimálních souborech. Na rozdíl od metody programování pomocí uisp není nutné nejprve smazat program a pak nahrát opět program. Umožňuje download i upload programu ze zařízení. Podporuje možnost programování paměti FLASH i EEPROM, dále umožňuje programovat propojky v mikrokontroléru. AVRDUDE provádí mazání mikrokontroléru automaticky před každým načtením programu do mikrokontroléru. Program má svůj konfigurační soubor, ve kterém jsou nadefinovány jednak parametry mnoha programovacích kabelů (včetně čísel vývodů LPT portu pro jednotlivé funkce) tak i parametry jednotlivých procesorů. [2], [10]

Podporované programátory:

- **atisp** AT-ISP V1.1 programming cable for AVR-SDK1 from <http://micro-research.co.th/>
- **avr109** Atmel AppNote AVR109 Boot Loader
- **avr910** Atmel Low Cost Serial Programmer
- **avr911** Atmel AppNote AVR911 AVROSP
- **avrisp** Atmel AVR ISP (an alias for stk500)
- **avrisp2** Atmel AVR ISP mkII (alias for stk500v2)
- **avrispmkII** Atmel AVR ISP mkII (alias for stk500v2)
- **avrispv2** Atmel AVR ISP, version 2.x firmware
- **bascom** Bascom SAMPLE programming cable
- **blaster** Altera ByteBlaster
- **bsd** Brian Dean's Programmer, <http://www.bsdhome.com/avrdude/>
- **butterfly** Atmel Butterfly Development Board
- **c2n232i** C2N232I, reset=dtr sck=!rts mosi=!txd miso=!cts, <http://www.ktverkko.fi/~msmakela/8bit/c2n232/hardware/index.en.html>
- **dapa** Direct AVR Parallel Access cable
- **dasa** serial port banging, reset=rts sck=dtr mosi=txd miso=cts
- **dasa3** serial port banging, reset=!dtr sck=rts mosi=txd miso=cts
- **dragon_dw** AVR Dragon in debugWire mode
- **dragon_hvsp** AVR Dragon in high-voltage serial programming mode
- **dragon_isp** AVR Dragon in ISP mode

- **dragon_jtag** AVR Dragon in JTAG mode
- **dragon_pp** AVR Dragon in (high-voltage) parallel programming
- **dt006** Dontronics DT006
- **ere-isp-avr** ERE ISP-AVR, <http://www.ere.co.th/download/sch050713.pdf>
- **frank-stk200** Frank's STK200 clone, <http://electropol.free.fr/spip/spip.php?article15>
- **futurlec** Futurlec.com programming cable
- **jtag1** Atmel JTAG ICE mkI, running at 115200 Bd
- **jtag1slow** Atmel JTAG ICE mkI, running at 19200 Bd
- **jtag2slow** Atmel JTAG ICE mkII (default speed 19200 Bd)
- **jtag2** Atmel JTAG ICE mkII, running at 115200 Bd
- **jtag2fast** Atmel JTAG ICE mkII, running at 115200 Bd
- **jtag2isp** Atmel JTAG ICE mkII in ISP mode.
- **jtag2dw** Atmel JTAG ICE mkII in debugWire mode.
- **jtagmkI** Atmel JTAG ICE mkI, running at 115200 Bd
- **jtagmkII** Atmel JTAG ICE mkII (default speed 19200 Bd)
- **pavr** Jason Kyle's pAVR Serial Programmer
- **picoweb** Picoweb Programming Cable, <http://www.picoweb.net/>
- **pony-stk200** Pony Prog STK200
- **ponyser** design ponyprog serial, reset=!txd sck=rts mosi=dtr miso=cts
- **sipro** Lancos SI-Prog, <http://www.lancos.com/siprogsch.html>
- **sp12** Steve Bolt's Programmer
- **stk200** STK200
- **stk500** Atmel STK500, probing for either version 1.x or 2.x firmware
- **stk500hvsp** Atmel STK500 high-voltage serial programming mode(version 2.x firmware)
- **stk500pp** Atmel STK500 in parallel programming mode (version 2.xfirmware)
- **stk500v1** Atmel STK500, running a version 1.x firmware
- **stk500v2** Atmel STK500, running a version 2.x firmware
- **usbasp** USBasp, <http://www.fischl.de/usbasp/>
- **usbtiny** USBtiny simple USB programmer, <http://www.ladyada.net/make/usbtinyisp/>
- **xil** Xilinx JTAG cable

Podporované mikrokontroléry:

| | | | | |
|-------------|-------------|-------------|-------------|-------------|
| AT90CAN128, | AT90PWM2, | AT90PWM3, | AT90S1200, | AT90S2313, |
| AT90S2333, | AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, |
| AT90S8515, | AT90S8535, | ATmega103, | ATmega128, | ATmega1280, |
| ATmega1281, | ATmega16, | ATmega161, | ATmega162, | ATmega163, |
| ATmega164, | ATmega169, | ATmega2560, | ATmega2561, | ATmega32, |
| ATmega324, | ATmega329, | ATmega3290, | ATmega48, | ATmega64, |
| ATmega640, | ATmega644, | ATmega649, | ATmega6490, | ATmega8, |
| ATmega8515, | ATmega8535, | ATmega88, | ATtiny12, | ATtiny13, |
| ATtiny15, | ATtiny2313, | ATtiny25, | ATtiny26, | ATtiny45, |
| ATtiny85, | | | | |

• 4.1.2.3 Avrp

Avrp – programovací nástroj, který umí jak sériově tak paralelně komunikovat s mikrokontroléry Atmel AVR. Avrp je vyvíjeno Jonem Anders Haugunen. Jeho nevýhodou je chybějící grafické rozhraní, ale dobře se s ním pracuje i v příkazové řádce. [6]

Podporované programátory:

- **avr910** Standard Atmel Serial Programmer (Low Cost)
- **avrprogger** Adrian's Parallel AVR programmer
<http://homepage.ntlworld.com/fangorn/projects/avr/avrprogger.html>

Podporované procesory:

Starší procesory řady AT89.

• 4.1.2.4 AVRprog

AVRprog je program který je užíván pro nahrávání nebo čtení FLASH a EEPROM paměti z Atmel AVR mikrokontrolérů na FreeBSD Unix, Linux, Windows. Program podporuje Atmel sériové programovací protokoly, používajícího PC paralelního rozhraní a může nahrávat buď klasický binární soubor nebo Intel hexadecimální formátový soubor. Avrpog může také užíván v interaktivním módu pro jednotlivé aktualizace jednotlivých EEPROM buňek, pojistky, nebo zámek (jestli jejich přístup je podporovaný Atmel AVR sériovým programovacím protokolem). Tento program má konzolovou podobu a také grafické rozhraní.

Podporované programátory:

- **Serial11** special selfmade serial adapter <http://avr.fenceline.de/serial11/serial11.html>
- **SerialSTK** Parallel Starter Kit STK200, STK300
<http://avr.fenceline.de/serialSTK/serialSTK.html>
- **avr910** Standard Atmel Serial Programmer (Low Cost)
(Atmel: http://www.atmel.com/dyn/resources/prod_documents/AVR910.ZIP)
(K. Leitner: <http://www.mikrocontroller-projekte.de/Mikrocontroller/AVR-Prog/AVR-Programmer.html>)
- **Akizuki** 'Akizuki' programmer <http://avr.fenceline.de/akizuki/akizuki.html>

Podporované mikrokontroléry:

AT90S1200-4, AT90S1200-12, AT90S2323, AT90LS2323, AT90S2343,
AT90LS2343, AT90S2313-4, AT90S2313-10, AT90S4433, AT90LS4433,
AT90S8515-4, AT90S8515-8, AT90S8535, AT90LS8535, AT90S2333,
AT90LS2333, AT90S4414-4, AT90S4414-8, AT90S4434, AT90LS4434,
ATmega161L, ATmega163, ATmega161, ATmega163L, ATmega323,
ATmega323L, ATmega103, ATmega103L, ATtiny22L, ATtiny12,
ATtiny12L, ATtiny12V, ATtiny15L.

4.1.3 Knihovny pro jazyk C AVR mikrokontrolérů

• 4.1.3.1 AVR Libc

AVR Libc je bezplatný software. Projekt jehož cíl je poskytnout vysoké kvalitní C knihovny pro použití s GCC, také assembler na mikroprocesory Atmel AVR. Podporuje celou řadu mikrokontrolérů AVR i jiné zařízení vyráběné firmou Atmel.

V této knihovně jsou definovány vstupní/výstupní piny, přerušení, kontrolu časovače, stará se i o obvod watchdog. Taktéž se stará i o obsazení paměti, funkci malloc a jiné podobné důležité vlastnosti mikrokontrolérů.[4]

Podporované mikrokontroléry:

| | | | | |
|-------------|--------------|--------------|-------------|-------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |
| AT90S2333, | AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, |
| AT90S8515, | AT90C8534, | AT90S8535, | ATmega103, | ATmega128, |
| ATmega1280, | ATmega1281, | ATmega16, | ATmega161, | ATmega162, |
| ATmega163, | ATmega164P, | ATmega165, | ATmega168, | ATmega169, |
| ATmega32, | ATmega323, | ATmega324P, | ATmega325, | ATmega3250, |
| ATmega329, | ATmega3290, | ATmega406, | ATmega48 | , ATmega64, |
| ATmega640, | ATmega644, | ATmega644P, | ATmega645, | ATmega6450, |
| ATmega649, | ATmega6490, | ATmega8, | ATmega8515, | ATmega8535, |
| ATmega88, | AT43USB1286, | AT43USB1287, | AT43USB320, | AT43USB355, |
| AT43USB646, | AT43USB647, | AT76C711, | AT86RF401, | AT90CAN32, |
| AT90CAN64, | AT90CAN128, | AT90PWM2, | AT90PWM3, | AT94K. |

• 4.1.3.2 Avr-binutils

Kolekce ve které jsou obsaženy binární utility a ostatní potřebné soubory pro práci s AVR mikrokontroléry. Například vkládá a obstarává funkce avr-objcopy pro kopírování a překládání objektových souborů. Také mezi další funkce patří avr-objdump, který zobrazuje informace z objektových souborů. Umožňuje i další možnosti od podpory assembleru až po kompletní podporu binárními utilitami pro programovací jazyk C.

Podporované mikrokontroléry:

| | | | | |
|-------------|-------------|-------------|-------------|--------------|
| ATtiny10, | ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, |
| ATtiny22, | ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, |
| ATtiny261, | ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, |
| ATtiny84, | ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, |
| AT90S2323, | AT90S2333, | AT90S2343, | AT90S4414, | AT90S4433, |
| AT90S4434, | AT90S8515, | AT90C8534, | AT90S8535, | ATmega103, |
| ATmega128, | ATmega1280, | ATmega1281, | ATmega16, | ATmega161, |
| ATmega162, | ATmega163, | ATmega164P, | ATmega165, | ATmega165P, |
| ATmega168, | ATmega169, | ATmega169P, | ATmega32, | ATmega323, |
| ATmega324P, | ATmega325, | ATmega3250, | ATmega329, | ATmega3290, |
| ATmega406, | ATmega48, | ATmega603, | ATmega64, | ATmega640, |
| ATmega644, | ATmega644P, | ATmega645, | ATmega6450, | ATmega649, |
| ATmega6490, | ATmega8, | ATmega83, | ATmega85, | ATmega8515, |
| ATmega8535, | ATmega88, | AT43USB320, | AT43USB355, | AT90USB1286, |

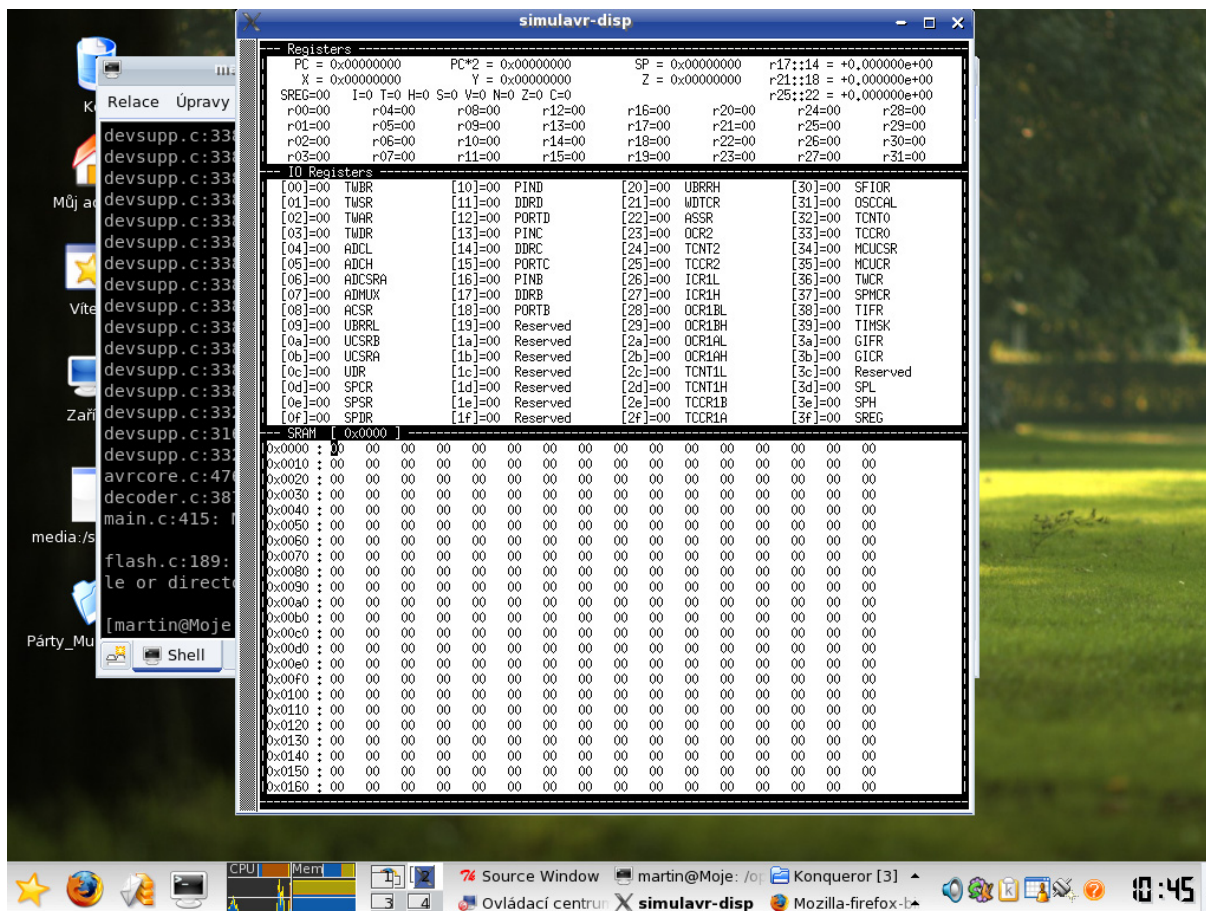
AT90USB1287, AT90USB646, AT90USB647, AT90CAN128, AT90CAN32,
 AT90CAN64, AT76C711, AT86RF401, AT94K, AT90PWM2,
 AT90PWM3.

4.1.4 Simulátory

Programy sloužící pro usnadnění vývoje aplikace bez nutnosti nahrávání software do mikrokontroléru s různými možnostmi sledovat obsahy registrů, různých jednotek jež jsou součástí mikrokontrolérů.

• 4.1.4.1 Simulavr

Simulavr je simulátor Atmel AVR procesorů. Tento projekt je ještě stále ve vývoji a není doporučen pro výrobní používání. V současné době jsou sice vydány záplaty na tento program ale ještě stále není 100% funkční. Funguje dobře v konzoli. Bohužel vývoj tohoto projektu začal stagnovat, a jeho další vývoj je nejistý. Program je ovládán pomocí konzole. Chybí u něho možnost libovolně ručně krokovat program, je krokován automaticky po určitém čase. Umožňuje nastavení si breakpointy a tak zastavit program na důležitém místě pro ladění programu.[5]



Obr 2. Simulavr v systémů Mandriva 2007

Podporované mikrokontroléry:

| | | | | |
|-------------|-------------|-------------|-------------|-------------|
| AT90S1200, | AT90S2313, | AT90S4414, | AT90S8515, | ATmega8, |
| ATmega16, | ATmega103, | ATmega128, | AT43USB320, | AT43USB325, |
| AT43USB326, | AT43USB351, | AT43USB353, | AT43USB355, | |

• 4.1.4.2 AVaRICE

AVaRICE – je simulační nástroj pro mikrokontroléry Atmel. Je založen na simulačním pluginu GDB. AVaRICE je vlastně interface mezi Atmel JTAG ICE a GDB, se zaměřením pro mikrokontroléry AVR. Je určen pro příkazovou řádku. Podporuje sériový protokol pro programování mikroprocesorů. Pro tento program existuje grafické rozhraní. Umožňuje taktéž programování programovatelných propojek. [7], [8]

Podporované mikrokontroléry:

| | | | | |
|------------|-------------|-------------|--------------|-------------|
| ATmega16, | ATmega162, | ATmega169, | ATmega323, | ATmega32, |
| ATmega64, | ATmega128, | AT90can128, | ATmega164, | ATmega324, |
| ATmega644, | ATmega329, | ATmega3290, | ATmega649, | ATmega6490, |
| ATmega640, | ATmega1280, | ATmega1281, | AT90USB1287, | ATmega48 |

• 4.1.4.3 Avr-GDB

GDB je utilita pro ladění software vytvořená v rámci projektu GNU. Umožňuje zjistit, co se děje při běhu testovaného software, případně umožňuje zjistit, proč došlo k případnému pádu software.

Avr-GDB – Simulátor určený pro mikrokontroléry Atmel AVR . Jde o modifikaci GDB simulátoru přímo pro potřeby simulace programů na mikrokontrolérech AVR. Je vyvíjen velkou skupinou lidí. Pracuje s příkazovou řádkou, která bohatě dostačuje pro ovládání tohoto simulátoru. Pro tento program existuje grafické rozhraní pro GNU Linux i pro OS Windows. Program umožňuje celou škálu možností simulování. Umožňuje program krokovat po řádcích programu pomocí příkazů *step* a *next*, nastavovat jednoduše breakpointy na zvolené lokace a tím usnadnit simulaci a nalést problémy při pádu a chybách aplikací. Breakpointy mohou být dokonce podmíněné, zadávané pomocí příkazu *break* lokace *if* výraz. Když program dorazí na uvedenou lokaci, dojde k vyhodnocení výrazu. Pokud se tento vyhodnotí na nenulovou hodnotu, dojde k zastavení programu. Umožňuje prohlížet obsahy registrů a nastavení jednotlivých jednotek mikrokontrolérů. AVR – GDB je integrováno i v AVR studiu, které výrobce mikrokontrolérů Atmel poskytuje zcela zdarma. Načítá soubory s koncovkou *.elf. Pro možnost simulace je nutné správně nastavit parametry překladače GCC, aby mohl debugger provést simulaci. Výhodou toho simulátoru je možnost vzdálené módu kdy simulace je ovládána vzdáleně. [8], [9]

Podporované mikrokontroléry:

| | | | | |
|------------|-------------|-------------|------------|--------------|
| ATtiny10, | ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, |
| ATtiny22, | ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, |
| ATtiny261, | ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, |
| ATtiny84, | ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, |
| AT90S2323, | AT90S2333, | AT90S2343, | AT90S4414 | , AT90S4433, |
| AT90S4434, | AT90S8515, | AT90C8534, | AT90S8535, | ATmega103, |
| ATmega128, | ATmega1280, | ATmega1281, | ATmega16, | ATmega161, |
| ATmega162, | ATmega163, | ATmega164P, | ATmega165, | ATmega165P, |
| ATmega168, | ATmega169, | ATmega169P, | ATmega32, | ATmega323, |

| | | | | |
|--------------|-------------|-------------|-------------|--------------|
| ATmega324P, | ATmega325, | ATmega3250, | ATmega329, | ATmega3290, |
| ATmega406, | ATmega48, | ATmega603, | ATmega64, | ATmega640, |
| ATmega644, | ATmega644P, | ATmega645, | ATmega6450, | ATmega649, |
| ATmega6490, | ATmega8, | ATmega83, | ATmega85, | ATmega8515, |
| ATmega8535, | ATmega88, | AT43USB320, | AT43USB355, | AT90USB1286, |
| AT90USB1287, | AT90USB646, | AT90USB647, | AT90CAN128, | AT90CAN32, |
| AT90CAN64, | AT76C711, | AT86RF401, | AT94K, | AT90PWM2, |
| AT90PWM3. | | | | |

4.1.5 Assembler

Assembler je programovací jazyk velice blízký strojovému kódu, nazývá se také jazyk symbolických adres. Název assembler se používá i pro překladač, který tento jazyk překládá do strojového kódu. Assembler je programovací jazyk nejnižší úrovně a je závislý na strojovém kódu procesoru. Každá rodina procesorů má svůj vlastní odlišný assembler, protože ve strojovém kódu různých rodin procesorů a možnosti rozdělování a adresování paměti bývají zásadní rozdíly. Každá firma vyrábějící procesory si definuje vlastní pravidla pro assembler svých procesorů, z kterých mohou (ale také nemusejí) vycházet nezávislí autoři a firmy.

• 4.1.5.1 Avr-avra

Avra je assembler pro rodinu mikrokontrolerů AVR. Je plně kompatibilní s assemblerem podporovaným procesory Atmel AVRASM32. Tento assembler podporuje vkládání speciálních direktiv a má vylepšenou podporu maker. [22]

Podporované mikrokontroléry:

| | | | | |
|------------|-------------|-------------|------------|------------|
| ATtiny11, | ATtiny12, | ATtiny15, | ATtiny22, | ATtiny26, |
| ATtiny28, | AT90S1200, | AT90S2313, | AT90S2323, | AT90S2333, |
| AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, | AT90S8515, |
| AT90C8534, | AT90S8535, | ATmega103, | ATmega104, | ATmega128, |
| ATmega16, | ATmega161, | ATmega162, | ATmega163, | ATmega169, |
| ATmega32, | ATmega323, | ATmega603, | ATmega64, | ATmega8, |
| ATmega83, | ATmega8515, | ATmega8535, | AT86RF401, | AT94K. |

• 4.1.5.2 Avr-tavrsm

Tavrsm je další assembler pro mikrokontroléry Atmel AVR. Používá zde podobnou syntaxi podobnou, kterou bylo možno najít v syntaxi Atmel AVR DOS assembleru. Podporuje vkládání makra do maker, umožňuje ukládat výstupní data v několika tvarech, například Intel HEX, Motorola S-record, a také podporuje binární výstup. Tavrasm je kompatibilní s AVR studiem od firmy Atmel. [22]

Podporované mikrokontroléry:

| | | | | |
|------------|------------|------------|------------|------------|
| AT90S1200, | AT90S2313, | AT90S2323, | AT90S2343, | AT90S4414, |
| AT90S4433, | AT90S4434, | AT90S8515, | AT90S8535, | ATmega103, |
| ATmega128, | ATmega16, | ATmega162, | ATmega161, | ATmega163, |
| ATmega168, | ATmega169, | ATmega32, | ATmega323, | ATmega48, |

ATmega8, ATmega8515, ATmega8535, ATmega88.

4.2 Vývojové prostředí IDE

Vývojové prostředí, (často se používá zkratka IDE z anglického Integrated development environment) je software usnadňující práci programátorů, většinou zaměřený na jeden konkrétní programovací jazyk. Obsahuje editor zdrojového kódu, kompilátor, případně interpret a většinou také debugger (simulátor). Většina vývojových prostředí používá programy vytvořené v rámci projektu AVR Cross Development Kit.

4.2.1 Vývojové prostředí pracující nativně pod systémem GNU/Linux

Programy které fungují přímo pod systémem GNU/Linux a nepoužívají žádné emulátory a ani žádné překladače API systému MS Windows.

4.2.1.1 KontrollerLab

KontrollerLab je program který v sobě shrnuje avr-gcc compiler, uisp, AVRDUDE. Takže je vhodný pro sériové i paralelní programování. Jako rozhraní používá grafické KDE prostředí. Spolupracuje s jazykem C,C++ i jazykem symbolických adres. Pracuje na všech operačních systémech založených na systému Unix. Prostředí umí programovat pomocí uisp, AVRDUDE, také i pomocí programátoru STK500 jenž se používají na ústavě radioelektroniky. V posledních verzích byl do tohoto vývojového prostředí integrován i simulátor s různými možnostmi simulace. [10]

Podporované mikrokontroléry:

| | | | | |
|--------------|--------------|--------------|-------------|-------------|
| AT43USB320, | AT43USB355, | AT76C711, | AT86rf401, | AT90C8534, |
| AT90CAN128, | AT90CAN32, | AT90CAN64, | AT90CAN646, | AT90PWM2, |
| AT90PWM3, | AT90S1200, | AT90S2313, | AT90S2323, | AT90S2343, |
| AT90S4414, | AT90S4433, | AT90S4434, | AT90S8515, | AT90S8535, |
| AT90USB1286, | AT90USB1287, | AT90USB1287, | AT90USB647, | AT94k, |
| ATmega103, | ATmega128, | ATmega1280, | ATmega1281, | ATmega16, |
| ATmega162, | ATmega161, | ATmega163, | ATmega164p, | ATmega165, |
| ATmega168, | ATmega169, | ATmega32, | ATmega323, | ATmega324p, |
| ATmega325, | ATmega3250, | ATmega329, | ATmega3290, | ATmega406, |
| ATmega48, | ATmega603, | ATmega64, | ATmega640, | ATmega644, |
| ATmega644p, | ATmega645, | ATmega649, | ATmega6490, | ATmega8, |
| ATmega8515, | ATmega8535, | ATmega88, | ATtiny11, | ATtiny12, |
| ATtiny13, | ATtiny15, | ATtiny22, | ATtiny2313, | ATtiny24, |
| ATtiny25, | ATtiny26, | ATtiny261, | ATtiny28, | ATtiny44, |
| ATtiny461, | ATtiny84, | ATtiny85, | ATtiny861, | |

4.2.1.2 PonyProg 2000

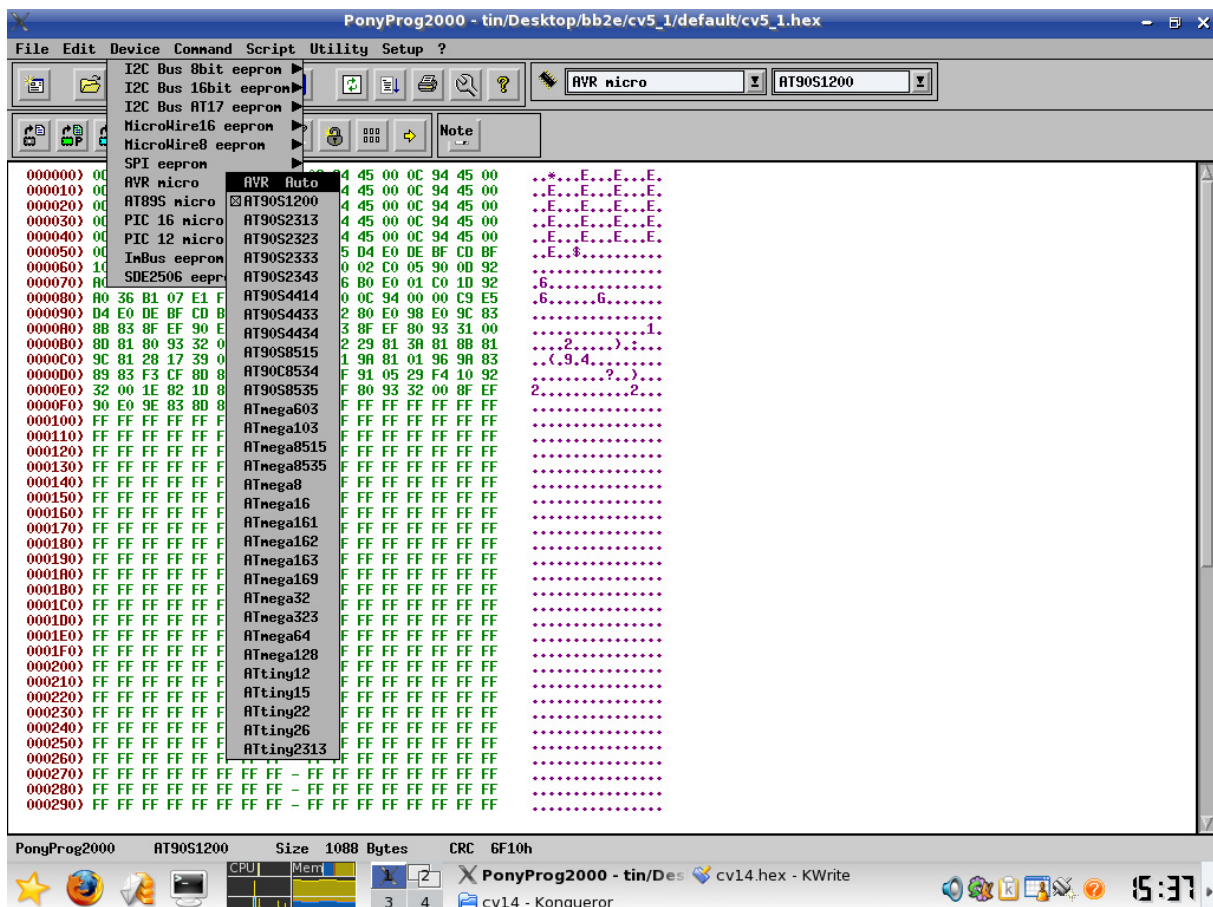
PonyProg - jednoduchý programátor na sériový a paralelní port, je dostupný jak pro operační systémy Windows tak také pro Linux. Používá „user friendly“ grafické rozhraní. Program by měl umět číst a zapisovat na každé sériové rozhraní. Nyní podporuje I²C Bus, Microwire, SPI eeprom. Pro programování AVR procesorů potřebuje speciální ISP programovací kit. PonyProg spolupracuje také s programátory STK200/300, měl by podporovat i verzi STK500 jež je na ústavu radioelektroniky. Využívá hardware portů

počítače a snaží se vnější elektroniku omezit na minimum. Vyniká velmi propracovaným softwarem a šíří podporovaných obvodů v rámci čehož jsou zahrnuty i AVR mikroprocesory. Oproti UISP je mnohem pomalejší (existuje podpora pouze pro programátory na par. port) což je poznat zvláště při programování mikroprocesorů ATmega. Starší verze programu obsahovaly chybu, která způsobovala špatné nastavení konfiguračních bitů a zablokování mikroprocesoru. Nevýhodou tohoto programovacího prostředí je při ovládání programových menu nutné stále stisknuté levé tlačítko myši pro možnost výběru položky. Výběr zařízení je zde velice jednoduchý a je možný přímo z hlavního okna.

Umí do zařízení nahrávat z několika typů souborů např: *.E2P, *.HEX, *.MOT, *.BIN, *.CSM, *.ROM, *.EEP. [2], [13]

Podporované mikrokontroléry AVR:

| | | | | |
|-------------|-------------|-------------|-------------|-------------|
| AT90S1200, | AT90S2313, | AT90S2323, | AT90S2343, | AT90S4414, |
| AT90S4434, | AT90S8515, | AT90S8535, | AT90S2323, | AT90S2343, |
| AT90S2333, | AT90S4433, | AT90S4434, | AT90S8535, | AT90S8534, |
| AT90can32, | AT90can64, | AT90can128, | ATmega103, | ATmega161, |
| ATmega163, | ATmega323, | ATmega128, | ATmega8, | ATmega16, |
| ATmega64, | ATmega32, | ATmega162, | ATmega169, | ATmega8515, |
| ATmega8535, | ATmega44, | ATmega88, | ATmega168, | ATmega164, |
| ATmega324, | ATmega644, | ATmega640, | ATmega1280, | ATmega1281, |
| ATmega2560, | ATmega2561, | ATtiny12, | ATtiny15, | ATtiny26, |
| ATtiny2313, | ATtiny13, | ATtiny25, | ATtiny45, | ATtiny85, |
| ATtiny261, | ATtiny461, | ATtiny861 | | |



Obr 3. Program PonyProg 2000 v systému Mandriva 2007 – ukázka výběru zařízení

4.2.1.3 AVR-Eclipse

AVR-Eclipse – jde o zásuvný modul do Eclipse. Eclipse je open source vývojová platforma, která je pro většinu lidí známa jako vývojové prostředí (IDE) určené pro programování v jazyce Java. Flexibilní návrh této platformy dovoluje rozšířit seznam podporovaných programovacích jazyků za pomoci pluginů, například o C++ nebo PHP. Právě pluginy umožňují toto vývojové prostředí rozšířit například o návrh UML, či zápis HTML nebo XML a také programování mikrokontrolérů Atmel AVR. AVR-Eclipse obsahuje 2 zásuvné moduly, jeden nezbytný pro zakládání standardních nových projektů a druhý plugin přidává položku v menu, která umožňuje stahovat hexadecimální soubory z aktuálního projektu k cílovému zařízení. Rovněž umožňuje i také simulaci pomocí nástroje GDB z AVR Cross Development Kit a pomocí klienta z Eclipse se připojíme do GDB serveru můžeme simulovat jako s GDB z příkazové řádky. Další možností simulace v tomto prostředí je použít opět nástroje z AVR Cross Development Kitu jako je simulavr a nebo také možnost JTAG spojení a s pomocí serveru GDB spustit aplikaci AVaRICE. Jako uploader používá Eclipse nástroj AVR dude.[11], [12]

Podporované mikrokontroléry:

| | | | | |
|--------------|--------------|--------------|--------------|-------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |
| AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, | AT90S8515, |
| AT90S8535, | ATmega103, | ATmega128, | ATmega1280, | ATmega1281, |
| ATmega16, | ATmega161, | ATmega162, | ATmega163, | ATmega164P, |
| ATmega165, | ATmega165P, | ATmega168, | ATmega168P, | ATmega169, |
| ATmega169P, | ATmega2560, | ATmega2561, | ATmega32, | ATmega323, |
| ATmega324P, | ATmega325, | ATmega3250, | ATmega3250P, | ATmega329, |
| ATmega329P, | ATmega3290, | ATmega3290P, | ATmega406, | ATmega48, |
| ATmega48P, | ATmega64, | ATmega640, | ATmega644, | ATmega644P, |
| ATmega645, | ATmega6450, | ATmega649, | ATmega6490, | ATmega8, |
| ATmega8515, | ATmega8535, | ATmega88, | ATmega88P, | AT43USB162, |
| AT90USB1286, | AT90USB1287, | AT90USB646, | AT90USB647, | AT90CAN128, |
| AT90CAN32, | AT90CAN64, | AT76C711, | AT86RF401, | AT94K, |
| AT90PWM2, | AT90PWM2B, | AT90PWM3, | AT90PWM3B. | |

4.2.2 Vývojové prostředí fungující pomocí programu WINE

Wine - je počítačový program, který umožňuje chod aplikací pro Microsoft Windows pod převážně unixovými operačními systémy. Wine nevyžaduje žádný operační systém Microsoft Windows, je to úplně volná alternativní implementace Windows API kde není žádná část kódu od firmy Microsoft. Nicméně tento program umožňuje volitelně užívat nativní knihovny Windows DLL pokud budou dostupné. Wine je pouze soustava API a proto neumí spouštět Windows aplikace na jiné architektuře, než je Intel x86.

4.2.2.1 AVR studio verze 3.56:

Programovací nástroj pro OS Windows. Lze provozovat i pomocí překladače wine-0.9.11-7. Emulace sice není příliš dokonalá, například nefunguje Help díky chybějícímu

Internet Exploreru, ale jinak simulace, kompilace a ostatní funkce studia pracují korektně. Pro práci s jazykem C je nutné pomocí Wine nainstalovat navíc WinAVR.

Podporované mikrokontroléry:

| | | | | |
|--------------|--------------|--------------|--------------|-------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |
| AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, | AT90S8515, |
| AT90S8535, | ATmega103, | ATmega128, | ATmega1280, | ATmega1281, |
| ATmega16, | ATmega161, | ATmega162, | ATmega163, | ATmega164P, |
| ATmega165, | ATmega165P, | ATmega168, | ATmega168P, | ATmega169, |
| ATmega169P, | ATmega2560, | ATmega2561, | ATmega32, | ATmega323, |
| ATmega324P, | ATmega325, | ATmega3250, | ATmega3250P, | ATmega329, |
| ATmega329P, | ATmega3290, | ATmega3290P, | ATmega406, | ATmega48, |
| ATmega48P, | ATmega64, | ATmega640, | ATmega644, | ATmega644P, |
| ATmega645, | ATmega6450, | ATmega649, | ATmega6490, | ATmega8, |
| ATmega8515, | ATmega8535, | ATmega88, | ATmega88P, | AT43USB162, |
| AT90USB1286, | AT90USB1287, | AT90USB646, | AT90USB647, | AT90CAN128, |
| AT90CAN32, | AT90CAN64, | AT76C711, | AT86RF401, | AT94K, |
| AT90PWM2, | AT90PWM2B, | AT90PWM3, | AT90PWM3B, | |

4.2.2.2 AVR studio verze 4.12:

Jedná se o novější verzi zdařilého vývojového prostředí pro mikrokontroléry AVR. Toto prostředí má zdařilý simulační program. Avšak pro jeho správný běh je potřeba nainstalovat do systému Linux i Internet Explorer 6 z produktu dílny Microsoft. Pro instalaci tohoto programu je nutné použít script IES4LINUX. Pro správný běh je také nutné stáhnout balíček cabextract. Cabextract je Free Software určený pro rozbalování cabinet souborů od firmy Microsoftu (tzv. .CAB soubory) na unixových systémech. Program je založen na knihovně libmspack a podporuje všechny speciální vlastnosti a komprimační algoritmy cabinet souborů. Je také potřeba zkopírovat fonty z OS Windows. I zde je nutné nainstalovat pomocí Wine WinAVR. [15]

Není vyloučena možnost správné funkce i novějších AVR studií, jen nebyli nainstalováni.

Podporované mikrokontroléry:

| | | | | |
|-------------|-------------|--------------|--------------|-------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |
| AT90S2343, | AT90S4414, | AT90S4433, | AT90S4434, | AT90S8515, |
| AT90S8535, | ATmega103, | ATmega128, | ATmega1280, | ATmega1281, |
| ATmega16, | ATmega161, | ATmega162, | ATmega163, | ATmega164P, |
| ATmega165, | ATmega165P, | ATmega168, | ATmega168P, | ATmega169, |
| ATmega169P, | ATmega2560, | ATmega2561, | ATmega32, | ATmega323, |
| ATmega324P, | ATmega325, | ATmega3250, | ATmega3250P, | ATmega329, |
| ATmega329P, | ATmega3290, | ATmega3290P, | ATmega406, | ATmega48, |
| ATmega48P, | ATmega64, | ATmega640, | ATmega644, | ATmega644P, |
| ATmega645, | ATmega6450, | ATmega649, | ATmega6490, | ATmega8, |

| | | | | |
|--------------|--------------|-------------|-------------|-------------|
| ATmega8515, | ATmega8535, | ATmega88, | ATmega88P, | AT43USB162, |
| AT90USB1286, | AT90USB1287, | AT90USB646, | AT90USB647, | AT90CAN128, |
| AT90CAN32, | AT90CAN64, | AT76C711, | AT86RF401, | AT94K, |
| AT90PWM2, | AT90PWM2B, | AT90PWM3, | AT90PWM3B. | |

4.2.2.3 Visual Micro Lab 3.12 for AVR:

Jde o software podobný vývojovému prostředí AVR Studio. Využívá taktéž AVR GCC knihovny pro překlad zdrojových souborů. Pro používání GCC knihovny je nutné nainstalovat WinAVR software. Podporuje programování jak v jazyce symbolických adres, tak v jazyce C. Obsahuje nástroje pro psaní zdrojových souborů, simulaci programů a nahrávání programu do mikrokontrolérů. Umožňuje spuštění více oken s programy, a tak usnadňuje práci při složitějších systémech s několika mikrokontroléry. Podporuje i vývojový kit firmy Atmel STK 500. [16]

Podporuje tyto typy mikrokontrolérů:

| | | | | |
|-------------|-------------|------------|------------|------------|
| ATtiny11, | ATtiny12, | ATtiny15, | ATtiny22, | AT90S2343, |
| AT90S2323, | AT90S1200, | AT90S2313, | AT90S4433, | AT90S4414, |
| AT90S8515, | AT90S4434, | AT90S8535, | ATmega8, | ATmega16, |
| ATmega161, | ATmega162, | ATmega32, | ATmega64, | ATmega128, |
| ATmega8515, | ATmega8535, | ATtiny2313 | | |

4.3 Další Simulátory:

Programy sloužící pro usnadnění vývoje aplikace bez nutnosti nahrávání software do mikrokontroléru s různými možnostmi sledovat obsahy registrů, různých jednotek jež jsou součástí mikrokontrolérů.

4.3.1 Avrora

Avrora je výzkumný projekt UCLA Compilers Group, hlavním vývojářem této aplikace je Ben L. Titzer. Jde o simulační a analyzační nástroj pro programy napsané pro mikrokontroléry AVR vyráběné firmou Atmel a Mica2 senzory. Avrora obsahuje flexibilní strukturu pro simulování a analýzy programů napsaných v jazyce symbolických adres – assembler a také podporuje jazyk C. Jeho nevýhodou je pouze možnost ovládání pomocí konzole. Tento projekt je stále vyvíjen a v poslední době bylo vytvořeno i grafické rozhraní pro ovládání simulací.

Pro svoji správnou činnost Avrora potřebuje nainstalovanou Javu. Program používá Java API. Také potřebuje balíček avr-GDB, kde využívá vlastností tohoto simulátoru. Také potřebuje mít nainstalovaný balíček avr-objdump a nezbytný balíček avr-GCC.

Pracuje s programy ve formátu *.od který se vytváří pomocí programu avr-objdump. [14]

Podporované mikrokontroléry:

| | | | | |
|-------------|------------|------------|------------|------------|
| ATtiny11, | ATtiny12, | ATtiny13, | ATtiny15, | ATtiny22, |
| ATtiny2313, | ATtiny24, | ATtiny25, | ATtiny26, | ATtiny261, |
| ATtiny28, | ATtiny44, | ATtiny45, | ATtiny461, | ATtiny84, |
| ATtiny85, | ATtiny861, | AT90S1200, | AT90S2313, | AT90S2323, |

| | | | | |
|-------------|--------------|--------------|-------------|-------------|
| ATmega1280, | ATmega1281, | ATmega16, | ATmega161, | ATmega162, |
| ATmega163, | ATmega164P, | ATmega165, | ATmega165P, | ATmega168, |
| ATmega169, | ATmega169P, | ATmega32, | ATmega323, | ATmega324P, |
| ATmega325, | ATmega3250, | ATmega329, | ATmega3290, | ATmega406, |
| ATmega48, | ATmega603, | ATmega64, | ATmega640, | ATmega644, |
| ATmega644P, | ATmega645, | ATmega6450, | ATmega649, | ATmega6490, |
| ATmega8, | ATmega8515, | ATmega8535, | ATmega88, | AT43USB320, |
| AT43USB355, | AT90USB1286, | AT90USB1287, | AT90USB646, | AT90USB647, |
| AT90CAN128, | AT90CAN32, | AT90CAN64, | AT76C711, | AT86RF401, |
| AT94K, | AT90PWM2, | AT90PWM3. | | |

5. Návrh Demo úlohy

V této kapitole jsou popsány základní návrhy demo úlohy: Zobrazení systémových informací z PC.

5.1 Použitý software:

Celá práce byla vytvořena a zkušena v operačním systému GNU/Linux distribuci Mandriva 2007, která později byla upgradována na novější systém Mandriva 2008.

5.1.1 KontrollerLab

Velice zdařilý, pěkně propracovaný grafický programátor mikrokontrolérů AVR. Ve standardním zobrazení má pracovní okno rozdělené na čtyři části viz Obr2.

Na pravé straně je pevně ukotveno okno Project manager, velice podobné jako se objevuje i v AVR Studiu firmy Atmel jen s tím rozdílem že u tohoto programu je na levé straně. Toto okno může být plovoucí tzn. můžeme ho přemístit dle libosti, kde se nám líbí, třeba na jinou pracovní plochu, jen ukotvená pozice je stále na levé straně hlavního okna programu.

Mezi Project managerem a oknem pro psaní zdrojových kódů, jak v jazyce ANSI C tak i v jazyce assembler, má místo serial terminal, kde je možno sledovat a řídit veškerou komunikaci při sériovém provozu pomocí sběrnice RS 232 i USB.

Okno pro psaní zdrojového kódu, které je na levé straně, lze maximalizovat i přemístit na libovolnou plochu. Zdrojový kód je psán v programu Kate. Zde je barevně rozlišen text, zda jde o hodnotu zapsanou v hexadecimálním tvaru či nějaký příkaz nebo klíčové slovo.

Program umí programovat pomocí nástroje UISP a také pomocí programu AVRDUDE. Programování UISP a AVRDUDE má v tomto programu širokou možnost nastavení komunikace. V programu je dále implementována knihovna jejímž autorem je Peter Fleury, pro nastavování a komunikaci s LCD displejem. Další možnost je grafické vytváření nových znaků na LCD displej, které nejsou v ASCII tabulce. Mimo jiné tento nástroj umožňuje nastavování programovatelných propojek pro různé mikrokontroléry AVR.

Program do mikrokontroléru jsou možné 2 způsoby programování a to UISP a AVRDUDE. Podporuje celou řadu programátorů ať jsou to komerčně vyráběné programátory od firmy ATMEL, a nebo programátory které lze snadno své pomocí vytvořit.

Nevýhodou je složitější vybírání právě programovaného mikrokontroléru. KontrollerLab má jednoduché ovládání pomocí ikon v hlavním panelu.

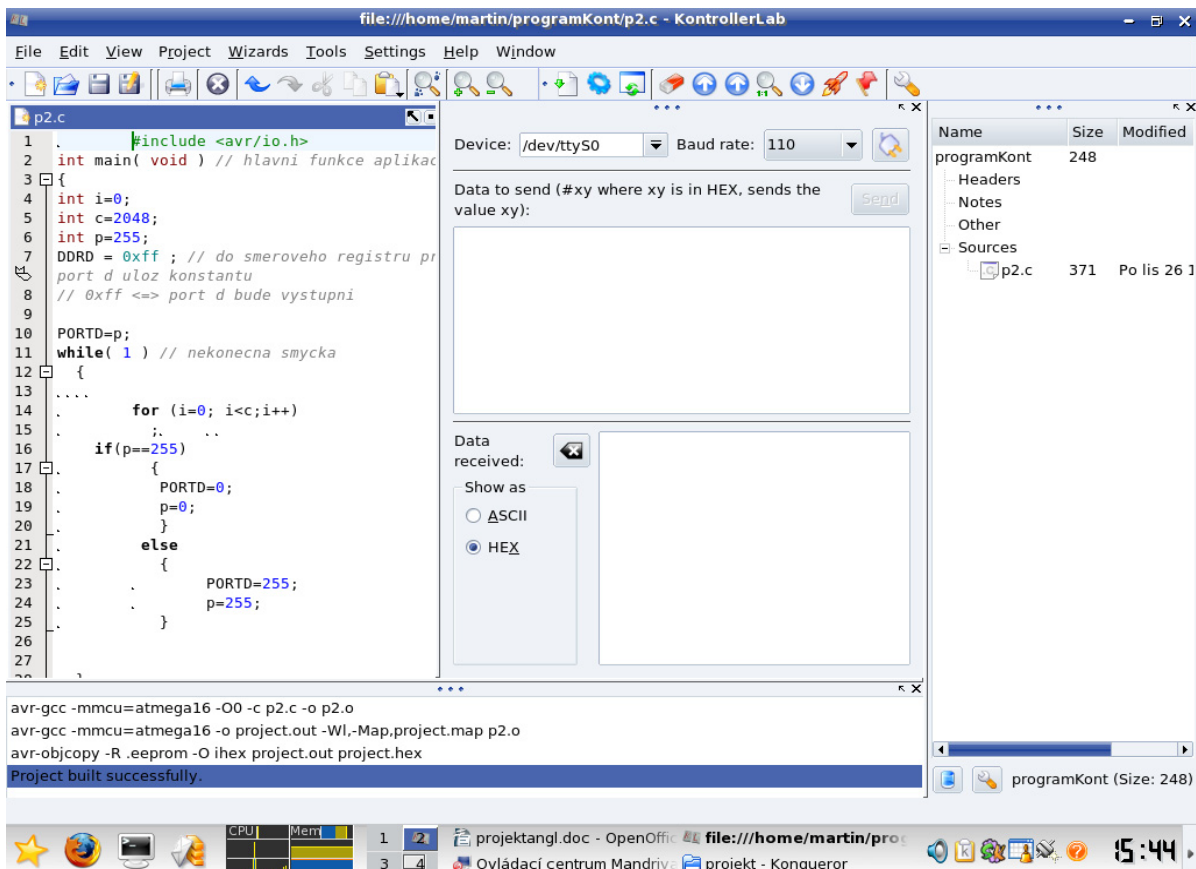
Pro rychlejší obsluhu a vývoj existuje na hlavním panelu tlačítko Ignite, které spustí kompilaci zdrojových souborů a zároveň nahraje program přímo do mikrokontroléru.

Výhodou programu je vlastnost při nastavování programovatelných propojek v tomto grafickém prostředí. U nezakřížkovaný čtverečku je nastavena propojka na hodnotu log. 1 to znamená, že propojka není nastavena, je –li označený čtvereček je programovatelná propojka v pozici log. 0 a propojka je nastavena.

Pro správný proces nahrávání programu do mikrokontroléru je taktéž nastavit práva uživateli pro přístup na paralelní port.

Toto vývojové prostředí má integrováno i přímé vložení knihovny na ovládání LCD displeje od Petera Fleuryho. Je zde možné přesně definovat, na kterém vývodu bude jaký řídící signál a na kterých pinech portu budou i datové vodiče pro komunikaci s radičem HD44780. Další možností u tohoto vývojového prostředí je také možnost nastavení počtu znaků na displeji i také počet řádků. Nevýhodou ovládání této funkce je absence možnosti nastavení portu, kde by mohl být připojen displej. Toto nastavení je nutné upravit až po ukončení nastavování v grafickém prostředí, přímo ve zdrojovém textu knihovny lcd.h . Po ukončení se automaticky vloží do projektu hlavičkový soubor i zdrojový text pro řízení zobrazování pomocí instrukcí radiče HD44780.

Velikost výsledného zkompilevaného textu se zobrazuje v okně Project manager. Ukáže pouze celkovou velikost souboru ale dále neřeší zda zkompilevaná hexa soubor není příliš velký a do mikrokontroléru se nevejde celý.

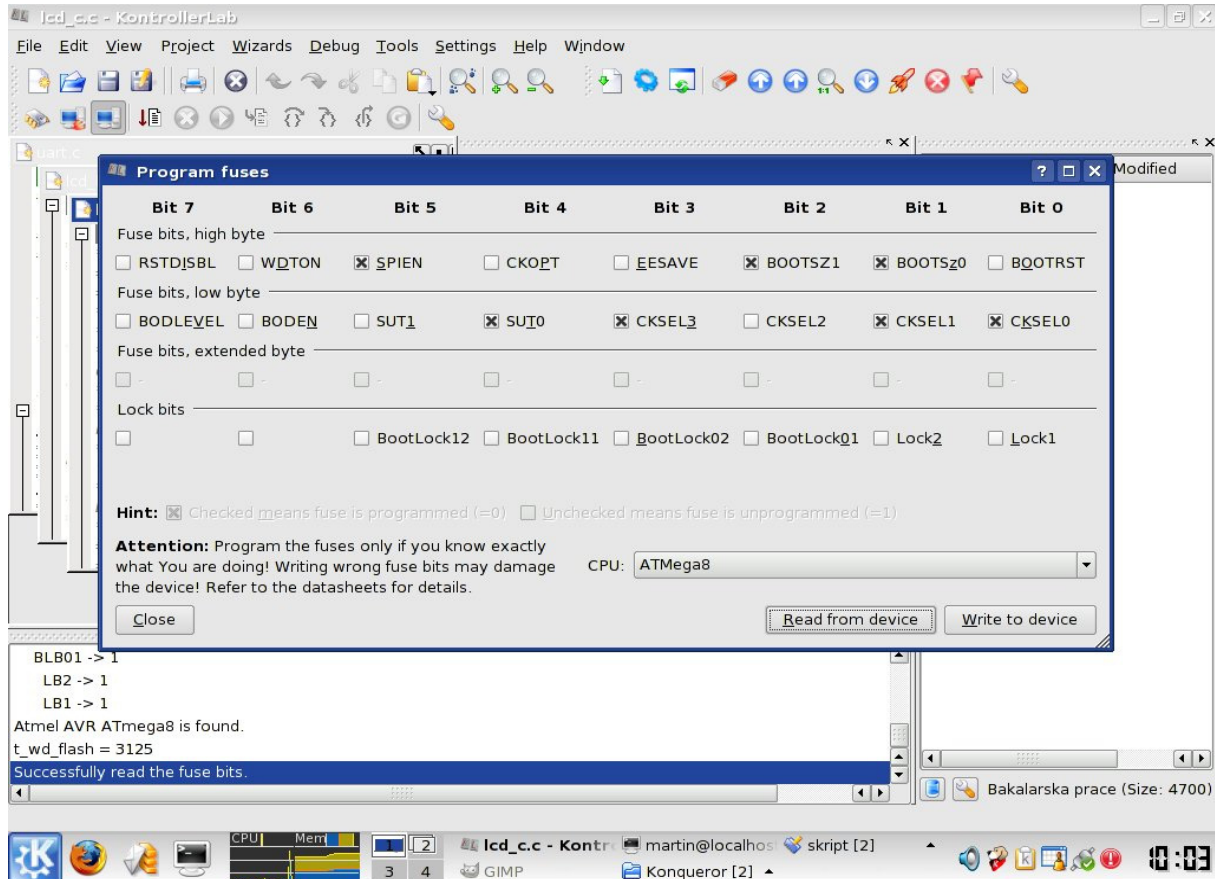


Obr 4. KontrollerLab v systému Mandriva 2007

Novinkou toho vývojového prostředí je i postupná integrace simulátoru do vývojového prostředí. Přibyl ještě jeden panel do celého prostředí kde je zobrazen obsah registrů mikrokontroléru. Tento simulátor s porovnáním se simulátory v AVR studiu je poněkud slabší a chybí třeba i zobrazení stavu integrovaných periférií jako je watchdog, výstupní

porty, také případných integrovaných čítačů/časovačů, stav jednotky UART a jiné. Bohužel během psaní této práce ještě nebyla zveřejněna dokumentace pro nastavení a používání simulátoru v tomto prostředí.

Pro práci s tímto vývojovým prostředím potřebujeme tyto balíčky: avr dude, avr libc, avr base, avr gcc, avr uisp, avr binutils.



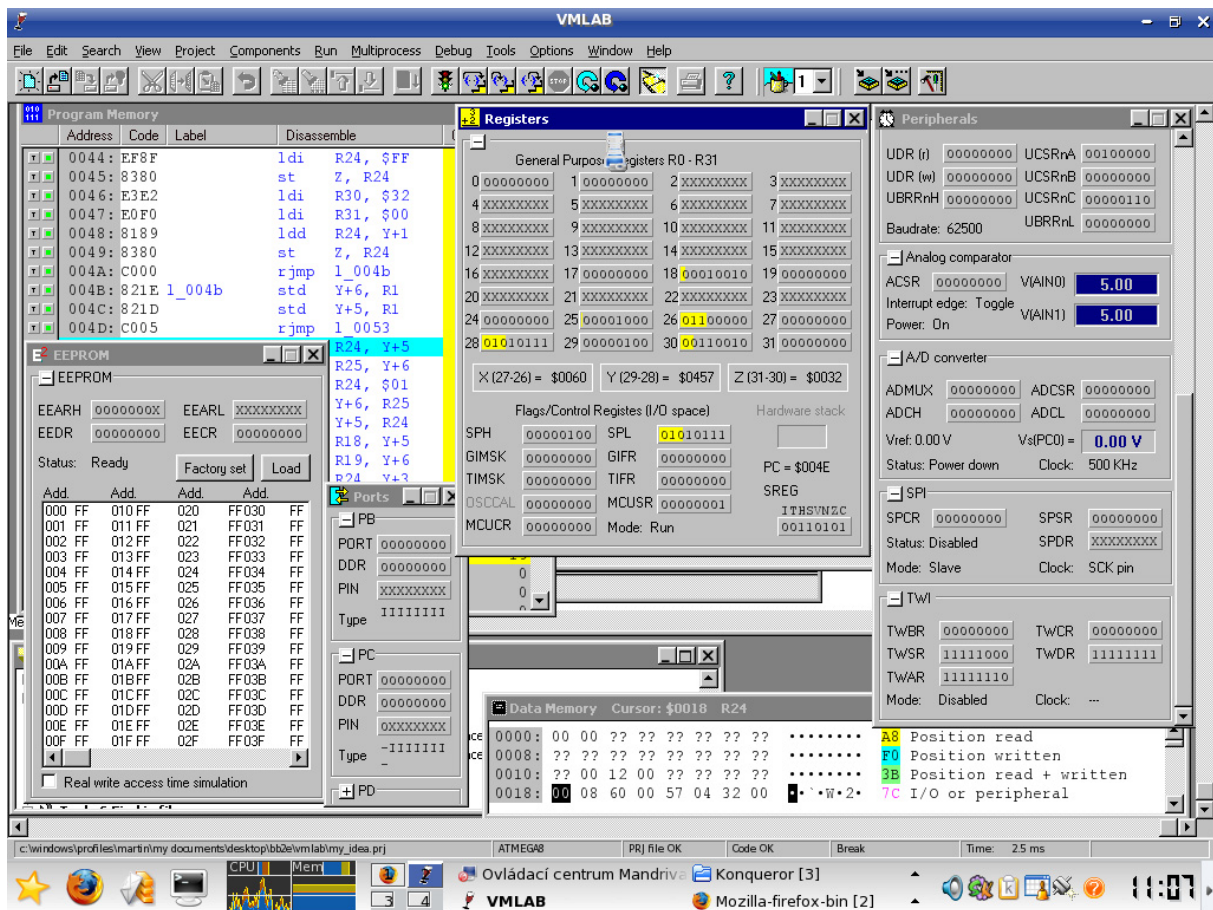
Obr 5. KontrollerLab (Mandriva 2008), možnost nastavení programovatelných propojek

5.1.2 Simulátor Visual Micro Lab 3.12

Nevýhodou tohoto programu je horší editor zdrojových kódu. Vypadá jako by po některých znacích nevynechával mezeru, a tak se zdrojový kód zdá být nečitelný. Tuto chybu je možné opravit vložením mezery mezi jednotlivé znaky. Program rovněž umí zobrazit jednotlivé obsahy pamětí mikrokontrolérů například datovou, programovou paměť. Je zde možnost zobrazit stav registrů, výstupních portů, také případných integrovaných čítačů/časovačů, stav jednotky UART pokud je v mikrokontroléru integrována.

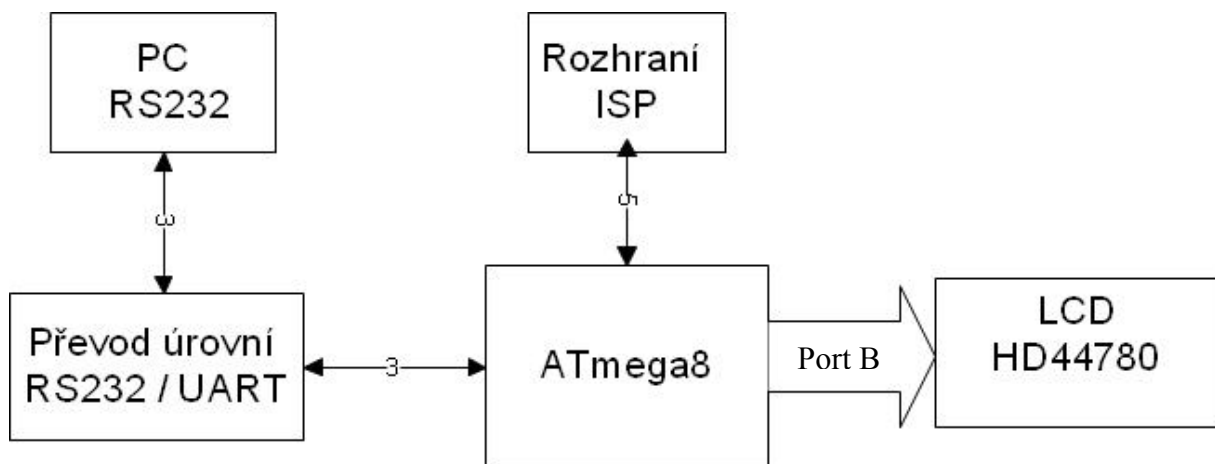
Umožňuje při simulaci krokování běhu programu také i spouštět celý vytvořený program. Nevýhodou tohoto vývojového prostředí vidím v tom, že se program nekrokuje přímo ve zdrojovém textu ale pouze v pamětech a v registrech případně na jiných perifériích. Tento program umožňuje i emulaci hardware jako například vícekanálový osciloskop, monitor sběrnice I2C LCD display, rozhraní RS232 a další.

Další možností simulátoru je použít AVR Studio pro windows opět s emulací pomocí programu wine nebo použít grafické rozhraní pro simulátor GDB.



Obr 6. Visual Micro Lab 3.12 v systému Mandriva 2007 pomocí Wine

5.2 Blokové schéma navrhované demo úlohy



Obr 7. Blokové schéma zobrazovače systémových informací z PC

Pro komunikaci mezi počítačem se sériovým rozhraním RS 232 je použita jen 3 vodičová struktura a to vodiče RxD, vývod 2 na konektoru DE9, a TxD, pin 3 na konektoru DE9, a jeden vodič pro propojení GND, vývod 5 na konektoru DE9, mezi mikrokontrolérem a PC. Následuje blok převodníku úrovní s pomocí obvodu MAX 232. Mikrokontrolér bude připojen taktéž třemi vodiči. Jeden společný (GND) a dva vodiče pro komunikaci přivedeny na PD0 (RxD) a PD1(TxD).

Krystal je použit pouze interní oscilátor, který umí pracovat na kmitočtech 1 MHz, 4 MHz a 8 MHz. Byl zvolen pracovní kmitočet 8 MHz.

Datová komunikace s LCD displejem s řídicím obvodem HD44780 bude používat redukovaný počet datových signálů tedy 4 vodiče a tři řídicí vodiče. Proto by byla možnost použít celý port C, který má jen 7 bitovou architekturu. Bohužel nelze tento port využít, protože poslední pin portu C PC6 je společný pro funkci /Reset a jako vstupně výstupní pin. Možnost využití tohoto pinu je možná, ale musel bych oželit pin /Reset, což není možné z praktického důvodu ztráty možnosti programování mikrokontroléru ATmega8 pomocí rozhraní ISP, které pro správnou funkci potřebuje tento resetovací vstup. Na místo portu C je použita pro komunikaci s LCD displejem port B. Na portu B jsou taktéž některé vývody, které jsou nutné pro ISP programování. Tyto vývody jsou ošetřeny pomocí rezistorů, přes které bude připojen LCD displej a to na pinu E a na dvou datových vodičích DB4 a DB5. Pro jednodušší použití knihoven pro komunikaci s LCD je dobré dodržet zásadu, aby datové vodiče byli na spodní nebo horní polovině portu. V této úloze jsou použity datové vodiče na dolní polovině portu tzn. DB4 je na pinu PB4, DB5 na pinu PB5 atd. Zjednoduší se tím komunikace pomocí externí knihovny pro ovládání displeje.

Pro komunikaci použiji knihovnu, jejímž autorem je Peter Fleury a zjednodušil ji Ing. Tomáš Frýza Ph. D. Tato knihovna je trochu upravena pro mikrokontrolér ATmega 8 a přímo pro zvolený port mikrokontroléru.

Programování mikrokontroléru je zajištěno pomocí sériové metody programování ISP. Pro správnou funkci programování je nutné použít 5 vodičové provedení. Z mikrokontroléru je použita část portu B. Budou to piny PB3, PB4, PB5 a dále propojení se zemí a jeden vodič na vstup /RST (vývod 1), který je nastaven na portu C pin 6.

[17]

5.3 Mikrokontrolér ATmega 8

Pro aplikaci zobrazování systémových informací z počítače byl zvolen mikrokontrolér od firmy Atmel – Atmega 8. Pro účely této demonstrační úlohy by stačil i mikrokontrolér ATtiny 44. ATmega 8 byl zvolen z důvodu špatné dostupnosti těchto mikrokontrolérů na českém trhu s elektronickými součástkami.

Mikrokontrolér je založen na RISCovém jádru AVR Hardwarové architektury. V tomto mikrokontroléru jsou integrovány tři „ plné „ porty, rozhraní pro komunikaci USART a další. Součásti jsou integrovány v 28 pinovém pouzdru vyráběné v provedení klasickém i pro povrchovou montáž SMD. V úloze je použita verze mikrokontroléru v provedení SMD. Blokové schéma je zobrazeno na Obr 1.

Základní vlastnosti mikrokontroléru ATmega 8:

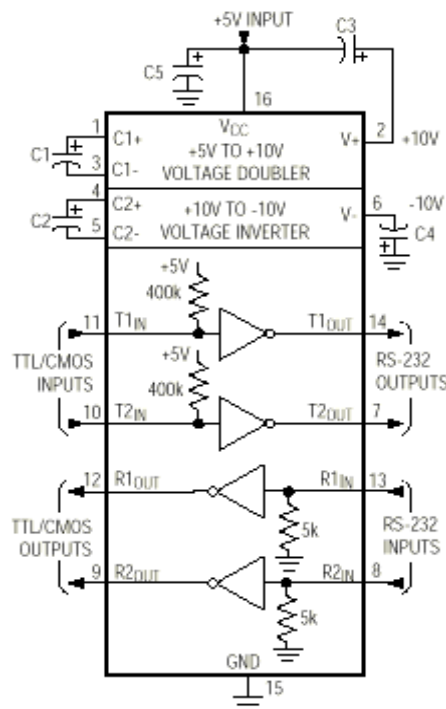
- 8 kB paměti FLASH
- 1kB RAM
- 512 B EEPROM

- dva 8 bitové čítač/časovač s předděličkou
- jeden 16 bitový čítač/časovač
- tři PWM kanály
- na čipu integrovaný 6x10bit A/D převodník
- programovatelný Watchdog
- USI – universální sériové rozhraní
- plně duplexní USART
- možnost programování ISP pomocí SPI portu
- analogový komparátor
- a další...

[18]

5.4 Převodník úrovní RS232 / UART – MAX 232

MAX 232 - jedná se o převodník úrovní TTL na RS 232. Obsahuje dvě dvojice oddělovačů konvertujících napěťové úrovně. Napětí pro RS 232 se získává pomocí nábojové pumpy. Integrovaný obvod MAX 232 sestává ze dvou budičů seriových linek, dvou přijmačů seriových linek a ze dvou měničů (násobiče a invertoru), které vytvářejí další napájecí napětí. MAX 232 je vhodný i pro aplikace vyžadující obousměrný přenos dat. Budiče i přijímače splňují specifikace norem EIA RS-232-C, vyhovují proto přenosovým rychlostem do 20000 bitů/s, nepřevyší-li se zatěžovací kapacita výstupu budiče 2500 pF. V technických datech integrovaného obvodu MAX 232 je uvedeno, že interní měniče generují napětí typicky -9V a +9V při napájecím napětí $5V \pm 10\%$ a zatěžovacím odporu budiče 3k. Kapacity kondenzátorů použité v obvodech nejsou kritické a mohou být za cenu zvětšení vnitřního odporu a zvlnění výstupního napětí měničů zmenšeny až na 2 μ F, pokud není vyžadován provoz obvodu při dolní hranici povolených pracovních hodnot. Zvýšené napětí je možno získat na pinech 2 a 6 a použít pro další obvody viz. Obr 8.



Obr 8. Aplikační zapojení obvodu MAX 232 [převzato 19]

5.5 LCD displej s řadičem HD44780

S řadičem HD44780 pracuje naprostá většina znakových displejů. Vyrábějí se provedení od 1x8 znaků do 4x40 znaků. V této úloze použijí verzi 2x16 znaků. K propojení s jednočipem je třeba 4 nebo 8 datových vodičů, jeden na přepínání zápisu instrukcí / dat (RS) a další s hodinovým signálem (E). Pro případné čtení obsahu displeje je třeba připojit ještě jeden (R/W), jinak je trvale uzemněn. Základní znakovou sadu lze doplnit osmi vlastními znaky, které jsou pak dostupné pod kódy 0-7 a 8-15.

V této úloze je zvolena 4 bitová komunikace. Při tomto druhu komunikace jsou využity jen 4 datové vodiče DB4-DB7 a zbytek je uzemněn. Data nebo instrukce jsou vysílány vždy nadvrát. Nejprve se na DB4-DB7 pošle horní polovina bytu, vygeneruje se kladný pulz na E, na sběrnici se pošle spodní polovina bytu a generuje se druhý pulz na E.

Vyjímku tvoří první instrukce, která inicializuje displej jako 4-bitový. Zde se pošle jen horní polovina inicializačního bytu a jen jeden pulz na E. Celý byte se pak vyšle znovu, ale už 4-bit způsobem, aby se nastavily i spodní 4 bity slova (počet řádků a font).

Pro jednodušší použití knihoven pro komunikaci s LCD je dobré dodržet zásadu, aby datové vodiče byli na spodní nebo horní polovině portu. [21]

Tab1 .Číslování a význam vývodu LCD displejů s řadičem HD44780

| Vývod | Název | Funkce |
|-------|---------|------------------------------------|
| 1 | Vss | GND |
| 2 | Vcc | napájení +5V |
| 3 | Vee, V0 | nastavení kontrastu |
| 4 | RS | volba mezi 0 - instrukce, 1 - data |
| 5 | R/W | volba mezi 0 - zápis, 1 - čtení |
| 6 | E | hodinový vstup |
| 7 | DB0 | data 0 |
| 8 | DB1 | data 1 |
| 9 | DB2 | data 2 |
| 10 | DB3 | data 3 |
| 11 | DB4 | data 4 |
| 12 | DB5 | data 5 |
| 13 | DB6 | data 6 |
| 14 | DB7 | data 7 |
| 15 | LED+ | anoda diody podsvětlení |
| 16 | LED- | katoda diody podsvětlení |

5.6 Programátor ISP

Jako programátor byl použit jednoduchý programátor, který lze snadno vytvořit. Je složen pouze z konektoru Canon 25 a 5 vodičů. Bylo vycházeno ze zapojení jehož autorem je Brian S. Dean. Toto zapojení je podporováno i v použitém vývojovém prostředí KontrollerLab. Zapojení totiž je vytvořeno pro programy UISP a také AVRDUDE které používá i toto vývojové prostředí.

Zapojení propojovacího kabelu:

Tab 2. Zapojení programovacího kabelu ISP

| Paralelní port | Funkce programátoru |
|----------------|---------------------|
| Data 6 (Pin 7) | RST |
| Data 7 (Pin 8) | SCK |
| Data 8 (Pin 9) | MOSI |
| /ACK (Pin 10) | MISO |
| GND (Pin 18) | GND |

Nevýhodou tohoto programátoru je už pomalu chybějící podpora paralelního portu od výrobců základních desek. V dnešní době jsou všichni nuceni přecházet na sériové rozhraní USB.

[20]

5.7 Napájecí zdroj

Odběr demo úlohy je cca. 75 mA a je napájena z 5V. Napájení přímo z linky RS232 je vyloučeno, protože z této sběrnice lze odebírat maximálně 15mA. Tyto parametry splňuje i možnost napájení z rozhraní USB v jeho základním módu.

Jako napájecí zdroj použijí možnost získat +5V z rozhraní počítače z USB(Universal Seriól Bus).

USB rozhraní se během posledních let stalo zcela běžnou součástí spotřební elektroniky připojitelné k počítači a již téměř vytlačilo klasický sériový port RS232 a dokonce i paralelní port. Koupit dnes notebook s klasickým sériovým rozhraní začíná být problém a proto jsou nuceni i výrobci a uživatelé speciálních aplikací postupně přecházet na USB. USB zařízení mohou být napájena přímo z USB sběrnice, pokud jejich odběr nepřekročí 100mA , případně 500mA a to může být maximálně jedno zařízení na celé USB sběrnici. Napájecí napětí je 5V. Vytvořil jsem si propojovací vodič pro propojení s mým výrobkem. Je zapojen na pinu číslo 1 jako napájecí napětí + 5V a na pinu 4 na zemnicí potenciál. [24]

Tab 3. Zapojení konektoru USB typ A

| Pin | Jméno | Popis |
|-----|-------|----------|
| 1 | VBus | + 5 V DC |
| 2 | D- | Data - |
| 3 | D+ | Data + |
| 4 | GND | Ground |

6. Firmware demo úlohy

6.1 Jednotka USART a její nastavení

V tomto zapojení je použita jednotka USART jen v asynchronním režimu UART.

Asynchronní přenos dat přenáší data jen v určitých sekvencích. Data jsou přenášena danou rychlostí uvozen startovací sekvencí, na kterou se synchronizují všechna přijímací zařízení. Obě strany ať přijmač nebo vysílač mají vlastní synchronizační oscilátor pomocí kterého odečítají data v přesně definovaných časových intervalech. Dále můžeme nastavit jednoduché zabezpečení pomocí paritních bitů. Paritu můžeme nastavit sudou nebo lichou. Po té jsou data ukončena jedním nebo dvěma Stop bity. Pro další komunikaci je nutné opět vyslat startovací bit. Asynchronní přenos není výhodný pro přenos velkého objemu dat. Je zde až o 20% menší přenosová rychlost užitečných dat při stejné rychlosti komunikace, vzhledem k nutnosti startovacích a paritních bitů.

Vlastnosti USART jednotky ATmega8:

- plně duplexní komunikace
- možnost volby asynchronní a synchronní komunikace
- možnost volit přenosovou rychlost pomocí nastavení předděličky
- volit si vlastnosti rámce UART (1 startbit, 1, nebo 2 stop bity, velikost sériových rámců 5 až 9 datových bitů, volitelný paritní bit)
- nastavení liché nebo sudé parity
- detekce falešného start bitu
- detekce chyb v rámci
- přerušení generované při ukončení vysílání(prázdný vysílací registr) nebo při přijetí celého rámce
- šumové filtry
- asynchronní mód komunikace s dvojnásobnou rychlostí

Nastavení přenosové rychlosti se provádí pomocí 16-ti bitového registru UBRR, který tvoří předděličku kmitočtu na kterém pracuje mikrokontrolér. Výpočet hodnoty se provádí pomocí vzorce v mém případě použiji normální režim přenosové rychlosti.

$$UBRR = \frac{f_{cpu}}{16 * SR} - 1 = \frac{8000000}{16 * 9600} - 1 \approx 51$$

UBRR – hodnota UBRR registru pro požadovanou přenosovou rychlost

f_{cpu} – frekvence procesoru v Hz

SR – symbolová rychlost v baudech

Zvolil jsem interní oscilátor s hodnotou 8 MHz. Pro můj případ spočtené hodnoty obsahu registru UBRR jsou v tabulce 2.

Tab 4. Hodnoty předděliče UBRR pro krystal 8 MHz [převzato 18]

| Baud Rate (bps) | $f_{osc} = 8.0000 \text{ MHz}$ | | | |
|--------------------|--------------------------------|-------|---------|-------|
| | U2X = 0 | | U2X = 1 | |
| | UBRR | Error | UBRR | Error |
| 2400 | 207 | 0.2% | 416 | -0.1% |
| 4800 | 103 | 0.2% | 207 | 0.2% |
| 9600 | 51 | 0.2% | 103 | 0.2% |
| 14.4k | 34 | -0.8% | 68 | 0.6% |
| 19.2k | 25 | 0.2% | 51 | 0.2% |
| 28.8k | 16 | 2.1% | 34 | -0.8% |
| 38.4k | 12 | 0.2% | 25 | 0.2% |
| 57.6k | 8 | -3.5% | 16 | 2.1% |
| 76.8k | 6 | -7.0% | 12 | 0.2% |
| 115.2k | 3 | 8.5% | 8 | -3.5% |
| 230.4k | 1 | 8.5% | 3 | 8.5% |
| 250k | 1 | 0.0% | 3 | 0.0% |
| 0.5M | 0 | 0.0% | 1 | 0.0% |
| 1M | - | - | 0 | 0.0% |
| Max ⁽¹⁾ | 0.5 Mbps | | 1 Mbps | |

Jednotka UART pracuje správně i při vyšších chybách hodnot předděličky. Není nutné používat externí přesný krystal. USART se nastavuje pomocí tří registrů UCSRA, UCSRB, UCSRC. Z těchto registrů se také čtou stavové informace potřebné k běhu programu.

Můj asynchronní přenos bude používat jeden stop bit. Symbolová rychlost je nastavena na 9600 Bd.



Obr 9. Struktura asynchronního rámce se dvěma stop bity a sudou paritou a sedmi datovými bity

6.2 Ovládání LCD displeje s řadičem HD 44780

Pro komunikaci s LCD displejem použiji knihovnu, která byla vytvořena Peterem Fleurem a dále ji pak pro potřeby předmětu Mikroprocesorová technika upravil Ing. Tomáš Frýza Ph.D. Tato knihovna je lehce upravena, protože byla vytvořena pro displej 16x4 a v zařízení je použit jen dvou řádkový displej.

6.3 Programovatelné konfigurační propojky

Kromě běžných pamětí EEPROM a FLASH má ATmega 8 ještě další registry kde se nastavují specifické vlastnosti mikrokontroléru pro danou aplikaci, které se jednou nastaví a pak dále není nutné jejich nastavení měnit. Jde pomocí těchto propojek dokonce zakázat čtení/zápis obsahu pamětí mikrokontroléru. Pokud jsou tyto propojky nenaprogramované mají hodnotu logické jedničky a pokud je propojka naprogramovaná je její hodnota logická nula.

Bylo použito nastavení programovatelných propojek pro nastavení vnitřního RC oscilátoru, protože nelze používat externí přesný krystal, protože jsou použity vstupy, na které se připojuje tento krystal jako výstupní piny pro řízení komunikace a přenosu dat mezi mikrokontrolérem a LCD displejem. Propojka CKOP bude nastavena na hodnotu 0 což určuje použití interního CR oscilátoru. Hodnota propojek CKSEL je nastavena na 8 MHz.

Ukázka nastavení propojek je v tab. 4.

Tab 5. Nastavení vnitřního RC oscilátoru ATmega8

| CKSEL 3.0 | Frekvence [MHz] |
|-----------|-----------------|
| 0001 | 1,0 |
| 0010 | 2,0 |
| 0011 | 4,0 |
| 0100 | 8,0 |

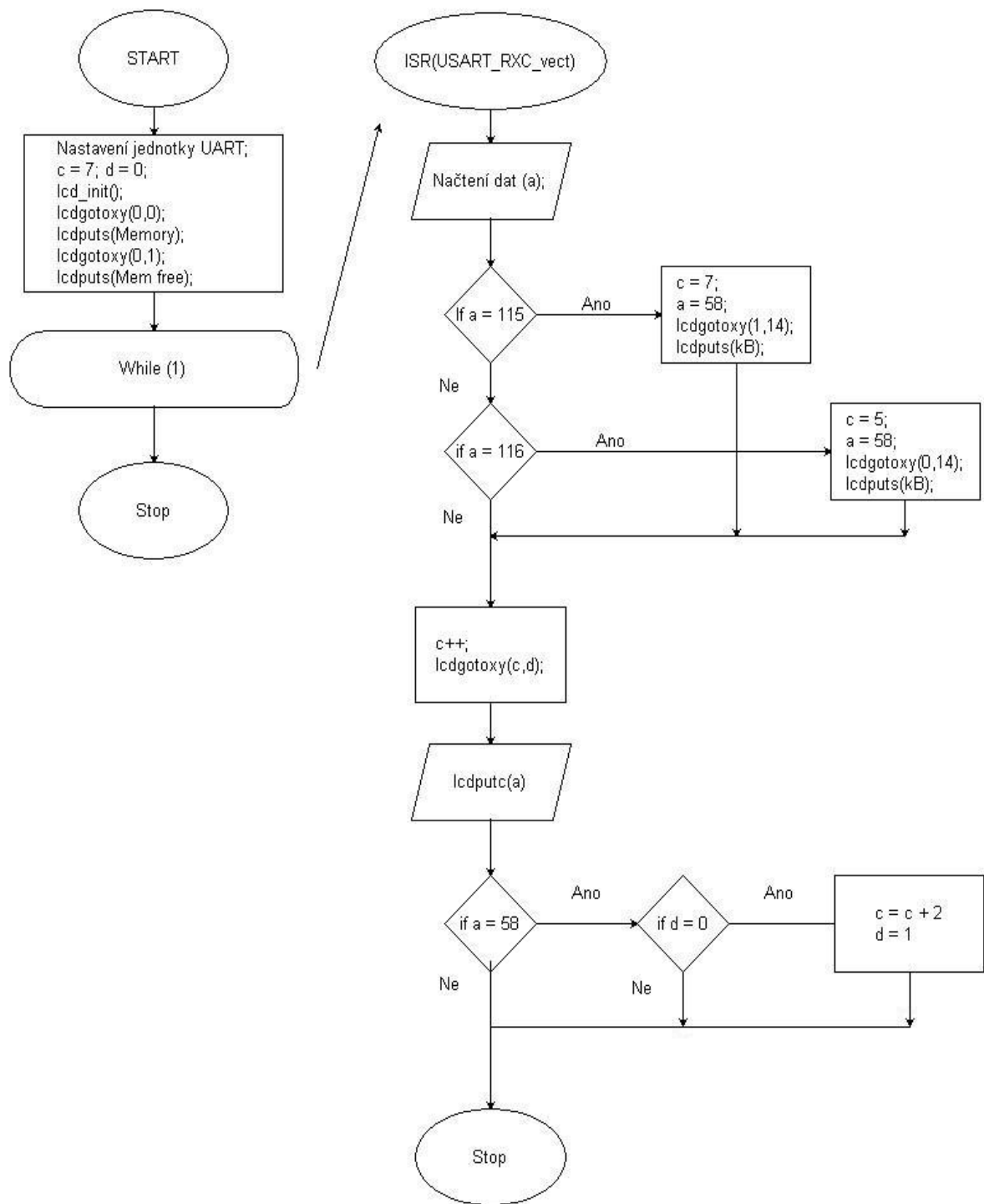
Ostatní nastavení programovatelných propojek zůstávají stejné jako originálně nastavené z výroby.

6.4 Skript posílající systémové informace

Pro posílání informací z počítače byl vytvořen jednoduchý skript v shellu, který používá informace ze souboru MEMINFO ve složce /proc. Skript v shellu je analogické s vytvářením dávkových souborů v DOSu. Skript pro shell se skládá z jednotlivých příkazů, které normálně píšete do příkazové řádky. Skript vezme hodnotu velikosti operační paměti ze tohoto souboru a pošle ji na výstup na port COM1. Po té mi pošle synchronizační znak o ukončení posílání velikosti operační paměti. Dále cyklicky bude posílat data o volné operační paměti v počítači a ty budou dále odeslána na sériový port společně se synchronizačním znakem. Jeho zdrojový kód je v příloze F a na příloženém CD.

6.5 Program mikrokontroléru

Jde o jednoduchý program, který řídí jednotku USART a LCD displej s řadičem HD44780. Přijímá data po rozhraní RS232 a dále je zobrazuje s popisem na LCD displeji. Zdrojový kód demo úlohy je přiložen v elektronické podobě na příloženém CD.



Obr. 10 Vývojový diagram programu mikrokontroléru

7. Porovnání řešení se stávajícími možnostmi na UREL

Programování a využívání zkušebních desek pro návrhy programů pro mikrokontroléry je velice důležité. Nejen že začátečníkům poskytují odrazový můstek pro psaní zdrojových kódů a vývojářům pro stavbu prototypů a testování nových aplikací.

7.1 Porovnání Software

Pro tvorbu zdrojových kódů lze použít libovolný textový editor. Některé textové editory v systému GNU/Linux, jako je například Kate, zároveň umožňují a i barevné odlišení textu, zda jde o klíčová slova, konstanty apod. Tímto způsobem rozlišení se velice zjednoduší a zpřehlední psaný zdrojový kód.

Kompilátor a linker pro jazyk C pro mikrokontroléry AVR je pro systém Windows převzat a používá se původně vyvíjený pro Linux AVR-GCC. Knihovny pro jazyk C jsou též využívány v programu WinAVR, který je nutné nainstalovat pro práci s jazykem C v AVR studiu.

Jako assembler lze na platformě Linux najít hned několik programů, které jsou plně kompatibilní s assembly pro mikrokontroléry AVR.

Programy pro nahrávání programu do mikrokontrolérů je v systému GNU/Linux velké množství. Podporují přímo originální vývojové kity od výrobce mikrokontrolérů, také různé zjednodušené verze programátorů, které jsou implementovány přímo v těchto programech. Pro praktickou ukázkou jsem použil jednoduchý programovací kabel, což je ukázkou nejjednodušší možnosti vytvoření programátoru.

Simulátory jsou největší slabinou podpory mikrokontrolérů v systému GNU/Linux. Simulátory běžící přímo pod Linuxem je několik, ale nedosahují takových kvalit jako je simulátor vyvíjený firmou Atmel, jež je v AVR studiu. Zajímavou alternativou je simulátor GDB. Tento simulátor se ovládá přes příkazovou řádku, i když započal i vývoj grafického rozhraní pro pohodlné ovládání toho programu. Výhodou je možnost simulace i přímo na mikrokontroléru pomocí JTAG a také možnost vzdáleného ovládání simulátoru přes počítačovou síť. Další možností simulace je projekt Avrora. Tento program je napsán v jazyce Java. Je stále vyvíjen a v poslední době je pro něj také vytvářeno grafické rozhraní. Další možností simulace je za použité simulátoru Simulavr. Bohužel tento projekt začal stagnovat a jeho vývoj je nejistý. Je ovládán z příkazové řádky a umožňuje i výpis do nového okna viz Obr 2.

V systému GNU/Linux jsou i celé vývojové prostředí. KontrollerLab je velice pěkné prostředí, v dnešní době i se simulátorem, které může nahradit i AVR studio. Bohužel nebyla zveřejněna dokumentace pro správné nastavení simulátoru v prostředí, takže jsem prakticky nevyzkoušel možnosti simulace. Další možností vývojového prostředí pracující nad emulátorem Wine je Visual Micro Lab. Toto prostředí je zdařilé a autoři tvrdí, že pracuje správně pod tímto emulátorem. Nevýhodou oproti KontrollerLab je slabá podpora programátorů. Pomocí emulátoru Wine funguje v systému GNU/Linux také AVR studio vyvíjené firmou Atmel.

7.2 Podpora Hardware

Na ústavu radioelektroniky jsou používány vývojové kity dva. Jeden je originálním výtvozem výrobce mikrokontrolérů AVR firmou Atmel. Jde o vývojový kit STK 500. Tento kit umožňuje programování ISP a také podporuje paralelní programování. Podporuje také paralelní i sériové programování vysokým napětím. Připojuje se k počítači pomocí sériové linky RS 232. Má také druhý konektor s RS 232 rozhraním pro použití v uživatelské aplikaci. Na desce jsou patice s 8, 20, 28 a 40 vývody pro připojení mikroprocesorů AVR. Vstupně výstupní porty jsou uživateli zpřístupněny prostřednictvím konektorů, takže je lze použít ve spojení s externími zařízeními, nebo je možné využít ovládacích tlačítek a signalizačních LED umístěných přímo na vývojové desce. Samostatný sériový port RS-232 může být připojen k jakýmkoliv I/O vývodům. Pro potřeby vývoje aplikací s různými mikroprocesory je vývojová deska STK500 koncipována jako široce variabilní, takže lze nastavovat prakticky vše od kmitočtu oscilátoru až po napájecí napětí mikroprocesorů pomocí zkratovacích propojek. K dispozici je též objímka pro připojení externího krystalu. Při použití interního oscilátoru desky však lze hodinovou frekvenci mikroprocesorů AVR a jejich napájecí napětí rovněž pohodlně nastavovat z prostředí AVR Studia. Ve vývojovém prostředí KontrollerLab by měl být dobře podporován, protože toto prostředí používá avrdude i uisp programy s podporou této vývojové desky. Bohužel se mi prakticky nepodařilo s tímto vývojovým kitem spojit a pracovat v něm, i když v manuálových stránkách je popsáno jak se se zařízením spojit. O spojení jsem se snažil v distribuci systému Linux Mandriva 2007, Mandriva 2008 a také Fedora 8. Další program s podporou vývojového kitu STK 500 je Visual Micro Lab.

Dále se na ústavu radioelektroniky používá vývojový kit jejímž autorem je Ing. Jiří Šebesta Ph.D. Tato deska používá pro komunikaci s AVR studiem modul AVRISP mkII. Tento modul se k počítači připojuje pomocí USB portu. Tento modul podporuje jen ISP programování se všemi jeho vymoženostmi. Umožňuje nastavování programovací frekvence 50 Hz – 8 MHz.

Podporu v systému GNU/Linux by měl mít taktéž jako vývojový kit STK 500. Bohužel nedošlo k praktickému vyzkoušení celého propojení a možnosti řízení a nahrávání programů do mikrokontrolérů. Modul, kterým probíhá komunikace s počítačem by měl být podporován v programu avrdude. Tento program je také implementován ve vývojovém prostředí, které jsme používal pro praktickou ukázkou.

8. Závěr:

Programovacích nástrojů v GNU/LINUX je mnoho, totéž platí i o programování mikrokontrolérů Atmel AVR. Existuje spousta nástrojů jak pro práci na konzole tak pro grafické rozhraní. Velice se rozšířil programovací kit AVR Cross Development Kit . Je to soubor programovacích nástrojů, jak pro simulaci tak kompilaci a nahrávání programu do mikrokontrolérů. Jde vlastně komplexní řešení s možnostmi plně programovat a pracovat s mikrokontroléry. Většina těchto programů pracuje v konzoli. Spoustu nástrojů z tohoto kitu bylo exportováno i do Windows jako program WinAVR.

Jako dobrý nástroj mi připadá také KontrollerLab, který jsme použil pro svoji praktickou část. Jeho nevýhodou je absence simulátoru. I když v poslední době autoři tohoto projektu začlenili simulátor do tohoto vývojového prostředí. Bohužel jsem prakticky nevyzkoušel, ještě nebyla zveřejněna dokumentace pro nastavení a používání toho simulátoru.

Největší slabinou práce s mikrokontroléry v systému GNU/LINUX jsou simulační nástroje. Je pravda, že v poslední době se zvýšila aktivita při vývoji. Nejlepší simlátor pod Linuxem je asi AVR studio a Visual Micro Lab, které pracují pomocí emulátoru Windows API Wine. Další možností je použít program z AVR Cross Development Kitu jako je SimulAvr nebo avr-gdb. Ty to simulátory zatím nemají plnohodnotné ovládání pomocí grafického prostředí. Simulátor avr-gdb umožňuje spousty možností i například vzdálenou simulaci apod.

Výhodou programování v tomto operačním systému je široká podpora programátorů. Není nutné zakupovat si drahé programátory, postačí například jen jednoduchý programovací kabel.

Nasazení systému GNU/LINUX pro výuku a vývoj aplikací ve srovnání s předmětem Mikroprocesorová technika by bylo ještě příliš předčasné. Není v tomto operačním systému vyřešena možnost simulace do jednoho vývojového prostředí. Ve vývojovém prostředí KontrollerLab nefungoval vývojový kit STK 500, i když podle autorů by měl správně pracovat. Je v tomto prostředí také podporován modul pro programování AVRISP mkII. Bohužel nebylo prakticky vyzkoušena možnost propojení a správné spolupráce. Pro vývojový kit STK 500 je ještě jedno možnost používání Visual Micro Lab, které má podporu pouze tohoto kitu a ST6 EPROMmer. Podporu modulu AVRISP mkII toto vývojové prostředí nemá. Poslední možností je zprovoznit pomocí programu Wine AVR studio. V dnešní době se pracuje na skriptu, který zjednodušuje spouštění AVR studia pod Linuxem. Pro správnou funkci je nutné také nainstalovat internetový prohlížeč Internet Explorer, pro správnou funkci nápovědy. Pokud by nebyl nainstalován tento webový prohlížeč program nahlásí chybu a jsme nuceni jej ukončit.

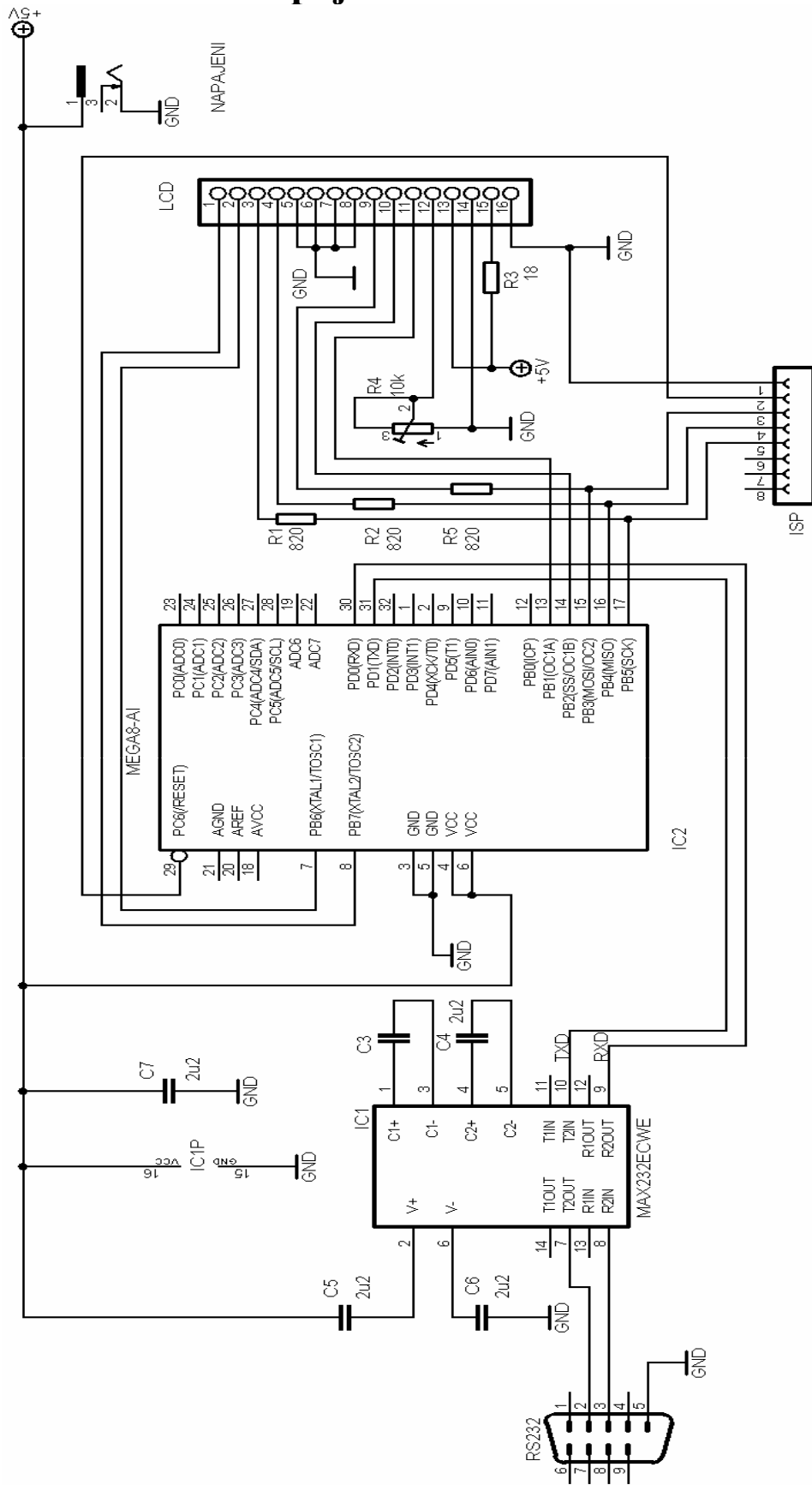
Většina popsaných projektů a programů je stále vyvíjena, proto nevylučuji možnost nasazení těchto nástrojů s postupem času kdy budou schopny plně nahradit stávající možnosti.

Použitá literatura a odkazy na zdroje:

- [1] - Martin Pokorný – Úvod do mikrokontrolerů Atmel AVR [online] dostupný z WWW: <<http://noel.feld.cvut.cz/vyu/scs/prezentace2005/ATmega/ATmega.pdf>> [cit. 20.12 2007]
- [2] - Pavel Čeleda - UISP - AVR In-System Programmer [online] dostupný z WWW: <<http://www.hw.cz/Teorie-a-praxe/Software/ART970-UISP---AVR-In-System-Programmer.html>> poslední úpravy 26.9. 2003 [cit. 20.12 2007]
- [3] - Programming the AVR microcontroller with GCC, libc 1.0.4 [online] dostupný z WWW: <<http://www.tuxgraphics.org/electronics/200411/article352.shtml>> [cit. 20.12 2007]
- [4] - AVR Libc 1.6.1 [online] dostupný z WWW: <<http://www.nongnu.org/avr-libc/user-manual/index.html>> [cit. 20.12 2007]
- [5] - Simulavr [online] dostupný z WWW: <<http://www.nongnu.org/simulavr/>> poslední úpravy 22.12. 2004 [cit. 20.12 2007]
- [6] - avrp – Atmel AVR programmer software for Linux and Other [online] dostupný z WWW: <<http://www.omegav.ntnu.no/~jonah/el/avrp.html>> [cit. 20.12 2007]
- [7] – AVaRICE [online] dostupný z WWW: <<http://avarice.sourceforge.net/>> poslední úpravy 7.12. 2004 [cit. 20.12 2007]
- [8] - Colin O'Flynn - Using AVR-GDB and AVaRICE Together [online] dostupný z WWW: <http://winavr.sourceforge.net/AVR-GDB_and_AVaRICE_Guide.pdf> [cit. 20.12 2007]
- [9] - AVR-GCC Programming Guide [online] dostupný z WWW: <<http://electrons.psychogenic.com/modules/arms/art/3/AVRGCCProgrammingGuide.php>> [cit. 20.12 2007]
- [10] - KontrolleLab [online] dostupný z WWW: <<http://www.cadmaniac.org/projectMain.php?projectName=kontrollerlab§ion=main>> [cit. 20.12 2007]
- [11] - Matthew R. McDougal - AVR Plugins for Elipse [online] dostupný z WWW: <<http://areciv.com/~doogie/index.php?aid=4>> [cit. 20.12 2007]
- [12] - Elipse (vývojové prostředí) [online] dostupný z WWW: <[http://cs.wikipedia.org/wiki/Eclipse_\(v%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD\)](http://cs.wikipedia.org/wiki/Eclipse_(v%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD))> poslední úpravy 15.12. 2007 [cit. 20.12 2007]
- [13] - PonyProg , Serial device programmer [online] dostupný v WWW: <<http://www.lancos.com/prog.html>> poslední úpravy 8.5. 2007 [cit. 20.12 2007]
- [14] - Aurora – The AVR Simulator [online] dostupný z WWW: <<http://compilers.cs.ucla.edu/aurora/simulator.html>> [cit. 20.12 2007]
- [15] - ninevoltz9. Running AVRStudio on Linux 23.9 2006 dostupné z <<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=41478>> [cit. 20.12 2007]
- [16] - VMLAB for ST6/AVR microcontrollers [online] dostupný z WWW: <<http://www.amctools.com/vmlab.htm>> [cit. 20.12 2007]
- [17] - Matoušek David. Práce s mikrokontroléry ATMEL AVR 1.vyd. Praha: BEN- technická literarura, 2003. 376s. ISBN 80-7300-088-1
- [18] - ATmega8 microcontroller [online], Atmel, 2007 dostupný z WWW: <http://atmel.com/dyn/resources/prod_documents/doc2486.pdf>
- [19] - Ing. NETUKA Jan, Integrovaný obvod MAX 232 a jeho použití, Amatérské radio, 1992, roč. 41,č.2,s. 68-69.
- [20] - Brian S. Dean AVRDUDE (Formerly AVRPROG) [online] dostupný z WWW: <<http://www.bsddhome.com/avrdude>>

- [21] – S. Maslan [online] - Ovládání znakových LCD s řadičem HD44780 dostupný z WWW:
< <http://elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>>
- [22] – CDK4AVR -- AVR Cross Development Kit [online] - dostupný z WWW:
<<http://cdk4avr.sourceforge.net/>> [cit. 19.5 2008]
- [23] – AVR STK 500 – User Guide[online] - dostupný z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc1925.pdf> [cit. 19.5 2008]
- [24] – Řehák Jan - USB - Universal Serial Bus - Popis rozhraní [online] - dostupný z WWW:
<<http://hw.cz/Teorie-a-praxe/Dokumentace/ART327-USB---Universal-Serial-Bus---Popis-rozhrani.html>> [cit. 19.5 2008]
- [25] – Kolektiv autorů –Assembler[online] - <<http://cs.wikipedia.org/>> [cit. 19.5 2008]

Příloha A – Schéma zapojení:



Příloha B – Tabulky zapojení konektorů:

Tab P1. Zapojení konektoru LCD

| Pin: | Zapojení: |
|------|-----------|
| 1. | DB7 |
| 2. | DB6 |
| 3. | DB5 |
| 4. | DB4 |
| 5. | DB3 |
| 6. | DB2 |
| 7. | DB1 |
| 8. | DB0 |
| 9. | E |
| 10. | R/W |
| 11. | RS |
| 12. | V0 |
| 13. | VDD |
| 14. | VSS |
| 15. | LED A |
| 16. | LED K |

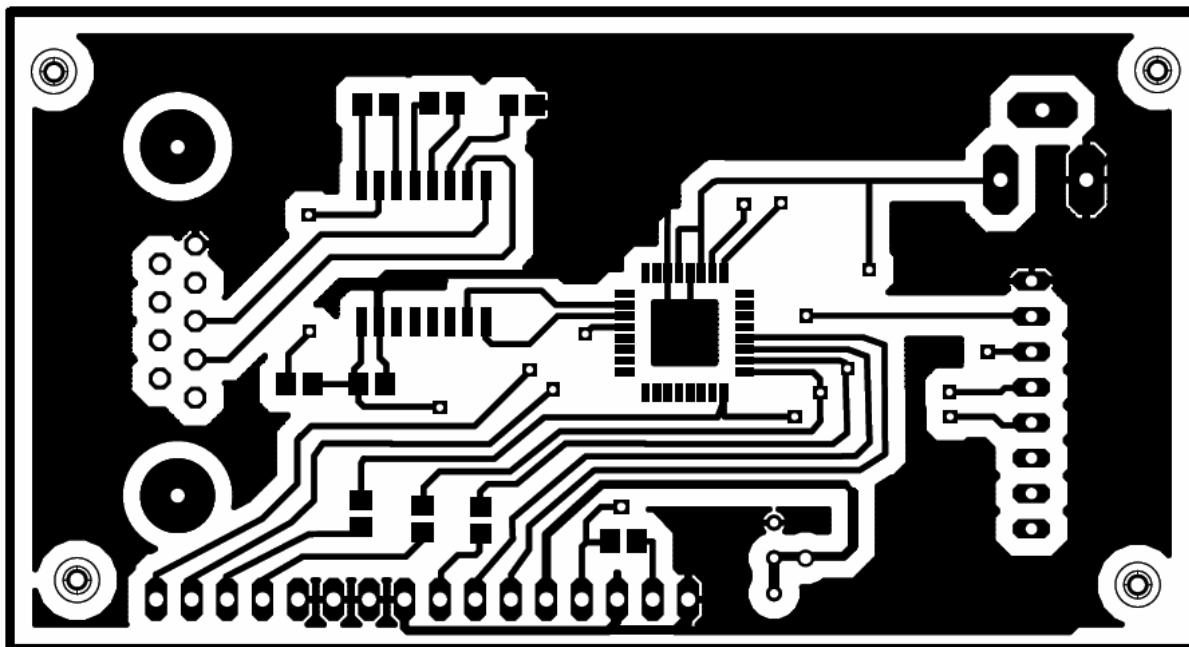
Tab P2. Zapojení konektoru ISP

| Pin: | Zapojení: |
|------|------------|
| 1. | GND |
| 2. | /RST |
| 3. | MOSI |
| 4. | MISO |
| 5. | SCK |
| 6. | nezapojeno |
| 7. | nezapojeno |
| 8. | nezapojeno |

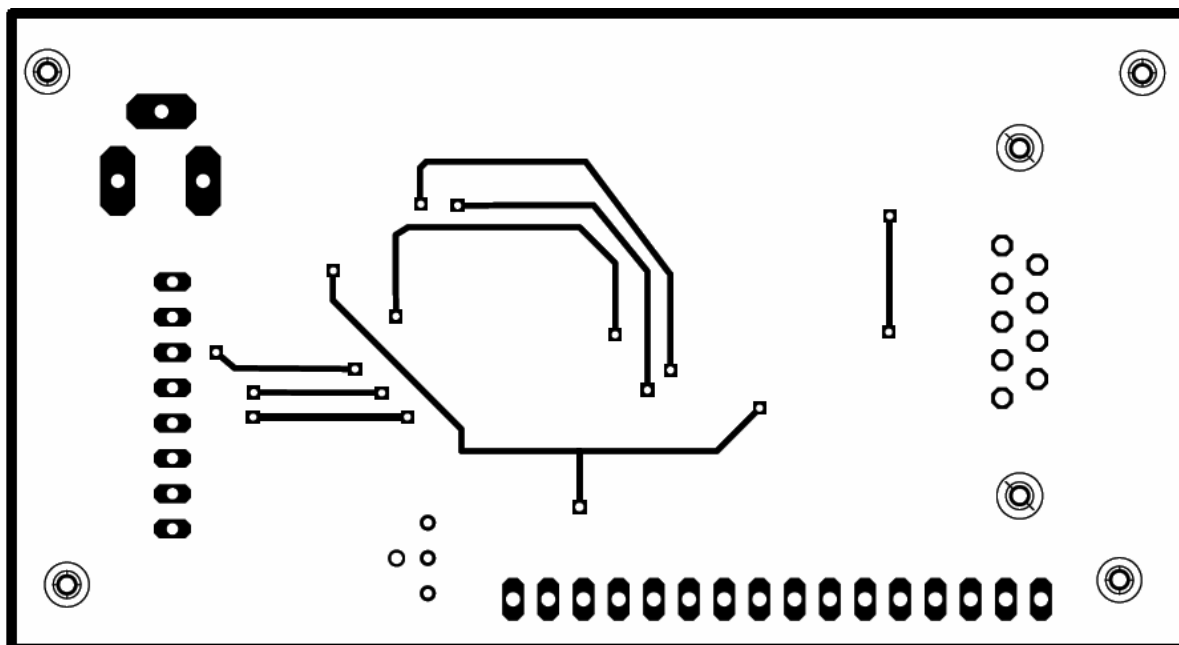
Konektor pro sériové programování ISP by bohatě stačil pouze s pěti piny, bohužel v době zhotovování výrobku jej nebylo možné v Brně zakoupit.

Příloha C – Desky plošných spojů :

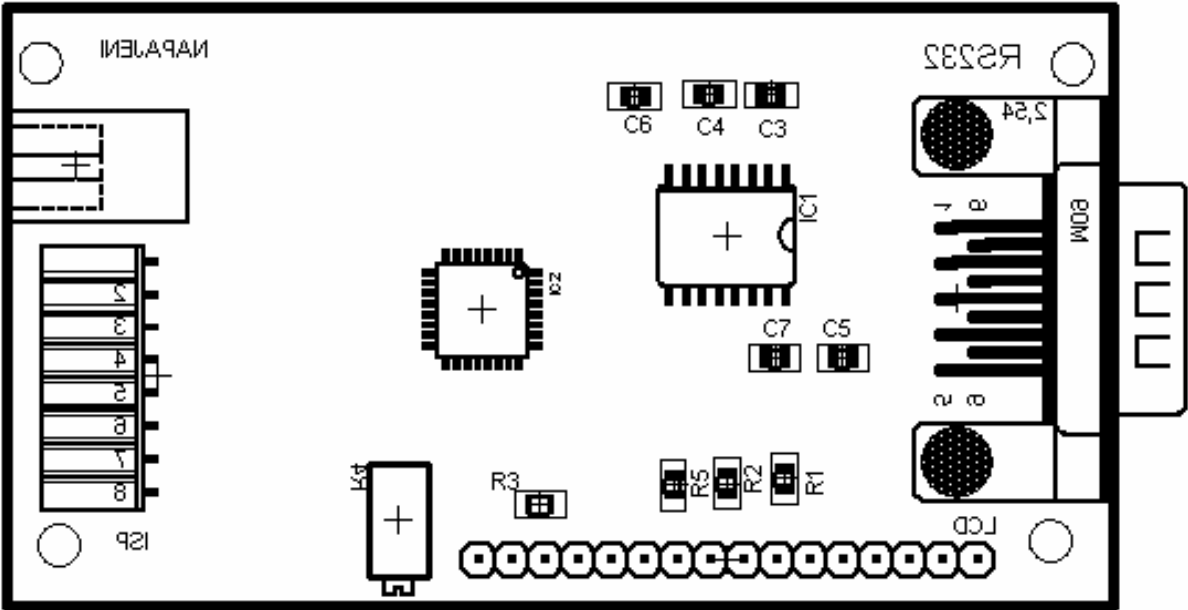
Deska plošných spojů pohled ze strany bottom (zvětšeno na 200 %)



Deska plošných spojů pohled ze strany top (zvětšeno na 200%)



Příloha D – Osazovací plán pohled ze strany bottom:



Příloha E – Seznam použitých součástek

| Part | value | Device | Package |
|----------|------------|--------------|------------|
| C3 | 2u2 | C-EUC0805 | C0805 |
| C4 | 2u2 | C-EUC0805 | C0805 |
| C5 | 2u2 | C-EUC0805 | C0805 |
| C6 | 2u2 | C-EUC0805 | C0805 |
| C7 | 2u2 | C-EUC0805 | C0805 |
| IC1 | MAX232ECWE | MAX232ECWE | SO16L |
| IC2 | MEGA8-AI | MEGA8-AI | TQFP32-08 |
| ISP | | FE08W | FE08W |
| LCD | | PINH0-1x16 | 1x16 |
| NAPAJENI | | 733980-62 | 733980-62 |
| R1 | 820 | R-EU_R0805 | R0805 |
| R2 | 820 | R-EU_R0805 | R0805 |
| R3 | 18 | R-EU_M0805 | M0805 |
| R4 | 10k | R-TRIMM3296X | RTRIM3296X |
| R5 | 820 | R-EU_R0805 | R0805 |
| RS232 | | M09H | M09H |

Příloha F – Zdrojový kód skriptu v počítači:

```
#!/bin/bash

stty -F /dev/ttyS0 cs8 -parenb -cstopb 9600
i=1
cesta_vstup="/proc/meminfo"
cesta_vystup="/dev/ttyS0"
memtotal=$(cat $cesta_vstup | cut -c 10-23 | sed -n 1p)
memfree=$(cat $cesta_vstup | cut -c 10-23 | sed -n 2p)
echo $memtotal "t" > $cesta_vystup
echo $memfree "s" >> $cesta_vystup
sleep 1s
while [ "$i" -ne 0 ]
do
memfree=$(cat $cesta_vstup | cut -c 10-23 | sed -n 2p)
echo $memfree "s" >> $cesta_vystup
sleep 1s
done
```

Příloha G – Obrázek funkčního zařízení:

