

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra elektroenergetiky a ekologie

DIPLOMOVÁ PRÁCE

Teplovní datalogger pro domovarní výrobu piva

Originál (kopie) zadání BP/DP

Abstrakt

Diplomová práce je zaměřena na návrh zařízení s nízkou spotřebou pro snímání teplotních a vlhkostních dat. Správnost návrhu je ověřena měřením spotřeby. Zařízení má sloužit k zlepšení efektivity technologického postupu vaření piva. Funkčnost zařízení je demonstrována na krátkém měření teploty a vlhkosti.

Klíčová slova

Datalogger, teplotní datalogger, domovarní výroba piva, mikrokontroler, SD karta

Abstract

The master theses presents design of low power device for measuring temperature and humidity. Correctness of this design is tested by measurement of power consumption. Device should improve efficiency of the process of brewing home-made beer. Funkcionality of the device is shown on short measurement of temperature and humidity.

Key words

Datalogger, temperature datalogger, home brewery, microcontroller, SD card

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 16.5.2016

Jiří Janouš

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Petrovi Burianovi, Ph.D. a konzultantovi Ing. Janu Krpálkovi, za cenné profesionální rady, připomínky a metodické vedení práce.

Obsah

OBSAH	7
SEZNAM SYMBOLŮ A ZKRATEK	8
ÚVOD	9
1 DOMÁCÍ VÝROBA PIVA	10
1.1 POSTUP PŘI POUŽITÍ ZÁKLADNÍCH SUROVIN.....	10
NÁROKY NA TEPLTNÍ DATALOGGER	12
2 NÁVRH ZAŘÍZENÍ	13
2.1 MIKROKONTROLÉR EFM ZERO GECKO.....	14
2.2 VÝVOJOVÁ DESKA	15
2.3 MĚŘENÍ TEPLoty A VLHKOSTI.....	16
2.4 UKLÁDÁNÍ DAT.....	19
3 NÁVRH VE VÝVOJOVÉM PROSTŘEDÍ	22
4 MĚŘENÍ SPOTŘEBY	26
4.1 MĚŘENÍ A SCHÉMA ZAPOJENÍ.....	26
4.2 CYKLUS MĚŘENÍ A ZÁPISU NA SD KARTU	27
DETAIL PRŮBĚHU:	28
4.2.1 <i>Porovnání spotřeby MCU v různých energy modech</i>	29
4.2.2 <i>Průběh proudu při zapnutí zařízení</i>	30
5 MĚŘENÍ VLHKOSTI A TEPLoty	30
ZÁVĚR	33
SEZNAM ZDROJŮ	34
SEZNAM OBRÁZKŮ	36
SEZNAM TABULEK	36
PŘÍLOHY	37

Seznam symbolů a zkratk

<i>ARM</i>	Acorn RISC Machine
<i>CPU</i>	Central processing unit
<i>CPU</i>	Central processing Unit
<i>DMA</i>	Direct Memory acces
<i>EFM</i>	Energy friendly microcontroller
<i>EM0-4</i>	Energy mode 0 až 4
<i>FAT</i>	File allocation table
<i>GPIO</i>	General Purpose Input Output
$I_{(t)}$	Okamžitý proud při startu zařízení [mA]
$I_{(t)AV}$	Průměrný proud při startu zařízení [mA]
<i>I2C</i>	Inter Integrated Circuit sběrnice
<i>IAR</i>	Švédská SW firma Ingenjörfirman Anders Rundgren
I_{em0}	Okamžitý proud v energy modu 0 [mA]
I_{em0AV}	Průměrný proud v energy modu 0 [mA]
I_{em2}	Okamžitý proud v energy modu 2 [mA]
I_{em2AV}	Průměrný proud v energy modu 2 [mA]
<i>LEUART</i>	Low Energy Universal Asynchronous Reciever Transciever
<i>MCU</i>	Micro Controlling unit
<i>RISC</i>	Reduced Instruction Set
<i>RTC</i>	Real time counter
<i>SD</i>	Secure Digital paměťová karta
<i>SPI</i>	Single peripheral interface

Úvod

Práce se zabývá návrhem zařízení, které by umožnilo sledovat a zefektivnit proces výroby piva v domácích podmínkách.

V první části je čtenář seznámen s problematikou výroby piva v domácích podmínkách a výhodami spojenými s možností monitorovat tento složitý technický proces. V další části je proveden výběr hardwarových komponent, které byly využity pro stavbu zařízení. Třetí kapitola se zabývá návrhem softwarové části, jsou zde zmíněny metody pro konfiguraci zařízení a zefektivnění měřicího procesu. Měřením je prozkoumána efektivita jednotlivých energetických modu. V závěru je ukázán výstup z dataloggeru.

1 Domácí výroba piva

Potřebné suroviny pro výrobu piva jsou obiloviny, voda, chmel a případně koření. Obiloviny jsou zdrojem škrobů, které se přemění při fermentaci na cukr alkohol a oxid uhličitý. Chmel dodává hořkost a pomáhá pivo konzervovat. Koření není nutností, ale dodává jedinečnost a napomáhá seberealizaci, proto se v domovarní výrobě často využívá. Velmi důležitá je při výrobě kvalitní voda. [5] [6]

Výrobu piva je možné provádět dvěma způsoby. Použitím základních surovin, nebo za použití sladinového extraktu. Sladinový extrakt je při nízkém tlaku zkondenzovaná sladina. [5] [6]

1.1 Postup při použití základních surovin

Sladování

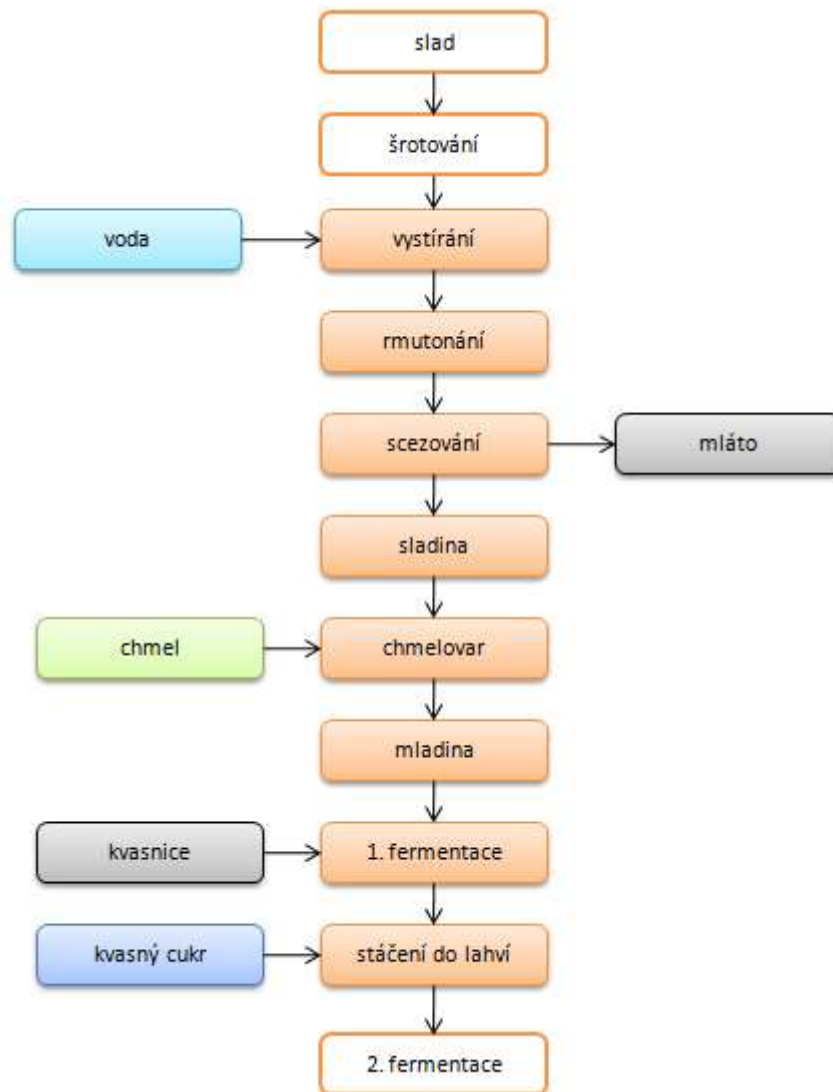
Existuje několik druhů sladu, postup je ale vždy obdobný. Ječmen se namáčí a nechá se kolem 3 dnů klíčit. Klíčení aktivuje enzymy v zrnech. Po dobu klíčení se musí ječmen pravidelně obracet. Po ukončení klíčení se ječmen suší při různých teplotách. Takto usušeným zrnům se říká slad. [5] [6]

Šrotování, vystírání, rmutování, chlazení

Slad se sešrotuje a smíchá se s vodou (vystírání). Takto získaná tekutina se uvede do teploty 63-72°C získají se tak cukry pro fermentaci, tekutina se scedí. Přidá se chmel a vaří se po dobu 120 minut. Pokud připravujeme pivo ze sladinového extraktu, tak po naředění extraktu se dostáváme do stejné fáze výroby jako v předchozím případě a postup je od této chvíle pro obě metody stejný. [5] [6]

Mladina se zchladí na kvasící teplotu a naplní se do nádoby, ve které bude fermentovat po dobu 1 až 2 týdnů. To záleží na teplotě a druhu piva, které vyrábíme. Fermentace je hotová, když se přestane uvolňovat oxid uhličitý a pivo přestane být kalné. Po ukončení prvotního kvašení se pivo slije do láhví, přidá se dextróza a pivo se nechá ještě 1 týden kvasit. Poté se pivo přemístí na chladné místo a nechá se ještě dozrát. Takto můžeme mít pivo uskladněno až půl roku. [5] [6]

Na následujícím obrázku jsou zobrazeny výše popsání kroky při výrobě piva



Obr. 1.1 vývojový diagram domácí výroby piva [vlastní zpracování]

Jak je patrné z popisu, vaření piva je technologicky poměrně náročný proces, je nutné přesně dodržovat přesně definované hodnoty vlhkosti a teploty. Kvůli náročnosti postupu je žádoucí vyrobit alespoň 30 a více litrů. Takové množství piva už obtížně uskladníme do lednice a kontrola teploty proto není možná.

V kontrolovaných podmínkách je fermentace ukončena po určité době. V domácích podmínkách je kvůli proměnlivé teplotě potřeba tento okamžik těžké odhadnout a to zejména při druhé fermentaci, kdy oxid uhličitý neprobublává z nádoby. Je proto potřeba proměnlivou teplotu zaznamenávat. Pokud se fermentační cyklus neodehrává podle předpokladu, je možné se podívat na průběh teplot a určit, zda teplota neklesla příliš nízko, což by negativně ovlivnilo aktivitu kvasinek, nebo zda nebyla příliš vysoko, což by znamenalo úhyn kvasinek. S těmito informacemi je pak možné přitopit v místnosti a stimulovat aktivitu kvasinek, nebo fermentační proces přerušit a uvolnit nádobu na další várku. Pokud budeme vyrábět vlastní

slad, tak je potřeba stejným způsobem sledovat průběh sladování. Pak je třeba sledovat nejen teplotu ale i vlhkost.

Zaznamenáváním teploty a vlhkosti je tak možné překonat jisté nedostatky domovarní výroby piva, kdy nemáme přesnou kontrolu nad teplotou a vlhkostí. Zaznamenané hodnoty poslouží k vyladění technického postupu.

Nároky na teplotní datalogger

V této kapitole jsou shrnuty jednotlivé požadavky, podle kterých bylo přistupováno při výběru vhodného hardwaru.

- Měření teploty. Pro správnou fermentaci je vhodné získávat teplotní data z několika míst. V různých částech místnosti nebo z několika nádob, ve kterých probíhá fermentace.
- Měření vlhkosti. Během sladování je potřeba měřit vlhkost, aby bylo možné analyzovat, zda vše probíhá podle technologického postupu.
- Možnost zaznamenávání dat. Teplotní data je nutné zaznamenávat po celou dobu fermentace do nějakého druhu paměti. Uživatel se potom spojí přímo se zařízením nebo k odnímatelné paměti, a data vyčte k dalšímu zpracování. Paměť musí být standardizována, aby užití dataloggeru nebylo limitováno technologií.
- Nízká spotřeba zařízení. Celé zařízení musí vykonávat svoji funkci po co nejdelší dobu, nejméně však alespoň jeden fermentační cyklus, který představuje nejdelší technologickou operaci procesu vaření piva.
- Konfigurace měření. Domovní prostory se budou lišit (sklep, stinný pokoj), budou v nich panovat jiné klimatické podmínky, teplota a vlhkost se budou měnit jiným tempem. Zařízení musí být opatřeno nějakým způsobem konfigurace měřícího procesu, abychom zohlednili tyto předpoklady.

2 Návrh zařízení

Při výběru mikrokontroleru byl brán ohled na spotřebu, a proto byl použit cortex M0 procesor. Jedná se o nejmenší řadu procesorů s ARM architekturou. Architektura byla vynalezena firmou Acorn a dnes je velmi rozšířena. Mikrokontrolery ARM jsou rozděleny podle použití na cortex A, R, M. Řada M je vytvořena pro průmyslové ovládání a je nejúspěšnější. Cortex M je rozdělen na řady M4-0, M0 je nejúspěšnější řada. [1]

Přehled mikrokontrolerů cortex 0

název	f [MHz]	typ jádra	Flash[KB]	SRAM[MB]	SPI	USART	UART	I2C	Ethernet
STM32F0	48	Cortex-M0	16-64	4-16	2	2	0	2	Ne
LPC800	30	Cortex-M0+	4-16	1-4	2	0	3	1	Ne
LPC11xx	50	Cortex-M0	4-128	1-12	2	0	1	1	Ne
LPC12xx	30	Cortex-M0	32-128	4-8	1	0	2	1	Ne
PSoC 4000	48	Cortex-M0	32	4	2	0	1	1	Ne
EFM32ZG	24	Cortex-M0	4-32	2-3	1	1	1	1	Ne
NUC 100	50	Cortex-M0	32-128	4-16	2	0	2	2	Ne
NUC 110	50	Cortex-M0	32-128	4-16	3	0	5	2	Ne
NUC 120	50	Cortex-M0	32-128	4-16	4	0	3	2	Ne
NUC 130	50	Cortex-M0	32-128	4-16	4	0	3	2	Ne
NUC 140	50	Cortex-M0	32-128	4-16	4	0	3	2	Ne
KL0	48	Cortex-M0	32	4	1	0	1	1	Ne
KL1	48	Cortex-M0	64	8	2	0	1	2	Ne
KL2	48	Cortex-M0	64	8	2	0	2	2	Ne

Tab. 2.1 přehled mikrokontrolerů ARM cortex0 [vlastní zpracování] [2] [7] [8] [19]

Přehled mikrokontrolerů cortex 0 druhá část

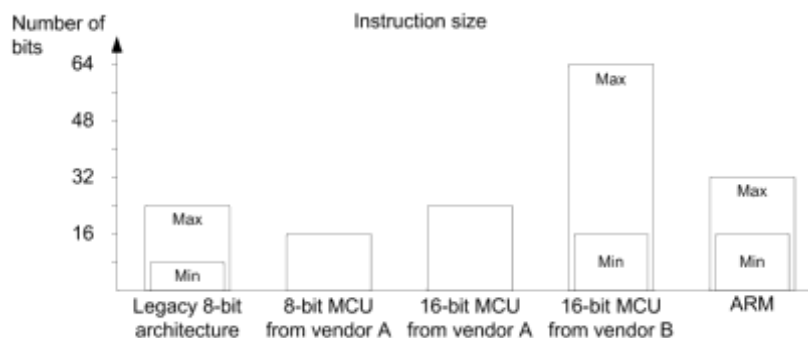
název	časovače	DMA	ADC/kanály	DAC	komparátor	RTC	PWM	vvyrobce
STM32F0	11	5	1/16	0	2	1	6	St microelectronic
LPC800	4	0	0	0	0	0	4	NXP semiconductor
LPC11xx	4	0	1/8	0	0	0	13	NXP semiconductor
LPC12xx	4	21	1/8	0	0	1	13	NXP semiconductor
PSoC 4000	4	0	1/8	2	2	0	0	cypress
EFM32ZG	4	4	1/4	0	1	1	6	silicon labs
NUC 100	4	9	1/8	2	2	1	4	Nuvotron
NUC 110	4	9	0	2	2	1	8	Nuvotron
NUC 120	4	9	1/8	2	2	1	8	Nuvotron
NUC 130	4	9	1/8	2	2	1	8	Nuvotron
NUC 140	4	9	1/8	2	2	1	8	Nuvotron
KL0	2	4	1/1	0	0	1	2	freescale
KL1	3	4	1/1	0	1	1	2	freescale
KL2	3	4	1/1	0	1	1	2	freescale

Tab. 2.2 přehled mikrokontrolerů ARM cortex0 druhá část [vlastní zpracování] [2] [7] [8] [19]

2.1 Mikrokontrolér EFM Zero Gecko

Jedná se o „low power“ mikrokontrolér, který je založen na 32-bitovém procesorovém jádru ARM Cortex-M0. To nabízí lepší hustotu kódu než 8-bit. a 16-bit. architektury, což má pozitivní dopad v podobě ušetřeného místa v programové paměti. [3] [4]

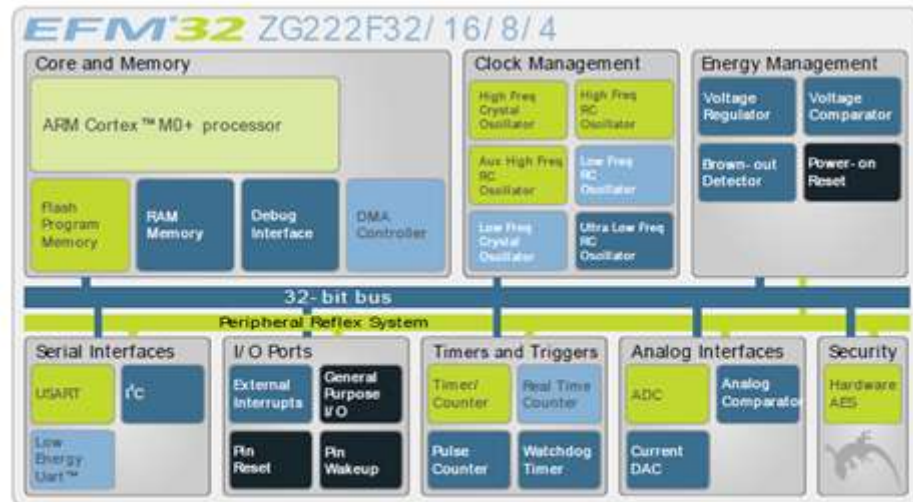
Je častým nedorozuměním, že 8 bitové mikrokontrolery používají 8 bitové instrukce a mikrokontrolery založené na procesorech ARM Cortex-M0+ používají 32 bitové instrukce. Velikosti instrukcí jsou na obrázku níže. [3] [4]



Obr. 2.1 přehled velikosti instrukcí [3]

Pro stejný úkol tak může být zapotřebí méně instrukcí, což znamená menší velikost kódu.

V dnešní době je vysoká poptávka po levných a sofistikovaných zařízeních, které dokážou rychle komunikovat s analogovými senzory. Pro některá 8-bitová zařízení by to znamenalo zvýšit frekvenci a to má za následek zvýšení spotřeby. [3] [4]



Obr. 2.2 blokové schéma mikrokontroleru EFM32ZG [11]

Mikrokontroler EFM Zero Gecko se skládá z procesorového jádra, statické RAM paměti, programové Flash paměti, EMU a dalších periférii. EFM je zkratkou Energy Friendly Microcontroller. [11]

Pozn. (Název gecko inspiroval David Attenborough, který udělal snímek o plazech a obojživelnících. Ti spotřebují 10% energie jako savec stejné velikosti.)

Spotřeba udávána výrobcem MCU je při frekvenci 24 MHz a napětí 3V 180 μ A/MHz při normálním průběhu kódu z flash paměti. Ve stand-by modu je spotřeba ještě nižší, výrobce garantuje dobu přechodu do energy modu EM0 během 2 μ s. Přístup do paměti DMA a komunikace LEUART je možné vykonávat bez zásahu MCU, to přispívá k hospodárnosti mikrokontroleru. Toto jsou však pouze deklarované hodnoty a jejich dosažení v praxi může být složité. Tyto faktory jsou zdokumentovány v kapitole měření spotřeby. [11]

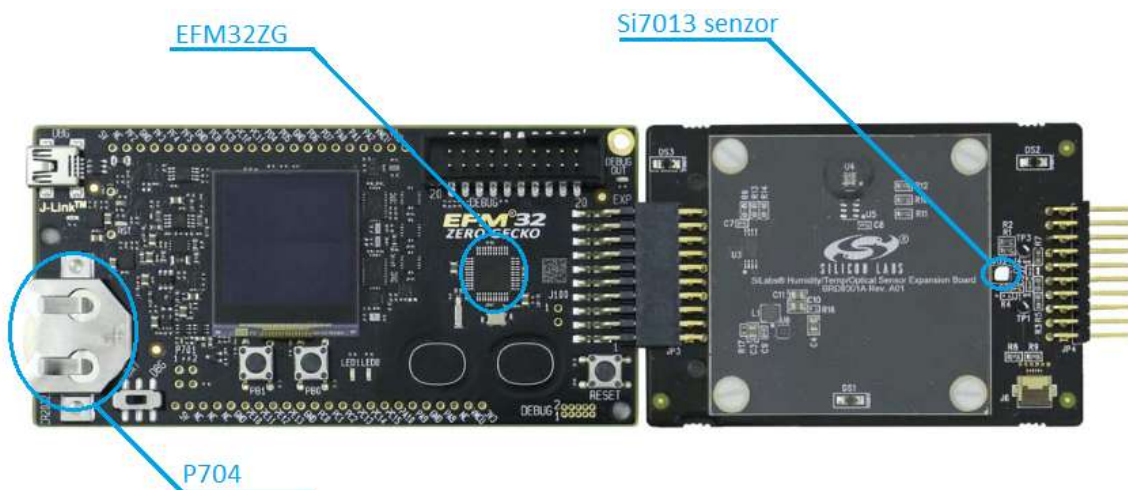
2.2 Vývojová deska

Pro realizaci samotného zařízení byla zvolena vývojová deska efm Zero Gecko od firmy Silicon Labs. Vývojová deska obsahuje mikroprocesor a prostředky potřebné pro využívání

dané platformy. Kvůli velkému množství ovládacích a měřicích prvků byla zvolena právě tato varianta.

Vyžití vývojové desky na začátku projektu velmi usnadňuje a akceleruje vývojové práce. Odpadá nutnost okamžitého návrhu a zhotovení desky plošných spojů a mikrokontrolerem senzorem a potřebnými perifériemi. Vývojář se tak může zaměřit na SW část projektu a finální hardware může navrhnout až v okamžiku, kdy bude verifikována základní funkcionality projektu na dané platformě. V některých případech není důvod nevyužít vývojovou desku jako finální HW, případně je možné využít její schéma jako východisko pro specializovaný HW, který bude splňovat specifické nároky aplikace (včetně mechanických parametrů – velikost, zapouzdření do krabičky atd.).

Na obrázku (2.3) jsou vidět jednotlivé části desky, deska obsahuje dvě kapacitní tlačítka, LCD display, 2 led diody, konektor pro baterii, in/out debug konektor. Debug konektor lze použít při debugování vlastního navrženého zařízení s mikrokontrolerem. [9]

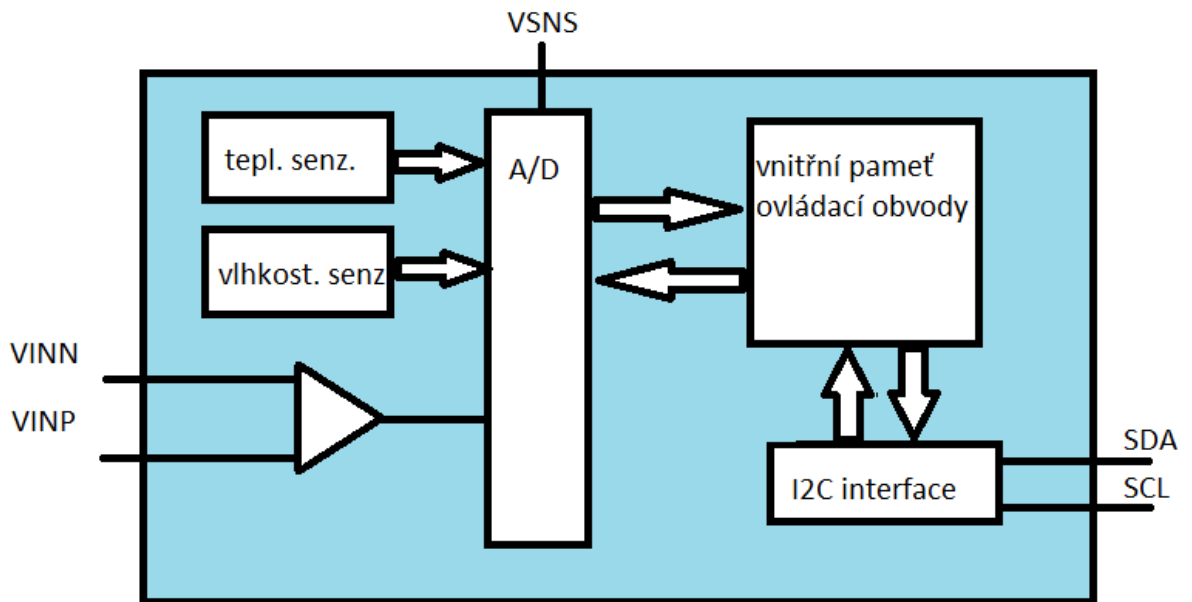


Obr. 2.3 vývojová deska a použité periferie [9]

2.3 Měření teploty a vlhkosti

Bylo by vhodné kromě teploty měřit i vlhkost vzduchu, zejména při výrobě vlastního sladu. Proto bylo zvoleno integrované čidlo teploty a vlhkosti Si7013, jedná se o teplotní I2C čidlo. Rozsah napájecího napětí je od 1,9 do 3,6 V. Vývojová deska je napájena akumulátorem s 3V. Výrobce uvádí přesnost teplotního senzoru $\pm 0,3\%$. Přesnost vlhkostního senzoru má

hysterezní charakter. Film na povrchu senzoru absorbuje vlhkost a tak při přechodu z nižších vlhkostí udává sensor hodnoty s negativní odchylkou a při přechodu z vyšších naopak. Čidlo má tovární kalibraci a není do něj nutné zasahovat. V případě nutnosti kalibrace obvod obsahuje vstupy pro externí senzory. [17]

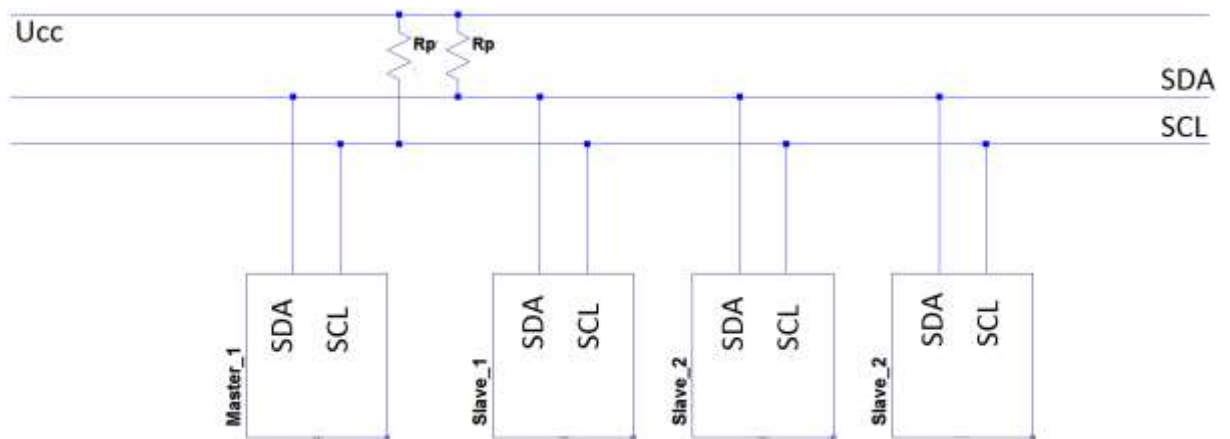


Obr. 2.4 blokové schéma teplotního čidla Si7013 [vlastní zpracování][17]

System využívá I2C sběrnici pro přenos dat, zařízení má 7 bitovou adresu, kde 6 nejvýznamnějších bitů je fixní, a poslední, LSB bit, je možné nastavit adresovacím pinem. Je tak možné připojit 2 senzory na jednu sběrnici.

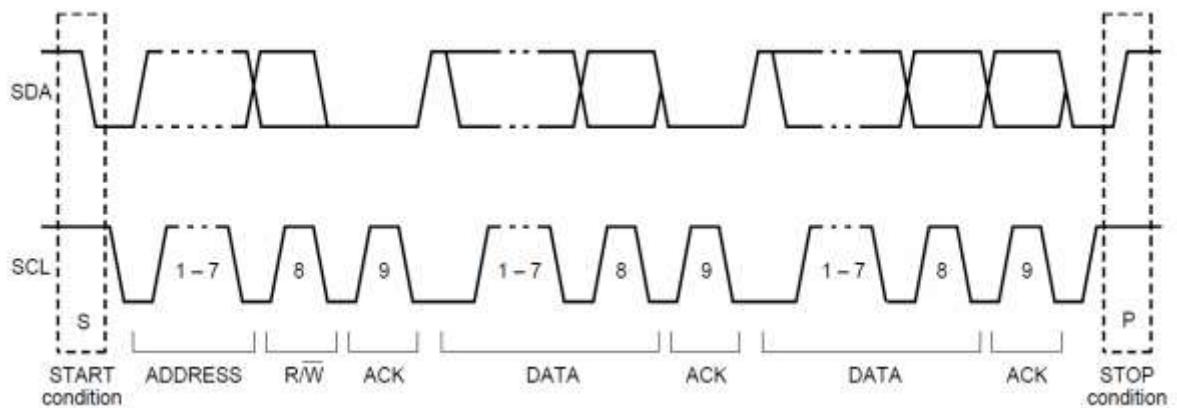
Přenos dat po I2C

Přenos dat probíhá po dvou vodičích SDA SCL. Vodiče jsou připojeny na napájecí napětí přes pullup rezistor. V klidovém stavu je na sběrnici napájecí napětím. Na sběrnici bývá jeden master a několik zařízení slave. Master řídí komunikaci na sběrnici. [21]



Obr. 2.5 blokové schéma zapojení I2C sběrnice [vlastní zpracování]

Na začátku komunikace je start bit, master stáhne sběrnici do nízkého stavu a oznámí ostatním zařízením, že bude vysílat. Následuje 7 bitová adresa s nastavitelným bitem, kdy master určí, se kterým slave zařízením hodlá komunikovat. Poté se vyšle read/write bit, kde master rozhodne, zda bude ze zařízení číst nebo zapisovat, vždy po 8 bitech následuje acknowledge bit. Pak následuje přenos dat, ten je zakončen stop bitem. [21]



Obr. 2.6 časové průběhy komunikace po I2C sběrnici [21]

2.4 Ukládání dat

Pro ukládání teplotních dat byla zvolena Secure Digital (SD) karta, která je dnes široce používaná pro přenosná zařízení (telefony, fotoaparáty, notebooky apod.). Její technologie je založena na flash paměti a poprvé byla zavedena v roce 2000. [15]

Základní způsoby přenosu dat na SD karty:

1bit sd

4bit sd

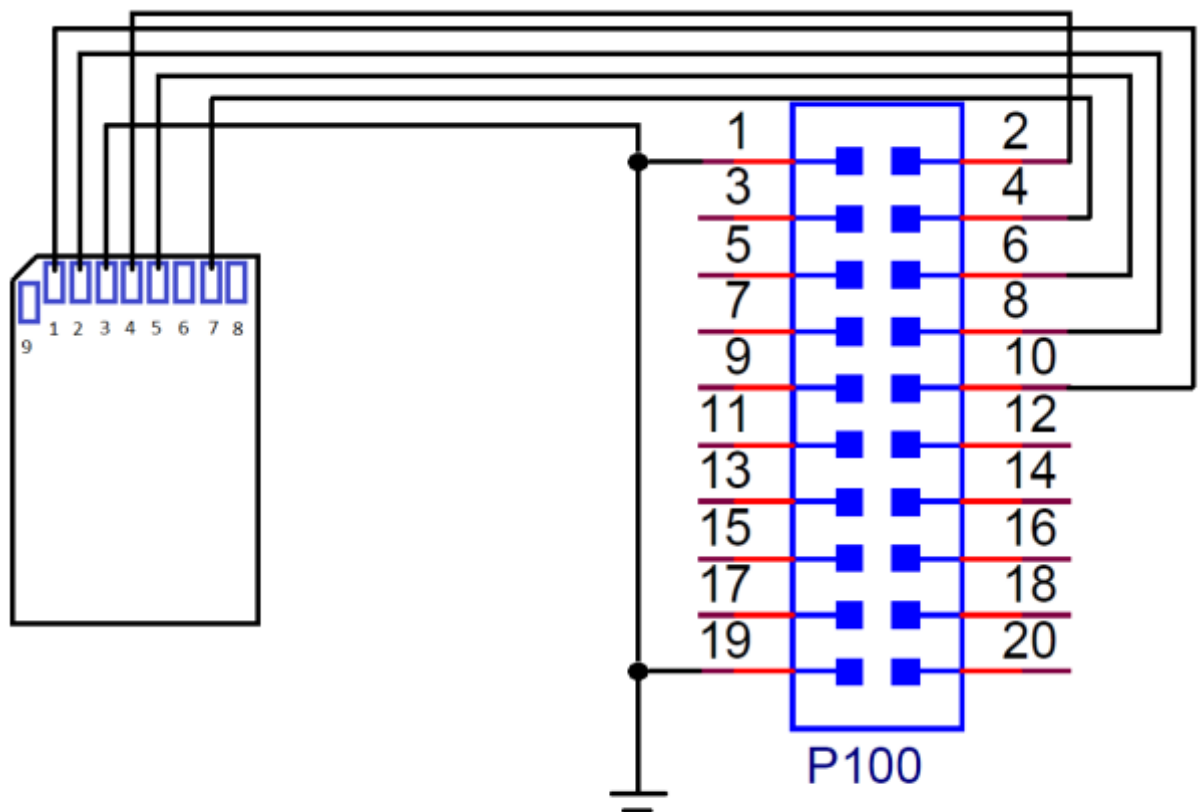
SPI

Všechny karty by měly podporovat tyto tři způsoby komunikace, kromě micro SD karet, pro které SPI není vyžadováno. V realizovaném dataloggeru je použita komunikace s SD kartou pomocí SPI rozhraní. Tato forma komunikace je často používaná právě v embedded systémech. [15]

Pro čtení a zápis na SD kartu byl použit LC studio breakout board. Důvody jsou stejné jako u vývojové desky. Cena socketu na kartu se pohybuje na podobné úrovni jako breakout board.



Obr. 2.7 LC studio breakout board na SD kartu [vlastní zpracování]



Obr. 2.8 zapojení SD karty na expansion header [vlastní zpracování]

Když přístroj pracuje z bateriového napájení, musí se breakout board napájet přímo z baterie nebo z konektoru VMCU. Pin 3,3 V na desce je napojen na regulátor napětí, které dává USB. [9]

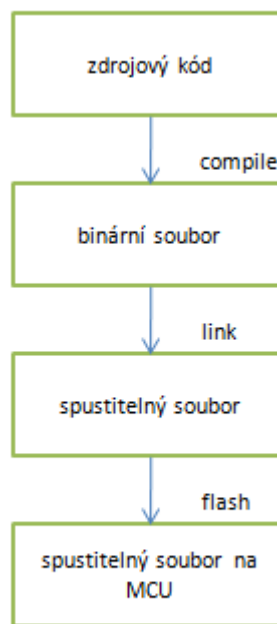
pin na desce	PD3	PD1	GND	TP704	PD2	GND	PD0		
pin na SD	1	2	3	4	5	6	7	8	9
význam	Chip select	data input	GND	Ucc	CLK	GND	Data output	reserv	reserv

Tab. 2.3 sobě odpovídající piny [vlastní tvorba]

SD karty používají FAT což je rozšířený systém adresování dat. Pro danou aplikaci byla využita verze FAT 32. Díky tomu lze paměť jednoduše přečíst na většině počítačů běžně dostupných téměř ve všech domácnostech. [15]

3 Návrh ve vývojovém prostředí

K návrhu projektu na mikrokontroleru je za potřebí několika nástrojů, ty jsou sdruženy do IDE (Integrated Development Enviroment). Nehledě na prostředí, které se používá pro vývoj, návrh aplikace prochází vždy stejnými kroky, které jsou zobrazeny na obrázku (3.1).



Obr. 3.1 kroky IED v návrhu projektu [vlastní zpracování]

Pro návrh dataloggeru bylo zvoleno Simplicity studio kvůli dostupnosti informací k EFM microcontrolerů. Simplicity studio není omezeno velikostí kompilovaného kódu. IAR je ekvivalentní prostředí jako simplicity studio. Disponuje také příkladnými projekty, kde jsou demonstrovány všechny periferie na desce. Na rozdíl od simplicity studia má však omezení velikosti kódu, je možné si zaregistrovat časově omezenou licenci, která nemá omezení velikosti kódu. Vývoj projektu však bývá časově náročný a do dvou týdnů omezené licence se nevejde.

Funkce programu začíná inicializací čipu. Inicializace čipu musí být hned na začátku main funkce.

Hodinový signál je nastaven na nejvyšší možnou frekvenci 24 MHz, aby se instrukce vykonaly v co nejkratší době a systém mohl přejít do EM2. Hodinový Signál je pro jednotlivé periferie rozdělován. Datalogger potřebuje hodinový signál pro GPIO a pro I2C.

V programu jsou nastaveny interrupty pro funkci delay, ta je použita v komunikaci s SD kartou.

Program pokračuje přečtením konfiguračního souboru, který umožní uživateli program přednastavit frekvenci RTC a tím rozestupy mezi jednotlivými čteními. Soubor musí být předem nahrán uživatelem na SD kartu a přesně pojmenován „config.txt“ pomocí getConfig() si mikrokontroler otevře soubor a přečte si údaje, které potřebuje k nastavení.

Nastaví se RTC counter a jeho hranice přetečení, RTC counter se inicializuje do výchozí inicializace tak, že counter počítá i v debug modu. Po dosažení porovnávací hladiny se vynuluje a počítá opět od nuly. Při frekvenci 32768 Hz se délka intervalu se doba mezi interrupty vypočítá následujícím způsobem.

$$\frac{x \times y \times m}{1000} - 1$$

X

LF counter frekvence

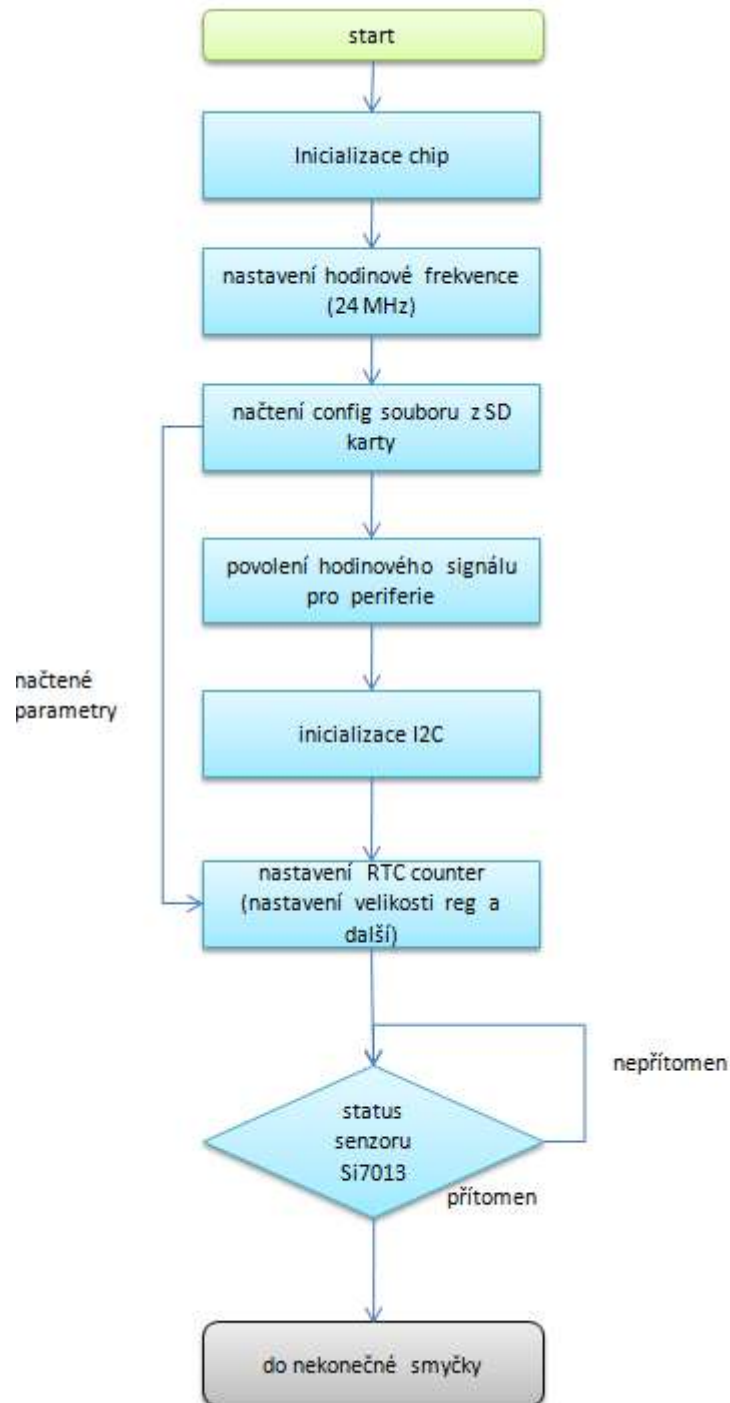
Y

požadovaný interval (ms)

m

„multiplier“ z SD karty

Program zinicizuje I2C protokol pro komunikaci s čidlem teploty a vlhkosti. Přítomnost čidla je ověřována ve smyčce. Dobu strávenou ve smyčce určuje parametr status. Jeho výchozí hodnota je true, v případě přítomnosti čidla se jeho hodnota změní na false a program pokračuje v dalším chodu.



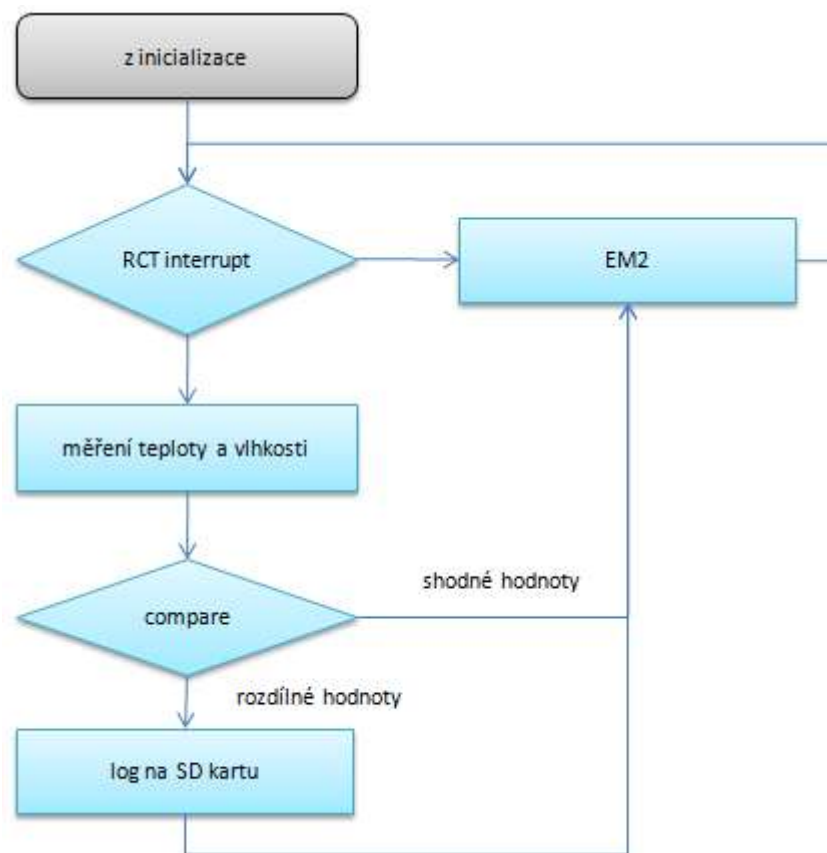
Obr. 3.2 vývojový diagram průběhu programu část 1. [vlastní zpracování]

V následujícím kroku program přejde do nekonečné smyčky, v které setrvá až do restartu tlačítkem, nebo restartu odebráním napájecího napětí. V prvním průchodu programu je status veličin pro měření a porovnávání naměřených dat nastaven do pravdivé hodnoty. Teplota a vlhkost jsou změřeny funkcí `MeasureTemp()`, výstupem této funkce jsou data teploty a vlhkosti. Teplotní data se porovnávají s předchozím měřením, parametr „předchozí teplota“ je nastavena na hodnotu 0°C v prvním průchodu programem, poté se do něj uloží současná

teplota. Jsou-li data shodná, zápis na kartu se neprovádí, aby se ušetřila energie, odebere se parametr „print“, program odskočí a přejde do EM2. Když jsou naměřená data rozdílná, zapíše se na kartu. Program vytvoří soubor s názvem „templog.txt“, napíše se úvodní řádek, který popisuje rozložení dat. Pokud již soubor existuje, program nalezne konec souboru a napíše řádek naměřených dat ve formátu

Dny:Hodiny:Minuty,teplota,vlhkost.

Po zápisu dat se nastaví parametry logToSD Compare a Print do hodnoty false. Program přejde do EM2 a čeká se na interrupt, který musí přijít od RTC. Po vyvolání interruptu, se výše zmíněné parametry nastaví do hodnoty true a celý běh programu se opakuje od začátku nekonečné smyčky.



Obr. 3.3 vývojový diagram průběhu programu část 2. [vlastní zpracování]

4 Měření spotřeby

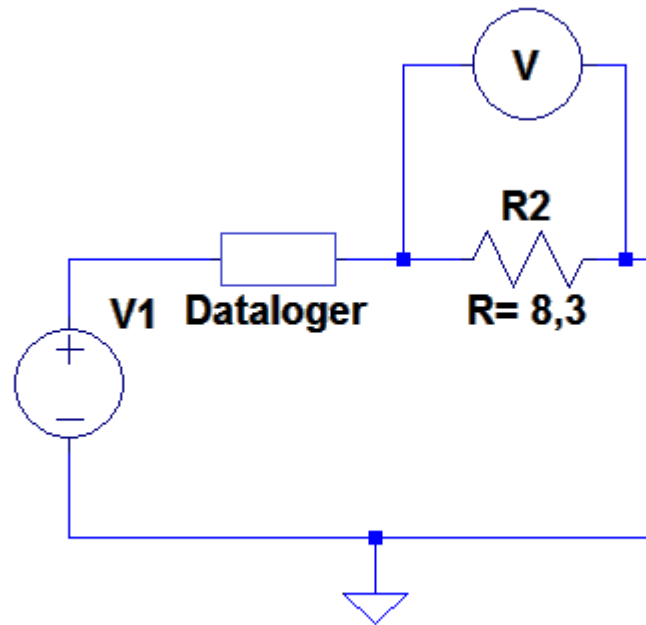
U bateriově napájených zařízení hraje spotřeba klíčovou roli. Proto je potřeba používat energy mody, které jsou dostupné od výrobce mikrokontroleru. Povolit hodinový signál jen pro části, které používáme. Hodinová frekvence je nastavena tak, aby se instrukce vykonala co nejrychleji a CPU pak počká v jiném energy módu. Snížení napájecího napětí na hodnotu, kterou dovoluje mikrokontroler také pozitivně ovlivní spotřebu. [14]

Přechody mezi jednotlivými energy mody má na starost EMU (Energy Management Unit). Mody mají index od 0 do 4, v nule má mikrokontroler k dispozici veškerou dostupnou funkcionalitu a se zvyšujícím se indexem se postupně odepínají některé systémy. Mode 0 je také výchozí pozicí po restartu, ostatní mody musí být zinicilizovány softwarem. [13]

Přechod do nižšího modu je pomocí software možný, přechod z vyšších modů do nižšího musí být vyvolán interruptem nebo událostí na pinu mikrokontroleru. Z modu 4 je přechod do 0 možný pouze po resetu, nebo pomocí události na pinu. [10]

4.1 Měření a schéma zapojení

Měření spotřeby bylo realizováno podle schématu na obrázku č3. Do série s vlastním zařízením byl zařazen odpor o velikosti 8,3 Ω . Pomocí osciloskopu byl snímán průběh napětí na tomto odporu.

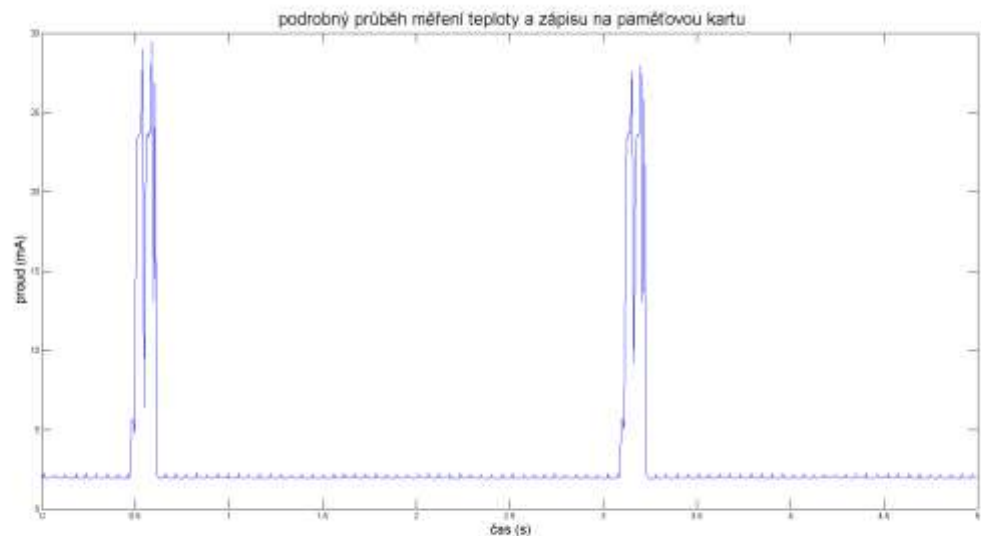


Obr. 4.1 schéma zapojení měření spotřeby. [vlastní zpracování]

4.2 Cyklus měření a zápisu na SD kartu

Zařízení loguje v několika vteřinových intervalech naměřenou teplotu a vlhkost. Po měření přejde do Energy modu 2, ve kterém je vyřazen z funkčnosti hlavní generátor hodin CPU. Nízko frekvenční generátor hodin je v provozu, pracuje na frekvenci 32,7kHz. Po uplynutí časového intervalu je generováno přerušení, po kterém je probuzena CPU a hlavní generátor hodinového signálu na frekvenci 24 MHz. Změří se teplota a data se zapíše na paměťovou kartu. Poté se celý cyklus opakuje. Podrobný popis funkce programu a vývojový diagram je popsán v další kapitole.

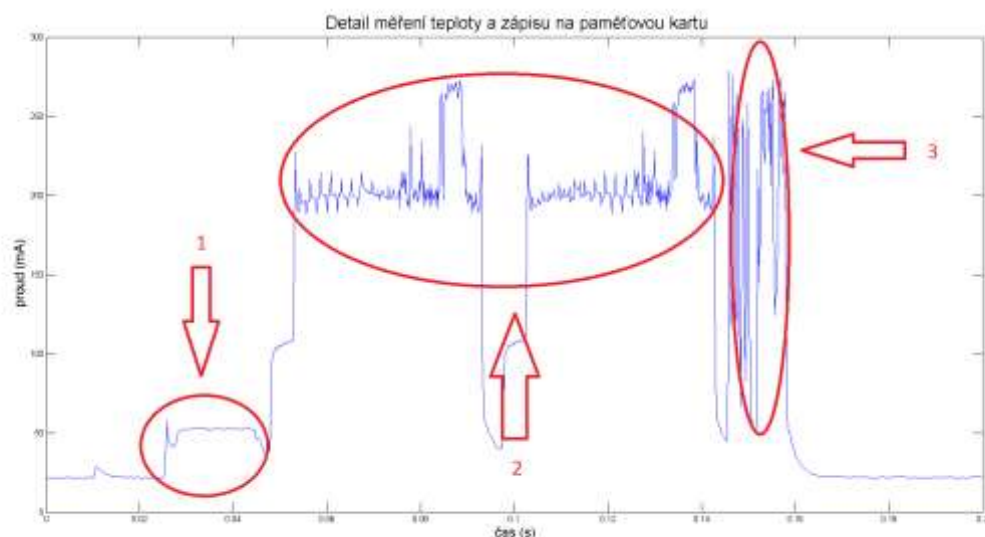
Obrázek (4.2) ukazuje průběh proudu v během měřicího cyklu a neaktivní periody. Na obrázku (4.3) je detailní pohled na měřicí a zápisový cyklus.



Obr. 4.2 průběh proudů během funkčních cyklů [vlastní zpracování]

Detail průběhu:

V grafu je zobrazen detailní průběh aktivní části měřicího a zápisového cyklu. V části 1 je vidět průběh při měření teploty, v druhé části je zápis dat na SD kartu, ten má dvakrát po sobě stejný průběh. Data se zapisují a zdvojují kvůli systému alokace dat FAT32, který karta používá. V 3 části je vidět konec komunikace, uzavření složky a přechod systému do energy modu 2.



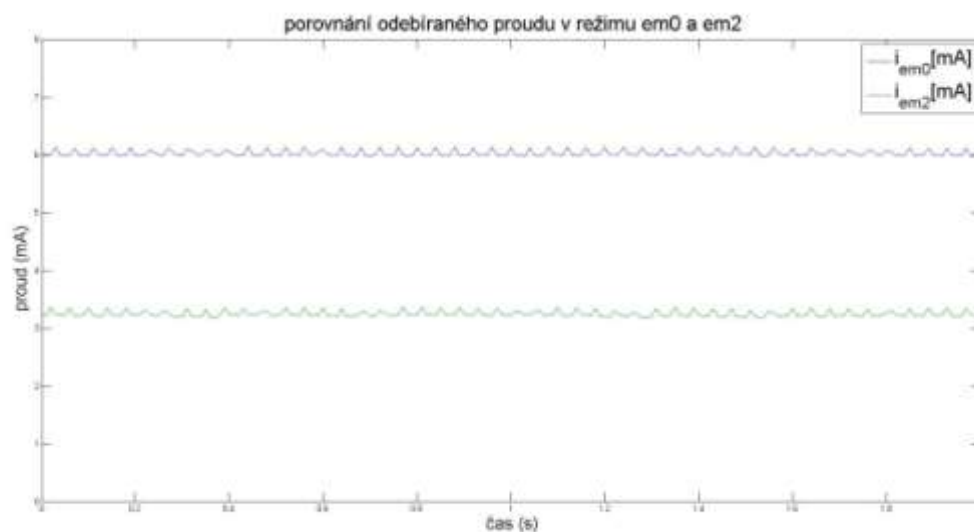
Obr. 4.3 detail na měření teploty a zápis na SD kartu [vlastní zpracování]

Spotřeba zařízení byla změřena a vypočítána v režimu zápisu každou vteřinu. Její hodnota je 4,93mAh, a datalogger pořídí 1800 měření a zápisů na paměťovou kartu. To je zbytečně velké, program tak byl upraven pouze pro účely měření, aby byl zajištěn dostatečný počet

měřících a zápisových cyklů v kratším měřicím okně. Při fermentaci se měří teplota relativně velkého množství piva, takovéto množství nezmění svou teplotu v krátkém časovém intervalu. Bylo by za potřebí působení velkého množství energie po delší dobu, a tak je žádoucí měřicí a zápisový interval prodloužit. Analýza takto získaného velkého množství dat je komplikovanější.

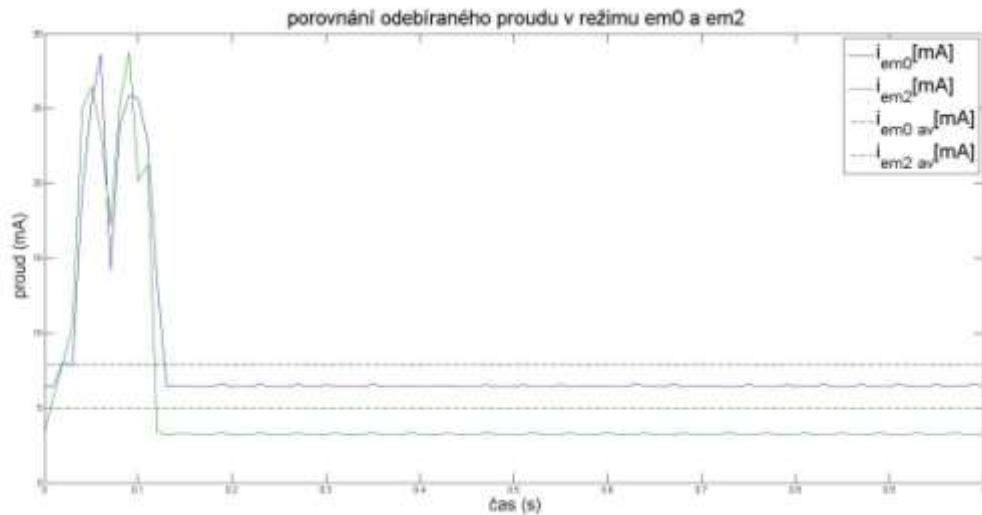
4.2.1 Porovnání spotřeby MCU v různých energy modech

Spotřeba MCU byla nejdříve změřena na vzorovém projektu „energy modes“, který je součástí dokumentace a zákaznické podpory z knihovny Simplicity studia. Cílem bylo zjistit, jakou spotřebu má MCU bez dalších periférií. Modrý průběh ukazuje proud, který vývojová deska odebírá při EM0. Frekvence jádra je nastavena na 24 MHz, program na mikrokontroleru počítá prvočísla. Zelený průběh představuje proud odebíraný vývojovou deskou v EM2. MCU je v „deep sleep“ modu generátor hlavního hodinového signálu je vypnut, v chodu je pouze generátor nízkofrekvenčního hodinového generátoru, který je nastaven na frekvenci 1Hz. Odebíraný proud je v EM2 téměř poloviční. Na vývojové desce je velké množství nevyužívaných periférií. Proud protéká přes pullup rezistory a zařízení má poměrně vysokou spotřebu i v klidovém modu.



Obr. 4.4 průběh proudu při použití vzorového projektu Emodes EM0 a EM2 [vlastní zpracování]

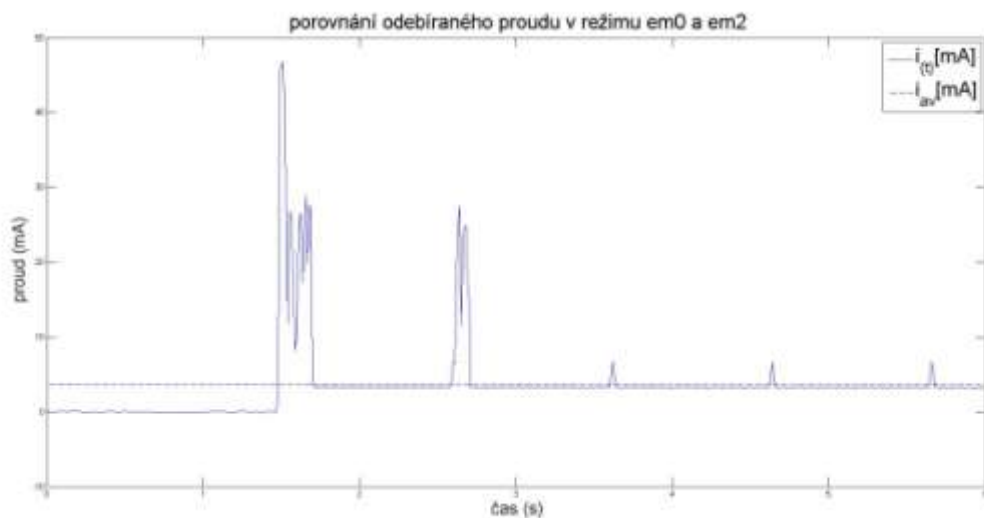
V dalším měření byl porovnán chod programu Dataloggeru. Opět se porovnává spotřeba v EM0 a EM2. Průměrná hodnota proudu je zobrazená čárkovaně. Pokud by datalogger zůstal v EM0, protékal by jím průměrný proud 7,4 mA. Přepínáním EM se protékající proud sníží o 2,5 mA. Zápis na kartu je energeticky nejnáročnější, v době zápisu může zařízením protékat proud až 29 mA.



Obr. 4.5 Průběh proudu dataloggeru se setrváváním v EM0 a s přecházením do EM0 [vlastní zpracování]

4.2.2 Průběh proudu při zapnutí zařízení

Po zapnutí má zařízení největší odběr, nabíjí se parazitní kapacity, zařízení nedělá jen zápis dat, ale vytváří se také složka pro teplotní data. Na průběhu je také vidět funkce porovnávání naměřených dat. Pokud se naměřená teplota dostatečně neliší, zařízení neprovádí zápis dat, proudová špička je pouze 6,75 mA.



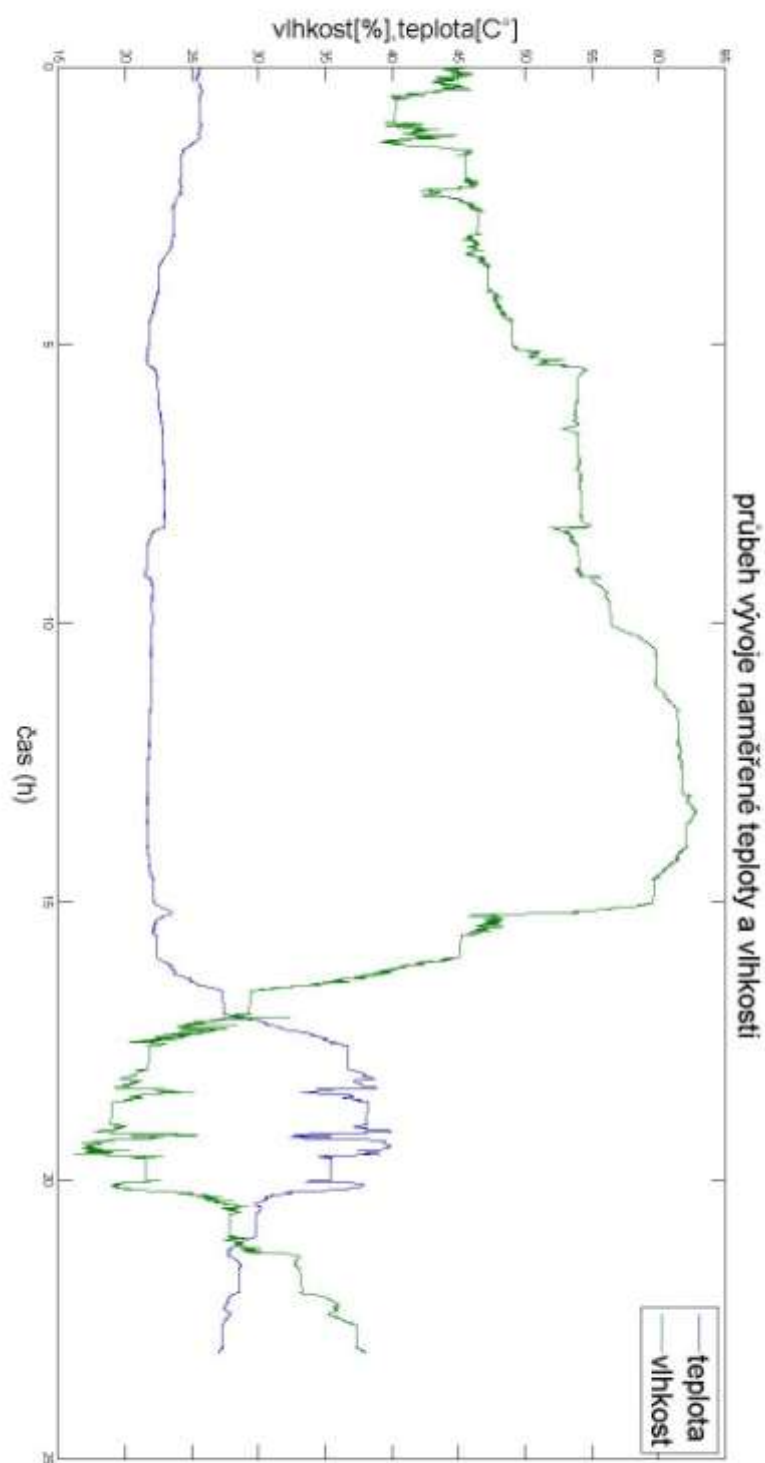
Obr. 4.6 průběh při zapnutí zařízení [vlastní zpracování]

5 Měření vlhkosti a teploty

Pomocí dataloggeru bylo provedeno měření vlhkosti a teploty v pokoji. Zařízení bylo umístěno za oknem. Zařízení zaznamenávalo 24h data. Přes noc spal v místnosti 1 člověk. Měření začalo v 17:00 a bylo druhý den ve stejnou dobu ukončeno.



Obr. 5.1 celé zařízení v chodu při měření teploty a vlhkosti. [vlastní zpracování]
Z měření je vidět že, proměna teploty je v podmínkách obydleného pokoje, je poměrně velká. Největší výkyv teploty byl způsoben tím, že na zařízení umístěné za oknem ráno svítlo slunce. Fermentační nádobu nebudeme umísťovat za okno, ale i v noci je výkyv teplot o několik stupňů.



Obr. 5.2 naměřené hodnoty teploty a vlhkosti[vlastní zpracování]

Závěr

Diplomová práce popisuje návrh teplotního dataloggeru z dílčích částí, jehož úkolem je zefektivnit proces výroby domácího piva, ale i jiných procesů, kde je za potřebí, dodržovat přesné technologické procesy.

Byl popsán návrh programu v Simplicity Studiu, metody pro zefektivnění a konfiguraci měřicího zařízení a byla tak potvrzena funkčnost tohoto zařízení, viz demonstrační měření. Otevřela se tak možnost realizovat zařízení přesně podle zapojení na vývojové desce. Pomocí měření byla ukázána energetická náročnost zařízení a byla demonstrována energetická úspora vzniklá použitím příslušného energy modu a zápisu pouze rozdílných měření. Rozdíl mezi jednotlivými proudovými špičkami byl 22mA. Z naměřených hodnot byl, pomocí programu MATLAB vypočítán proud, který zařízení odebírá při měření každou vteřinu. Zařízení odebírá proud o hodnotě 4,93mAh. Při použití CRV3 baterie o kapacitě 3000 mAh by zařízení vydrželo pracovat 608,5 hodin.

Bylo demonstrováno měření pomocí dataloggeru a ukázána proměnlivost teploty v domácích podmínkách.

Z časových důvodů se bohužel nepodařilo zařízení umístit do krabičky a chybí funkční display kvůli omezení paměti mikrokontroleru, což limituje zpětnou vazbu s uživatelem. Ta je v současné době pouze skrz paměťovou kartu.

Dalším krokem k zlepšení funkčnosti zařízení by bylo doplnění chybějícího displeje. Buď jiným displejem než je momentálně na desce nebo upravením knihovny displeje aby se vešla do paměti mikrokontroleru.

Seznam zdrojů

- [1] Cortex-M0+ Processor. *ARM Ltd.* [online]. ©1995-2016 [cit. 2016-05-16]. Dostupné z: <http://www.arm.com/products/processors/cortex-m/cortex-m0plus.php>
- [2] *Cypress Semiconductor Corporation* [online]. [cit. 2016-05-16]. Dostupné z: <http://www.cypress.com>
- [3] Evan Schulz. Choosing low-power microcontrollers by the numbers. *Embedded* [online]. Silicon Labs, 2013 [cit. 2016-05-16]. Dostupné z: <http://www.embedded.com/design/mcus-processors-and-socs/4413368/Choosing-low-power-microcontrollers-by-the-numbers>
- [4] Ingar Fredriksen. Choosing a MCU for your next design; 8 bit or 32 bit? *Atmel Corporation* [online]. San Jose, © 2014, , 6 [cit. 2016-05-16]. Dostupné z: http://www.atmel.com/Images/45107A-Choosing-a-MCU-Fredriksen_Article_103114.pdf
- [5] Jak na výrobu piva I. *VASEPIVO.CZ* [online]. 2016 [cit. 2016-05-15]. Dostupné z: <http://www.vasepivo.cz/sekce/jak-na-vyrobu-piva-I-S60.html>
- [6] Návod na vaření domácího piva. *JAK PRACOVAT DOMA: PRÁCE Z DOMOVA SNADNO* [online]. [cit. 2016-05-15]. Dostupné z: <http://www.jakpracovatdoma.cz/vareni-piva-doma/>
- [7] *Nuvoton Technology Corporation* [online]. ©2008-2016 [cit. 2016-05-16]. Dostupné z: <http://www.nuvoton.com/>
- [8] *NXP Semiconductors* [online]. ©2006-2016 [cit. 2016-05-16]. Dostupné z: <http://www.nxp.com/>
- [9] SILICON LABS. *EFM32 Zero Gecko STK* [online]. In: . 2015, s. 10 [cit. 2016-05-16]. Dostupné z: http://www.silabs.com/Support%20Documents/TechnicalDocs/BRD2010A_B01.pdf
- [10] SILICON LABS. *EFM32ZG Reference Manual: Zero Gecko Series* [online]. In: . 2015, s. 394 [cit. 2016-05-16]. Dostupné z: <http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32ZG-RM.pdf>
- [11] SILICON LABS. *EFM32ZG222 DATASHEET: F32/F16/F8/F4* [online]. In: . 2015, s. 70 [cit. 2016-05-16]. Dostupné z: <http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32ZG222.pdf>
- [12] SILICON LABS. *Energy Micro University: UM003 - Setting Up Development Environments* [online]. In: . 2012, s. 24 [cit. 2016-05-16]. Dostupné z: <http://www.silabs.com/Marcom%20Documents/Resources/um003-ides.pdf>
- [13] SILICON LABS. *Energy Micro University: UM006 - Energy Modes* [online]. In: . 2012, s. 21 [cit. 2016-05-16]. Dostupné z: <http://www.silabs.com/Marcom%20Documents/Resources/um006-energy-modes.pdf>
- [14] SILICON LABS. *Energy Optimization: AN0027 - Application Note* [online]. In: . 2013, s. 20 [cit. 2016-05-16]. Dostupné z: <http://www.silabs.com/Support%20Documents/TechnicalDocs/AN0027.pdf>

- [15] SILICON LABS. *FAT on SD Card: AN0030 - Application Note* [online]. In: . 2013, s. 18 [cit. 2016-05-16]. Dostupné z:
<http://www.silabs.com/Support%20Documents/TechnicalDocs/AN0030.pdf>
- [16] SILICON LABS. *Real Time Counters: AN0005 - Application Note* [online]. In: . 2013, s. 10 [cit. 2016-05-16]. Dostupné z:
<http://www.silabs.com/Support%20Documents/TechnicalDocs/AN0005.pdf>
- [17] SILICON LABS. *Si7013-A20: I²C HUMIDITY AND TWO -Z ONE TEMPERATURE SENSOR* [online]. In: . © 2016, s. 45 [cit. 2016-05-16]. Dostupné z:
<https://www.silabs.com/Support%20Documents/TechnicalDocs/Si7013-A20.pdf>
- [18] SILICON LABS. *USER MANUAL: Starter Kit EFM32ZG-STK3200* [online]. In: . 2013, s. 28 [cit. 2016-05-16]. Dostupné z:
<http://www.silabs.com/Support%20Documents/TechnicalDocs/efm32zg-stk3200-ug.pdf>
- [19] *STMicroelectronics* [online]. 2016 [cit. 2016-05-16]. Dostupné z:
<http://www.st.com/web/en/home.html>
- [20] Zero Gecko follow up: talking to a microSD card via SPI. *Element14 community* [online]. ©2009-2016 [cit. 2016-05-16]. Dostupné z:
<https://www.element14.com/community/groups/roadtest/blog/2014/11/17/zero-gecko-follow-up-talking-to-a-microsd-card-via-spi>
- [21] Bus Specification. *I2C info* [online]. 2016 [cit. 2016-05-16]. Dostupné z:
<http://i2c.info/i2c-bus-specification>

Seznam obrázků

OBR. 1.1 VÝVOJOVÝ DIAGRAM DOMÁCÍ VÝROBY PIVA [VLASTNÍ ZPRACOVÁNÍ]	11
OBR. 2.1 PŘEHLED VELIKOSTI INSTRUKCÍ [3].....	14
OBR. 2.2 BLOKOVÉ SCHÉMA MIKROKONTROLERU EFM32ZG [11].....	15
OBR. 2.3 VÝVOJOVÁ DESKA A POUŽITÉ PERIFERIE [9].....	16
OBR. 2.4 BLOKOVÉ SCHÉMA TEPLONÍHO ČIDLA SI7013 [VLASTNÍ ZPRACOVÁNÍ][17]	17
OBR. 2.5 BLOKOVÉ SCHÉMA ZAPOJENÍ I2C SBĚRNICE [VLASTNÍ ZPRACOVÁNÍ].....	18
OBR. 2.6 ČASOVÉ PRŮBĚHY KOMUNIKACE PO I2C SBĚRNICI [21]	18
OBR. 2.7 LC STUDIO BREAKOUT BOARD NA SD KARTU [VLASTNÍ ZPRACOVÁNÍ].....	20
OBR. 2.8 ZAPOJENÍ SD KARTY NA EXPANSION HEADER [VLASTNÍ ZPRACOVÁNÍ].....	21
OBR. 3.1 KROKY IED V NÁVRHU PROJEKTU [VLASTNÍ ZPRACOVÁNÍ]	22
OBR. 3.2 VÝVOJOVÝ DIAGRAM PRŮBĚHU PROGRAMU ČÁST 1. [VLASTNÍ ZPRACOVÁNÍ]	24
OBR. 3.3 VÝVOJOVÝ DIAGRAM PRŮBĚHU PROGRAMU ČÁST 2. [VLASTNÍ ZPRACOVÁNÍ]	25
OBR. 4.1 SCHÉMA ZAPOJENÍ MĚŘENÍ SPOTŘEBY. [VLASTNÍ ZPRACOVÁNÍ]	27
OBR. 4.2 PRŮBĚH PROUDŮ BĚHEM FUNKČNÍCH CYKLŮ [VLASTNÍ ZPRACOVÁNÍ]	28
OBR. 4.3DETAIL NA MĚŘENÍ TEPLoty A ZÁPIS NA SD KARTU [VLASTNÍ ZPRACOVÁNÍ].....	28
OBR. 4.4 PRŮBĚH PROUDU PŘI POUŽITÍ VZOROVÉHO PROJEKTU EMODES EM0 A EM2 [VLASTNÍ ZPRACOVÁNÍ]	29
OBR. 4.5 PRŮBĚH PROUDU DATALOGGERU SE SETRVÁVÁNÍM V EM0 A S PŘECHÁZENÍM DO EM0 [VLASTNÍ ZPRACOVÁNÍ]	30
OBR. 4.6 PRŮBĚH PŘI ZAPNUTÍ ZAŘÍZENÍ [VLASTNÍ ZPRACOVÁNÍ]	30
OBR. 5.1 CÉLÉ ZAŘÍZENÍ V CHODU PŘI MĚŘENÍ TEPLoty A VLHKOSTI. [VLASTNÍ ZPRACOVÁNÍ]	31
OBR. 5.2 NAMĚŘENÉ HODNOTY TEPLoty A VLHKOSTI[VLASTNÍ ZPRACOVÁNÍ]	32

Seznam tabulek

TAB. 2.1PŘEHLED MIKROKONTROLERŮ ARM CORETEX0 [VLASTNÍ ZPRACOVÁNÍ] [2] [7] [8] [19]	13
TAB. 2.2 PŘEHLED MIKROKONTROLERŮ ARM CORETEX0 DRUHÁ ČÁST [VLASTNÍ ZPRACOVÁNÍ] [2] [7] [8] [19] ...	14
TAB. 2.3 SOBĚ ODPOVÍDAJÍCÍ PINY [VLASTNÍ TVORBA].....	21

přílohy

Příloha 1

Projekt dataloggeru vypracovaný v Simplicity studiu na přiloženém CD

Příloha 2

Záznam měření z dataloggeru na přiloženém CD

Příloha 3

Kompletní schéma zapojení na přiloženém CD

Příloha 4

Datasheet a reference manual mikrokontroleru na přiloženém CD