

JIHOČESKÁ UNIVERZITA

PŘÍRODOVĚDECKÁ FAKULTA

Ústav fyziky a biofyziky



Bakalářská práce

NÁVRH A REALIZACE DATALOGGERU NA BÁZI PROCESORU ATMEL AVR
PRO FOTOVOLTAICKÉ PANELY.

České Budějovice 2011

Vedoucí práce:
Ing. Michal Šerý

Autor práce:
Bohumír Tomášek

Bibliografický záznam

TOMÁŠEK Bohumír, Návrh a realizace dataloggeru na bázi procesoru Atmel AVR pro fotovoltaické panely: Bakalářská práce, České Budějovice: Jihočeská univerzita, Přírodovědecká fakulta, Ústav fyziky a biofyziky 2011. 58 listů. Vedoucí bakalářské práce: Ing. Michal Šerý.

Anotace

Bakalářská práce „Návrh a realizace dataloggeru na bázi procesoru Atmel AVR pro fotovoltaické panely“ pojednává o využití procesoru Atmel AVR pro sběr dat teploty s následnou prezentací zpracovaných dat na PC.

Annotation

Bachelor thesis "Design and implementation of data logger based on Atmel AVR for photovoltaic panels," deals with using Atmel AVR for data acquisition temperature and presentation of data on the PC.

Klíčová slova

Současné trendy, fotovoltaické panely, datalogger, měřící systémy spolupracující s PC, teplotní čidla, operační zesilovač, jednočipové procesory ATMEL AVR.

Keywords

Current trends, photovoltaic panels, data logger, measuring systems, working with computer, temperature-sensitive element, an operational amplifier, single-chip processors ATMEL AVR.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě (v úpravě vzniklé vypuštěním vyznačených částí archivovaných Přírodovědeckou fakultou), elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Č. Budějovicích 20. dubna 2011

.....
Bohumír TOMÁŠEK

Děkuji vedoucímu bakalářské práce Ing. Michalu Šerému, za odborné vedení, metodickou pomoc, cenné rady a připomínky, které mi v průběhu bakalářské práce poskytl. Dále děkuji všem ostatním, kteří mi byli při tvorbě této práce jakkoliv nápomocni.

Obsah:

Úvod.....	1
1 Využití slunečního záření	2
1.1 Přeměna sluneční energie na elektrickou energii.....	2
2 Historický vývoj fotovoltaického článku.....	2
2.1 Princip fotovoltaického článku.	3
2.2 Výroba solárních článků	4
2.3 Stupeň účinnosti fotovoltaického článku	5
2.4 Fotovoltaické systémy	5
3 Datalogger	5
4 Jednotlivé části vývojového Kitu EvB 4.3 [6]	6
4.1 Napájení	6
4.2 Port (rozhraní) USB	6
4.3 Pole s LED indikátory.....	7
4.4 Tlačítka	7
4.5 Slot pro paměťovou kartu SD/MMC	8
4.6 LCD displej s podsvětlením.....	8
4.7 Hodiny reálného času – RTC, paměť EEPROM.....	9
4.8 IR přijímač a teplotní senzor [3]	10
4.9 Stručný popis AVR	23
4.10 Rozhraní a připojení mikrokontroléru	23
5 Vývojové nástroje	27
5.9 Bascom AVR	27
6 Závěr	30
7 Použitá literatura:	31
7.1 Webové zdroje:	31
Seznam pojmů a zkratk	32
8 Seznam obrázků	33
9 Seznam tabulek	34
10 Přílohy	35

ÚVOD

Cílem této práce je vytvořit zařízení pro sběr dat z fotovoltaických systémů a navrhnout možnosti využití jednočipového procesoru Atmel AVR spolupracujícího s PC při zpracování a prezentaci nashromážděných dat. Datalogger je sestaven a určen pro sběr teplotních dat z fotovoltaického článku. V tomto případě se jedná o uložení teploty fotovoltaického článku, který je vystaven osvětlení a zatížen odběrem. Datalogger slouží také k uložení těchto dat na paměťovou kartu a při následném přenesení dat do PC k jejich vyhodnocení - závislosti teploty, osvětlení a výkonu fotovoltaického článku. Datalogger funguje samostatně a když nedojde k neočekávané poruše, nepotřebuje pro svoji činnost žádnou obsluhu.

Fotovoltaika je pro současnost velice zajímavé téma, a to jak z ekonomického, ekologického tak i elektrotechnického hlediska. Poslední dobou tento obor zažívá obrovský rozmach díky neustále zlevňujícím technologiím a i ekonomickým aspektům. V současnosti můžeme na mnoha místech vidět vyrůstat nové fotovoltaické elektrárny, ať už na střechách rodinných domů, výrobních halách nebo zemědělských budovách či jinak nevyužitých plochách.

1 Využití slunečního záření

Přímá nebo nepřímá přeměna na elektrickou energii. K přímé přeměně dochází ve fotovoltaických článcích, nepřímo se elektrická energie získává soustředěním paprsků do ohniska tepelných slunečních elektráren.

1.1 Přeměna sluneční energie na elektrickou energii

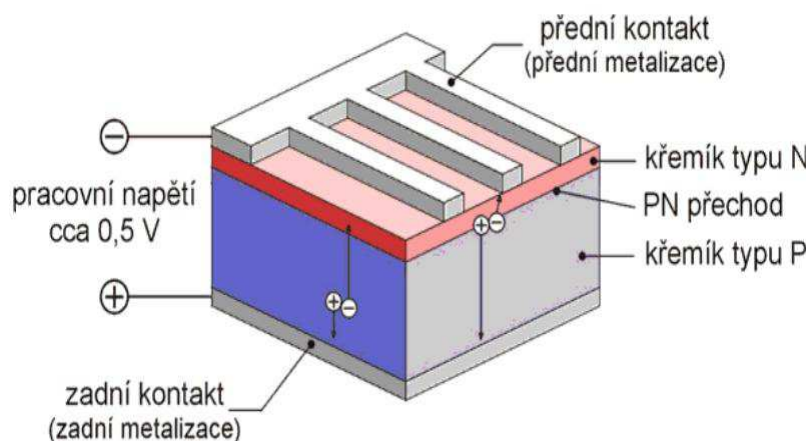
Přeměna světla na elektrickou energii je nazývána fotovoltaika. Toto označení pochází z řeckého slova a skládá se ze dvou slov "foto" = světlo a "volt" – jednotka elektrického napětí. Zásadní impuls k využití a studiu fotovoltaiky přinesl teprve rozvoj polovodičové techniky. Nejpoužívanějším materiálem pro výrobu fotovoltaických článků je křemík, který dosahuje poměrně vysoké účinnosti přeměny energie záření. K širšímu využití fotovoltaiky došlo teprve začátkem 60. let minulého století s nástupem kosmonautiky. Sluneční články patří k hlavním zdrojům elektrické energie pro družice, kosmické stanice i výzkumné sondy. Dalším podnětem pro rozsáhlý výzkum fotovoltaiky byla celosvětová ropná krize v 70. letech. Vzrostl význam obnovitelných zdrojů elektrické energie a mezi nimi i energie slunečního záření. Fotovoltaické zdroje dnes nacházejí své uplatnění v mnoha oblastech. Malé solární články napájí kapesní kalkulačky, větší solární panely mohou sloužit jako zdroj elektrické energie v místech bez připojení k síti. Velké fotovoltaické systémy jsou schopny dodávat energii do běžné rozvodné sítě i pro vědecké přístroje na kosmických stanicích.

2 Historický vývoj fotovoltaického článku.

Fotovoltaický jev byl objeven již v roce 1839 francouzským fyzikem **Alexandrem Edmondem Becquerelem**. Fotovoltaický článek byl ale sestaven až v roce 1883 **Charlesem Frittem**. Ten tenkou vrstvou zlata potáhl selenem. Při této konstrukci dosahovala účinnost pouze 1 %. Později, konkrétně roku 1946, si nechal konstrukci článku patentovat **Russel Ohl**, ale to ještě nedosahoval dnešní podoby. První fotovoltaický článek byl vyroben v roce 1954 v Bellových laboratořích. Od této doby se vědci stále snaží zdokonalit vlastnosti fotovoltaických článků, především zvýšit jejich účinnost.

Fotovoltaické články (sluneční baterie) jsou polovodičové prvky, které mění světelnou energii v energii elektrickou. Přímou přeměnou světla na elektrickou

energií se dnes zabývá samostatná specializace. Fotoelektrický efekt vysvětluje vznik volných elektrických nosičů dopadem záření. Celkově se daří přeměnit v elektrickou energii jen asi 17 % energie dopadajícího záření. Solární články jsou tvořeny polovodičovými plátkami tenčími než 1mm. Na spodní straně je plošná průchozí elektroda. Horní elektroda má plošné uspořádání tvaru dlouhých prstů zasahujících do plochy. Tak může světlo na plochu svítit. Povrch solárního článku je chráněn skleněnou vrstvou. Sloužící jako antireflexní vrstva a zabezpečuje tak, aby co nejvíce světla vniklo do polovodiče. Antireflexní vrstvy se většinou tvoří napařením oxidu titanu. Tím získá článek svůj tmavomodrý vzhled. Jako polovodičový materiál se používá převážně křemík. Jiné polovodičové materiály, např. galium arsenid, kadmium-sulfid, kadmium-tellurid, selenid mědi a india, nebo sirič galia se zatím zkoušejí. Krycí sklo chrání povrch solárních článků i před vlivy prostředí.

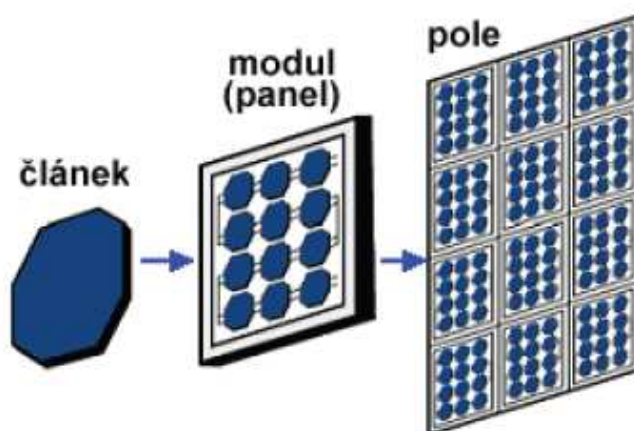


Obr. 1: Princip fotovoltaického jevu [4]

2.1 Princip fotovoltaického článku.

Fotovoltaický článek je v podstatě polovodičová dioda. Jeho základem je tenká křemíková destička s vodivostí typu P. Na ní se při výrobě vytvoří tenká vrstva polovodiče typu N, obě vrstvy jsou odděleny tzv. přechodem P-N. Osvětlením článku vznikne v polovodiči vnitřní fotoelektrický jev a v polovodiči se z krystalové mřížky začnou uvolňovat záporné elektrony. Na přechodu P-N se vytvoří elektrické napětí, které dosahuje u křemíkových článků velikosti zhruba 0,5 V. Energie dopadajícího světla se v článku mění na elektrickou energii. Připojíme-li k článku pomocí vodičů spotřebič (například miniaturní elektromotor), začnou se kladné a záporné náboje vyrovnávat a obvodem začne procházet elektrický proud. Je-li třeba větší napětí nebo

proud, zapojují se jednotlivé články sériově či paralelně a sestavují se z nich fotovoltaické panely.



Obr. 2: Sestavení panelu z jednotlivých článků [5]

2.2 Výroba solárních článků

Solární články se vyrábějí z křemenného písku, který je nejprve zbaven nečistot a poté zpracován na monokrystal křemíku. Křemíkový krystal je ponořen do horkého, tekutého křemíku. Tekutý křemík se spojí s ponořeným křemíkovým krystalem, zatímco je tento pomalu vytahován z pánve. Takto vznikají křemíkové tyče s délkou přes 1 metr a průměrem cca. 12 cm. U polykrystalických článků je horký křemík odléván do formy a pozvolna ochlazován. Také u této metody vznikají křemíkové tyče.

Nyní jsou tyto tyče, které vznikají při obou metodách, řezány na tenoučké plátky (<0,5 mm). Každý plátek je leptáním a broušením vyhlazován. Poté je jedna strana plátku obohacena malým množstvím pětimocného chemického prvku - vznikne polovodič typu N (prvek P, As), zatímco druhá strana je obohacena prvkem trojmocným - vznikne polovodič typu P (převážně B). Toto obohacení se nazývá *dotace* (řízené zavádění příměsí).

Zadní strana článku se pak potáhne velmi tenkou vrstvou hliníku, která slouží jako kladný pól. Přední strana je potažena stříbrem, ovšem nikoliv plošně, nýbrž kovová vrstva představuje jen úzké vodivé dráhy, aby mohlo světlo dále dopadat na křemík.

Dosažené napětí na jednom článku je v rozmezí 0,6 ÷ 0,7 V, a proto se články zapojují sériově popř. sériovým a paralelním spojením pomocí vodivých pásků. V případě sériového zapojení je vodivý pásek připájen k přední straně jednoho článku (kladný pól) a zároveň k zadní straně (záporný pól) druhého článku.

2.3 Stupeň účinnosti fotovoltaického článku

Měření, které určuje stupeň účinnosti solárního článku, je prováděno v laboratoři. Přitom musí být dodržena různá kritéria. Světelné ozáření činí během měření 1000W/m². Kromě toho musí být dodržena teplota článků 25 °C. Vedle toho je co nejpřesněji kontrolována vlhkost vzduchu. Tato kritéria, která musí všichni výrobci dodržet, umožňují porovnat různé solární články nejodlišnějších zhotovení. Ale co je vlastně stupeň účinnosti? Stupeň účinnosti určuje vztah mezi napájenou energií a získanou energií v procentech. Například, když 1000W vstupního výkonu vyrobí 100W výstupního výkonu, potom činí stupeň účinnosti 10 %.

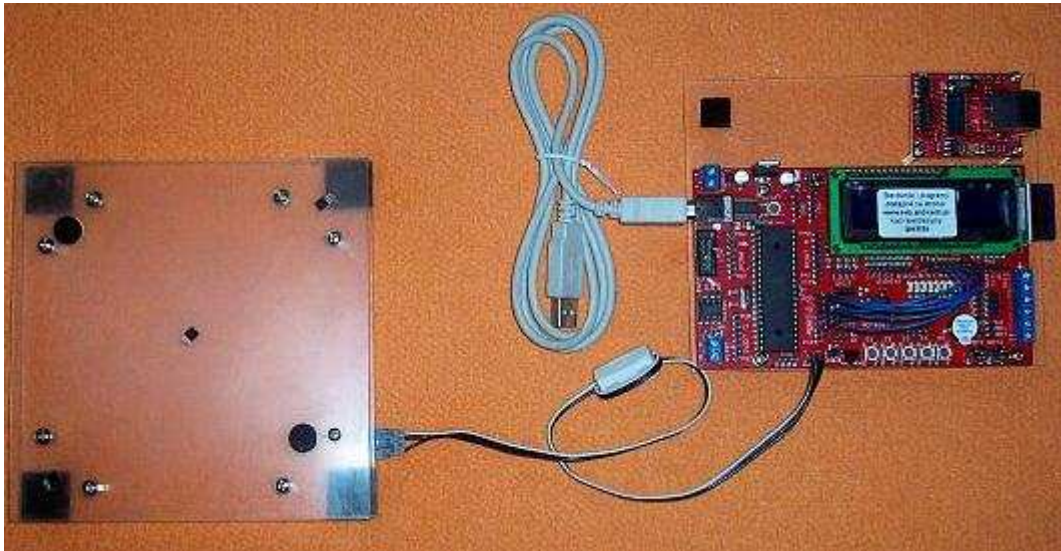
2.4 Fotovoltaické systémy

Jediný fotovoltaický článek má jen velmi malé využití. Výstupní napětí i výkon je pro většinu aplikací příliš malý. Proto se články podle požadovaného napětí a odebíraného proudu spojují a vytvářejí fotovoltaický modul (panel). Spojením více modulů vzniká rozměrné fotovoltaické pole, které se instaluje například na střechu nebo fasádu budovy. Pro dosažení vysoké životnosti se moduly ukládají do hermeticky uzavřených pouzder, která jsou opatřena vysoce průhledným tvrzeným sklem. Tato úprava chrání moduly před povětrnostními vlivy, udávaná životnost je 20 ÷ 30 let.

3 Datalogger

Na sestavení dataloggeru byl použit vývojový kit EvB 4.3 s jednočipovým procesorem ATmega32 od ATMEL. Na vývojové desce jsou osazeny i další obvody a komponenty, které budou využity při sestavení dataloggeru a popsány níže.

K dataloggeru jsou připojené senzory pro měření teploty fotovoltaického článku a venkovní teploty. Měření napětí a proudu fotovoltaického článku není součástí tohoto projektu a v případě potřeby by bylo možné tuto alternativu dopracovat. Datalogger naměřenou teplotu ukládá v pravidelných intervalech na paměťovou kartu, kterou je vybavený. Časový interval je možné změnit od 1 do 30 minut a nebo tlačítkem S2 hodnoty uložit ručně.

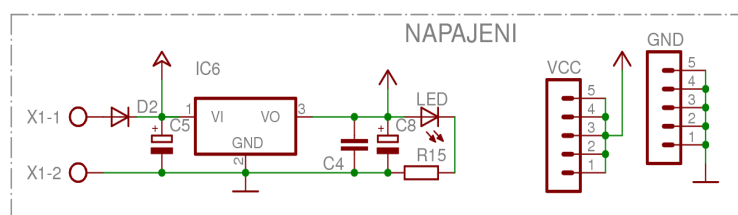


Obr. 3: celkový pohled na sestavu dataloggeru s podložkou pro FV články

4 Jednotlivé části vývojového Kitu EvB 4.3 [6]

4.1 Napájení

1. jednak přes USB port (v tomto případě musí být osazena propojka “jumper“ na pinech USB Vcc)
2. z externího napájecího zdroje, který musí dodávat minimálně +9V DC. V tom případě musí být propojka na USB Vcc rozpojena



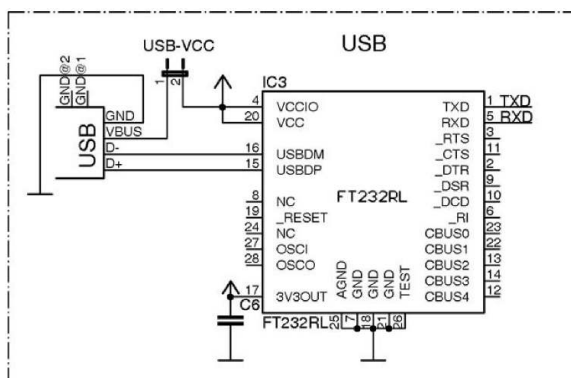
Obr. 4: Schéma zapojení napájení vývojového modulu EvB

4.2 Port (rozhraní) USB

Komunikaci modulu (kitu) EvB 4.3 s počítačem umožňuje vestavěné USB rozhraní realizované převodníkem USB / UART FT232RL (virtuální sériový port - COMx). Obvod FT232RL je svými signály na desce trvale propojen s komunikačními kanály RX a TX, proto není nutné při realizaci zapojení dodatečně

nic propojovat. Pro připojení extra zařízení na RX a TX signály můžeme použít hřebínku na portu D, konkrétně piny PD0 – RxD a PD1 – TxD.

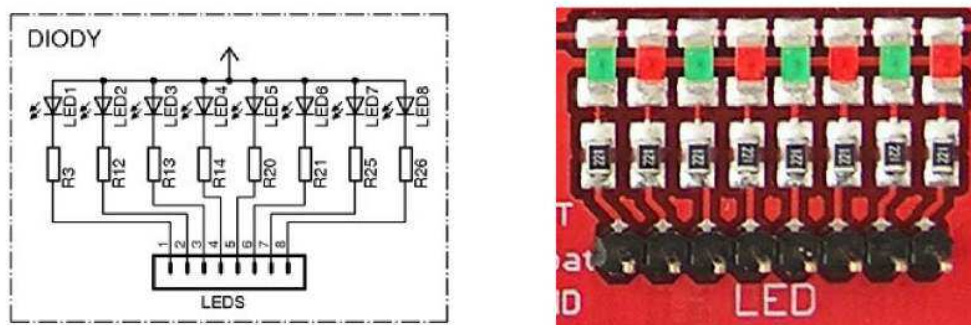
Na speciálním hřebínku ve spodní části desky najdete zbylé signály virtuálního portu, které můžete použít v dalších konstrukcích – jedná se o signály CTS (TS), DSR (SR), DCD (CD) a RI. Tyto signály mohou být použity pro naprogramování úplně nového procesoru bez bootloaderu.



Obr. 5: Schéma zapojení USB rozhraní s napájením

4.3 Pole s LED indikátory

Pro indikační účely je deska EvB 4.3 osazena osmi LED v barvách zelená a červená. Diody jsou společně napojeny na V_{cc} , ovládají se tedy připojením logické úrovně 0 (propojení příslušné LED s GND)



Obr. 6: Schéma zapojení LED indikátorů s obrázkem umístění na desce

4.4 Tlačítka

Vývojová deska EvB 4.3 je osazena 5 tlačítky se společnou zemí. Dá se tak využít možnosti mikrokontrolérů AVR nastavit vstupní pin s Pull-up odporem, který potom není nutné osazovat do zapojení. Tlačítka nejsou zapojena přes omezovače zákmitů ani jiné tvarovací obvody, různé nežádoucí stavy je třeba ošetřit programově.

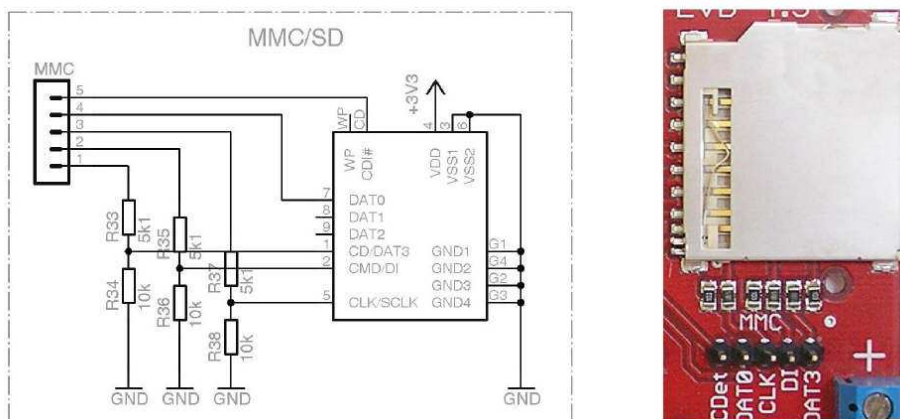
V programu jsou užitá tlačítka S1, S2, S3 pro nastavení hodin, počtu čidel měření teploty a pro přímé uložení dat (S2).



Obr. 7: Schéma zapojení tlačítek a obrázek tlačítek na desce

4.5 Slot pro paměťovou kartu SD/MMC

Uchování hodnot, popřípadě čtení konfigurace nebo rozsáhlých datových souborů může být realizováno přístupem na SD/MMC kartu. Vývojová deska EvB 4.3 je osazena patičí (slotem) pro připojení tohoto typu paměťové karty. Napájení karty je zajištěno napěťovým stabilizátorem 3.3V umístěným v komunikačním obvodu FT232RL, pro spolupráci s mikrokontrolérem napájeným napětím 5 V jsou vstupní piny rozhraní karty opatřeny odporovými děliči 5.1 k Ω spolu s 10 k Ω pro snížení napětí. Výstupní signál z karty do mikrokontroléru je zapojen přímo. Ze slotu pro kartu jsou vyvedeny signály CLK, DI, DATA0 a DATA3 a signál pro oznámení vložení karty CDet.



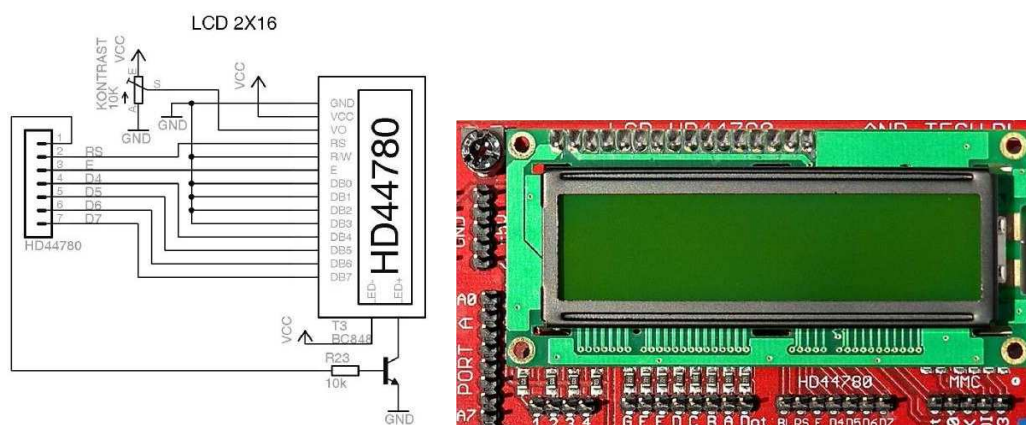
Obr. 8: Schéma zapojení SD karty a obrázek slotu

4.6 LCD displej s podsvětlením

Možnost jak může vývojová deska komunikovat s okolím nebo obsluhou je podsvětlený LCD display v konfiguraci 2 x 16 znaků. Pro připojení displeje je

v blízkosti LED zobrazovače umístěno 16 pinové připojovací pole se standardní pinovou roztečí. Displej je osazen standardním řadičem HITACHI HD44780, kdy je zvolen 4 bitový způsob pro komunikaci. Potřebné signály jsou vyvedené na připojovací piny, kontrast lze regulovat trimrem umístěným na desce v blízkosti displeje. Na vývodu BL aktivujeme přivedením +5 V DC podsvětlení, jehož barva závisí na typu displeje.

Signál R/W je trvale připojen na GND a není ho možné používat. Některé programy pro spolupráci s displejem – např. CodeVisionAVR – vyžadují pro korektní činnost možnost číst příznak BUSY, kterým řadič korektně oznamuje připravenost s dalším příkazům. V tomto případě nemůžeme použít originální knihovny, ale je třeba použít vlastní. Knihovny poskytované jako podpora pro tuto desku pracují s tímto zapojením bez komplikací. (Nepoužívají čtení příznaku BUSY, neboť čekají pevný čas mezi odeslanými příkazy)

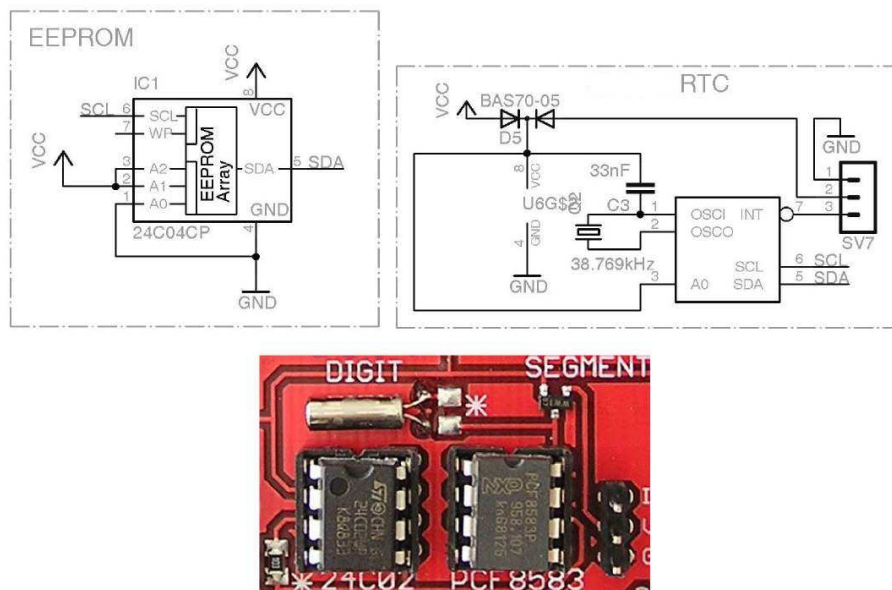


Obr. 9: Schéma zapojení zobrazovací jednotky a její obrázek na desce

4.7 Hodiny reálného času – RTC, paměť EEPROM

První pokusy a seznámení se sběrnici I2C je vhodné začít na typických obvodech, tedy RTC a sériové paměti. Vývojová deska je osazena pamětí EEPROM 24C02 o velikosti 2 Kbit, umístěné na adrese 172 (0xAC) pro zápis a adrese 173 (0xAD) pro čtení, druhým obvodem na sběrnici I2C je obvod reálného času PCF8583 na adresách 162 (0xA2) pro instrukce zápisu a 163 (0xA3) pro čtení. Navíc je z obvodu RTC vyveden na samostatný konektor výstup pro vyvolání přerušení INT využitelný např. pro mikrokontrolér a také samostatný vstup pro zálohovací bateriové napájení, připojený přes ochranné diody, takže stačí pouze připojit zálohovací baterii. V případě obsluhy sběrnice I2C programovým prostředím BASCOM bude pravděpodobně nutné osadit navíc „zdvihací“ rezistory 4.7 k Ω nebo 10 k Ω v místech

označených hvězdičkou. Toto je způsobené tím, že BASCOM ve své obsluze I2C nepoužívá interní Pull-Up rezistory v mikrokontroléru.



Obr. 10: Schéma zapojení EEPROM a RTC PCF8583, obrázek osazení na desce

4.8 IR přijímač a teplotní senzor [3]

Pro měření teploty okolí je vývojová deska vybavena digitálním teplotním senzorem DS18B20 komunikujícím pomocí 1 drátové signálové sběrnice 1wire™ od firmy DALLAS. Čidlo je pevně spojeno s napájením, stačí připojit k vývodu mikrokontroléru pouze datový vodič. Na desce je již připojen Pull-Up rezistor 10 kΩ pro správnou činnost. Sběrnice umožňuje zapojit paralelně více senzorů, pro tyto účely jsou vyvedené separátně potřebné signály – tedy +5 V DC, Data a GND. Na pinech je vyznačena polarita pro správné připojení. IR přijímač – TSOP4836 je možné použít pro spolupráci s klasickým IR ovládačem pro spotřební elektroniku. Je použit typ pro 36 kHz, nicméně šířka filtrů v přijímači umožňuje bezproblémovou spolupráci s dálkovými ovládači využívajícími 38 kHz nebo 40 kHz nosný kmitočet. Obvod je taktéž pevně připojen na napájecí napětí, stačí připojit pouze datový vodič.

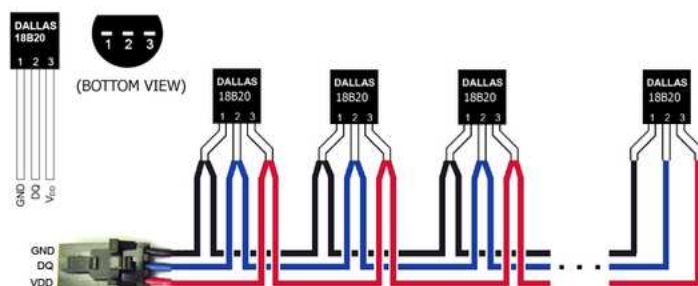
Na samostatné desce pro měření teploty fotovoltaického článku jsou osazeny tři teplotní čidla DALAS DS18B20, které sledují teplotu na krajích a středu fotovoltaického článku.



Obr. 11: Podložka pro měření teploty fotovoltaického článku

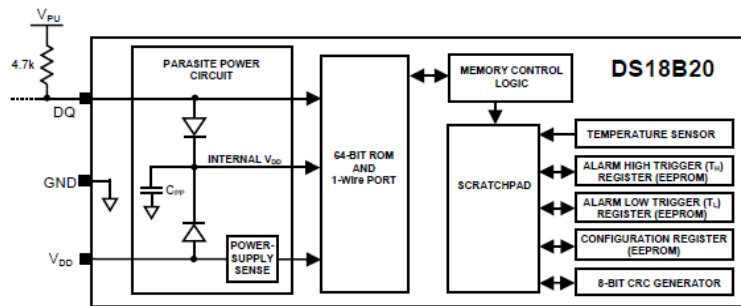
Součástka DS18B20 je běžně dostupný digitální teploměr, který je schopný měřit v rozsahu $-55\text{ }^{\circ}\text{C}$ až $+125\text{ }^{\circ}\text{C}$. Výrobce garantuje přesnost $0.5\text{ }^{\circ}\text{C}$ v teplotním rozsahu od $-10\text{ }^{\circ}\text{C}$ do $+85\text{ }^{\circ}\text{C}$. Rozlišení měření je volitelná v rozmezí 9 - 12 bitů (tzn. rozlišení $0.5 \div 0.0625\text{ }^{\circ}\text{C}$). Senzor se připojuje pomocí sériového rozhraní označované jako 1wire.

Pro komunikaci se používá jeden datový a dva napájecí vodiče (GND, VDD, DQ). Napájecí napětí VDD se může pohybovat v rozmezí 3.3 V až 5.5 V. Mimo to lze senzor provozovat v tzv. parazitním režimu využívajícím pouze dvou vodičů (GND, DQ).



Obr. 12: propojení čidla DS18B20 [3]

Teplota je snímána a následně zobrazena na LCD panelu a potom uložena na SD kartu. Data z SD karty jsou případně “stažena“ do PC a vyhodnocena. Teplotní čidla DS18B20 jsou řízena částí programu (viz níže) a teplota je zjišťována s rozlišením na $0,065\text{ }^{\circ}\text{C}$.



Obr. 13: Blokové schéma čidla DALAS DS 18B20[3]

V této části bych probral snímač teploty Dallas 18B20 podrobněji:

Operace měření teploty

Jádrem funkce čidla DS18B20 je přímá digitální komunikace snímače teploty s masterem. Výsledek převodu teploty je uživatelem nastavitelný na 9,10,11 nebo 12 bitů, a tomu odpovídá rozlišení 0,5 °C (9bitů); 0, 25 °C (10bitů); 0,125 °C (11bitů); 0,0625 °C (12bitů). Výchozí nastavení po zapnutí napájení je 12bitový převod. DS18B20 je po zapnutí v neurčitěm stavu. K zahájení měření teploty a převodu A/D musí master jednotka (většinou to bývá mikroprocesor) vyslat příkaz ConvertT (44h)“. Následuje převod a výsledek je uložen ve dvou „teplotních registrech“ v paměti „Scratchpad“ a DS18B20 se vrátí do neurčitěho stavu. Pokud je DS18B20 napájeno z externího zdroje, master jednotka může vysílat čtecí časové úseky (read time slots), (viz 1wire bus system) po příkazu ConvertT(44h) a DS18B20 bude odpovídat log.0 při probíhání převodu teploty a log.1 až bude převod dokončen. Pokud je čidlo napájeno přes sběrnici 1wire bus system, toto oznámení o dokončení převodu nemůže být použito a na sběrnici musí být od začátku převodu log.1. Požadavky na napájení přes 1wire bus system je detailně vysvětlen v kapitole powering the DS18B20 datasheet. Výstupní data (teplota) jsou ve °C, pro aplikace s použitím °F musí být použit převod softwarově. Převedená teplota na data je uložena jako 16bitový výsledek - rozšířený dvojkový doplněk – číslo v teplotních registrech – viz Tab. 1 Znaménkový bit (S) signalizuje, zda je teplota kladná nebo záporná. Pro kladné teploty je S=0, pro záporné teploty S=1. Pokud je DS18B20 nastaven na 12bit převod – výsledek – všechny bity v registrech obsahují platná data. Při převodu 11 bitovém není použit bit0, při převodu 10 bitovém nejsou použity bity 1 a 0, při převodu 9bitovém nejsou použity bity 2, 1 a 0. Tabulka 2 ukazuje příklady digitálních výstupních dat a odpovídajících teplot při 12 bitovém převodu.

Formát teplotních registrů

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2 ⁶	2 ⁵	2 ⁴

S = SIGN

Tab. 1: Formát teplotních registrů [3]

Vztah teplota/data (12BIT)

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

*Po zapnutí napájení je hodnota v registrech odpovídající +85 °C

Tab. 2: Vztah teplota/data [3]

Transakční sekvence

Transakční sekvence pro přístup do čidla je následující:

Krok 1. Inicializace

Krok 2. Příkaz ROM (následovaný nějakou požadovanou výměnou dat)

Krok 3. DS18B20 funkční příkaz (následovaný nějakou požadovanou výměnou dat)

Je velice důležité dodržet tuto sekvenci vždy při komunikaci, jinak čidlo nebude odpovídat, pokud nějaký krok bude vynechán. Výjimku tvoří příkazy Search ROM [F0h] a Alarm Search [ECh]. Po vyslání jednoho z těchto příkazů master musí zpět ke kroku 1 v sekvenci.

Krok 1. - Inicializace

Všechny transakce na 1-wire sběrnici začínají s inicializační sekvencí. Inicializační sekvence obsahuje RESET puls od masteru a následovaným presentačním pulsem (pulsy) zasláným čidlem (čidly DS18B20). Presentační puls dává masteru vědět o zařízeních na sběrnici nalezené jako DS18B20 připravené k práci. Časování RESET a present pulsů jsou detailně popsány v kapitole 1-wire signaling.

Krok 2. – Příkazy ROM

Po detekci presentačního pulsu může master vyslat příkazy ROM. Tyto příkazy vyvolají unikátní 64bit ROM kód z každého zařízení na sběrnici a povolí masteru jednoznačně specifikovat zařízení, pokud jich je na sběrnici více. Tyto příkazy tedy dovolí masteru určit kolik a jaké typy zařízení jsou nalezeny na sběrnici nebo kolik zařízení má nastavenou podmínku alarmu. Je pět příkazů ROM a každý příkaz je 8 bitový. Mikroprocesor musí vyslat příslušný příkaz ROM před vysláním funkčního příkazu. Vývojový diagram pro operaci příkazů ROM je na obr. 14.

SEARCH ROM (F0h)

Když je systém po zapnutí napájení inicializovaný, mikroprocesor musí identifikovat kódy ROM všech SLAVE zařízení na sběrnici, které dovolí mikroprocesoru určit číslo čidla a jeho typ. Master se naučí kódy ROM během procesu vyloučení (eliminace), který požaduje k vykonání SearchROM cyklu, tj. příkaz SEARCH ROM a následná výměna dat bude tolikrát, kolikrát je potřeba k identifikaci všech čidel. Když je na sběrnici pouze jedno čidlo, může být použit místo příkazu SearchROM příkaz ReadROM (viz níže).

Po každém cyklu SearchROM se musí mikroprocesor vrátit na krok 1 inicializace transakční sekvenci

READ ROM (33h)

Tento příkaz může být použit pouze pokud je na sběrnici pouze jedno zařízení – čidlo. To dovolí mikroprocesoru číst 64bit ROM kód čidla bez použití procedury SearchROM. Pokud je tento příkaz použit, když je na sběrnici víc jak jedno zařízení, nastane kolize dat, protože všechna čidla se pokusí vyslat data současně.

MATCH ROM (55h)

Příkaz MatchROM a následné vyslání 64bitového ROM kódu dovolí masteru komunikaci s požadovaným slave-zařízením na sběrnici multidrop nebo singledrop. Pouze slave zařízení, kterému odpovídá 64bitový ROM kód, bude odpovídat na funkční příkazy vyslané masterem. Všechny ostatní slave (čidla) zařízení na sběrnici budou čekat na resetovací impuls.

SKIP ROM (CCh)

Mikroprocesor může použít tento příkaz na sběrnici i bez vyslání informačních ROM kódů. Např. mikroprocesor může na sběrnici vyvolat převod teploty vysláním příkazu SKIP ROM a následně příkazem ConvertT(44h).

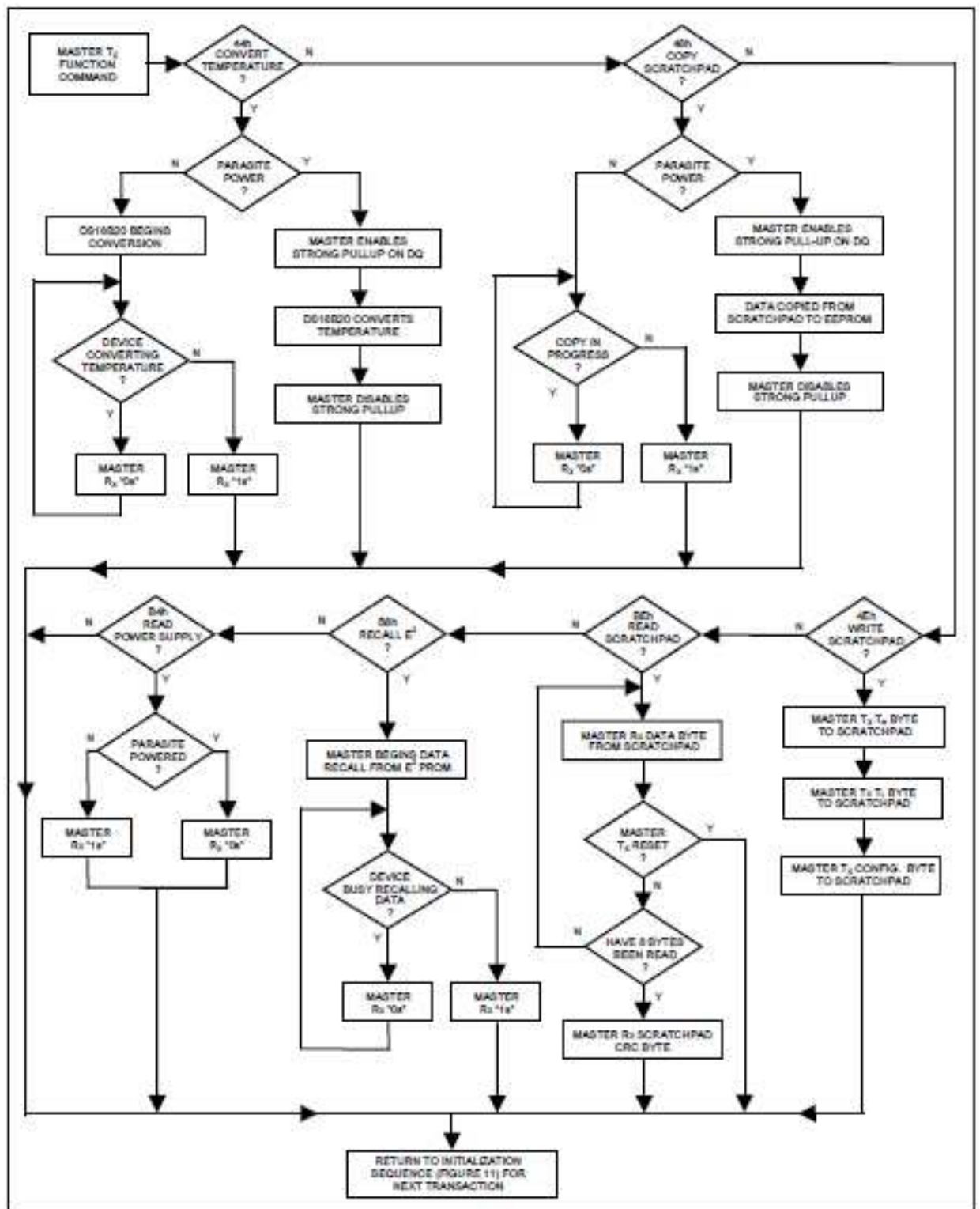
Pozn. Příkaz ReadScratchpad může následovat za příkazem SkipROM pouze pokud je pouze jedno zařízení na sběrnici. V tomto případě je ušetřeno to, že mikroprocesor může číst ze slave bez zaslání 64bitového ROM kódu zařízení. Pokud by bylo na sběrnici víc než jedno slave (čidlo), příkaz SkipROM a následný příkaz Read Scratchpad bude příčinou kolize dat na sběrnici, protože se budou pokoušet posílat data všechny slave současně.

ALARM SEARCH (ECh)

Operace tímto příkazem je stejná jak operace příkazem Search ROM kromě toho, že budou odpovídat pouze čidla s nastaveným příznakem alarm. Tento příkaz umožní masteru zjistit, zda nějaké zařízení DS18B20 splnilo podmínku „alarmu“ v posledním teplotním převodu. Po každém cyklu ALARM SEARCH (tj. Příkaz Alarm Search a následná výměna dat) se musí master vrátit na krok 1 – Inicializace v transakční sekvenci. V kapitole OPERATION - ALARM SEARCH je vysvětlena operace s příznakem alarmu.

Krok 3. – Funkční příkazy DS18B20

Po použití příkazu ROM k adresaci čidla DS18B20, se kterým budeme komunikovat, master může vyslat jeden funkční příkaz pro čidlo DS18B20. Tyto příkazy dovolí masteru zapisovat do a číst z paměti Scratchpad, zahájit převod teploty a zjistit druh napájení. Funkční příkazy čidla DS18B20, které jsou níže napsány a seřazeny v tabulce 3 a jsou představeny vývojovým diagramem na obr. 14.



Obr. 14: Funkční příkazy čidla DS18B20, vývojovým diagram [3]

CONVERT T (44h) – převod teploty

Tento příkaz zahájí jeden převod teploty. Převod teploty - výsledná „teplotní“ data se uloží do dvou 8bit registrů ve scratchpad paměti a čidlo DS18B20 se vrátí do „klidového stavu“ (bude čekat na inicializační sekvenci). Když je zařízení zapojeno v parazitním módu, master musí během 10 μ s po tomto příkazu uvolnit 1wire sběrnici, aby se na sběrnici „udělala“ log.1 a mohlo se napájet čidlo po dobu konání převodu t_{conv} , jak je popsáno v kapitole NAPÁJENÍ (POWERING THE) DS18B20. Když je čidlo DS18B20 napájeno externím zdrojem, master může po příkazu ConvertT vysílat „čtecí časové úseky“ a čidlo DS18B20 bude odpovídat log.0 během převodu teploty a log.1 po jeho ukončení. V parazitním módu tato informace o průběhu převodu není možná, protože sběrnice musí být v log.1 během celého převodu.

WRITE SCRATCHPAD [4Eh] – zápis do paměti

Tento příkaz dovolí masteru zapsat 3 bajty dat do paměti scratchpad čidla. První bajt je zapsán do registru TH (bajt 2 v paměti scratchpadu), druhý bajt do registru TL (bajt 3) a třetí do Konfiguračního registru (bajt 4). Data musí být poslána - první bit je LSB. Všechny tři bajt MUSÍ být zapsány před vykováním resetu masterem, neboť data by se nezapsala správně!!!

READ SCRATCHPAD [BEh] – čtení paměti

Tento příkaz dovolí masteru číst obsah paměti scratchpad. Přenášená data se začínají číst LSB bajtem 0 a pokračují až do devátého bajtu (bajt 8 – CRC). Master smí vyslat RESET pro ukončení čtení kdykoli, pokud chce pouze část dat z paměti.

COPY SCRATCHPAD [48h] – kopírování paměti

Tento příkaz zkopíruje obsah paměti scratchpad TH, TL a konfigurační registry (bajty 2, 3 a 4) do EEPROM v čidlu. Když je slave používán v parazitním módu, během 10 μ s (max) po tomto příkazu musí master uvolnit 1wire sběrnici (aby se na sběrnici „udělala“ log.1 a mohlo se napájet čidlo) nejméně na 10 ms jak je napsáno v kapitole POWERING THE DS18B20 – (napájení DS18B20)

RECALL E² [B8h]

Tento příkaz vyvolá hodnoty TH, TL a konfigurační reg. z EEPROM čidla a uloží je na místa bajtů 2, 3, a 4 v paměti scratchpad. Master může po příkazu Recall E² vysílat „čtecí časové úseky“ a čidlo DS18B20 bude odpovídat log.0, pokud bude proces probíhat a log.1, pokud bude proces dokončen. Recall E² operace proběhne automaticky po zapnutí napájení, aby byly připraveny pro práci aktuální data.

READ POWER SUPPLY [B4h]

Master vyšle tento příkaz pro určení druhu napájení čidla. To zjistíme tak, že po vyslání příkazu Read Power Supply vyšle master „čtecí časový úsek“ – při parazitním napájení bude „čtený časový úsek“ log.0 a při externím napájení bude log.1. V kapitole Napájení - POWERING THE DS18B20 jsou informace k použití pro tento příkaz.

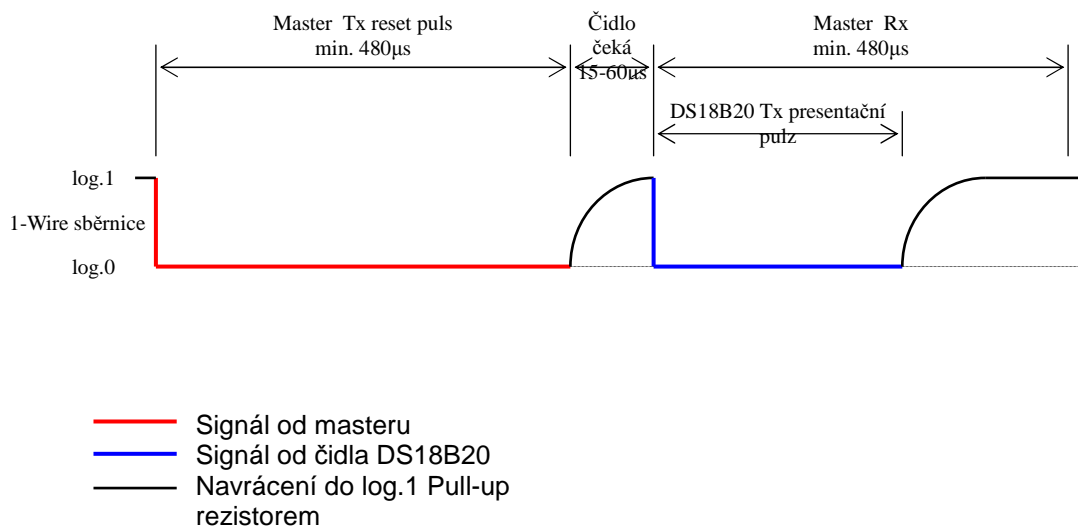
1wire SIGNALING – časování signálů

DS18B20 používá komunikační protokol „1wire communication“ k zabezpečení správného přenosu dat. Několik typů signálů z tohoto protokolu je tu popsáno. Jsou to: Reset puls, presentační puls, zápis 0, zápis 1, čtení 0 a čtení 1. Master začíná všechny tyto signály (pulsy) s výjimkou presentačního pulsu.

Inicializace: Reset a Presentační puls

Všechna komunikace s DS18B20 začíná inicializační sekvencí, která obsahuje Reset puls a za ním následuje presentační puls z čidla DS18B20 (viz obr. 14). Když DS18B20 pošle inicializační puls na základě Reset pulsu je to potvrzení správného připojení a připravenosti čidla DS18B20 komunikovat s masterem. Během inicializační sekvence master zasílá (Tx) Reset puls – 1wire sběrnice v log.0 minimálně 480μs. Master poté uvolní sběrnici a přejde do režimu přijímače (Rx). Když je sběrnice uvolněna, 5kΩ pull-up rezistor uvede 1wire sběrnici do log.1. Když DS18B20 detekuje náběžnou hranu, počká ještě 15μs až 60μs a poté pošle presentační impuls - 1wire sběrnice do log.0 na 60μs to 240μs.

Časování inicializační sekvence



Obr. 15: Časování inicializační sekvence

READ/WRITE (čtecí/zapisovací) časový průběh

Master zapisuje/čte data do/z čidla DS18B20 během časového úseku. Jeden datový bit je posílán přes 1wire sběrnici za jeden časový úsek.

WRITE (zapisovací) časový úsek

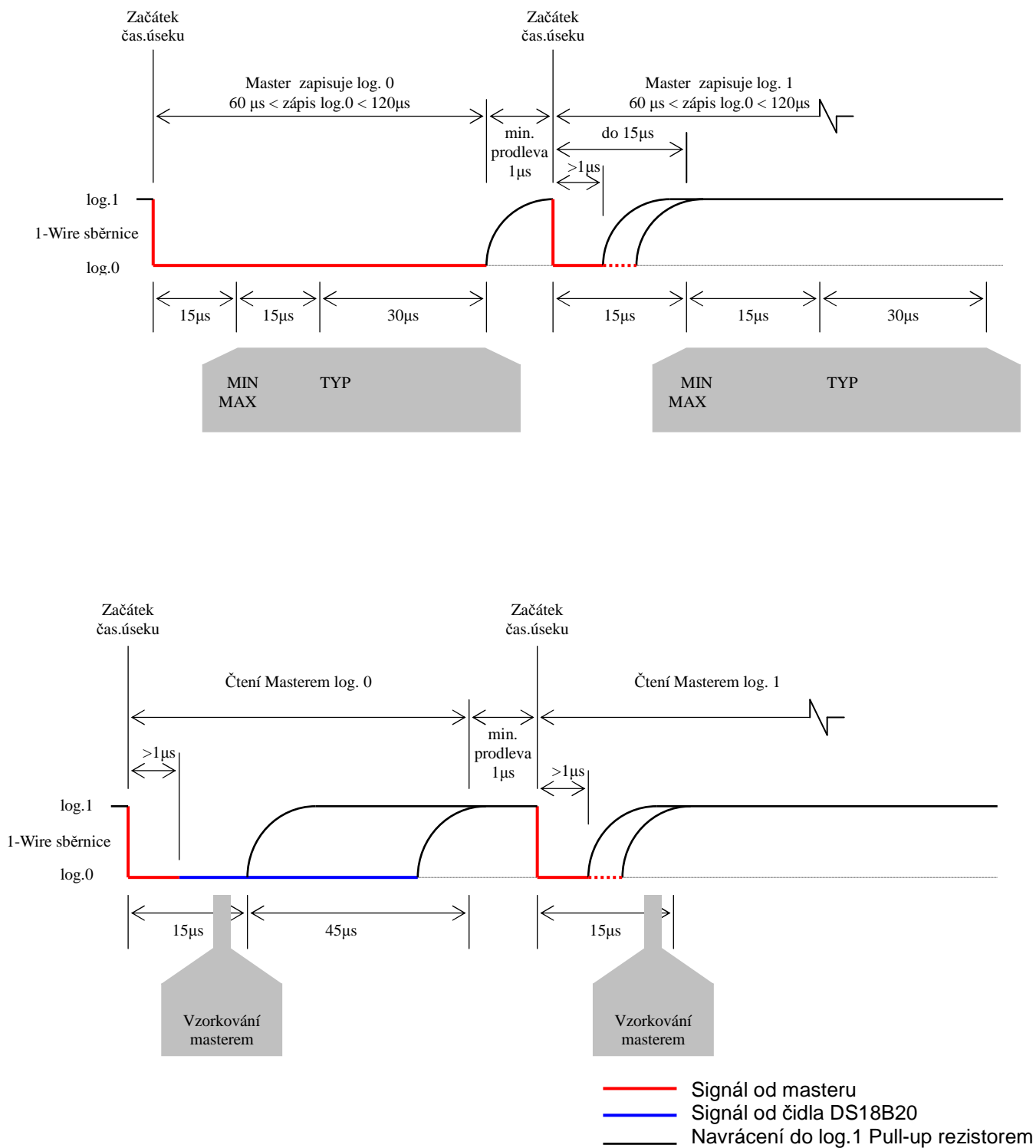
Zde máme dva typy: „Zápis 1” a „Zápis 0”. Master používá logiku „Zápis 1=log.1“ a „Zápis 0=log.0“. Všechny zapisovací časové průběhy musí být minimálně dlouhé 60µs s minimálně 1µs prodlevou pro navrácení sběrnice do log.1 mezi jednotlivými časovými úseky. Oba dva úseky zápisu začínají tak, že master udělá log.0 na 1wire sběrnici (viz obr. 15).

Pro vytvoření Zápisu log.1 puls začíná tak, že master uvede sběrnici do log.0 a během 15 µs po začátku časového úseku - musí sběrnici uvést do log.1.

Pro vytvoření Zápisu log.0 puls začíná tak, že master uvede sběrnici do log.0 a tato úroveň musí na sběrnici trvat více jak 60µs a méně než 120µs.

Čidlo DS18B20 vzorkuje 1wire sběrnici a podle poslední zjištěné hodnoty mezi 15µs až 60µs po zahajovací log.0 master vyhodnotí hodnotu bitu. Když nadetekuje 1, tuto informaci bere čidlo jako log.1., když nadetekuje 0, tuto informaci bere jako log.0.

READ/WRITE průběh časového úseku



Obr. 16: Read/Write průběh časového úseku

READ (čtecí) časový úsek

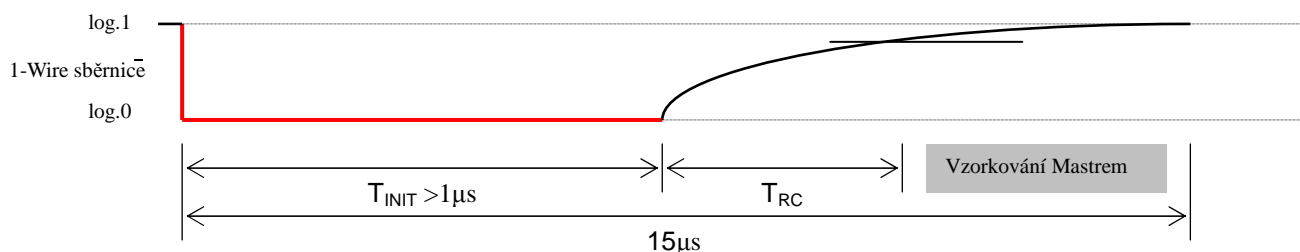
Čidlo DS18B20 může posílat data do masteru pouze, když master vyšle „Čtení časového úseku“. Proto master musí generovat čtecí časový úsek ihned po vyslání příkazu Read Scratchpad [BEh] nebo Read Power Supply [B4h], takže DS18B20 může poskytnout potřebná data. Navíc master může generovat čtecí časové úseky po vyslání příkazu Convert T [44h] nebo Recall E² [B8h] pro zjištění stavu operace jak je vysvětleno v kapitole DS18B20 FUNCTION COMMAND.

Všechny čtecí časové úseky musí být dlouhé minimálně 60 μ s s minimální prodlevou mezi pulsy 1 μ s. Čtecí časové úseky začínají vysláním log.0 na sběrnici na dobu min. 1 μ s a poté musí master sběrnici uvolnit (viz obr.14). Po začátku čtecího časového úseku čidlo DS18B20 začne posílat 1 nebo 0 po sběrnici. DS18B20 posílá 1 tak, že pustí sběrnici do log.1 a 0 posílá držením sběrnice v log.0. Když je posílaná 0, čidlo DS18B20 uvolní sběrnici po ukončení časového úseku a sběrnice bude navracena do log.1 pull-up rezistorem. Výstupní data z čidla DS18B20 jsou platná po 15 μ s sestupné hrany generované masterem (začátek čtení). Proto master musí uvolnit sběrnici a vzorkovat její úroveň (stav) po 15 μ s od začátku čtecího časového úseku (odstartovaným masterem).

Z obr. 16 je patrné, že součet T_{INIT} , T_{RC} , a $T_{VZORKOVANÍ}$ musí být menší, než 15 μ s pro čtení časového úseku.

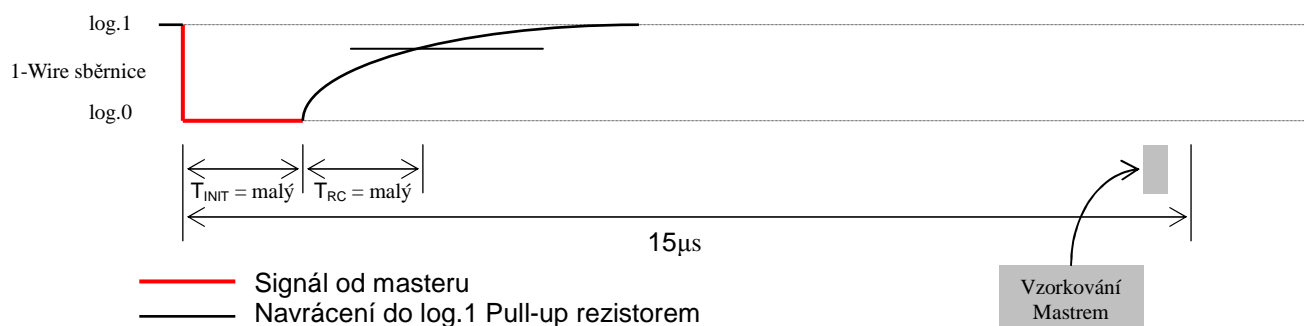
Obr. 17 ukazuje, že systémové časování okraje je maximalizovaný s držením T_{INIT} a T_{RC} tak krátce, jak je to možné a umístění vzorkování master během čtecího časového úseku je ke konci 15 μ s periody.

Detailní časování čtení masteru



Obr.16: Detailní časování masteru

Doporučené časování čtení log.1



Obr.17: Doporučené časování čtení log.1

DS18B20 Popis funkčních příkazů [3]

Příkaz	Popis	Protokol (kód)	Stav 1-Wire sběrnice po použití příkazu	Pozn.
Příkazy převodu teploty				
Convert T	Zahájení převodu teploty	44h	DS18B20 vysílá informaci o stavu převodu pro mastera (pouze při externím napájení čidla)	1
Příkazy pro paměť				
Read Scratchpad	Čte veškerý Scratchpad obsahující i bajt CRC	BEh	DS18B20 posílá až všech 9 bajtů Scratchpadu do mastera	2
Write Scratchpad	Zapisuje data bajtů 2,3,4 (T_H , T_L , konfig.reg.) do Scratchpadu	4Eh	Master posílá 3 datové bajty do čidla DS18B20	3
Copy Scratchpad	Kopíruje T_H , T_L , konfig.reg. ze Scratchpadu do EEPROM čidla	48h	Není	1
Recall E ²	Vybírá T_H , T_L , konfig.reg. z EEPROM čidla do Scratchpadu	B8h	Čidlo DS18B20 posílá informace o stavu transakce Recall E ²	
Read Power supply	Signály pro master ke zjištění módu napájení	B4h	DS18B20 posílá druh napájení pro mastera	

Pozn.:

- 1 - Pro parazitní napájení čidel master musí povolit pull-up na sběrnici během převodu teploty a kopírování ze Scratchpadu do EEPROM čidla. Žádné další (činnosti) příkazy toto nepotřebují.
- 2 - Master může přerušit přenos dat kdykoliv pomocí resetu
- 3 - Všechny tři bajty musí být zapsány před použitím resetu.

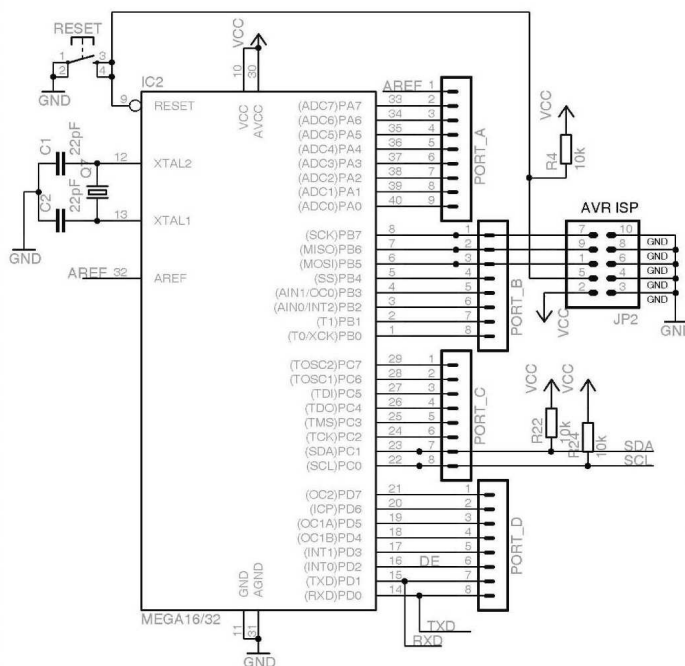
Tab. 3: DS18B20 Popis funkčních příkazů

4.9 Stručný popis AVR

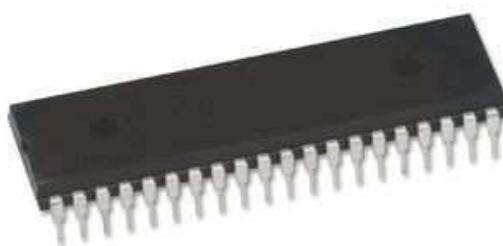
Jádro AVR je RISC. Skládá se ze 32 stejných 8 bitových registrů, které mohou obsahovat jak data, tak i adresy. Vzhledem k propojení registrů s ALU (aritmeticko logická jednotka) provede ALU za jeden hodinový cyklus jednu operaci. Mikroprocesory AVR využívají koncepci harwardské architektury. To znamená, že mají oddělenou paměť pro program a pro data.

4.10 Rozhraní a připojení mikrokontroléru

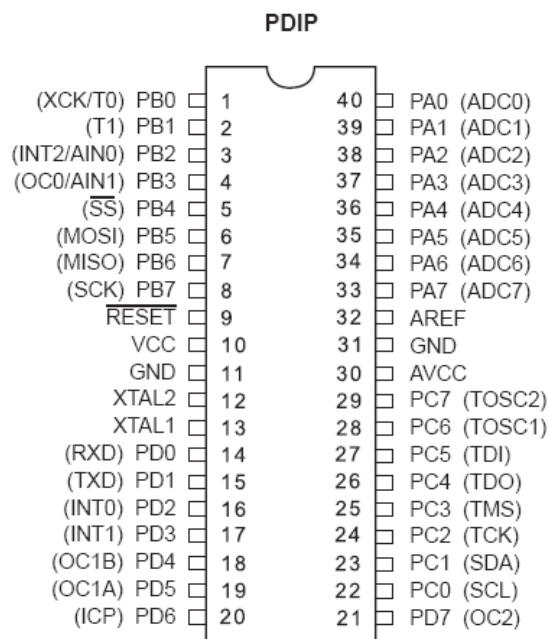
Všechny signály mikrokontroléru jsou připojeny na externí piny včetně vstupu AREF používaného pro práci s AD převodníky – připojení napěťové reference. Popisy signálů a příslušné propojení je zobrazeno na schématu na této straně. Tlačítko RESET je umístěno v blízkosti USB konektoru. Krystal pro využití oscilátoru mikrokontroléru má hodnotu 16 MHz.



Obr. 19: Schéma zapojení mikrokontroleru ATmega32 na desce [6]



Obr. 20: Pouzdro DIL40 mikroprocesoru AVR ATmega32



Obr. 21: Rozložení vývodů ATmega32 – pouzdro DIP40 [11]

AVR-Rodina

Jednočipový mikropočítač (MCU). Procesor s univerzálním jádrem, s kterým jsou současně zaintegrovány základní periferní obvody, takže je schopen samostatné funkce. Za průkopníky v této kategorii můžeme považovat 8bitový procesor Intel i8051, který poprvé integroval všechny základní periferie (jádro procesoru, paměť RAM, EEPROM, čítače a časovače) na jediném čipu a 16bitový technologický procesor Siemens SAB 80C166, který poprvé integroval A/D převodníky, komunikační linky a masivní systém čítačů/časovačů/přerušení (následníky řady 80166 dnes vyrábí Infineon (řada C167 a C166 SV2) a SGS Thomson (řada ST10)). AVR je označení pro rodinu 8bitových mikročipů typu RISC s harvardskou architekturou od firmy Atmel. Za zrodem AVR stojí studenti Alf-Egil Bogen a Vegard Wollan z Norského technického institutu. Na trhu se tyto mikroprocesory začaly objevovat od roku 1997.

RISC Architektura

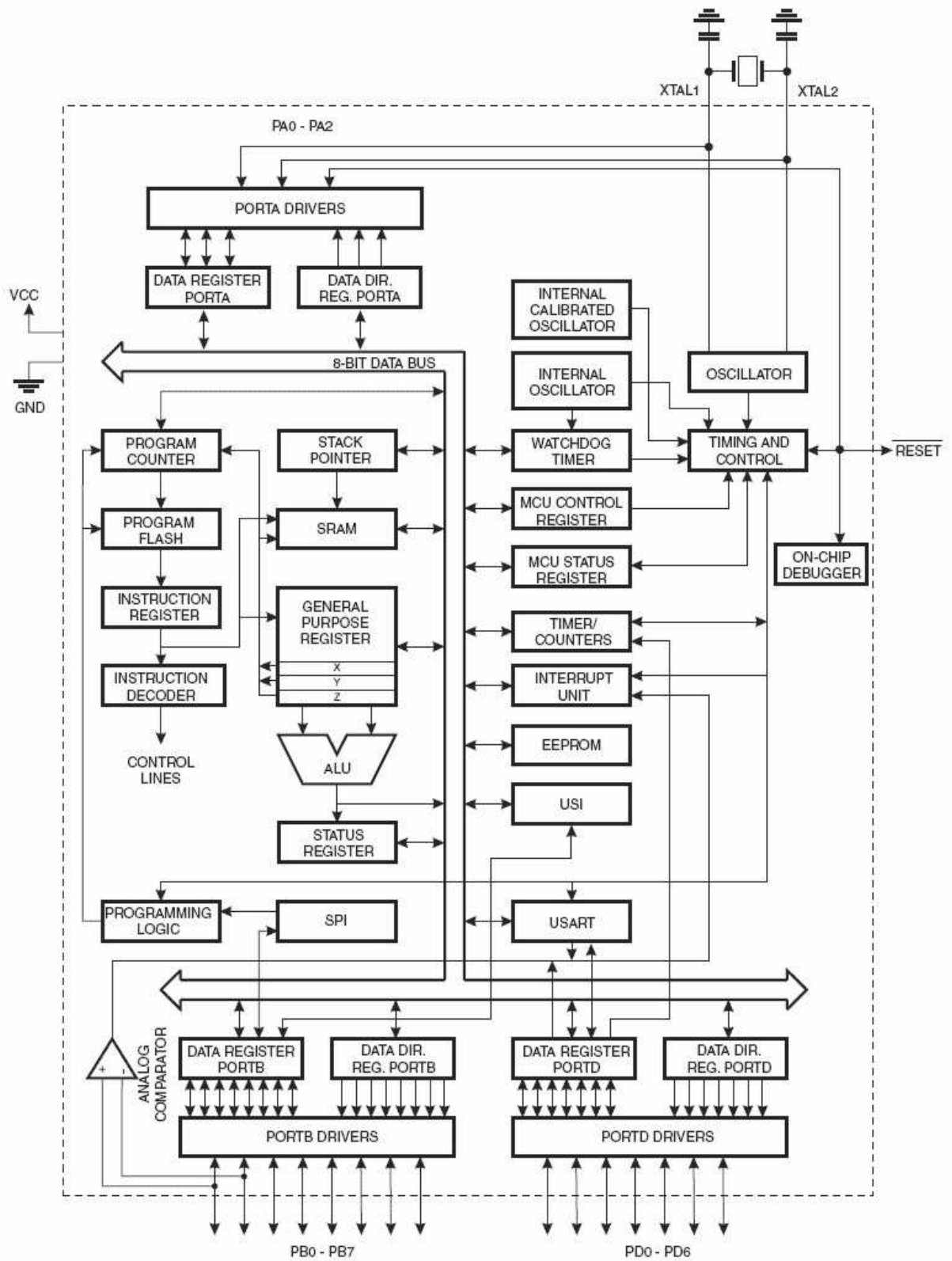
RISC - zkratka pochází z anglického originálu Reduced Instruction Set Computer, přeloženo: počítač s redukovanou instrukční sadou.

Během 70. let 20. století vědci ukázali, že většina programů prováděných na tehdejších počítačích využívala pouze malou část ze všech dostupných strojových instrukcí procesoru (tehdejší překladače nedokázaly efektivněji využít všech

instrukcí). Také složitý přístup do paměti zpomaloval provádění operací. Z toho vyplynulo, že složitější operace (mikro kód) efektivněji vykoná posloupnost jednodušších instrukcí, které lze provádět s vyšší frekvencí.

Shrnutí typických rysů RISC procesorů:

- procesor komunikuje s pamětí po sběrnici
- redukovaná sada strojových instrukcí obsahuje hlavně jednoduché instrukce
- délka provádění jedné instrukce je vždy jeden cyklus
- mikroinstrukce jsou HW implementovány na procesoru tím je zvýšena rychlost jejich provádění
- využívají řetězení instrukcí
- instrukční soubor obsahuje 131 instrukcí
- 32 8bitových registrů
- čtyři 8bitové vstupní/výstupní porty
- hodinový kmitočet 0-8 MHz výpočetní výkon 8 MIPS
- napájecí napětí 2,7 – 5,5 V
- programová paměť typu flash 32 Kbyte, až 10 000 cyklů přepsání
- 2 Kb datové SRAM Paměti
- 1 Kb EEPROM 100 000 cyklů přepsání
- programování pomocí SPI nebo JTAG rozhraní
- 2x 8bitový čítač, 2x 16bitový čítač, watchdog
- 4x PWM kanál
- analogový komparátor
- 8x 10bitový A/D převodník
- vestavěné rozhraní: USART, SPI, I2C, micro-wire
- interní RC oscilátor (možnost softwarové kalibrace)
- spotřeba < 1μA ve Standby režimu



Obr. 22: RISC Architektura [8]

5 Vývojové nástroje

Pro práci a vývoj aplikací s mikroprocesory AVR jsou přímo určeny softwarové a hardwarové nástroje.

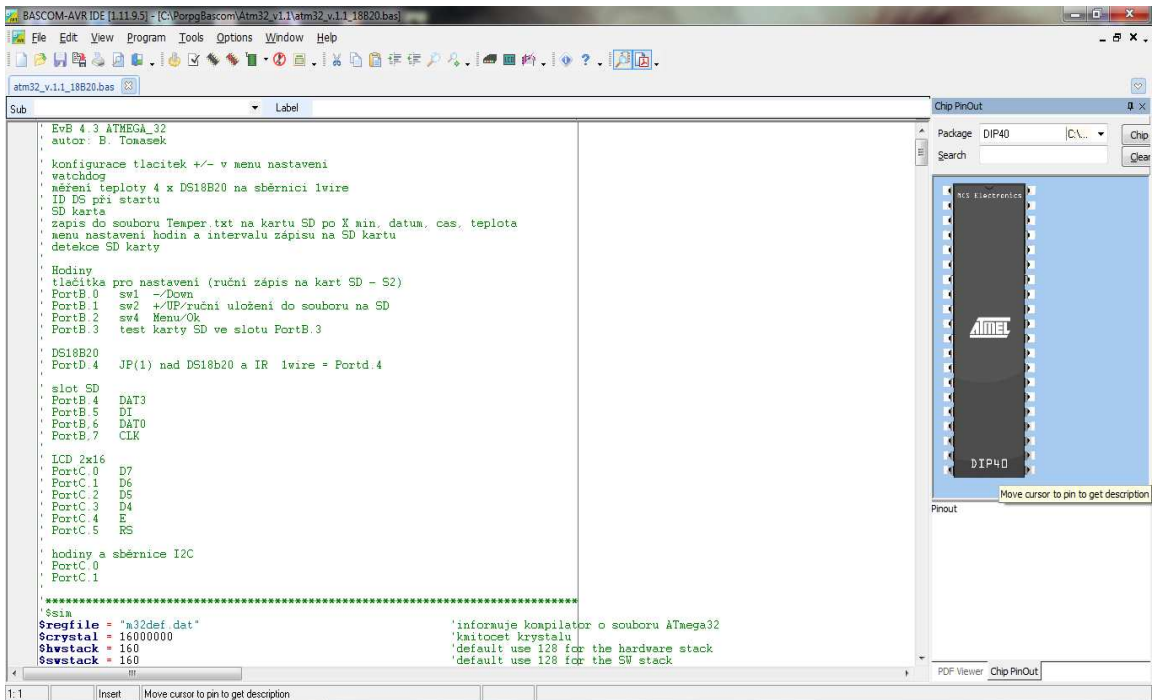
Softwarové nástroje podle svého druhu umožňují psát aplikace v různých programovacích jazycích (Assembler, C/C++, Pascal, BASCOM atd.)

Hardwarové nástroje jsou většinou koncipovány jako základní desky obsahující podpůrné obvody pro programování a ladění mikroprocesorových aplikací.

5.9 Bascom AVR

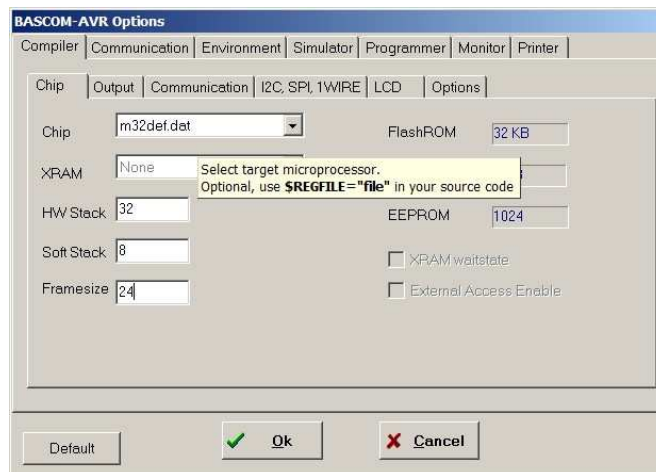
S rozšiřující se nabídkou a dostupností programovatelných obvodů se objevuje potřeba uživatelů bez hlubších programovacích znalostí mít jednoduchý programovací jazyk. V roce 1995 vytvořil Mark Alberts nový programovací jazyk pojmenovaný Bascom. Tento jednoduchý jazyk umožňuje i začátečníkům napsat program pro jednočipové mikropočítače a mikrokontroléry. Proto také byl Bascom použit v této práci a je v něm vytvořen ovládací program pro měření teploty. Pro tento účel je použita demo verze, kterou nabízí firma MCS electronics. Tato verze je plně funkční jen velikost kódu je omezena na max. velikost 4 kB, což je pro aplikaci měření teploty plně dostačující. Ovládání vývojové prostředí je plně intuitivní. Výchozím oknem programu je integrované vývojové prostředí (IDE), odtud jsou dostupné veškeré funkce programu (editace kódu, překlad, debugging, atd.)

Bascom je vývojové prostředí pro 8bitové mikroprocesory pracující pod Windows. Existuje ve dvou variantách 8051 a AVR. Starší varianta 8051 je kompatibilní s procesory x51. Novější AVR je pro procesory RISC architektury, které mají větší výkon a jsou často užity v konstrukcích modelářů, robotiků a ostatních. Jazyk má syntaxi Basic, takže pokud znáte alespoň jednu z jeho verzí, Bascom by pro neměl být problém.

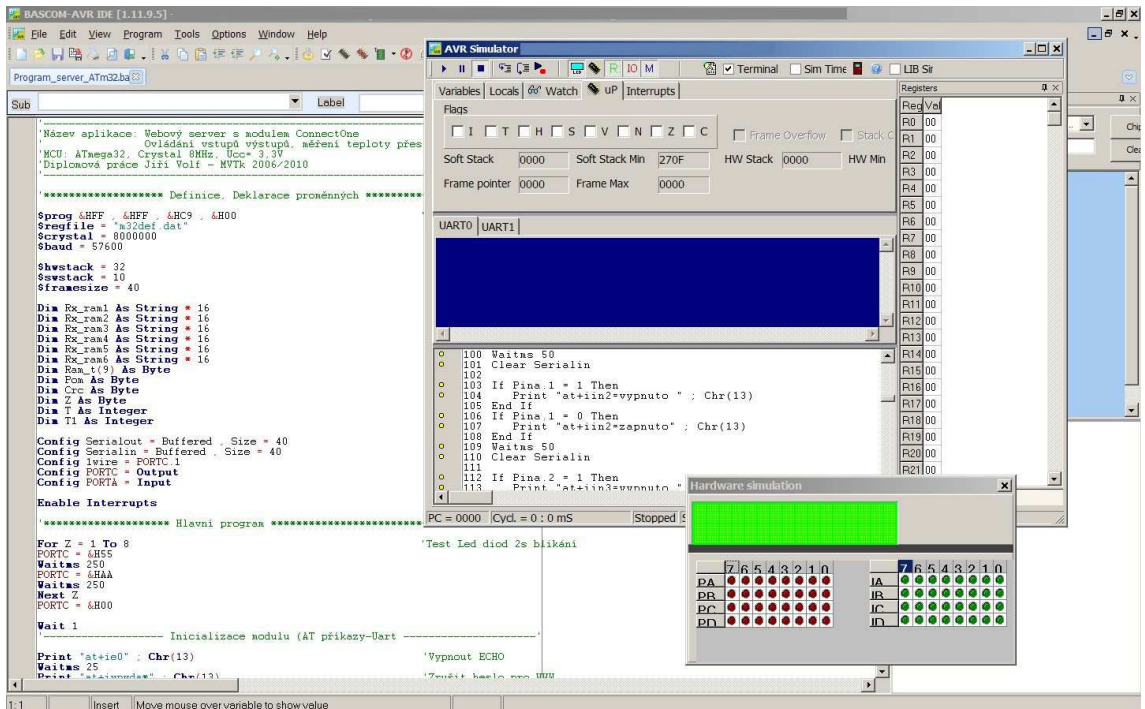


Obr. 23: BASCOM základní okno programu

Před samotným psaním programu je nutné správně zvolit typ mikroprocesoru se kterým budeme pracovat (Obr. č.24) tím je zajištěno správné pojmenování použitelných registrů. Toto je důležité například při použití jiného typu mikroprocesoru. Není nutné složité přepisování programu. Například pro přímý přístup na portA mikroprocesoru, slouží v Bascom příkaz: portA, ale každý mikroprocesor má tento port na jiné adrese a tu bychom museli přepisovat na každém místě programu, tímto krokem je zajištěna jednoduchá kompatibilita programu pro více mikroprocesorů bez složitého úprav hlavního programu.

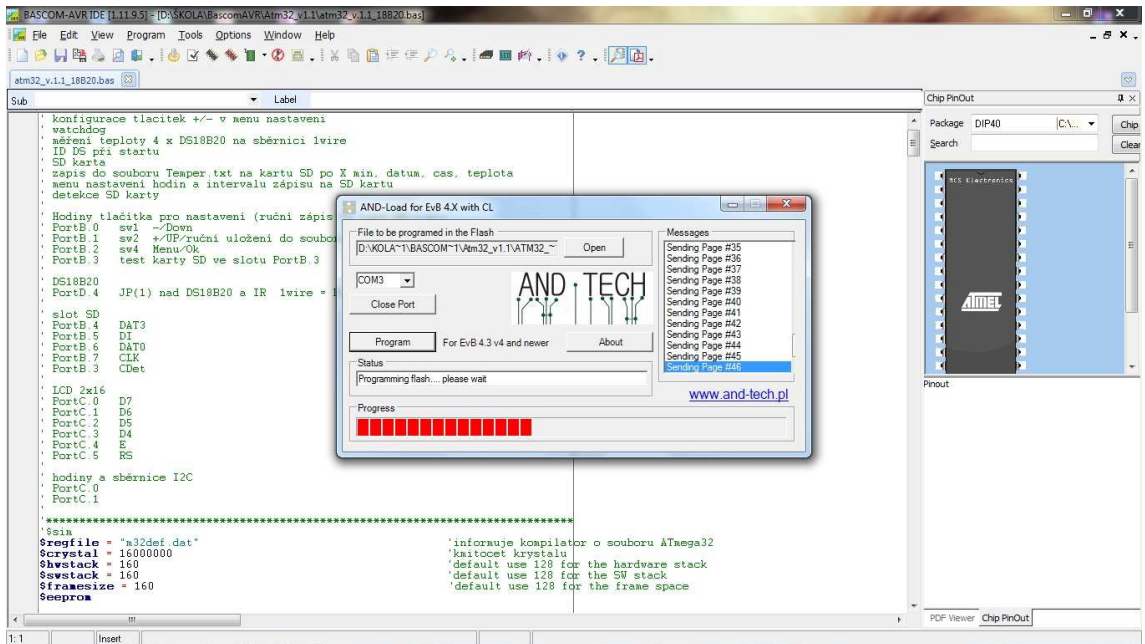


Obr. 24: BASCOM nastavení použitého procesoru



Obr. 25: okno Bascom – simulátor

Pro testování a odladění programu je možné využít simulátor mikroprocesorů, který je součástí prostředí BASCOM. Pomocí grafického znázornění je možné program procházet po jednotlivých krocích a sledovat stav hardwaru (registry, porty, atd.). Po kompilaci programu se provede jeho nahrání pomocí AND-Load for EvB do paměti mikroprocesoru.



Obr. 26: Nahrání programu do mikroprocesoru ATmega32

6 Závěr

Závěrem bych chtěl říci, že já osobně jsem s touto vývojovou deskou spokojen. Cílem práce bylo sestavit teplotní datalogger na vývojovém kitu EvB 4.3 s mikroprocesorem ATMEL ATmega32. To se pomocí periférií umístěných na desce EvB podařilo. Tato vývojová deska je velice univerzální díky přídavným perifériím. Tudíž použití a rozšíření této desky se meze nekladou.

Data pořízená měřením teploty jsou ukládána na SD kartu a po přenesení do osobního počítače zpracována a vyhodnocena dalšími nástroji. Funkčnost celé desky je odzkoušena a mohu říci, že funguje bezchybně. Do budoucna plánuji vylepšení desky. Ze začátku je programování malinko těžké, ale s touto vývojovou deskou je to zábava a hlavně základy BASCOM se naučí takřka hned.

Kalibrace senzorů teploty

Pro zjištění přesnosti údajů teplotních senzorů na podložce pro měření teploty fotovoltaického článku bylo provedeno srovnávací měření. Pro srovnávací měření byl použit přesný laboratorní rtuťový teploměr EXATHERM s rozsahem $-10\text{ }^{\circ}\text{C} \div +30\text{ }^{\circ}\text{C}$. Byla zvolena metoda Ice-Point, což je směs destilované vody a ledové tříště. Tato směs má po ustálení teplotu $0\text{ }^{\circ}\text{C}$

Senzor č.	teplota[$^{\circ}\text{C}$]	etalon [$^{\circ}\text{C}$]	rozdíl [$^{\circ}\text{C}$]
1	2,0000	0,4000	1,6000
2	1,0625	0,4000	0,6625
3	1,5625	0,4000	1,1625
4	1,5625	0,4000	1,1625

Tab. 4: Tabulka naměřených teplot při kalibraci

Rozdíl neodpovídá přesnosti udávané výrobcem ($\pm 0,5\text{ }^{\circ}\text{C}$) a bylo by potřeba vybírat z většího počtu senzorů, aby vyhověly dané toleranci.

7 Použitá literatura:

- [1] Občanské sdružení Eurosolar.cz Český sluneční informační server - Přehled obnovitelných zdrojů [online]. Poslední modifikace: 13.12.2007.
[cit. 27.12.2007]. Dostupné na URL: <http://www.eurosolar.cz/phprs/showpage.php?name=prehledoz>
- [2] Příručka k vývojovému kitu EvB 4.3 rev. 4, listopad 2010
- [3] Datasheet čidla teploty DALAS DS18B20
- [4] Fotovoltaika, Libra M., Poulek V., ISBN: 978-80-904311-0-2 ILSA
- [5] Fotovoltaika - elektřina ze slunce, Murtinger, Beranovský, Tomeš
ISBN: 978-80-7366-100-7 ERA
- [6] BASCOM – AVR User Manual, MCS Electronics 1995-2010
- [7] Elektrická měření, Vladimír Haasz, Vydavatelství ČVUT, Praha 2003
- [8] Matoušek, D.: Práce s mikrokontroléry ATMEL AT89C2051 1.díl. Praha, BEN,2002, 248 s. ISBN 80-7300-048-2
- [9] Matoušek, D.: Práce s mikrokontroléry ATMEL AVR 3.díl. Praha, BEN, 2003. 376 ISBN 80-7300-088-1
- [10] Matoušek, D.: Práce s mikrokontroléry ATMEL AVR ATmega16 4.díl. Praha, BEN, 2006, 320 s. ISBN 80-7300-174-8
- [11] Matoušek, D.: Práce s mikrokontroléry ATMEL AT89LP2052 5.díl. Praha, BEN, 2006. 200 s. ISBN 80-7300-205-1
- [12] Hrbáček, J.: Komunikace mikrokontroléru s okolím. Praha, BEN, 1999. 152 s. ISBN 80-86056-36-8
- [13] Váňa, V.: Mikrokontroléry ATMEL AVR - Bascom. Praha, BEN, 2004. 144 s. ISBN 80-7300-115-2

7.1 Webové zdroje:

<http://www.atmel.com>

<http://www.hw.cz>

<http://www.onpa.cz>

<http://www.earchiv.cz>

<http://www.mcselec.com>

<http://www.spezial.cz>

<http://www.mcselec.com>

<http://www.wikipedia.org>

Seznam pojmů a zkratk

Scratchpad	paměť ROM v čidle, kde se nachází potřebné registry pro práci
1wire pull-up rezistor	rezistor zajišťující log. 1 na sběrnici
1wire	sběrnice firmy Dallas semiconductor, komunikace po 1 drátu
DS18B20	čidlo firmy Dallas semiconductor
Cpp	vnitřní kondenzátor, který se využívá při parazitním napájení
Master	řídící zařízení 1wire sběrnice, většinou μ procesor
rodinný kód	kód druhu čidla (např. řada čidel DS18xXX)
MSB	více významný bit, u 8bit. registrů je to bit 7
LSB	poslední významný bit, u 8bit. registrů je to bit 0
convert (44h)	převod teploty
write scratchpad[4Eh]	zápis do paměti
read scatchpad[BEh]	čtení paměti
copy scatchpad[48h]	kopírování paměti
Datalogger	zařízení na záznam zjišťovaných dat
USB	universal serial bus, univerzální sériová sběrnice
RISC	Reduced Instruction Set Computer, redukováná inst. sada
EvB4.3	deska vývojového kitu
SDA	vodič u I2C sběrnice, po kterém se posílá adresa a data
SCL	vodič u I2C sběrnice, po kterém se posílá pracovní frekvence
JUMPER	propojka
EEPROM	elektricky mazatelná paměť
PWM	regulace pomocí pulsně šířkové modulace
A/D	převodník, převádí analogové hodnoty na digitální
IcePoint	metoda měření teploty 0 °C

8 Seznam obrázků

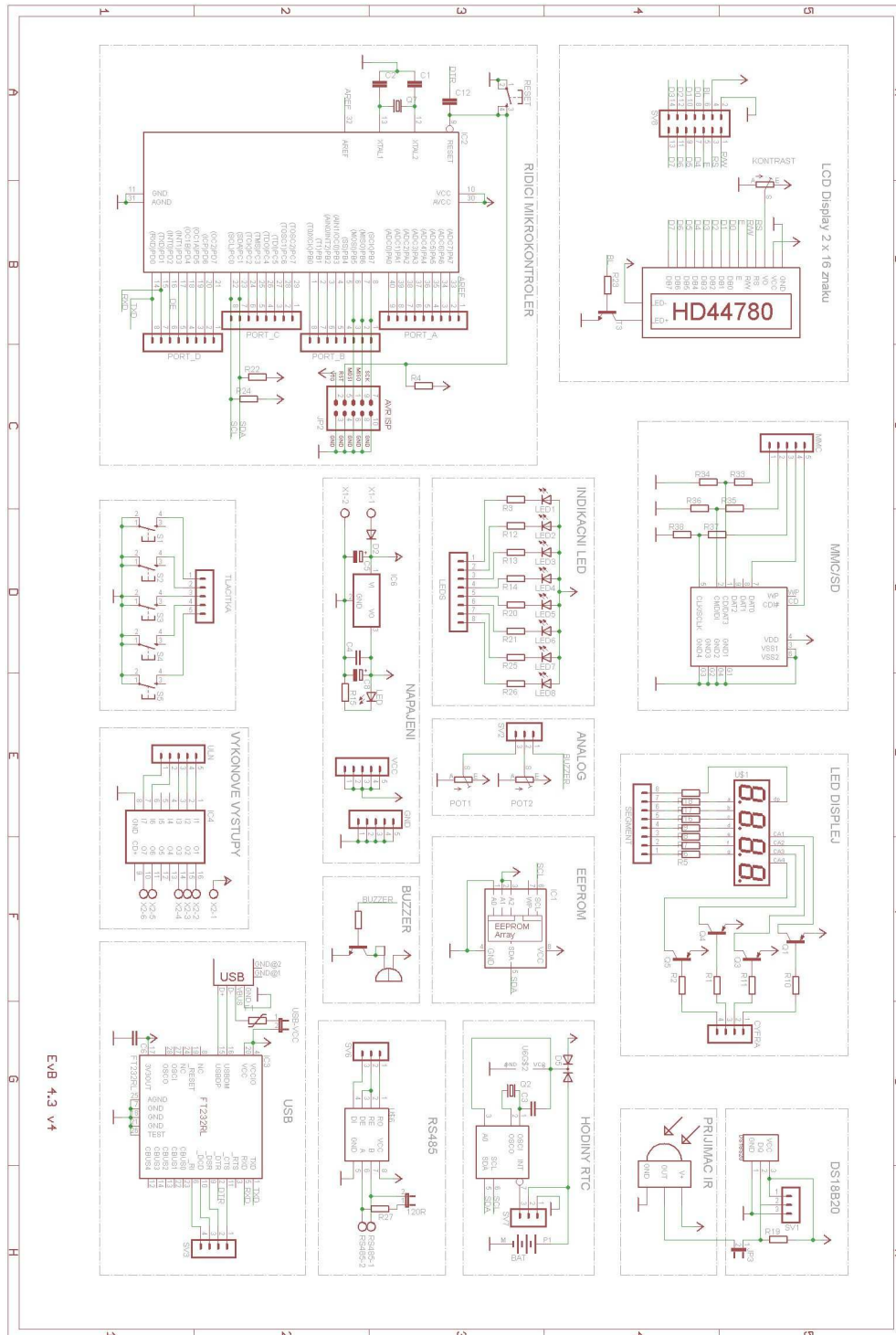
Obr. 1: Princip fotovoltaického jevu	Str.: 3
Obr. 2: Sestavení panelu z jednotlivých článků	4
Obr. 3: Celkový pohled na sestavu dataloggeru s podložkou pro FV článek	6
Obr. 4: Schéma zapojení napájení vývojového modulu EvB	6
Obr. 5: Schéma zapojení USB rozhraní s napájením	7
Obr. 6: Schéma zapojení LED indikátorů s obrázkem umístění na desce	7
Obr. 7: Schéma zapojení tlačítek a obrázek tlačítek na desce	8
Obr. 8: Schéma zapojení SD karty a obrázek slotu	8
Obr. 9: Schéma zapojení zobrazovací jednotky a její obrázek na desce	9
Obr. 10: Schéma zapojení EEPROM a PCF8583, obrázek osazení na desce	10
Obr. 11: Podložka pro měření teploty fotovoltaického článku	11
Obr. 12: propojení čidla DS18B20	11
Obr. 13: Blokové schéma čidla DALAS DS 18B20	12
Obr. 14: Funkční příkazy čidla DS18B20, vývojovým diagram	16
Obr. 15: Časování inicializační sekvence	19
Obr. 16: Read/Write průběh časového úseku	20
Obr. 17: Detailní časování masteru	21
Obr. 18: Doporučené časování čtení log. 1	22
Obr. 19: Schéma zapojení mikrokontroleru ATmega32 na desce	23
Obr. 20: Pouzdro DIL40 mikroprocesoru AVR ATmega32	23
Obr. 21: Rozložení vývodů ATmega32 – pouzdro DIP40	24
Obr. 22: RISC Architektura	26
Obr. 23: BASCOM základní okno programu	28
Obr. 24: BASCOM nastavení použitého procesoru	28
Obr. 25: okno Bascom – simulátor	29
Obr. 26: Nahrání programu do mikroprocesoru ATmega32	29

9 Seznam tabulek

Tab. 1: Formát teplotních registrů	Str.	13
Tab. 2: Vztah teplota/data		13
Tab. 3: DS18B20 Popis funkčních příkazů		22
Tab. 4: Tabulka naměřených teplot při kalibraci		30

10 Přílohy

- Příloha 1: Blokové schéma vývojové desky EvB 4.3 v4 [2]
- Příloha 2: Výpis programu
- Příloha 3: Ukázka výpisu uložených dat na MMC/SD kartě
- Příloha 4: Ukázka dat exportovaných do MS Excel
- Příloha 5: Grafy z MS Excel (závislost čas/teplota)



Příloha 1: Blokové schéma vývojové desky EvB 4.3 v4 [2]

Příloha 2: Výpis programu

```
' EvB 4.3 ATMEGA_32
'
' konfigurace tlacitek +/- v menu nastaveni
' watchdog
' měření teploty 4 x DS18B20 na sběrnici I2C
' ID DS při startu
' SD karta
' zapis do souboru Temper.txt na kartu SD po X min, datum, cas, teplota
' menu nastavení hodin a intervalu zápisu na SD kartu
' detekce SD karty
'
' Hodiny tlačítka pro nastavení (ruční zápis na kart SD - S2)
' PortB.0   sw1  -/Down
' PortB.1   sw2  +/UP/ruční uložení do souboru na SD
' PortB.2   sw4  Menu/Ok
' PortB.3   test karty SD ve slotu PortB.3
'
' DS18B20
' PortD.4   JP(1) nad DS18B20 a IR   I2C = Portd.4
'
' slot SD
' PortB.3   CDet
' PortB.4   DAT3
' PortB.5   DI
' PortB.6   DAT0
' PortB.7   CLK
'
' LCD 2x16
' PortC.0   D7
' PortC.1   D6
' PortC.2   D5
' PortC.3   D4
' PortC.4   E
' PortC.5   RS
'
' hodiny a sběrnice I2C
' PortC.0
' PortC.1
'
'*****
'$sim
$regfile = "m32def.dat"           'informuje kompilator o souboru
ATmega32
$crystal = 16000000              'kmitocet krystalu
$hwstack = 160                   'default use 128 for the hardware stack
$swstack = 160                   'default use 128 for the SW stack
$framesize = 160                 'default use 128 for the frame space
$eeprom

Config Watchdog = 2048           'reset after 2048 mSec

#include "Config_MMC.bas"         'Include AVR-DOS Configuration and library
#include "Config_AVR-DOS.BAS"

'*****

Declare Sub Inc_sd
```

```

Declare Sub Record_file(byval Uloz_data2 As String )
Declare Sub Testerrors(byval N As Byte )

Config Date = Mdy , Separator = .
Dim Btemp1 As Byte

Config Portb.3 = Input 'stav karty ve slotu
Sd_in_slot Alias Pinb.3
Set Portb.3 'stav karty ve slotu

Dim Uloz_data As String * 80

'nastavení tlačítek
Sw1 Alias Pinb.0
Sw2 Alias Pinb.1
Sw3 Alias Pinb.2

Set Portb.0
Set Portb.1
Set Portb.2

Config Lcd = 16 * 2 'konfigurace LCD
Config Lcdpin = Pin , Db4 = Portc.4 , Db5 = Portc.5 , Db6 = Portc.6 , Db7 =
Portc.7 , E = Portc.3 , Rs = Portc.2 'konfigurace LCD

Deflcdchar 1 , 4 , 10 , 4 , 32 , 32 , 32 , 32 , 32 'znak stupně

Declare Sub Settime
Declare Sub Gettime
Declare Sub Menu_setup 'konfigurace, menu setupu
Declare Sub Nast_hod 'Pod Menu nastavení hodin

Enable Interrupts

'=====

Declare Sub Setup_2

Dim Sekundy As Word
Dim Hist_off As Byte
Dim Hist_off_e As Byte
Dim Histereza As Word
Dim Histereza_e As Word
Dim Delta As Integer

'***** config the use port pin for I2C *****

Config I2cdelay = 5 'default slow
mode
Config Sda = Portc.1 'Portc.1
Config Scl = Portc.0 'Portc.0
I2cinit 'we need to set the pins in the proper state

'declare vairables
Dim S As Byte , M As Byte , H As Byte , D As Byte , Month As Byte
Dim Wm As Byte , Yd As Byte

```

```

Dim Year As Byte

'blikani : v hodinach
Dim Dvojtec As Bit
Dim Sw_ok As Bit , Sw_sav As Bit 'sw_sav 0min 15min 30min 45min co 15 min
Dim Time_out As Integer

'zapis po X minut
Dim Xmin As Byte 'zapis po X minut
Dim M_memory As Byte
Dim M_reszta As Byte

Xmin = 3
M_memory = 42

Declare Sub Up_set
Declare Sub Down_set
Dim Temp_zm As Byte
Dim B1 As Byte
Set B1.3

Config Debounce = 20

'##### DS1820 #####

Dim Nr_e_ds As Word 'pocet cidel ulozeno v EPROM
Dim Nr_ds As Word
Dim lwire_b1 As Byte , lwire_b2 As Byte , lwire_r As Byte
Dim lwire_t1 As Single
Dim lwire_t2 As Single
Dim lwire_tmp As String * 6
Dim lwire_znak As String * 1
Dim lwire_temperatura(2) As Byte '2 prvkove pole
Dim lwire_id(8) As Byte '8 prvkove pole
Dim lwire_i As Byte
Dim lwire_j As Byte
Dim lwire_k As Byte
Dim lwire_pocet_ds As Byte 'proměná do které je ulozen pocet snimacu na
sbernici lwire
lwire_r = &HFF
Config lwire = Portd.4 'konfigurace lwire na portd.4
Declare Sub Odcz_temp(snimac As Byte) 'procedura odedctu teploty
Declare Sub Zapis_id 'procedura zapisu cisla ID
Declare Sub Pocet_cidel(byval Silent As Byte) 'procedura poctu snimacu

Cursor Off Noblink
Cls
Lcd "SD DS18B20 v.1.1"
Locate 2 , 1
Lcd " Bohumir Tomasek"
Wait 2

Cls
Call Pocet_cidel(0) 'testu snimacu
Nr_e_ds = lwire_pocet_ds
Nr_ds = Nr_e_ds

If Nr_ds > lwire_pocet_ds Then

```

```

Cls
Lcd "Nastav snimace"
Lowerline
Lcd "press MENU"
Do
    Wait 1
    Loop Until Sw3 = 0                'pouze pri stisknuti tlacitka SW3
    Call Setup_2                    'podprogram nastavení
End If

'##### End DS1820 #####

Year = 011
Sw_sav = 0

Start Watchdog                    'start the watchdog timer

Do
    Reset Watchdog                'resetuje watchdog, 2s od ted
    Call Gettime                    'get time ,
    lwire_pocet_ds = lwirecount()  'pocet snimacu

    If Nr_ds > lwire_pocet_ds Then
        Set Porta.1
        Set Porta.0
        Cls
        Lcd "Pripoj snimac DS"
        Lowerline

        Wait 1
        Reset Watchdog            'resetuje watchdog, 2s od
ted
    End If

    If lwire_pocet_ds > 0 Then
        lwire_k = 0
        Do
            Incr lwire_k                'pocet snimacu lwire
            Call Odcz_temp(lwire_k)    'volani procedury mereni
        Loop Until lwire_k = lwire_pocet_ds
    End If

'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx procedura nastaveni xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

    Debounce Pinb.2 , 0 , Menu_setup , Sub

'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx zapis na kartu SD po X minutach xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

    If Sd_in_slot = 1 Then            'karta SD neni ve slotu
        Locate 1 , 15
        Lcd "SD"
        Locate 2 , 15
        Lcd "er"
    Else                                'karta SD je ve slotu
        Locate 1 , 15
        Lcd "SD"
        Locate 2 , 15
        Lcd "in"
        Debounce Pinb.1 , 0 , Inc_sd , Sub 'tl. s2 vyvola zapis na kartu SD

```

```

'----- test zapisu po X minutach -----
M = Makedec(m)
If M_memory <> M Then
    M_reszta = M Mod Xmin
    If M_reszta = 0 Then
        M_memory = M
        Call Inc_sd()
    End If
End If
End If
Loop

End

'***** 162 (0xA2) adresa zapisu a 163 adresa cteni (0xA3) *****
Sub Gettim
    Wm = 0
    I2cstart           'generate start
    I2cwbyte 162       'write address of PCF8583
    I2cwbyte 2         'select second register
    I2cstart           'generate repeated start
    I2cwbyte 163       'write address for reading info
    I2crbyte S , Ack   'read seconds
    I2crbyte M , Ack   'read minuts
    I2crbyte H , Ack   'read hours
    I2crbyte Yd , Ack  'read year and days
    I2crbyte Wm , Nack 'read weekday and month
    I2cstop            'generate stop
    Locate 1 , 1
    If Dvojtec = 1 Then
        Lcd Bcd(h) ; ":" ; Bcd(m)           '; ":" ; Bcd(s)  blikani
    dvojtecky
        Dvojtec = 0
    Else
        Lcd Bcd(h) ; " " ; Bcd(m)           '; ":" ; Bcd(s)
    blikani dvojtecky
        Dvojtec = 1
    End If

    If Year < 10 Then
        Uloz_data = "200" + Str(year) + "-" + Bcd(wm) + "-" + Bcd(yd) + ";" +
        Bcd(h) + ":" + Bcd(m) + ";"
    Else
        Uloz_data = "20" + Str(year) + "-" + Bcd(wm) + "-" + Bcd(yd) + ";" +
        Bcd(h) + ":" + Bcd(m) + ";"
    End If
    Locate 2 , 1
    Lcd Bcd(yd) ; "."           'zobrazeni dne
    Locate 2 , 4
    Lcd Bcd(wm)                 'zobrazeni mesice
End Sub

Sub Settime
' (s_1 As Byte , M_1 As Byte , H_1 As Byte , D_1 As Byte , Month_1 As Byte)
' values are stored as BCD values so convert the values first
' S = Makebcd(s_1)           'seconds
' M = Makebcd(m_1)           'minuts
' H = Makebcd(h_1)           'hours

```

```

' D = Makebcd(d_1) 'days
' Month = Makebcd(month_1) 'months
I2cstart 'generate start
I2cwbyte 162 'write address
I2cwbyte 0 'select control register
I2cwbyte B1 'set year and day bit for
masking
I2cstop 'generate stop
I2cstart 'generate start
I2cwbyte 162 'write mode
I2cwbyte 2 'select seconds Register
I2cwbyte S 'write seconds
I2cwbyte M 'write minuts
I2cwbyte H 'write hours
I2cwbyte Yd 'write days
I2cwbyte Wm 'write months
I2cstop
End Sub
Set_sw_ok:
Sw_ok = 1
Return
'***** MENU Setup *****

Sub Menu_setup

    Stop Watchdog ' test if the WD will stop
    Cls
    Lcd "Verze v1.1"
    Wait 1

'----- nastaveni hodin -----
    Cls
    Locate 1 , 1
    Temp_zm = 0
    Sw_ok = 0
    Time_out = 0
    Do
        Locate 1 , 1
        Lcd "Nastav hod: " ; Temp_zm ; " "
        Locate 2 , 1
        Lcd "1 Ano, 0 Ne"
        Debounce Pinb.1 , 0 , Up_set , Sub
        Debounce Pinb.0 , 0 , Down_set , Sub
        If Temp_zm > 1 Then Temp_zm = 0
        Debounce Pinb.2 , 0 , Set_sw_ok , Sub
        Time_out = Time_out + 1 'zabezpeceni ukonceni pokud se nic nedeje...
        If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1 'pouze tehdy, pokud nedošlo k vypršení časového
limitu, nebo žádné reakci provést nastavení nastavení času
    If Temp_zm = 1 Then
        ' S = Makebcd(0)
        Call Nast_hod
        Waitms 400
    End If

'----- nastaveni DS1820 -----

    Cls

```



```

Locate 1 , 1
Temp_zm = 0
Sw_ok = 0
Time_out = 0
Do
  Locate 1 , 1
  Lcd "Nastav ID DS: " ; Temp_zm ; " "
  Locate 2 , 1
  Lcd "1 Ano, 0 Ne"
  Debounce Pinb.1 , 0 , Up_set , Sub
  Debounce Pinb.0 , 0 , Down_set , Sub
  If Temp_zm > 1 Then Temp_zm = 0
  Debounce Pinb.2 , 0 , Set_sw_ok , Sub
  Time_out = Time_out + 1 'zabezpeceni ukonceni pokud se nic nedeje
  If Time_out > 1000 Then Sw_ok = 1
  Loop Until Sw_ok = 1 'pouze tehdy, pokud nedošlo k vypršení časového
limitu, nebo žádně reakci provést nastavení nastavení času
  If Temp_zm = 1 Then
    Call Setup_2
    Waitms 400
  End If
'----- RESET programu -----
Cls
Locate 1 , 1
Temp_zm = 0
Sw_ok = 0
Time_out = 0
Do
  Locate 1 , 1
  Lcd "Restart ? : " ; Temp_zm ; " "
  Locate 2 , 1
  Lcd "1 Ano, 0 Ne"
  Debounce Pinb.1 , 0 , Up_set , Sub
  Debounce Pinb.0 , 0 , Down_set , Sub
  If Temp_zm > 1 Then Temp_zm = 0
  Debounce Pinb.2 , 0 , Set_sw_ok , Sub
  Time_out = Time_out + 1 'zabezpeceni ukonceni pokud se nic nedeje
  If Time_out > 1000 Then Sw_ok = 1
  Loop Until Sw_ok = 1 'pouze tehdy, pokud nedošlo k vypršení časového
limitu, nebo žádně reakci provést nastavení nastavení času
  If Temp_zm = 1 Then
    Config Watchdog = 1024 'reconfig to 4 sec
    Cls
    Lcd "Reset - cekej... "
    Start Watchdog 'CONFIG WATCHDOG will disable the WD so start it
    Wait 5
    Cls
    Lcd "po resetu ?"
  End If

Temp_zm = Xmin
Sw_ok = 0
Cls
Do
  Locate 1 , 1
  Lcd "Zapis na SD"
  Locate 2 , 1
  Lcd "po min:" ; Temp_zm ; " "
  Debounce Pinb.1 , 0 , Up_set , Sub

```

```

    Debounce Pinb.0 , 0 , Down_set , Sub
    If Temp_zm > 30 Then Temp_zm = 0
    Debounce Pinb.2 , 0 , Set_sw_ok , Sub'zabezpeceni ukonceni pokud se nic
nedeje
    Time_out = Time_out + 1
    If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1
    Xmin = Temp_zm
    Waitms 400
    Cls
    Lcd "Konec Menu"
    Wait 2
    Cls
    Start Watchdog ' test if the WD will start
End Sub
'***** END MENU Setup *****

'***** Sub Setup_2 *****

Sub Setup_2

    Cls
    Lcd "Instal " ; lwire_pocet_ds ; " DS18B20"
    Lowerline
    Sekundy = 40
    If lwire_pocet_ds = 1 Then
        ' czujka pieca 1
        Nr_e_ds = 1
        Writeeprom Nr_e_ds , 21
        Nr_ds = Nr_e_ds
        Wait 2
    Else
        Do
            Locate 2 , 1
            Lcd "Pocet snimacu=" ; Nr_ds
            If Sw2 = 0 Then
                Sekundy = 40
                Incr Nr_ds
                If Nr_ds > lwire_pocet_ds Then Nr_ds = lwire_pocet_ds
            End If
            If Sw1 = 0 Then
                Sekundy = 40
                Decr Nr_ds
                If Nr_ds < 1 Then Nr_ds = 1
            End If
            Waitms 250
            Decr Sekundy
        Loop Until Sw3 = 0 Or Sekundy <= 2
        Nr_e_ds = Nr_ds
        Writeeprom Nr_e_ds , 21
    End If
    Wait 1

End Sub

'***** End Sub Setup_2 *****

```

```

***** Sub Nast_hod *****
Sub Nast_hod
'----- nastaveni minut -----
  Cls
  Locate 1 , 1
  Temp_zm = Makedec(m)
  Sw_ok = 0
  Time_out = 0
  Do
    Locate 1 , 1
    Lcd "Minut: " ; Temp_zm ; " "
    Debounce Pinb.1 , 0 , Up_set , Sub
    Debounce Pinb.0 , 0 , Down_set , Sub
    If Temp_zm > 59 Then Temp_zm = 0
    Debounce Pinb.2 , 0 , Set_sw_ok , Sub
    Time_out = Time_out + 1 'zabezpeceni ukonceni pokud se nic nedeje...
    If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1 'pouze tehdy, pokud nedošlo k vypršení časového
    limitu, nebo žádně reakci provést nastavení nastavení času
    If Time_out < 1000 Then
      M = Makebcd(temp_zm)
      Call Settime
      Waitms 400
    End If
    Cls
    Lcd "Ukladam"
    Wait 2
'----- nastaveni hodin -----
  Cls
  Locate 1 , 1
  Temp_zm = Makedec(h)
  Sw_ok = 0
  Do
    Locate 1 , 1
    Lcd "Hodin: " ; Temp_zm ; " "
    Debounce Pinb.1 , 0 , Up_set , Sub
    Debounce Pinb.0 , 0 , Down_set , Sub
    If Temp_zm > 23 Then Temp_zm = 0
    Debounce Pinb.2 , 0 , Set_sw_ok , Sub
    Time_out = Time_out + 1 'zabezpeceni ukonceni pokud se nic
nedeje...
    If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1
    H = Makebcd(temp_zm)
    Call Settime
    Waitms 400
    Cls
    Lcd "Ukladam"
    Wait 2
'----- nastaveni dne -----
  Cls
  Locate 1 , 1
  ' maska pro odecet roku z bitu 7,6 1100 0000 Year = &B11000000 And Yd Shift
Year , Right , 6
  Temp_zm = Makedec(yd)
  Sw_ok = 0
  Do
    Locate 1 , 1
    Lcd "Den: " ; Temp_zm ; " "

```

```

    Debounce Pinb.1 , 0 , Up_set , Sub
    Debounce Pinb.0 , 0 , Down_set , Sub
    If Temp_zm > 31 Then Temp_zm = 1
    Debounce Pinb.2 , 0 , Set_sw_ok , Sub
    Time_out = Time_out + 1      'zabezpeceni ukonceni pokud se nic
nedeje...
    If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1
    Yd = Makebcd(temp_zm)
    Call Settime
    Waitms 400
    Cls
    Lcd "Ukladam"
    Wait 2

'----- nastavení mesice -----

    Cls
    Locate 1 , 1
    Temp_zm = Makedec(wm)
    Sw_ok = 0
Do
    Locate 1 , 1
    Lcd "Mesic: " ; Temp_zm ; " "
    Debounce Pinb.1 , 0 , Up_set , Sub
    Debounce Pinb.0 , 0 , Down_set , Sub
    If Temp_zm > 12 Then Temp_zm = 1
    Debounce Pinb.2 , 0 , Set_sw_ok , Sub
    Time_out = Time_out + 1      'zabezpeceni ukonceni pokud se nic
nedeje...
    If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1
    Wm = Makebcd(temp_zm)
    Call Settime
    Waitms 400
    Cls
    Lcd "Ukladam"
    Wait 2

'----- nastaveni roku -----

    Cls
    Locate 1 , 1
    Temp_zm = Year
    Sw_ok = 0
Do
    Locate 1 , 1
    If Temp_zm < 10 Then
    Lcd "rok 200" ; Temp_zm ; " "
    Else
    Lcd "rok 20" ; Temp_zm ; " "
    End If
    Debounce Pinb.1 , 0 , Up_set , Sub
    Debounce Pinb.0 , 0 , Down_set , Sub
    If Temp_zm > 15 Then Temp_zm = 0
    Debounce Pinb.2 , 0 , Set_sw_ok , Sub
    Time_out = Time_out + 1      'zabezpeceni ukonceni pokud se nic nedeje...
    If Time_out > 1000 Then Sw_ok = 1
    Loop Until Sw_ok = 1
    Year = Temp_zm
    Waitms 400
    Cls
    Lcd "ulozeni HODIN"

```

```

    Wait 2
End Sub
'*****End Sub Nast_hod *****

Up_set:
    Temp_zm = Temp_zm + 1
    Time_out = 0
Return

Down_set:
    Temp_zm = Temp_zm - 1
    Time_out = 0
Return

'ooooooooooooooooooooo lwires DS18B20 nacteni snimacu oooooooooooooooooooooo

Sub Odcz_temp(snimac As Byte)          'procedura pomiaru temperatury
    lwire_i = Snimac * 8
    lwire_i = lwire_i - 7
    lwire_j = 0
Do
    Incr lwire_j
    Readeeprom lwire_id(lwire_j) , lwire_i 'cist ID snimace z pameti EEPROM
    Incr lwire_i                          'increase by one
Loop Until lwire_j = 8
lwreset
Call Zapis_id
lwwrite &H44
Waitms 350
lwreset                                  'reset lwire
Call Zapis_id

lwwrite &HBE
lwire_temperatura(1) = lwread(1)         'odecet zmerene teploty
lwire_temperatura(2) = lwread(1)
lwreset                                  'reset lwire
If lwire_temperatura(2) >= 248 Then      'v pripade zapornych teplot
    lwire_temperatura(1) = lwire_r - lwire_temperatura(1)
    lwire_temperatura(2) = lwire_r - lwire_temperatura(2)

    lwire_znak = "-"
Else
    lwire_znak = "+"
End If
lwire_t1 = lwire_temperatura(1) / 16     'presunuti o 4 bajty vpravo
lwire_t2 = lwire_temperatura(2) * 16     'presunuti o 4 bajty vlevo
lwire_t1 = lwire_t1 + lwire_t2
lwire_tmp = Fusing(lwire_t1 , "#.##&")
Uloz_data = Uloz_data + Str(snimac) + ";" + Str(lwire_t1) + ";"
Locate 1 , 7
Lcd "Cidlo " ; Str(snimac)
Locate 2 , 7
Lcd lwire_znak ; lwire_tmp ; Chr(1)
End Sub                                  'konec procedury mereni

```

```

'----- procedura cteni tabulky ID snimace -----

Sub Zapis_id                                     'definice procedury zapisu
  lwwrite &H55                                   'zaslani prikaru k zapsani
  For lwire_i = 1 To 8                           'cyklus vykonani I = 1 do
    lwwrite lwire_id(lwire_i)
  Next lwire_i
End Sub                                          'konec procedury

'----- procedura poctu snimacu na lwire -----

Sub Pocet_cidel(byval Silent As Byte)
  lwire_pocet_ds = lwirecount()                 'pocet snimacu lwire
  If Silent <> 0 Then
    Cls
    Lcd "Pocet snimacu: " ; lwire_pocet_ds      'LCD pocet snimacu
    Wait 1
  End If
  If lwire_pocet_ds > 0 Then                     'kdyz je pocet snimacu
vetsi 0 tak
    lwire_id(1) = lwsearchfirst()               'najdi prvni snimac
    For lwire_i = 1 To 8                         'cyklus 8x
      Writeeprom lwire_id(lwire_i) , lwire_i    'uloz ID do EEPROM 1...8
    Next lwire_i                                'inc I o jeden
  End If
  If lwire_pocet_ds > 1 Then                     'kdyz je pocet snimacu vetsi 1 tak
    lwire_id(1) = lwsearchnext()                'nacteni SN druheho DS1820
    lwire_j = 0                                  'pocatecni hodnota promene
    For lwire_i = 9 To 16                       'cxklus 8x I = 9 do I = 16
      Incr lwire_j                               'Inc J o jeden
      Writeeprom lwire_id(lwire_j) , lwire_i    'ulozeni ID do EEPROM
    Next lwire_i                                'Inc o jeden I
  End If
  If lwire_pocet_ds > 2 Then                     'kdyz je pocet snimacu vetsi 2 tak
    lwire_id(1) = lwsearchnext()                'nacteni SN tretiho DS1820
    lwire_j = 0                                  'pocatecni hodnota promene
    For lwire_i = 17 To 24                      'cyklus 8x od I = 17 do I = 24
      Incr lwire_j                               'Inc J o jeden
      Writeeprom lwire_id(lwire_j) , lwire_i    'ulozeni ID tretiho EEPROM
17...24
    Next lwire_i                                'Inc o jeden I
  End If
  If lwire_pocet_ds > 3 Then                     'kdyz je pocet snimacu vetsi 3, to
    lwire_id(1) = lwsearchnext()                'nacteni SN ctvrteho DS1820
    lwire_j = 0                                  'pocatecni hodnota
    For lwire_i = 25 To 32                     'cyklus 8x od I = 25 do I = 32
      Incr lwire_j                               'Inc J o jeden
      Writeeprom lwire_id(lwire_j) , lwire_i    'ulozeni ID ctvrteho EEPROM
25...32
    Next lwire_i                                'Inc o jeden I
  End If
  If Silent <> 0 Then
    Lowerline                                   'kursor na druhe radce LCD
    Lcd "Prog ok ID DS"                         'ukonceni nacteni ID snimacu
    Wait 1
    Cls
  End If

```

```

End Sub
'ooooooooooooooooooooooooooooo END lwires oooooooooooooooooooooooooooooo

'xxxxxxxxxxxxxxxxxxxxxxxxxxxxx zapis na SD kartu xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Sub Inc_sd
  Gbdriveerror = Driveinit()           'Inciace karty
  If Gbdriveerror = 0 Then
    Btemp1 = Initfilesystem(1)         ' Partition 1
    If Btemp1 <> 0 Then
      Cls
      Lcd "error: " ; Btemp1
      Locate 2 , 1
      Lcd "at Init f. system"
      Wait 2
    Else
      Locate 1 , 6
      Lcd "S"
      Call Record_file(uloz_data)
      Locate 1 , 6
      Lcd " "
    End If
  Else
    Print "Error during Drive Init: " ; Gbdriveerror
    Cls
    Lcd "Err Init:" ; Gbdriveerror
    Wait 2
  End If
End Sub

'*****
'*
'* Archives a file in the sorted directory tree, creating new directories
'* if necessary
'*
'*****
Sub Record_file(byval Uloz_data2 As String )
  Reset Watchdog
  Open "Temper.txt" For Append As #1
  Write #1 , Uloz_data2           ' write is also supported
  Flush #1
  Close #1
  Call Testerrors(1)
  Reset Watchdog
  Wait 1
  Locate 2 , 6
  Lcd "O"
  Reset Watchdog
  Wait 1
  Locate 2 , 6
  Lcd " "
  Reset Watchdog
End Sub

```

```
*****
'* Debug routine, test AVR-DOS error flag. If errors found signals user
'* flashing a LED and writes the error position code
*****

Sub Testerrors(byval N As Byte )
  While Gbdoserror <> 0
    Cls
    Lcd "err:" ; Gbdoserror ; " posit:" ; N
    Waitms 100
  Wend
End Sub
' xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx end SD karta xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Příloha 3: Ukázka výpisu uložených dat na MMC/SD kartě

Název souboru: temper.txt

```
"2011-04-13;16:45;1;24.9375;2;23.625;3;23.625;4;23.75;"
"2011-04-13;16:48;1;24.9375;2;23.5625;3;23.5625;4;23.6875;"
"2011-04-13;16:51;1;24.875;2;23.5625;3;23.5625;4;23.6875;"
"2011-04-13;17:12;1;24.5;2;23.5;3;23.4375;4;23.5625;"
"2011-04-13;17:15;1;24.75;2;23.5625;3;23.5625;4;23.6875;"
"2011-04-13;17:18;1;24.9375;2;23.6875;3;23.625;4;23.75;"
"2011-04-13;17:21;1;25.0625;2;23.75;3;23.6875;4;23.75;"
"2011-04-13;17:24;1;25.25;2;23.875;3;23.8125;4;23.875;"
"2011-04-13;17:27;1;25.4375;2;23.9375;3;23.9375;4;24.0;"
"2011-04-13;17:30;1;25.5;2;24.125;3;24.0625;4;24.0625;"
"2011-04-13;17:33;1;25.625;2;24.125;3;24.125;4;24.125;"
"2011-04-13;17:36;1;25.6875;2;24.25;3;24.1875;4;24.1875;"
"2011-04-13;17:39;1;25.6875;2;24.3125;3;24.25;4;24.3125;"
"2011-04-13;17:42;1;25.75;2;24.3125;3;24.3125;4;24.3125;"
"2011-04-13;17:45;1;25.9375;2;24.375;3;24.375;4;24.375;"
"2011-04-13;17:48;1;25.875;2;24.4375;3;24.375;4;24.4375;"
"2011-04-13;17:51;1;25.9375;2;24.5;3;24.4375;4;24.5;"
```

Formát zápisu je datum, čas, číslo senzoru (1,2,3,4), teplota. Hodnoty jsou odděleny středníkem. Nastaveno uložení po třech minutách.

Příloha 4: Ukázka dat exportovaných do MS Excel

Datum	Čas	Senzor okolni	Teplota 1 [°C]	Senzor kraj	Teplota 2 [°C]	Senzor kraj	Teplota 3 [°C]	Senzor střed	Teplota 4 [°C]
13.4.2011	16:30	1	24,6250	2	23,6250	3	23,5000	4	23,6875
13.4.2011	16:45	1	24,9375	2	23,6250	3	23,6250	4	23,7500
13.4.2011	16:48	1	24,9375	2	23,5625	3	23,5625	4	23,6875
13.4.2011	16:51	1	24,8750	2	23,5625	3	23,5625	4	23,6875
13.4.2011	17:12	1	24,5000	2	23,5000	3	23,4375	4	23,5625
13.4.2011	17:15	1	24,7500	2	23,5625	3	23,5625	4	23,6875
13.4.2011	17:18	1	24,9375	2	23,6875	3	23,6250	4	23,7500
13.4.2011	17:21	1	25,0625	2	23,7500	3	23,6875	4	23,7500
13.4.2011	17:24	1	25,2500	2	23,8750	3	23,8125	4	23,8750
13.4.2011	17:27	1	25,4375	2	23,9375	3	23,9375	4	24,0000
13.4.2011	17:30	1	25,5000	2	24,1250	3	24,0625	4	24,0625
13.4.2011	17:33	1	25,6250	2	24,1250	3	24,1250	4	24,1250
13.4.2011	17:36	1	25,6875	2	24,2500	3	24,1875	4	24,1875
13.4.2011	17:39	1	25,6875	2	24,3125	3	24,2500	4	24,3125
13.4.2011	17:42	1	25,7500	2	24,3125	3	24,3125	4	24,3125
13.4.2011	17:45	1	25,9375	2	24,3750	3	24,3750	4	24,3750
13.4.2011	17:48	1	25,8750	2	24,4375	3	24,3750	4	24,4375
13.4.2011	17:51	1	25,9375	2	24,5000	3	24,4375	4	24,5000
13.4.2011	17:54	1	26,0000	2	24,5000	3	24,5000	4	24,5000
13.4.2011	17:57	1	26,0625	2	24,5625	3	24,5625	4	24,5625
13.4.2011	18:00	1	26,0625	2	24,6250	3	24,6250	4	24,6250
13.4.2011	18:03	1	26,1250	2	24,6250	3	24,6250	4	24,6250
13.4.2011	18:06	1	26,1875	2	24,6875	3	24,6875	4	24,6875
13.4.2011	18:09	1	26,1875	2	24,7500	3	24,6875	4	24,6875
13.4.2011	18:12	1	26,2500	2	24,7500	3	24,7500	4	24,7500
13.4.2011	18:15	1	26,1875	2	24,8125	3	24,7500	4	24,7500
13.4.2011	18:18	1	26,2500	2	24,8125	3	24,8125	4	24,8125
13.4.2011	18:21	1	26,3125	2	24,8750	3	24,8125	4	24,8125
13.4.2011	18:24	1	26,3750	2	24,8750	3	24,8750	4	24,8750
13.4.2011	18:27	1	26,3125	2	24,8750	3	24,8750	4	24,8750
13.4.2011	18:30	1	26,3125	2	24,8750	3	24,9375	4	24,8750
13.4.2011	18:33	1	26,3750	2	24,9375	3	24,9375	4	24,9375
13.4.2011	18:36	1	26,3750	2	24,9375	3	24,9375	4	24,9375
13.4.2011	18:39	1	26,4375	2	24,9375	3	24,9375	4	24,9375
13.4.2011	18:42	1	26,4375	2	25,0000	3	25,0000	4	24,9375
13.4.2011	18:45	1	26,5625	2	25,0000	3	25,0000	4	25,0000
13.4.2011	18:48	1	26,5625	2	24,9375	3	25,0000	4	25,0000
13.4.2011	18:51	1	26,5000	2	25,0000	3	25,0000	4	25,0000
13.4.2011	18:54	1	26,5625	2	25,0000	3	25,0000	4	25,0000
13.4.2011	18:57	1	26,5625	2	25,0625	3	25,0625	4	25,0625
13.4.2011	19:00	1	26,5625	2	25,0000	3	25,0625	4	25,0625
13.4.2011	19:03	1	26,6250	2	25,0625	3	25,0625	4	25,0625
13.4.2011	19:06	1	26,6250	2	24,9375	3	25,0000	4	24,9375
13.4.2011	19:09	1	26,1875	2	24,7500	3	24,8125	4	24,8125
13.4.2011	19:12	1	26,1250	2	24,6250	3	24,6875	4	24,6875
13.4.2011	19:15	1	25,9375	2	24,5000	3	24,6250	4	24,6250
13.4.2011	19:18	1	25,8125	2	24,3750	3	24,5000	4	24,5000
13.4.2011	19:21	1	25,7500	2	24,3125	3	24,4375	4	24,5000
13.4.2011	19:24	1	25,7500	2	24,2500	3	24,3750	4	24,4375
13.4.2011	19:27	1	25,6875	2	24,2500	3	24,3750	4	24,4375
13.4.2011	19:30	1	25,6875	2	24,2500	3	24,4375	4	24,4375
13.4.2011	19:33	1	25,6875	2	24,3125	3	24,4375	4	24,4375
13.4.2011	19:36	1	25,6875	2	24,2500	3	24,3750	4	24,3750
13.4.2011	19:39	1	25,8125	2	24,3125	3	24,5000	4	24,4375

Příloha 5: Grafy z MS Excel (závislost teplota/čas)

