

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**  
**FAKULTA ELEKTROTECHNICKÁ**

**Katedra elektroenergetiky a ekologie**

**DIPLOMOVÁ PRÁCE**

**Teplovní datalogger pro domovarní výrobu piva**

*Originál (kopie) zadání BP/DP*

## **Abstrakt**

Práce se zabývá problematikou návrhu zařízení s nízkou spotřebou energie, pro měření teploty a vlhkosti. Zařízení má sloužit k zlepšení efektivity technologického postupu fermentace. Spotřeba zařízení je změřena v sérii krátkých měření. 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. The device should increase efficiency of technological process of fermentation. Consumption of the device is measured in series of short measurements. Functionality of the device is demonstrated in shot measurement of humidity and temperature.

## **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 14.5.2017

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

<b>OBSAH</b> .....	<b>7</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>8</b>
<b>ÚVOD</b> .....	<b>9</b>
<b>1 DOMÁCÍ VÝROBA PIVA</b> .....	<b>10</b>
1.1 POSTUP PŘI DOMÁCÍ VÝROBĚ PIVA.....	10
1.2 NÁROKY NA TEPLTNÍ DATALOGGER .....	12
<b>2 NÁVRH ZAŘÍZENÍ</b> .....	<b>14</b>
2.1 MIKROKONTROLÉR EFM PEARL GECKO .....	14
2.2 VÝVOJOVÁ DESKA .....	14
2.3 MĚŘENÍ TEPLoty A VLHKOSTI.....	15
2.4 SOUČASNÝ CHOD VÍCE SENZORŮ .....	18
2.5 UKLÁDÁNÍ NAMĚŘENÝCH DAT .....	19
<b>3 NÁVRH PROGRAMU</b> .....	<b>26</b>
3.1 HLAVNÍ FUNKCE PROGRAMU .....	26
3.2 ZPRACOVÁNÍ PŘERUŠENÍ .....	29
<b>4 MANAGEMENT NÍZKÉ SPOTŘEBY</b> .....	<b>31</b>
4.1 MĚŘENÍ SPOTŘEBY .....	31
4.2 CYKLUS MĚŘENÍ A ZÁPISU NA SD KARTU .....	32
4.3 DETAIL PRŮBĚHU:.....	33
4.4 POROVNÁNÍ SPOTŘEBY ZAŘÍZENÍ V RŮZNÝCH REŽIMECH .....	34
4.5 PRŮBĚH PROUDU PŘI ZAPNUTÍ ZAŘÍZENÍ .....	37
<b>5 MĚŘENÍ VLHKOSTI A TEPLoty</b> .....	<b>39</b>
<b>ZÁVĚR</b> .....	<b>41</b>
<b>SEZNAM ZDROJŮ</b> .....	<b>42</b>
<b>SEZNAM OBRÁZKŮ</b> .....	<b>44</b>
<b>SEZNAM TABULEK</b> .....	<b>45</b>
<b>PŘÍLOHY</b> .....	<b>46</b>

## 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>I2C</i>	Inter Integrated Circuit sběrnice
<i>MCU</i>	Micro Controlling unit
<i>MISO</i>	Master in slave out
<i>MOSI</i>	Master out slave in
<i>RISC</i>	Reduced Instruction Set
<i>RTC</i>	Real time counter
<i>SCLK</i>	Serial clock(hodinový signal pro SD kartu)
<i>SD</i>	Secure Digital paměťová karta
<i>SPI</i>	Single peripheral interface
<i>SS</i>	Slave slect



## 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 popis 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 úsporných režimů. 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 domácí výrobě piva

### 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]

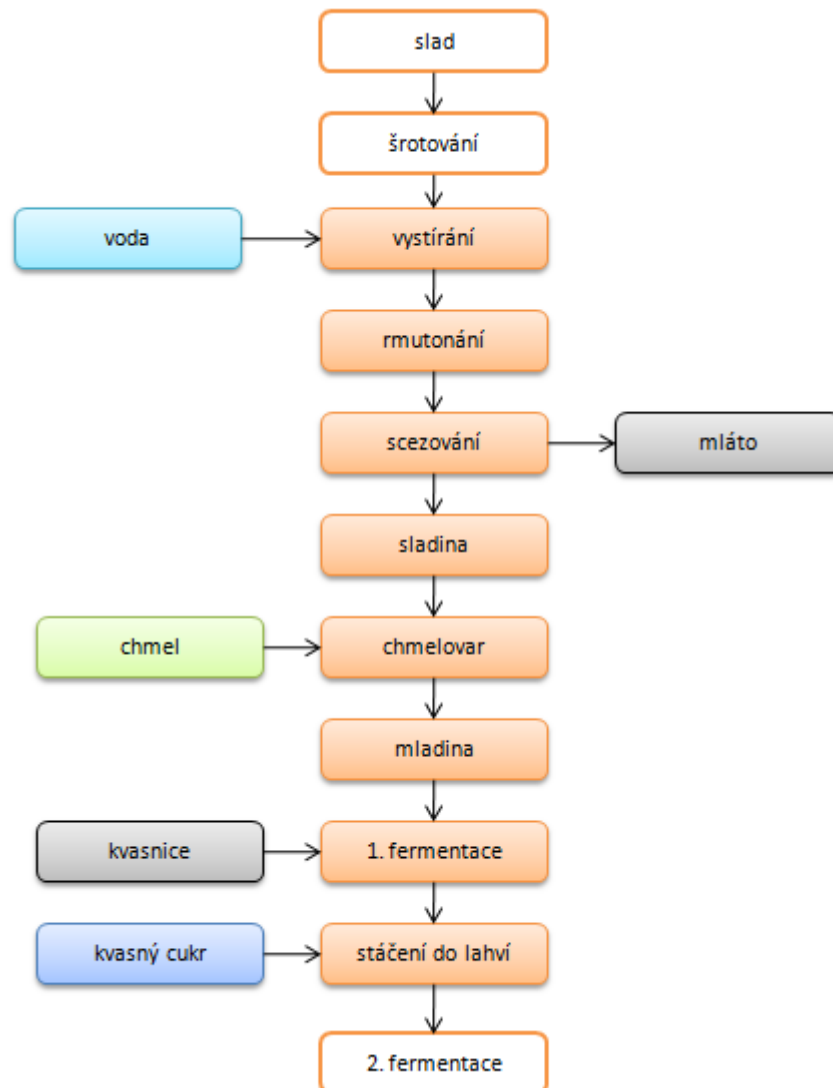
### Šrotování, vystírání, rmutování, chlazení

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 se 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 jeden 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 popsané 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 nízkou, 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 technologického postupu.

## **1.2 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 je slad sušen 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. Zařízení bude použito mimo dosah zdroje stabilní elektrické energie. Musí být schopné provozu s vlastním zdrojem elektrické energie.

**Konfigurace měření**

Domovní prostory se budou lišit (sklep, stinný pokoj), budou v nich panovat rozdílné klimatické podmínky. Teplota a vlhkost se budou měnit různým 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 coretex M4 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 coretex A, R, M. Řada M je vytvořena pro průmyslové ovládání a je nejúspornější. Coretex M je rozdělen na řady M4-0. [3]

### 2.1 Mikrokontrolér EFM Pearl Gecko

Jedná se o „low power“ mikrokontrolér, který je založen na 32-bitovém procesorovém jádru ARM Cortex-M4. Mikrokontroler EFM Pearl 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  $\mu$ 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 $\mu$ 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 Pearl Gecko od firmy Silicon Labs. Vývojová deska obsahuje mikroprocesor a prostředky potřebné pro využívání dané platformy. Kvůli velkému množství ovládacích a měřících 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 programovou 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 hardware, případně je možné využít její schéma jako

východisko pro specializovaný hardware, který bude splňovat specifické nároky aplikace (včetně mechanických parametrů – velikost, zapouzdření do krabičky atd.).

Na obrázku č. 2.1 jsou vidět jednotlivé části desky. Deska obsahuje dvě tlačítka, LCD display, konektor pro baterii, in/out debug konektor, teplotní a vlhkostní čidlo. Po stranách vývojové desky jsou vyvedeny jednotlivé signály mikrokontroleru. [9]



Obr. 2.1 Vývojová deska a použité periferie [9]

### 2.3 Měření teploty a vlhkosti

Nárokem na čidlo byla nízká spotřeba energie. V domácích podmínkách není potřeba překonávat velké vzdálenosti a tak budou čidla umístěna v blízkosti zařízení. Byl proto zvolen způsob komunikace I<sup>2</sup>C, navíc mikrokontroler má zabudovaný řadič I<sup>2</sup>C. Bylo by možné komunikaci realizovat programově, to by však zatížilo mikrokontroler a negativně by se projevilo na spotřebě. Je žádoucí, aby byla snímána vlhkost i teplota zároveň. Z těchto důvodů bylo vybráno čidlo Si7021.

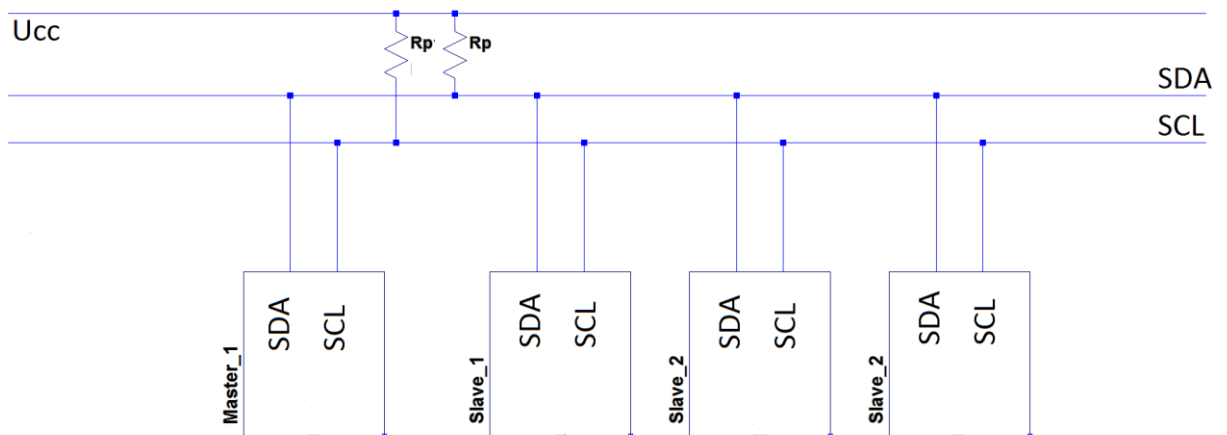
Jedná se o teplotní a vlhkostní čidlo od firmy Silicon labs. Rozsah měření teplot je (-40 až 85°C) to je dostačující pro zamýšlenou funkcionalitu. Vývojová deska s mikrokontrolerem je napájena akumulátorem s 3V. Rozsah napájecího napětí je od 1,9 do 3,6 V. Proces fermentace je nutné individuálně vyladit podle potřeb uživatele, není tak velký nárok na přesnost. Výrobce uvádí přesnost teplotního senzoru +/-0,3%. To je pro zamýšlenou funkcionalitu dostačující. Přesnost vlhkostního senzoru má hysterezní charakter. Film na povrchu senzoru absorbuje vlhkost a tak při přechodu z nižších vlhkostí udává senzor hodnoty s negativní odchylkou a při přechodu z vyšších naopak. Senzor také obsahuje odporové topné těleso, které je možné aktivovat zápisem do příslušného registru.

Těleso pomáhá vysoušet povrch, pokud byl senzor vystaven vysoké vlhkosti. Toho by se dalo využít v aplikacích s dynamickou změnou teplot. [17]

### 2.3.1 Přenos dat po I<sup>2</sup>C

Přenos dat probíhá po dvou vodičích SDA a SCL. Vodiče jsou připojeny na napájecí napětí přes dvojici tzv. zdvihacích (pullup) rezistorů. V klidovém stavu je na sběrnici napájecí napětí. Budiče sběrnice jsou typu otevřený kolektor, což umožňuje jednoduchou arbitráž bez nebezpečí destruktivních konfliktů. Sběrnice umožňuje multimaster konfigurace, ale většinou se používá rozložení: jedno zařízení typu master a více zařízení typu slave. Stav logická 0 může díky otevřeným kolektorům nastavit jakýkoliv obvod, nehrozí přitom elektricky destruktivní kolize, pouze porušení dat. To umožňuje zpomalit vysílání obvodům, které mají různou rychlost zpracování dat zpomalit přenos. Obvod například může uvést stav na hodinovém signálu do log 0, dokud nezpracuje data, nebo vyřídí přerušení. [21]

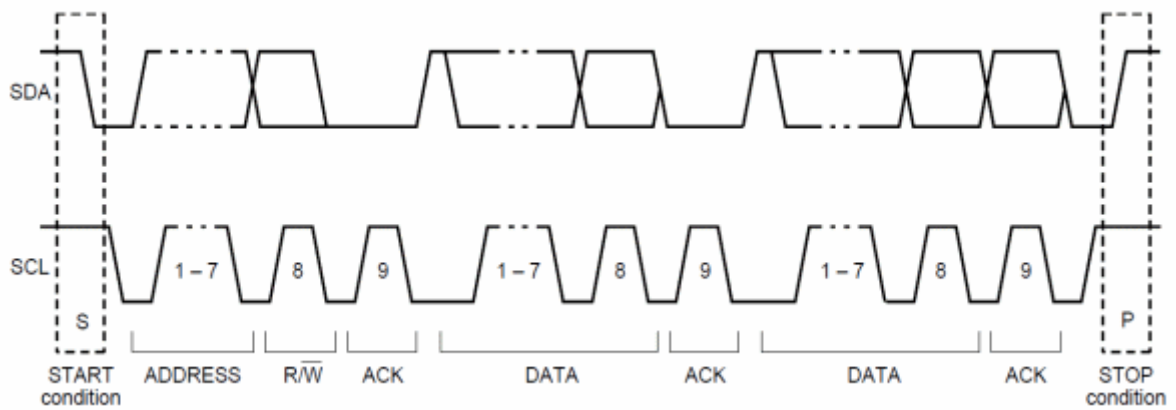
Na obrázku 2.2 je blokové schéma zapojení výše popsané sběrnice I<sup>2</sup>C



Obr. 2.2 Blokové schéma zapojení I<sup>2</sup>C sběrnice [vlastní zpracování]

Na začátku komunikace výše master start bit, je potřeba určit s kterým obvodem bude master komunikovat k tomu slouží adresa. Adresa mívá většinou 7 až 10 bitů. Pak nastává vlastní komunikace mezi dvěma zařízeními. Po skončení komunikace je odvysílán stop bit. Aby bylo možné kontrolovat, zda byla vyslaná data řádně přijata, vysílá se po každém datovém bitu potvrzení (ACKnowledge). Master uvede SDA do stavu 1 a čeká, pokud zařízení uvede SDA do stavu 0, znamená to, že data jsou v pořádku přijata.

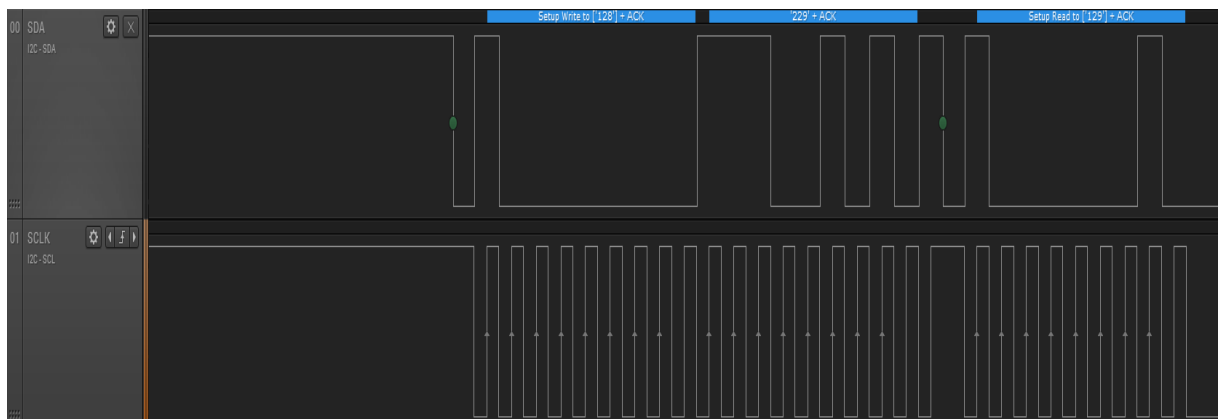




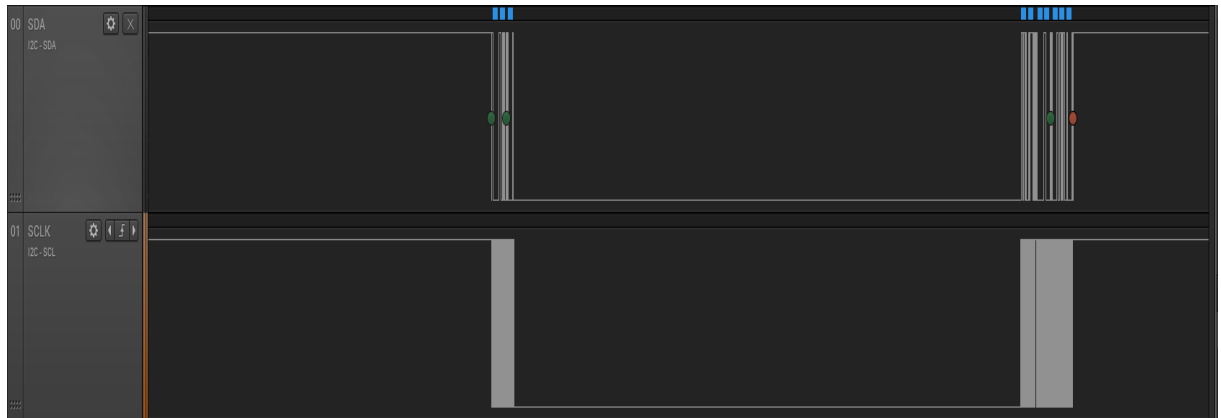
Obr. 2.3 Časové průběhy komunikace po I<sup>2</sup>C sběrnici [21]

Logickým analyzátozem je možné se popsany průběh komunikace prohlédnout. Po příkazech pro měření vlhkosti a teploty drží zařízení SDA ve stavu 0 po dobu 16ms poté jsou data poslána mikroprocesoru a přenos je zakončen stop bitem.

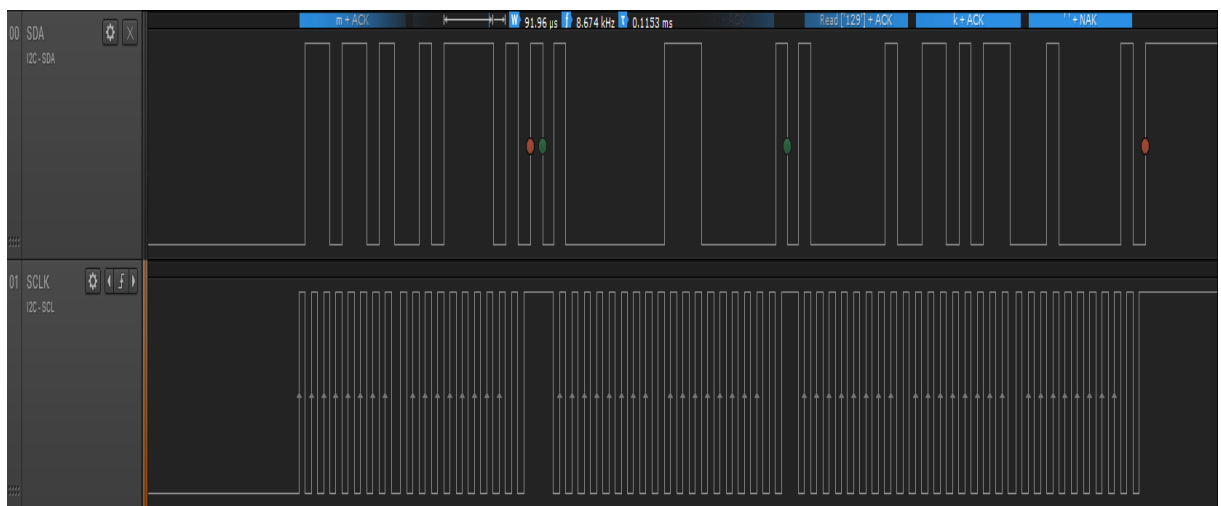
Pro prozkoumání komunikace byl použit logický analyzátor od firmy SALEAE.



Obr. 2.4 Začátek komunikace s teplotním a vlhkostním čidlem (příkazy pro měření teploty a vlhkosti)



Obr. 2.5 Přehled celé komunikace. Obvod drží hodinový signál v log 0 do doby, než vyřídí příkazy.



Obr. 2.6 Konec komunikace s datovým blokem obsahujícím teplotní a vlhkostní data

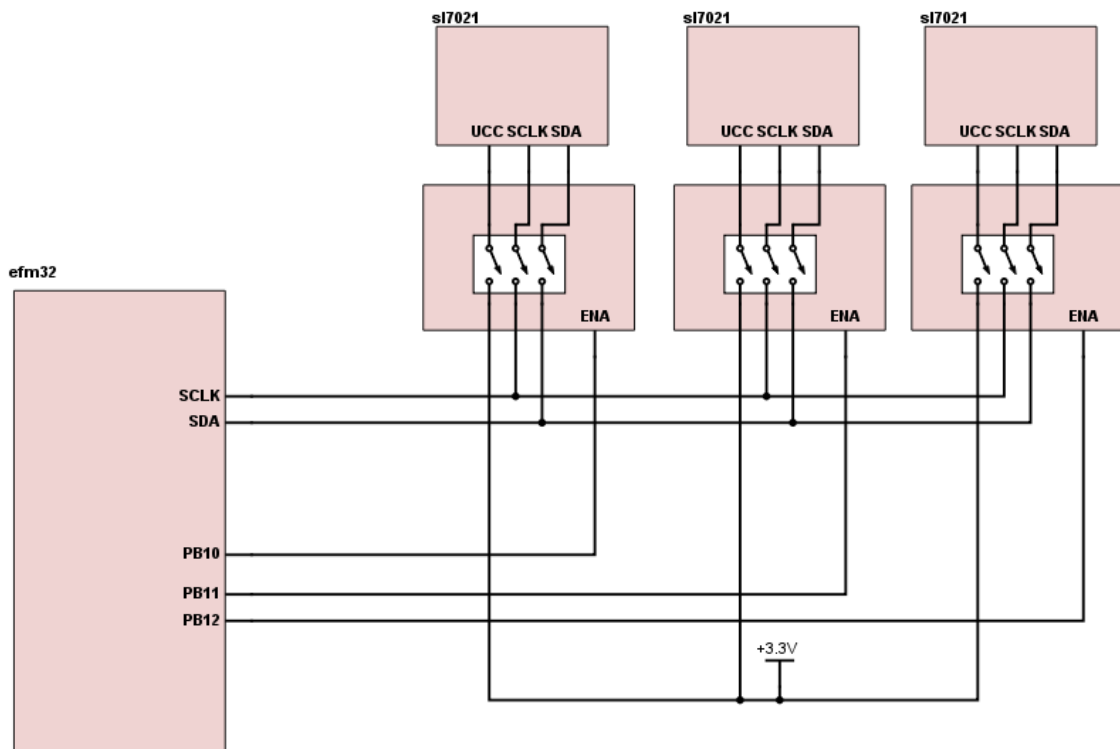
V programu je měření teploty a vlhkosti zajištěno voláním funkce z knihovny Si7013. Tato knihovna obsahuje funkce pro komunikaci se senzorem a je dostupná jako součást použitého vývojového prostředí. Příkazy pro měření teploty vlhkosti, status senzoru. Po vyslání příkazu senzoru obdrží procesor 16ti bitovou hodnotu kterou je potřeba konvertovat. Tato konverze je v knihovně obsažena na tisíce stupně.

## 2.4 Současný chod více senzorů

K monitorování procesu fermentace piva je zapotřebí zaznamenávat teplotu z několika zdrojů a získat tak komplexnější obraz o celém procesu. Je tedy potřeba více čidel napojených na sběrnici. Toto je na I<sup>2</sup>C možné, ale vybraný senzor nemá k dispozici adresovací pin.

Součástí mikrokontroleru jsou 2 řadiče I<sup>2</sup>C je tak možné adresovat ho na další pozice a získat tak 2 čidla. Více senzorů není možné tímto způsobem přidat.

Další řešení je implementováno přímo na vývojové desce. Všechny pracovní vodiče senzoru jsou připojeny přes vypínač TS3A4751 viz obrázek č 2.5. Pokud není přiveden spouštějící impuls z PB10, není senzor připojen. Tímto způsobem je možné přidat více čidel omezující je zde dostatečný počet pinů mikrokontroleru, které přivádí spouštějící impuls a zvyšující se kapacita na vodičích sběrnice. Nabízí se také kombinace obou popsaných řešení. Ta je náročnější na počet pinů mikrokontroleru, ale kapacita vodičů na sběrnici by byla rozložena.[1][7]



Obr. 2.7 Zapojení několika senzorů na sběrnici.

## 2.5 Ukládání naměřených dat

Pro ukládání teplotních a vlhkostních dat byla zvolena Secure Digital (SD) karta, která je dnes široce používaná pro přenosná zařízení (telefony, fotoaparáty, notebooky apod.). Karty podporují tři způsoby komunikace, 1bit SD, 4 bit SD, SPI. Mikrokontroler obsahuje řadič pro SPI, proto je zvolen způsob komunikace po SPI.[15]

SPI je druhořadý komunikační protokol, z hlediska komunikace s SD kartami. Výhodou je široká podpora v mnoha zřízeních, nevýhodou je nižší rychlost oproti použití 1bit SD, 4 bit SD. Data je možné předávat mezi mikroprocesorem („master“) a několika SD kartami („slave“). Komunikace s SD kartou je založena na příkazech a odpovědích. Příkaz má 48 bitů, jsou to: start bit, číslo příkazu, argument, CRC a stop bit. Každý příkaz začíná uvedením signálu chip select do nízkého stavu. Tím se stanoví, s kterou SD kartou bude probíhat komunikace. Na každý přijatý příkaz odpoví karta. Pokud nebyla data v pořádku obdržena, odvysílá karta chybový rámeček. Příkaz odpověď nebo datový blok má délku násobku osmi bitů.[8]

Každá karta začíná komunikaci v SD modu. Rámeček komunikace je vybrán v inicializační sekvenci. Ta je volána v knihovně `microsd.c`, která odkazuje do knihovny `diskio.c`. Knihovny jsou součástí vývojového prostředí.

Knihovna `microsd.c` byla upravena, aby ji bylo možné použít pro mikrokontroler Pearl gecko. Knihovna používá registr ROUTE (routing register), tento registr je použit u většiny mikrokontrolerů od firmy Silicon labs, Pearl gecko k nim bohužel nepatří. Místo toho jsou použity registry ROUTEPEN a ROUTELOC0, k nastavení jednotlivých pinů SPI komunikace. Upravená knihovna je součástí projektu přiloženého na CD. Knihovnu `diskio.c` je možné použít tak, jak je.

Inicializační sekvence je následující. Nejprve je inicializován hodinový signál na nízké frekvenci. Je poslán příkaz CMD0, který vyvolá reset karty. Příkazem CMD8 se zjistí platnost pracovních podmínek. Příkaz CMD58 zjistí kompatibilitu pracovního napětí karty. Příkaz CMD41 navrátí, jestli karta podporuje HCS (High Capacity Support). Příkazem CMD16 se nastaví délka zápisových respektive čtecích bloků. Hodinová frekvence se přepne na vysokou frekvenci. Na vysoké hodinové frekvenci je pak vykonávána komunikace mezi zařízeními, datové bloky ty mají stejnou popsanou podobu.[9][8]

```
DSTATUS disk_initialize (  
BYTE drv /* Physical drive number (0) */  
)  
{  
BYTE n, cmd, ty, ocr[4];
```

```

if (drv) return STA_NOINIT; /* Supportonly single drive
*/

if (stat & STA_NODISK) return stat; /* No card in the socket */

MICROSD_PowerOn(); /* Forcesocket power on */

MICROSD_SpiClkSlow(); /* Start with low SPI clock.
*/

for (n = 10; n; n--) MICROSD_XferSpi(0xff); /* 80 dummy clocks */

    ty = 0;

if (MICROSD_SendCmd(CMD0, 0) == 1) { /* Enter Idle state */

MICROSD_TimeOutSet(1000); /* Initialization timeout of 1000
msec */

if (MICROSD_SendCmd(CMD8, 0x1AA) == 1) { /* SDv2? */

for (n = 0; n < 4; n++) ocr[n] = MICROSD_XferSpi(0xff); /* Get trailing
return value of R7 resp */

if (ocr[2] == 0x01 && ocr[3] == 0xAA) { /* The card can work at vdd range of 2.7-
3.6V */

while (!MICROSD_TimeOutElapsed() && MICROSD_SendCmd(ACMD41, 0x40000000)); /*
Wait for leaving idle state (ACMD41 with HCS bit) */

if (!MICROSD_TimeOutElapsed() && MICROSD_SendCmd(CMD58, 0) == 0) { /*
Check CCS bit in the OCR */

for (n = 0; n < 4; n++) ocr[n] = MICROSD_XferSpi(0xff);

        ty = (ocr[0] & 0x40) ? CT_SD2 | CT_BLOCK : CT_SD2; /* SDv2 */

    }

}

} else { /* SDv1 or MMCv3 */

if (MICROSD_SendCmd(ACMD41, 0) <= 1) {

        ty = CT_SD1; cmd = ACMD41; /* SDv1 */

    } else {

        ty = CT_MMC; cmd = CMD1; /* MMCv3 */

    }

while (!MICROSD_TimeOutElapsed() && MICROSD_SendCmd(cmd, 0)); /*
Wait for leaving idle state */

if (MICROSD_TimeOutElapsed() || MICROSD_SendCmd(CMD16, 512) != 0) /* Set

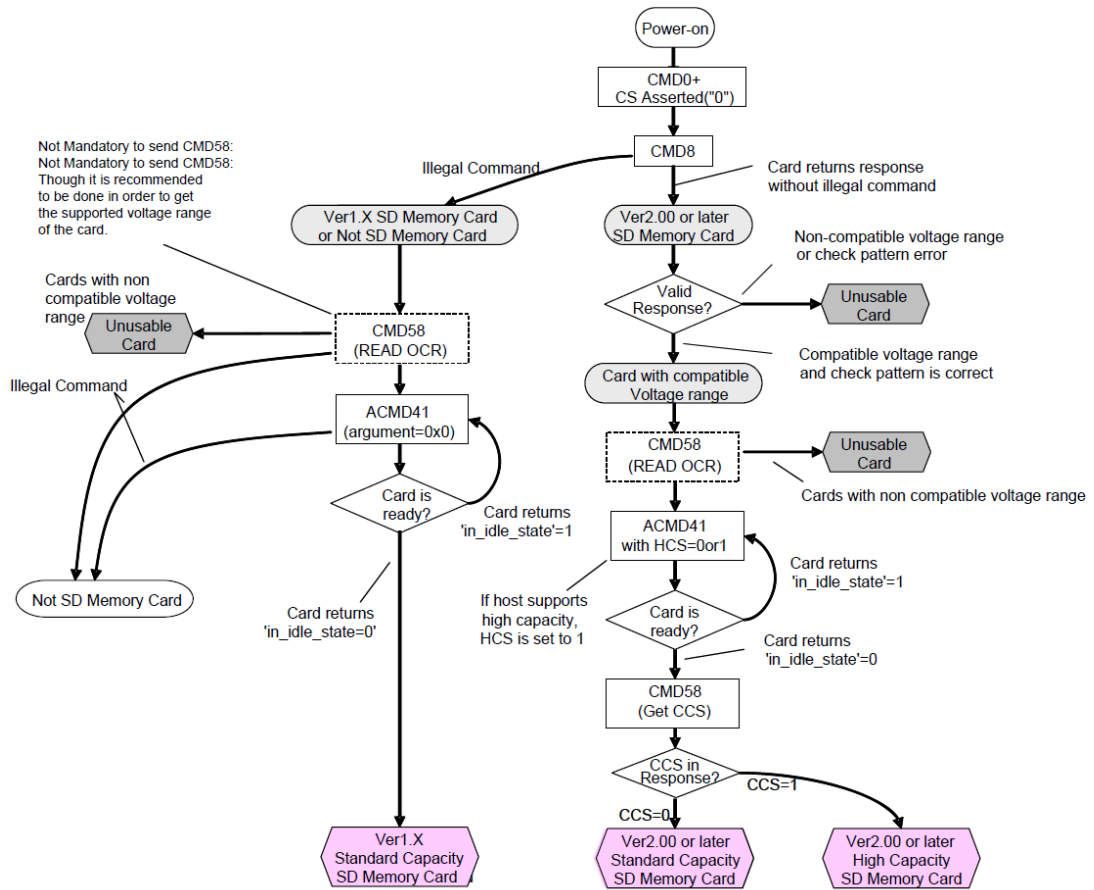
```

```
read/writeblocklength to 512 */
    ty = 0;
}
}
CardType = ty;
MICROSD_Deselect();

if (ty) {                               /* Initializationsucceeded */
stat&= ~STA_NOINIT;                       /* Clear STA_NOINIT */
MICROSD_SpiClkFast();                   /* Speed up SPI clock. */
    } else {                               /* Initializationfailed */
MICROSD_PowerOff();
stat |= STA_NOINIT;                       /* Set STA_NOINIT */
    }

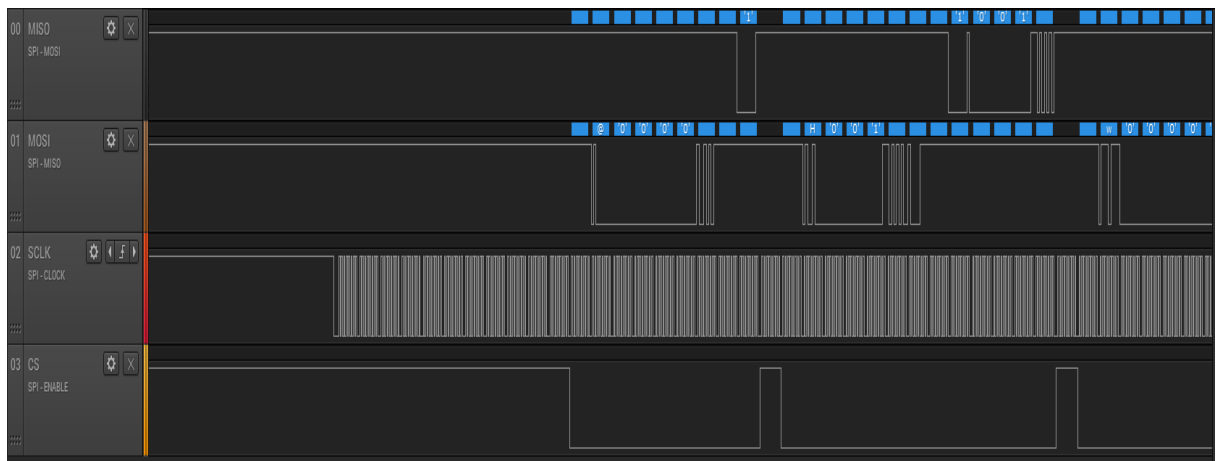
returnstat;
}
```

Na obrázku 2.8 je zobrazen vývojový diagram posloupnosti příkazů z dokumentu „SD Physical Layer Specification“.

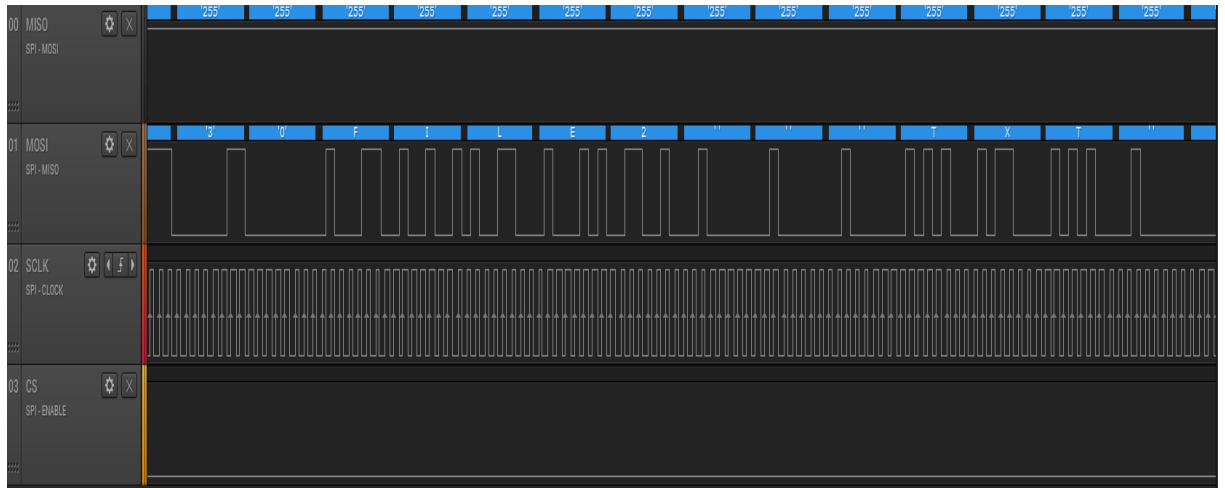


Obr. 2.8 Inicializace komunikace s SD kartou

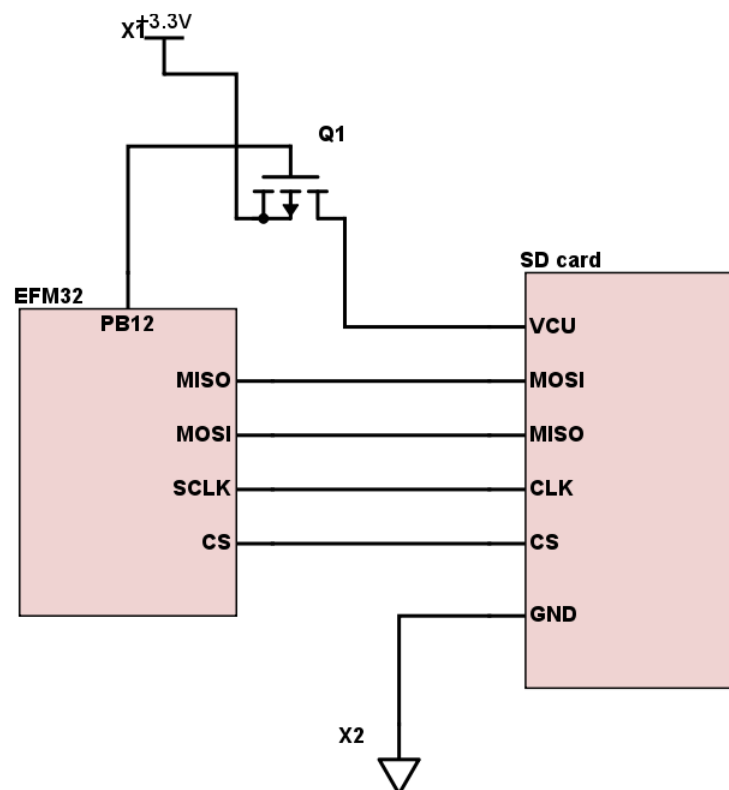
Komunikace byla také zanalyzována jako v předchozím případě pomocí logického analyzátoru. Díky tomu je možné prohlédnout výše popsanou komunikaci v obrázcích č. 2.9-10



Obr. 2.9 Začátek komunikace, přenos příkazů inicializace



Obr. 2.10 Přenos dat na SD kartu



Obr. 2.11 Zapojení SD karty na mikrokontroler [vlastní zpracování]

Když přístroj pracuje z bateriového napájení, musí se SD karta napájet přímo z baterie nebo z konektoru VMCU. Pin 3,3 V na desce je napojen na regulátor napětí, které dává USB. Napájení karty je připojeno přes PMOS tranzistor. Na bránu tranzistoru jsou přiváděny spínací impulzy z portu PB12. Napájení je tak aktivní pouze když je vyvolána funkce **sdlog**.



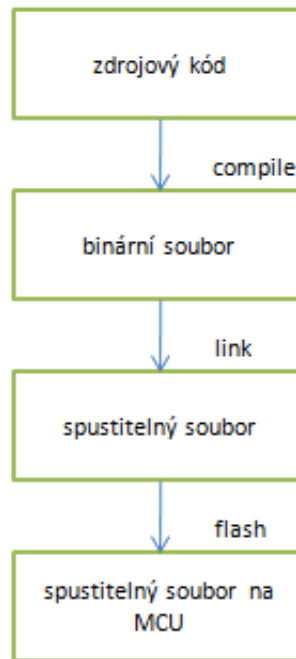
Na začátku funkce je sepnut tranzistor a tím napájecí napětí karty. Karta odebírá v klidovém stavu proud v řádu několika mA. To je pro zařízení s nízkou spotřebou nepřijatelné. Po dokončení zápisu je tranzistor rozepnut, zároveň je ještě nutné deinitializovat datové vodiče.[9]

**Tab. 1** Sobě odpovídající piny [vlastní tvorba]

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

### 3 Návrh programu

K návrhu programu 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.



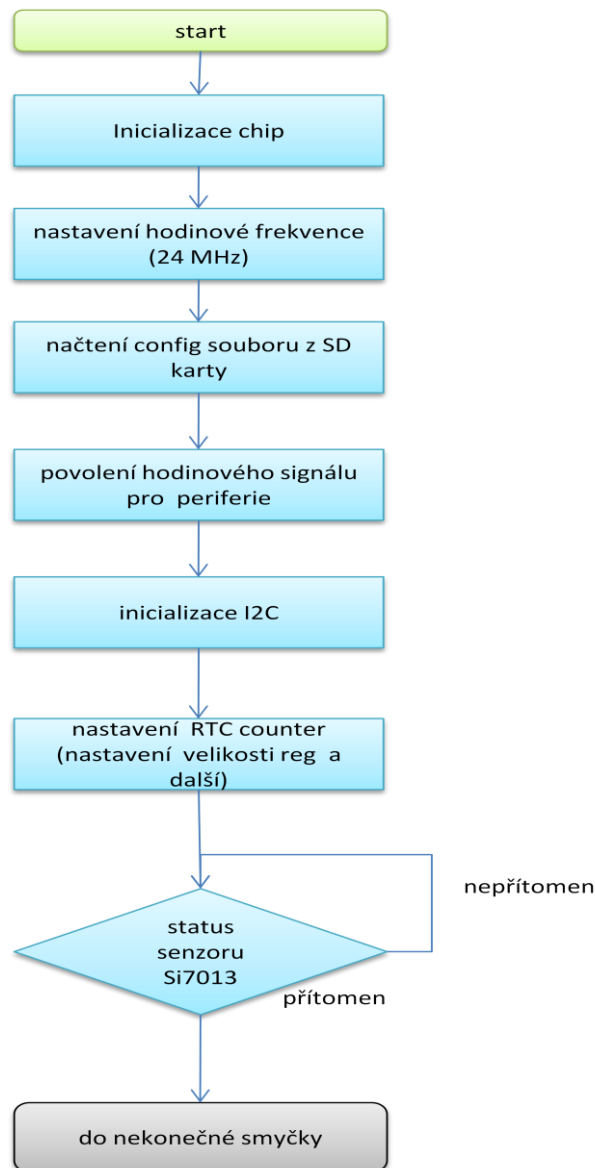
Obr. 3.1 Kroky IED v návrhu projektu [vlastní zpracování]

Návrh programu dataloggeru je proveden v jazyce C, bylo zvoleno Simplicity studio z důvodů uživatelské podpory. Simplicity studio není omezeno velikostí kompilovaného kódu. Druhou variantou je IAR Workbench prostředí jako simplicity studio. Disponuje také příkladnými projekty, kde jsou demonstrovány všechny periferie na desce. Na rozdíl od vybraného vývojového prostředí má volná licence 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.

#### 3.1 Hlavní funkce programu

Hlavní funkce programu je velmi jednoduchá. Na začátku je deklarace proměnných. Teplota a vlhkost se ukládají do *rhdata* a *tempdata*. V program uchovává minimální a maximální teplotu, k tomu slouží *maxtemp* a *mintemp*. Teplota předávána z teplotního čidla je v m°C, proto je jejich počáteční hodnota nastavena na 0 a 99999999 aby zachytili očekávané rozmezí

teplot okolo 10-30°C . *I* a *pbdelay* jsou pomocné proměnné, které obstarávají práci s tlačítky. Následují inicializace oscilátoru, a hodinových signálů pro jednotlivé periférie. Zjišťuje se status senzoru si7021. I když program pro práci se senzorem nese jméno podle jiného senzoru od firmy silicon labs, je používán i pro tento senzor.



Obr. 3.2 začátek hlavní funkce [vlastní zpracování]

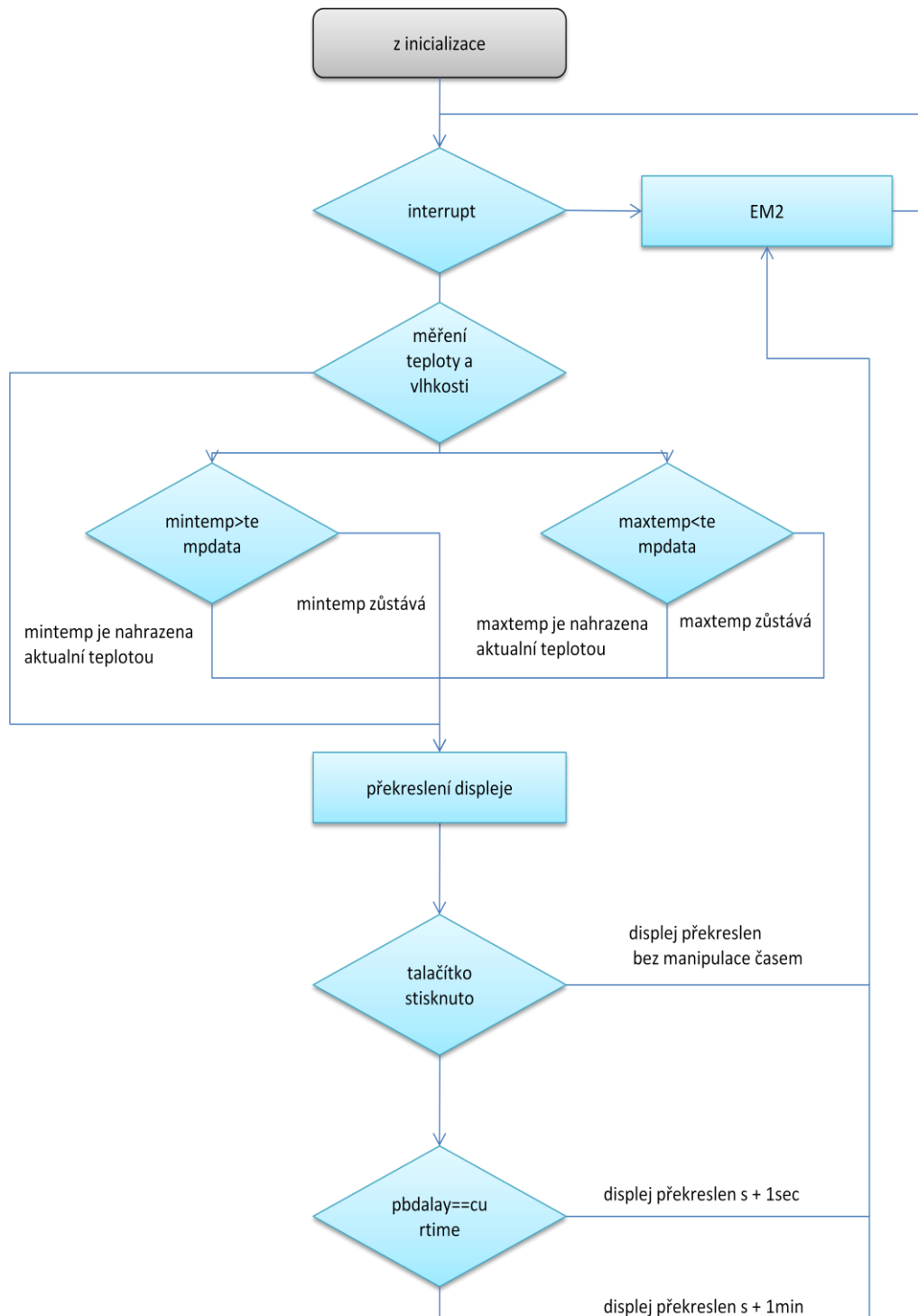
V druhé polovině hlavní funkce programu je nekonečná smyčka pro práci s tlačítky a volání dalších funkcí programu. Pokud je *updateMeasurement* v hodnotě true, provede se měření a přepne se na hodnotu false. Hodnota je přepínána ve funkci, která obsluhuje přerušení. Ty jsou generovány každou vteřinu. Tento časový interval je ovlivněn frekvencí oscilátoru, který byl nastaven v předchozí části programu. Byl takto zvolen kvůli zobrazování času s přesností na vteřiny. Minimální a maximální teplota jsou obslouženy velmi jednoduchou

podmínkou.

Pro nastavení času je použito tlačítko PB0, při stisknutí je inkrementován čas o jednu vteřinu. Pokud je tlačítko stisknuto alespoň jednu vteřinu inkrementuje se čas o minuty. Tato funkce je dosažena smyčkou, hodnota ***pbdelay*** je nastavena o jednu vteřinu větší, než je aktuální čas. Uvnitř smyčky je podmínka, která kontroluje, zda je tlačítko stisknuto. Pokud je tlačítko v nestisknuté poloze, příkaz *break* vrátí běh programu mimo tělo smyčky. Program setrvá ve smyčce, dokud není generováno přerušení, ve kterém se inkrementuje čas o jednu vteřinu.

Následuje smyčka, jejíž podmínka je splněna v okamžiku, kdy je tlačítko stisknuto. Zde se inkrementuje čas o minuty a volá se funkce ***GRAPHICS\_Draw***, která překreslí display.

Pokud vše proběhlo jak má, je mikrokontroler uveden do úsporného stavu pomocí příkazu ***EMU\_EnterEM2***, ve kterém setrvá, dokud se nevygeneruje další přerušení.

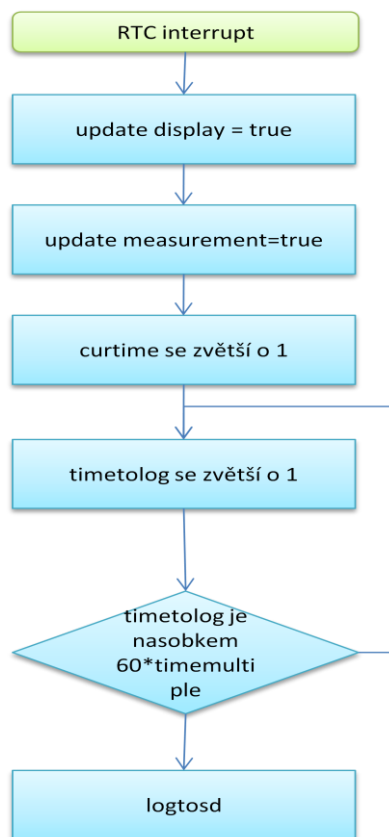


Obr. 3.3 Vývojový diagram hlavní smyčky programu [vlastní zpracování]

### 3.2 Zpracování přerušení

Přerušení jsou zpracovávána ve funkci *periodicUpdateCallback*. Proměnné, které byly vynulovány v hlavní funkci, jsou nastaveny zpět do původních hodnot. Je tak vyvoláno opět měření nebo vykreslení displeje. Zápis dat na SD kartu je velmi energeticky náročný, a tak je prováděn v řádech minut. Přesné nastavení je možné určit pomocí konfiguračního souboru

na SD kartě, ze kterého je získán parametr *timemultiple*, pomocí funkce *getConfig*, ta je volána v hlavní funkci během inicializací.



Obr. 3.4 Vývojový diagram generace přerušení [vlastní zpracování]

## 4 Management nízké spotřeby

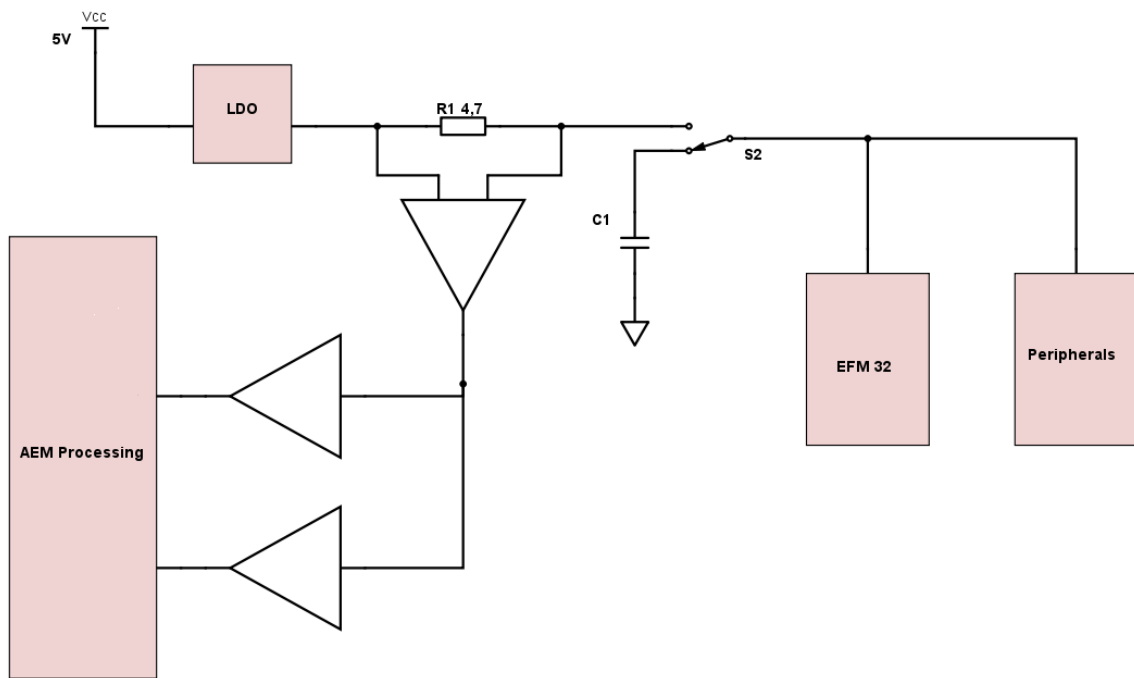
U bateriově napájených zařízení hraje spotřeba klíčovou roli. Proto je vhodné používat úsporné módy. Mikrokontroler vykoná všechny instrukce, jak nejrychleji to jde, poté se přepne do úsporného módu, ve kterém má řádově nižší spotřebu, dokud není vyvoláno přerušení a není potřeba zpracovat další instrukce. Současně je nutné, aby byly řádně vypnuty (deinicializovány) všechny nepotřebné periférie (hodinový signál, data, chip select, apod). [14]

Přechody mezi jednotlivými úspornými módy 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. Režim 0 je také výchozí pozicí po restartu, ostatní módy musí být zinicizovány softwarem. [13]

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

### 4.1 Měření spotřeby

Měření spotřeby bylo realizováno podle schématu na obrázku č.4.1. Jedná se o popis tzv. „advance energy monitoring“, které je součástí vývojové desky a slouží k vyladění programu z hlediska spotřeby. Do série s vlastním zařízením byl zařazen odpor o velikosti 4,7  $\Omega$ . Na odpor je připojen proudový zesilovač, který je napojen na zesilovače s nízkým a vysokým zesílením. Takto získaný proud je označen časem a instrukcemi od mikrokontroleru a je odeslán do počítače přes USB port. Rozsah snímaného proudu je 1  $\mu\text{A}$  až 50 mA, hranice přepínání mezi zesilovači je 250  $\mu\text{A}$ . Absolutní přesnost je udávána 1  $\mu\text{A}$  v rozsahu do 250  $\mu\text{A}$  a 0.1 mA v rozsahu nad 250  $\mu\text{A}$ . [1]



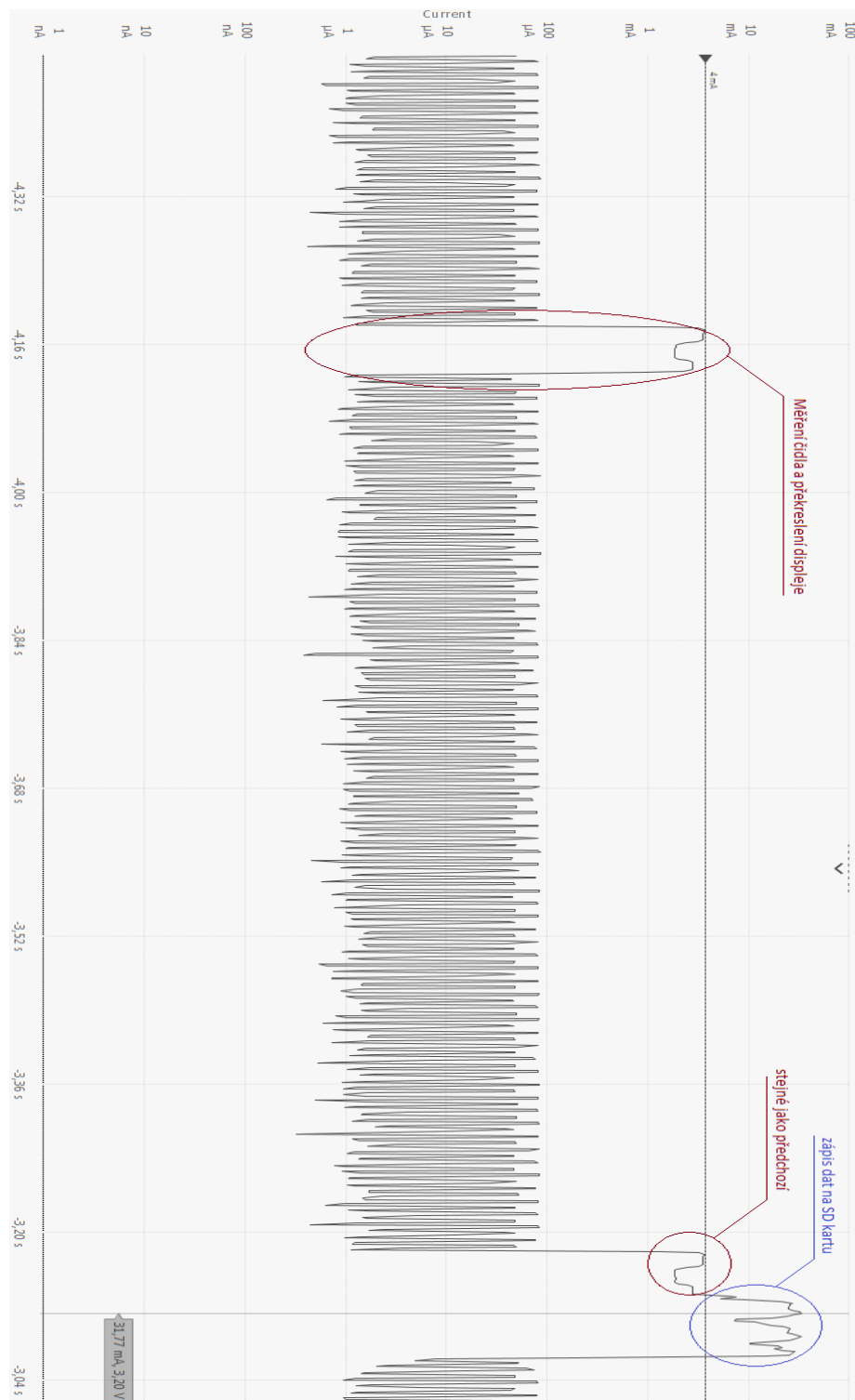
Obr. 4.1 schéma zapojení měření spotřeby. [vlastní zpracování]

## 4.2 Cyklus měření a zápisu na SD kartu

Zařízení zapisuje v několika vteřinových intervalech naměřenou teplotu a vlhkost. Po měření přejde do úsporného režimu 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 zapíší na paměťovou kartu. Poté se celý cyklus opakuje. Podrobný popis funkce programu a vývojový diagram je popsán v předchozí kapitole.

Pokud probíhá jen přeměření teplot a vykreslení displeje bez zápisu na SD kartu, proudová špička trvá jen okolo 55ms a její maximální hodnota je 4mA. Na druhé špičce je vidět přeměření teplot s vykreslením displeje a zápisem na SD kartu. V tomto případě trvá špička téměř dvojnásobnou dobu a její maximální proud může dosahovat velikosti okolo 30 mA. Je vidět, že zápis nebo čtení z SD karty je energeticky nejnáročnějším procesem v celé činnosti programu, a proto je nutné ho dělat pokud možno co nejméně. V předchozí kapitole byl popsán nastavitelný parametr, kterým může uživatel podle potřeby zkrátit interval mezi jednotlivými zápisy.



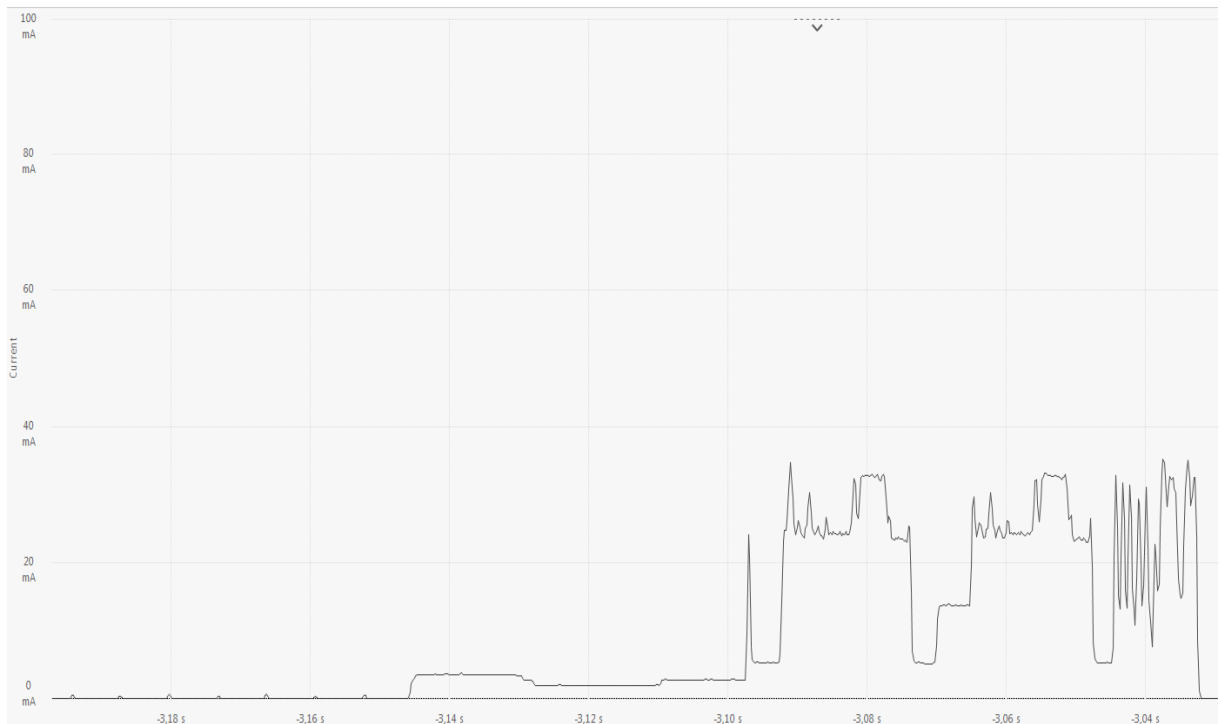


Obr. 4.2 Průběh proudů během funkčních cyklů [vlastní zpracování]

### 4.3 Detail průběhu:

V grafu je zobrazen detailní průběh aktivní části měřicího a zápisového cyklu. V první části je vidět průběh při měření teploty, v druhé části je zápis dat na SD kartu. V třetí části je vidět

konec komunikace, uzavření složky a přechod systému do úsporného režimu 2.

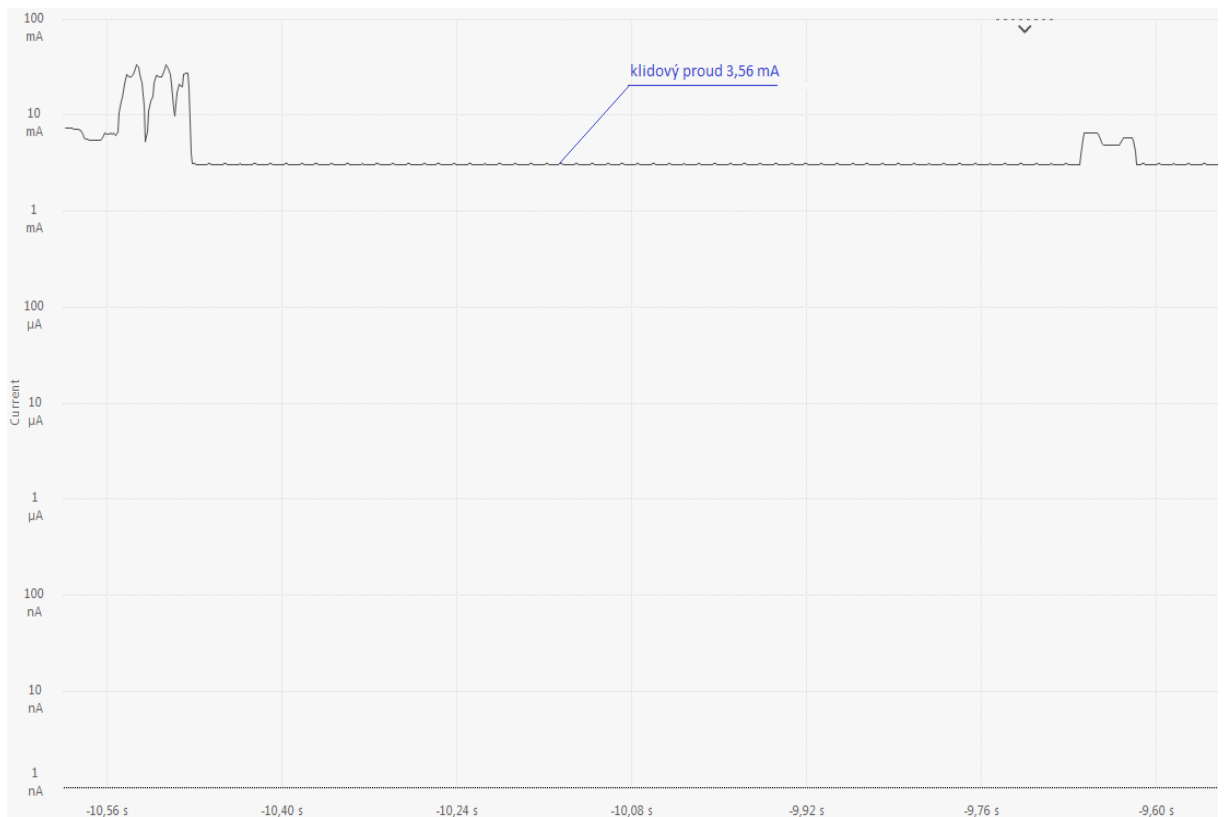


Obr. 4.3 Detail na měření teploty a zápis na SD kartu [vlastní zpracování]

#### 4.4 Porovnání spotřeby zařízení v různých režimech

Největší podíl na spotřebě, jak již bylo zmíněno, má SD karta. A to jak v aktivním režimu, tak i v klidovém stavu. Proto jak bylo popsáno v kapitole o ukládání dat napájení SD karty je odepínáno pomocí PMOS tranzistoru. Klidový proud na kartě použité v teplotním dataloggeru byl naměřen 3,15 mA. Kartu však nestačí pouze odepínat. Je zapotřebí také nastavit ostatní I/O piny do stavu low, v opačném případě je klidový proud stále vysoký.

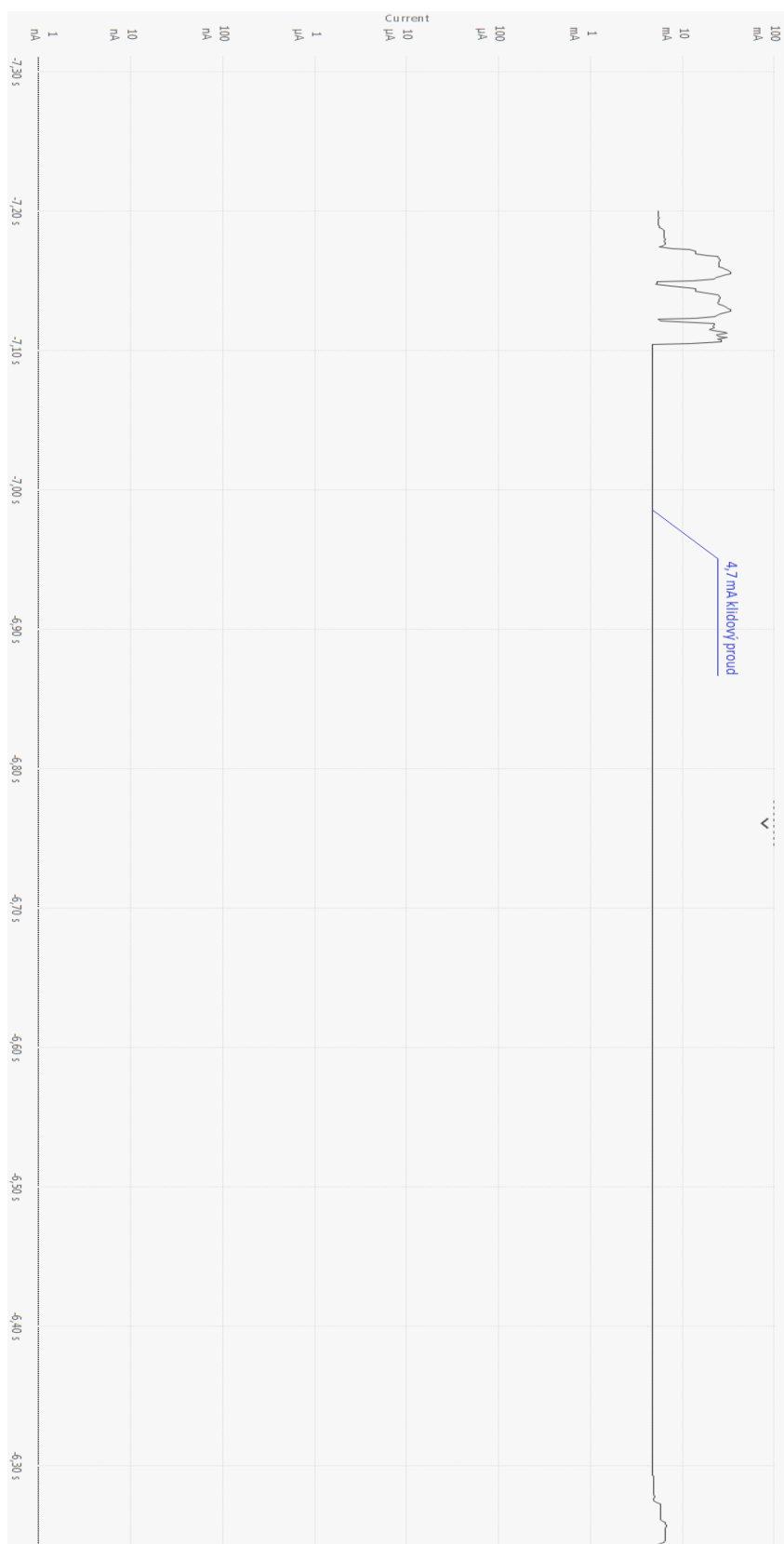
Na obrázku č. 4.4 je vidět průběh bez vypnuté inicializace. Přestože je nyní napájení odepnuto od zdroje, klidový proud zařízení je i přesto vysoký.



**Obr. 4.4 Průběh s nesprávnou deinitializací karty**

Druhou největší příčinou spotřeby zařízení je přecházení mezi jednotlivými úspornými režimy. Pokud mikrokontroler není přepnut do úsporného režimu, zařízení odebírá 1,55 mA navíc.

Na obrázku č 4.5 je zachycen průběh proudu s nesprávně odepnutou SD kartou a v klidových stavech zařízení setrvává ve stejném režimu. Proudové špičky téměř splývají s klidovým stavem, ve kterém odebírá 4,7 mA.



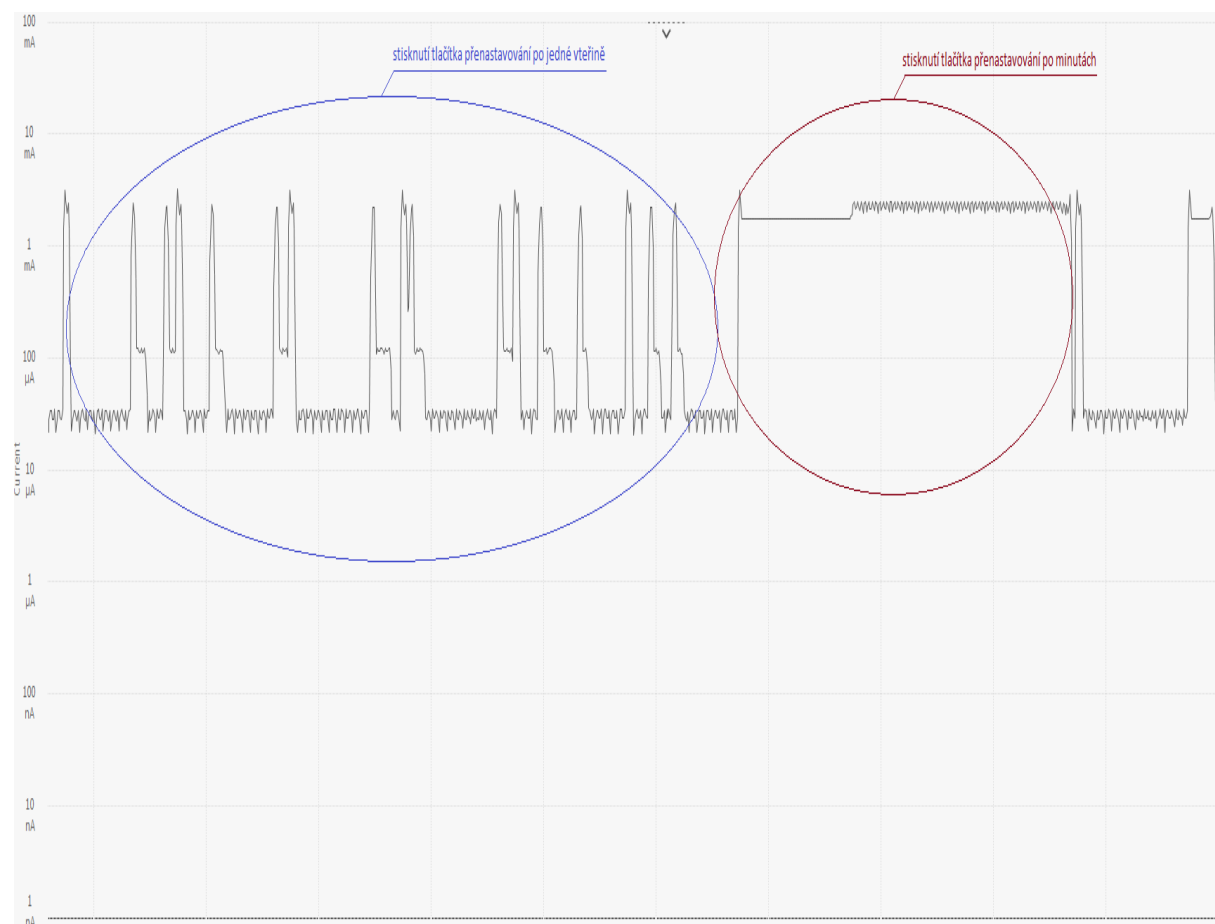
**Obr. 4.5** průběh proudu při setrvání v EM0, s nesprávnou deinicializací karty

## 4.5 Průběh proudu při zapnutí zařízení

Po zapnutí má zařízení největší odběr, nabíjí se kondenzátory a parazitní kapacity. Zařízení nedělá jen zápis dat, ale vytváří se také složka pro teplotní data. Načítají se parametry z konfiguračního souboru a inicializují se všechny periferní obvody. Na obrázku č. 4.7 je vidět, jak ovlivní napájení překreslování displeje, pomocí tlačítek. Stisknutí tlačítka způsobuje krátké impulzy při listování mezi minimální a maximální teplotou, nebo dlouhý impulz při nastavování času. Proud se při těchto impulzech pohybuje okolo 2mA.



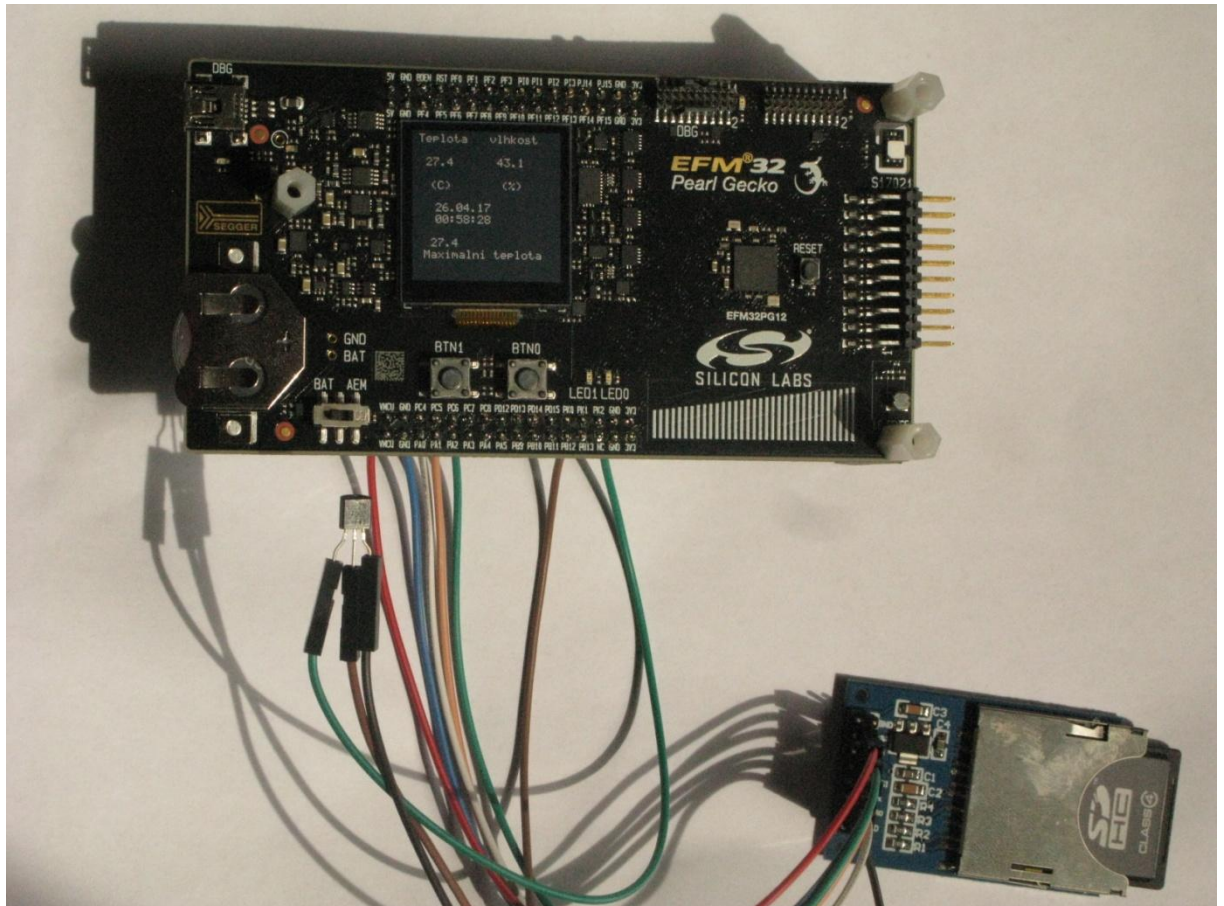
Obr. 4.6 Průběh při zapnutí zařízení



Obr. 4.7 Průběh při práci s displejem[vlastní zpracování]

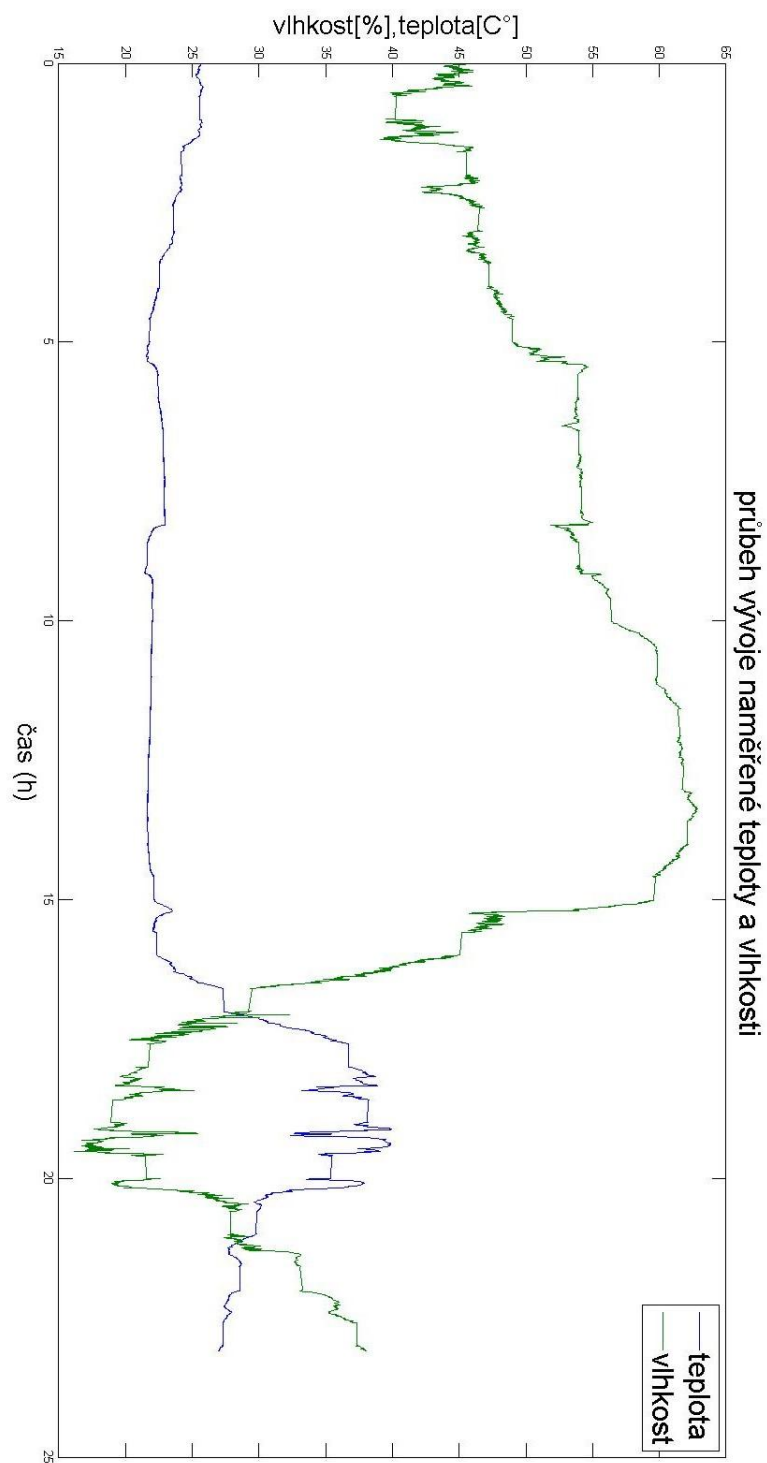
## 5 Měření vlhkosti a teploty

Pomocí dataloggeru bylo provedeno měření vlhkosti a teploty v pokoji. Zařízení bylo umístěno za oknem. Zařízení zaznamenávalo 24 hodin data. Přes noc spal v místnosti jeden 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 obydlého pokoje, je poměrně veliká.



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 úsporného režimu a správnou deinitializací nepoužívaných periférií. Rozdíl mezi jednotlivými proudovými špičkami byl 30mA. Zařízení odebírá průměrný proud o hodnotě 55  $\mu$ A. Při použití CR2032 baterie o kapacitě 220 mAh by zařízení vydrželo pracovat 4 000 hodin. To by postačilo na několik fermentačních cyklů a zařízení by pořídilo 240 000 záznamů teploty a vlhkosti.

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

## Seznam zdrojů

- [1] SILICON LABS. *UG257: EFM32 Pearl Gecko PG12 Starter Kit User's Guide* [online]. s. 29 [cit. 2017-05-14]. Dostupné z: <http://www.silabs.com/documents/public/user-guides/ug257-stk3402-usersguide.pdf>
- [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](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] BURIAN, Petr. Software pro monitorování a záznam zpráv na sběrnici I2C. Plzeň, 2005. Bakalářská. ZČU Plzeň. Vedoucí práce Vít Martin.
- [8] *Lecture 12: SPI and SD cards* [online]. 2013 [cit. 2017-05-14]. Dostupné z: [http://www.dejazzer.com/ee379/lecture\\_notes/lec12\\_sd\\_card.pdf](http://www.dejazzer.com/ee379/lecture_notes/lec12_sd_card.pdf)
- [9] *Physical Layer Simplified Specification: SD Specifications* [online]. ©2001-2006, [cit. 2017-05-14]. Dostupné z: [http://users.ece.utexas.edu/~valvano/EE345M/SD\\_Physical\\_Layer\\_Spec.pdf](http://users.ece.utexas.edu/~valvano/EE345M/SD_Physical_Layer_Spec.pdf)
- [10] PINKER, Jiří. *Mikroprocesory a mikropočítače*. Praha: BEN - technická literatura, 2004. ISBN 80-7300-110-1.
- [11] SILICON LABS. *EFM32PG12 Pearl Gecko Family Reference Manual* [online]. s. 1193 [cit. 2017-05-14]. Dostupné z: <http://www.silabs.com/documents/public/reference-manuals/efm32pg12-rm.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: I<sup>2</sup>C 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 VÝVOJOVÁ DESKA A POUŽITÉ PERIFERIE [9] .....	15
OBR. 2.2 BLOKOVÉ SCHÉMA ZAPOJENÍ I <sup>2</sup> C SBĚRNICE [VLASTNÍ ZPRACOVÁNÍ] .....	16
OBR. 2.3 ČASOVÉ PRŮBĚHY KOMUNIKACE PO I <sup>2</sup> C SBĚRNICI [21] .....	17
OBR. 2.4 ZAČÁTEK KOMUNIKACE S TEPLTNÍM A VLHKOSTNÍM ČIDLEM (PŘÍKAZY PRO MĚŘENÍ TEPLoty A VLHKOSTI) .....	17
OBR. 2.5 PŘEHLED CELÉ KOMUNIKACE. OBVOD DRŽÍ HODINOVÝ SIGNÁL V LOG 0 DO DOBY, NEŽ NEVYŘÍDÍ PŘÍKAZY. ....	18
OBR. 2.6 KONEC KOMUNIKACE S DATOVÝM BLOKEM OBSAHUJÍCÍM TEPLTNÍ A VLHKOSTNÍ DATA .....	18
OBR. 2.7 ZAPOJENÍ NĚKOLIKA SENZORŮ NA SBĚRNICI. ....	19
OBR. 2.8 INICIALIZACE KOMUNIKACE S SD KARTOU .....	23
OBR. 2.9 ZAČÁTEK KOMUNIKACE, PŘENOS PŘÍKAZŮ INICIALIZACE .....	23
OBR. 2.10 PŘENOS DAT NA SD KARTU .....	24
OBR. 2.11 ZAPOJENÍ SD KARTY NA MIKROKONTROLER [VLASTNÍ ZPRACOVÁNÍ] .....	24
OBR. 3.1 KROKY IED V NÁVRHU PROJEKTU [VLASTNÍ ZPRACOVÁNÍ] .....	26
OBR. 3.2.....	27
OBR. 3.3 VÝVOJOVÝ DIAGRAM HLAVNÍ SMYČKY PROGRAMU .....	29
OBR. 3.4 VÝVOJOVÝ DIAGRAM GENERACE PŘERUŠENÍ .....	30
OBR. 4.1 SCHÉMA ZAPOJENÍ MĚŘENÍ SPOTŘEBY. [VLASTNÍ ZPRACOVÁNÍ] .....	32
OBR. 4.2 PRŮBĚH PROUDŮ BĚHEM FUNKČNÍCH CYKLŮ [VLASTNÍ ZPRACOVÁNÍ] .....	33
OBR. 4.3 DETAIL NA MĚŘENÍ TEPLoty A ZÁPIS NA SD KARTU [VLASTNÍ ZPRACOVÁNÍ] .....	34
OBR. 4.4 PRŮBĚH S NESPRÁVNOU DEINICIALIZACÍ KARTY .....	35
OBR. 4.5 PRŮBĚH PROUDU PŘI SETRVÁNÍ V EM0, S NESPRÁVNOU DEINICIALIZACÍ KARTY .....	36
OBR. 4.6 PRŮBĚH PŘI ZAPNUTÍ ZAŘÍZENÍ .....	37
OBR. 4.7 PRŮBĚH PŘI PRÁCI S DISPLEJEM [VLASTNÍ ZPRACOVÁNÍ] .....	38
OBR. 5.1 CELÉ ZAŘÍZENÍ V CHODU PŘI MĚŘENÍ TEPLoty A VLHKOSTI. [VLASTNÍ ZPRACOVÁNÍ] .....	39
OBR. 5.2 NAMĚŘENÉ HODNOTY TEPLoty A VLHKOSTI [VLASTNÍ ZPRACOVÁNÍ] .....	40

## **Seznam tabulek**

TAB. 1 SOBĚ ODPOVÍDAJÍCÍ PINY [VLASTNÍ TVORBA].....	25
---	----

## **Přílohy**

### **Příloha 1**

Projekt dataloggeru vypracovaný v IDE Simplicity studio na přiloženém CD

### **Příloha 2**

Kompletní schéma zapojení na přiloženém CD

### **Příloha 3**

Datasheet a reference manual mikrokontroleru na přiloženém CD