



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra elektromagnetického pole

Využití tabulkového procesoru pro sběr a záznam dat

Data collection using spreadsheet application

Bakalářská práce

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Komunikační technika

Vedoucí práce: Ing. Lukáš Vojtěch Ph.D.

Daniel Dousek

Praha 2017

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Dousek** Jméno: **Daniel** Osobní číslo: **434711**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektromagnetického pole**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Komunikační technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití tabulkového procesoru pro sběr a záznam dat

Název bakalářské práce anglicky:

Data collection using spreadsheet application

Pokyny pro vypracování:

Navrhněte a zrealizujte makra pro sběr a zpracovávání dat v tabulkovém procesoru. Uvažujte připojení vybraných zařízení pomocí sériové komunikační linky (RS232/USB). Realizujte makra pro odečet veličin ze spínaného laboratorního zdroje MANSON série HCS (elektrické napětí a proud) a makra pro měření teploty s vybraným USB teploměrem. Preferujte open source tabulkové procesory. Detaily sdělí vedoucí práce.

Seznam doporučené literatury:

- [1] Laurenčík, M.: Programování v Excelu 2010 a 2013, Grada Publishing, ISBN: 978-80-247-5033-0 .
- [2] Dokumentace dostupná na <http://www.manson.com> [on-line]

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Lukáš Vojtěch Ph.D., katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **17.02.2017**

Termín odevzdání bakalářské práce: **26.05.2017**

Platnost zadání bakalářské práce: **30.09.2018**

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou.

Datum: 25. 5. 2017

.....

Podpis

Poděkování

Chtěl bych velice poděkovat svému vedoucímu práce, Ing. Lukášovi Vojtěchovi Ph.D, za jeho velikou trpělivost a ochotnou pomoc při řešení této práce.

Abstrakt

Cílem této práce je podat informace o platformách pro zabudované systémy, o použitých přístrojích, jejich způsobu komunikace a o tabulkových procesorech a jejich možnostech. Dále popíšu, jak jsem vytvořil makro pro sběr dat z laboratorního zdroje a teploměru. Na konci pohovořím, jak by se správně měl okomentovat kód.

Klíčová slova

Sběr dat, zabudovaná platforma, makro, skript

Abstract

The purpose of this work is to give information about platforms for embedded systems, about applied devices, their form of communication and about spreadsheets and their possibilities. Next I will talk about, how I created macro for gathering data from laboratory power supply and thermometer. In the end I will talk about correct way to comment code.

Keywords

Data collection, embedded platform, macro, script

Obsah

1	Úvod do problematiky	12
2	Teoretická část.....	13
2.1	Platformy pro zabudované systémy (embedded systems).....	13
2.2	Tabulkové procesory	15
2.3	Sériová komunikace	16
2.4	1-Wire.....	17
2.5	RTC.....	18
2.6	Komentování kódu	19
3	Realizační část.....	20
3.1	Návrh zapojení.....	20
3.2	Raspberry Pi	21
3.3	Makro pro sběr dat.....	22
3.4	Laboratorní zdroj Manson HCS-3400-USB.....	24
3.5	Teploměr Dallas DS18B20.....	26
3.6	Dotykový displej	27
4	Závěr.....	28
4.1	Seznam použité literatury	29
4.2	Seznam obrázků	31

1 Úvod do problematiky

Sběr dat je důležitá procedura současného světa, jejímž cílem je získat a zapsat informace tak, aby se následně daly zpracovat a vytěžit z nich nějaký výsledek. Probíhá v různých odvětvích od výzkumu přes podnikání až po společenské vědy.

Způsob sběru se už dlouhou dobu vylepšuje od ručního opisování na listy papíru, s rozvojem výpočetní techniky přišlo digitální zapisování a načítání dat ze senzorů a jiných přístrojů. Mezi další možnosti patří čtení čárových kódů a dnes bezkontaktní čtení RFID čipů. Žádná z těchto metod se ale neobejde bez lidského zásahu. Pořád musíme určit, jaké hodnoty z jakých přístrojů budeme získávat a za jakým účelem je sbíráme. Zároveň musíme jednotlivé proměnné označit a připravit strukturu, kam se data v digitální formě zapíší.

Důležitými faktory jsou jak kvantita, tak kvalita nashromážděných dat. Kvantitativní sběr slouží je vhodný převážně pro statistické účely v biologii, sociálních vědách a ekonomice, kde není příliš kladen důraz na kvalitu vzorků nebo odpovědí, ale jejich množství. Na druhou stranu kvalitativní sběr dat se zaměřuje na přesnost a autentičnost získaných dat a jeho záměr je získat kompletní přehled o nějaké závislosti nebo problému. Z těchto dat se pak dají vytvářet modely, teorie nebo se pomocí nich kontrolují a vyvíjejí přístroje pro běžné použití nebo přesné měření. Obě metody jsou spolu provázané neboť v případě, že by dat bylo malé množství, tak i sebelepší kvalita dat by neuvědla celou pravdu a naopak celostátní průzkum, kde si polovinu odpovědí vymyslí, by byl úplně zbytečný a může vést k negativním efektům, které byly rozhodnuty právě na základě provedeného průzkumu.

Pro samotné shromažďování informací pak slouží takzvané systémy pro sběr dat neboli DCS. DCS je nějakou aplikací na počítači, nejčastěji to bývá tabulkový procesor nebo databáze, do které se data systematicky zapisují do námi zvolené struktury. To umožní jejich následnou analýzu a možné budoucí použití.

V současné době je velký rozvoj platform pro drony neboli bezpilotní letadla, které nalézají uplatnění u policie, hasičů, armádě, ale i jako zábava běžných obyvatel planety.

Mezi konkrétní platformy patří BeagleBoard X15, disponující výkonným hardwarem, kompatibilitu Linuxem nebo Androidem a širokou škálou připojitelných periférií. Mezi první využití patřili 3D tiskárny nebo robotika.

Dalším nováčkem je Alexa od Amazonu. Jedná se o novou inteligentní asistentku, kterou hned po představení zařadili do svých systémů velké firmy jako Ford, Samsung nebo LG. Tato novinka inspirovala i Arduino, které začalo pro asistentku vytvářet platformu STAR Otto, která by ostatním firmám usnadnila implementovat inteligentní program do svých systémů.

Mezi novým nástupcem na trh mikropočítačů se snaží prorazit ASUS Tinker Board, který stávajícímu vůdci Raspberry Pi3 konkuruje hlavně výrazně vyšším výkonem, ale zároveň ne o tolik vyšší cenou.

Posledním zmíněným je Systém on Chip procesor ATOM E(A)3900 od Intelu, který by měl být hlavní základnou pro Internet of Things systémy.

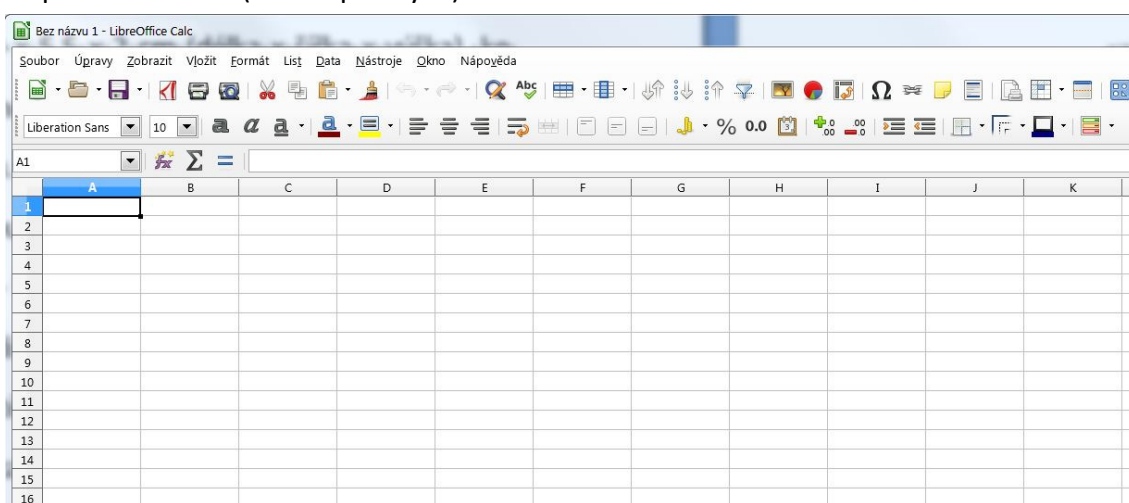
2.2 Tabulkové procesory

Již z názvu plyne, že se jedná o program, který má za úkol zpracovávat data v tabulce. Aplikace umí počítat základní aritmetické a matematické funkce pomocí vzorců napsaných v buňkách tabulky. Při změně proměnné pak automaticky přehodnotí výsledek v reálném čase. Kromě počítání může být využit pro vytváření grafů nebo i pro obyčejné seznamy dat, jako například seznam zaměstnanců, kde bude zaznamenáno jejich jméno, příjmení, bydliště a mnoho dalších. Všechny tyto funkce provádí na základě tří základních typů hodnot, které se do buněk tabulky zapisují: text pro databázové účely a popisy ostatních buněk, dále čísla pro použití ve výpočtech a funkce pro samotné zpracovávání hodnot v tabulce. Pokročilejší procesory disponují i dalšími funkcemi vhodnými pro finanční nebo statistické odvětví. |20|

Programy jsou dnes vybaveny skriptovacím prostředím pro tvorbu maker nejčastěji využívajícím různé verze programovacího jazyku BASIC. Tato makra mají nejčastěji za úkol vykonávat repetitivní úkony, dále například přeformátování textu nebo vkládání aktuálního data a času.

Nejnámějšími zástupci jsou dnes Excel z kancelářského balíčku Microsoft Office využívající jako programovací jazyk Visual Basic for Applications nebo volně dostupný Calc z dnes již nevyvíjeného OpenOffice a jeho nástupce LibreOffice od Apache. V obou se makra píšou v programovacím jazyku BASIC.

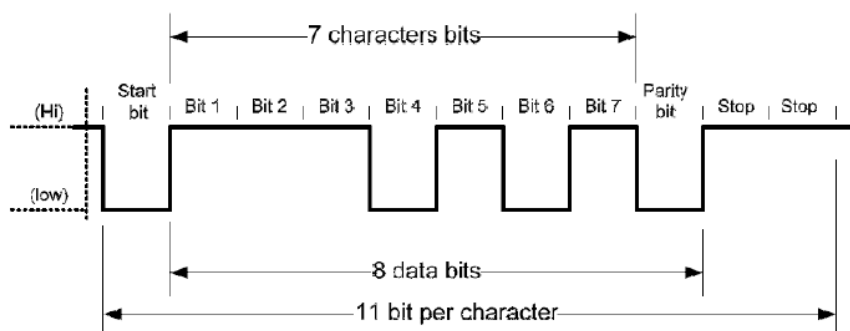
Pro zápis dat jsem použil právě LibreOffice Calc, který mi běží na operačním systému Raspbian s Pixelem (viz. Raspberry Pi).



OBRÁZEK 2 - LIBREOFFICE CALC

2.3 Sériová komunikace

Je způsob komunikace po datové lince. Na rozdíl od paralelní komunikace se bity neposílají paralelně vedle sebe, nýbrž v sérii za sebou. Bity se posílají sekvenčně, neboli jeden bit v jeden časový okamžik. To ubere počet potřebných pinů, ale zvýší složitost.



OBRÁZEK 3 - PŘENOS CHARAKTERU PO LINCE [22]

Komunikace je vedena buď synchronně, nebo asynchronně. Synchronní přenos vyžaduje, aby jedna strana určila bitovou rychlost přenosu dat, jinak by došlo ke čtení špatných hodnot kvůli desynchronizaci posílaných bitů. Asynchronní komunikace předpokládá, že obě strany mají totožně nastavené parametry komunikace. Mezi ně patří: bitová rychlost, která určuje, jak často se bity objevují, dále kolik bitů s daty se odešle a nakonec startovní a konečný bit, který obklopuje datové bity.

Mezi sériové porty patří jako hlavní zástupce RS-232. Jedná se o standard pro sériovou komunikaci v telekomunikacích. V současnosti samotný port RS-232 přestal být podporován a nahradili ho jeho nástupci jako například Ethernet nebo USB neboli Universal Serial Bus, kteří s malými výjimkami pracují na stejném principu, ale s vyššími rychlostmi a délkami kabelů. [19]

2.4 1-Wire

Neboli jednodrátové rozhraní je systém pro komunikaci se zařízením navrhnutý firmou Dallas Semiconductors. Podstatou je přenos dat, signalizace a napájení po jednom drátu. Sensory mají navíc jeden drát svedený na zem a jeden pro možnost externího napájení.

Komunikační protokol je založen na principu master-slave, aby na jednom drátě nedocházelo ke kolizím. Master je nějaký počítač nebo mikrokontroler, který zahajuje komunikaci. Slave je pak nějaký senzor, který čeká, než mu dá master pokyn, aby něco udělal. Slave pak podle typu příkazu vrací žádané hodnoty.

Jednodrátové zařízení mají zároveň svoje unikátní sériové číslo, které umožní mít na jednom drátě umístěno více senzorů najednou. Takové unikátní číslo má 8 bitů pro typ zařízení, 8 bitů pro Cyclic Redundancy Check a zbytek je výrobní číslovaného typu senzoru. [24]

2.5 RTC

RTC nebo Real-Time Clock je zařízení, které má za úkol udržovat na přístroji aktuální čas a datum i mimo napájení. Základní požadavky jsou hlavně nízká spotřeba energie, velikost a přesnost udržovaného času. Pokud je RTC bez napájení musí se spolehnout na svojí vnitřní nejčastěji Lithiové baterie. V současné době se objevují na trhu takzvané superkondenzátory. [15] Superkondenzátory jsou speciální verze elektrolytických kondenzátorů s uhlíkovými elektrodami. Charakterizuje je kapacita v řádu Faradů, krátká doba dobíjení a nízké napětí. [17] Přesnost hodin zajišťuje krystalový oscilátor, který je velmi stabilní při pokojové teplotě. Pokud bychom chtěli větší přesnost a výdrž, přidává se teplotní kompenzace za cenu velikosti zařízení.

Protože moje makro zapisuje aktuální datum a čas měření a Raspberry Pi RTC nemá, je potřeba zajistit, aby byl na zařízení přesný čas. Nabízejí se tři možnosti: První je ručně datum a čas nastavit po každém zapnutí. To je neefektivní a vyžaduje zásah uživatele do systémového nastavení. Druhá možnost je na desku RTC dodat, ale třetí možnost je podle mého názoru nejlepší a to je mít Raspberry připojené k internetu, to si pak automaticky načte čas ze serveru. Připojení k internetu je pro nějakou laboratoř, kde by se moje zařízení používalo, samozřejmostí a přinese kromě přesného času i další možnosti jako nahrávání nasbíraných dat na cloud nebo přímé odesílání dat zákazníkovi.

2.6 Komentování kódu

Moje praktická část ze značné části spočívala v napsání makra a skriptů, proto bych přidal do mé práce něco o komentování kódů.

Komentování je psaní poznámek ke kódu, tak aby bylo jasné, co se uvnitř programu děje. Text komentářů sám o sobě nemá vliv na běh programu. Psaní poznámek má mnoho důvodů, první z nich je snadné dohledání, co která část kódu dělá a zároveň hledání chyb nebo psaní úprav. Dalším důvodem je, když někdo jiný přečte váš kód, aby pochopil, k čemu slouží a co jednotlivé funkce dělají.

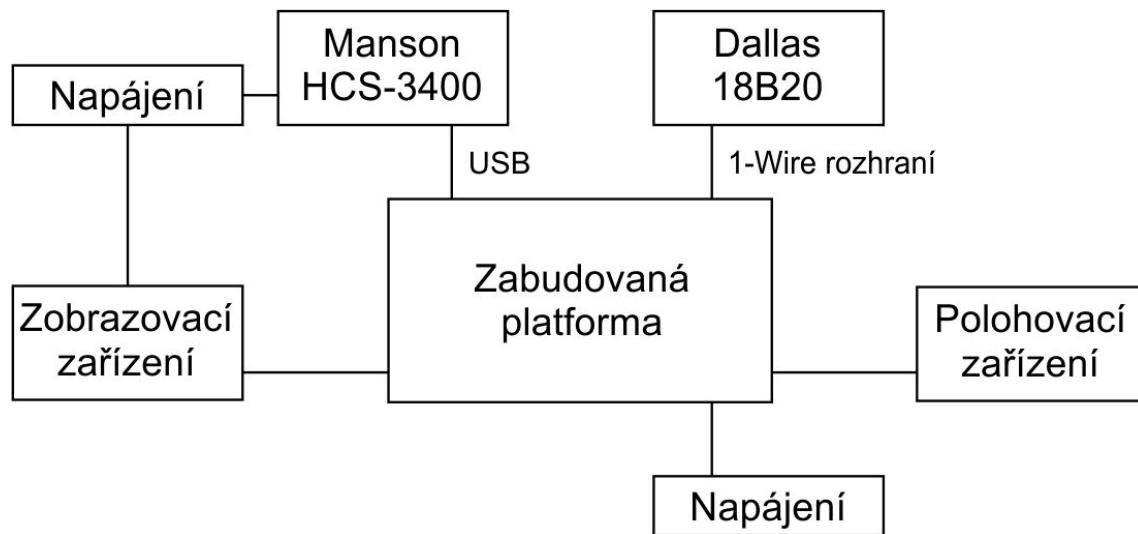
Komentování by mělo sloužit jako osnova, podle které se kód napíše, tím se zajistí, že výsledný program bude vykonávat vše, co má a nic v něm nebude chybět. Komentáře by měli být stručné a jasné a měli by se objevovat na začátku programu, kde stručně popíší, co aplikace dělá, dále nad funkcemi a nakonec případné poznámky pro složitější operace, kde není na první pohled jasné, co se děje. Je například zbytečné okomentovat „pocetBananu=5“, každý už od pohledu předpokládá, že počet banánů je 5.

To naráží i na problém pojmenování proměnných a funkcí. Pokud nazveme proměnnou „p“, nikdo neví, co obsahuje. Na druhou stranu název proměnné „teplota“ jasně říká, že je v proměnné uložena nějaká teplota. Programátor by se měl také vyvarovat psaní příliš dlouhých funkcí, které jsou nepřehledné a nikdo neví, jak vlastně fungují.

Shrnutí je, že pokud si někdo kód po nás přečte, mělo by mu být jasné co je cílem aplikace, co jednotlivé funkce dělají a případně jak upravit kód tak, aby ho mohl použít k řešení podobného problému. |3|

3 Realizační část

3.1 Návrh zapojení



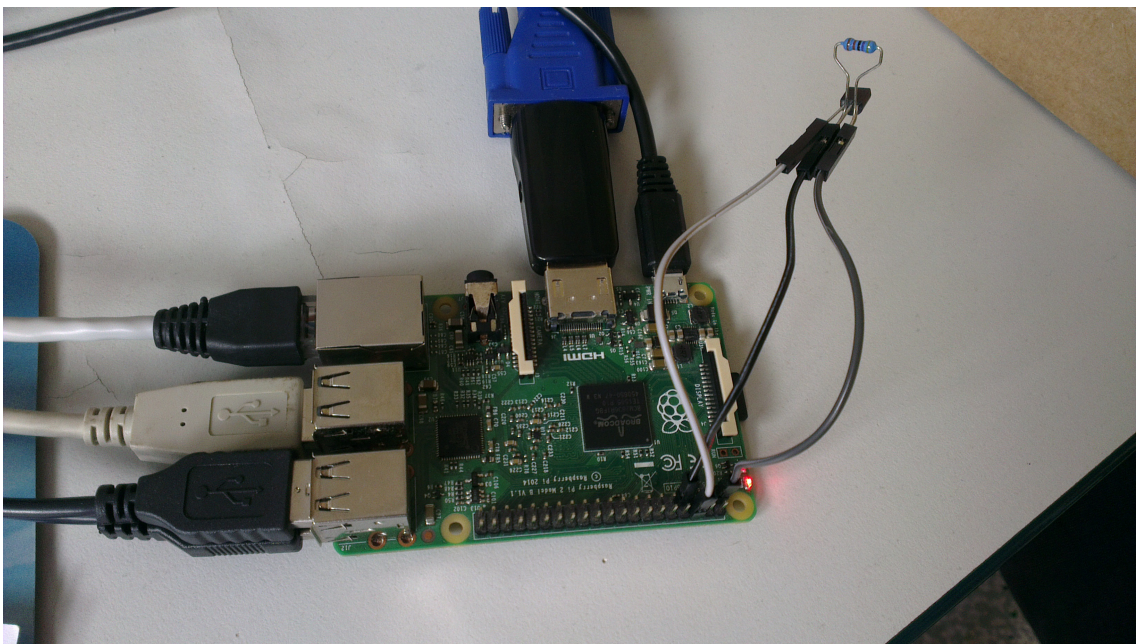
OBRÁZEK 4 - BLOKOVÉ SCHÉMA

Řešení jak sběr dat z přístrojů provést vedl na zabudované platformy. Zdroj Manson je samostatně napájený ze sítě a komunikace s ním probíhá přes USB rozhraní, teploměr Dallas má 1-Wire rozhraní s potřebou napájení 3,3V. To vedlo na platformu Raspberry, která všechny zadané periferie dokáže připojit respektive zdroj k USB portu a teploměr ke 3 GPIO pinům. Jako zobrazovací zařízení slouží počítačový monitor připojený přes rozhraní HDMI, jako polohovací zařízení jsou zvoleny myš a klávesnice připojené do USB portů a připojení k internetu je přes rozhraní Ethernet.

3.2 Raspberry Pi

Platformu Raspberry jsem zvolil ze dvou důvodů, první byl open source operační systém a druhý dostačující připojitelnou periferií.

Raspberry Pi je mikropočítač o rozměrech 9 x 5,5 x 2 cm (délka x šířka x výška), ke kterému stačí připojit klávesnici, myš a displej a je ho možné začít používat. Cenová kategorie je do 35\$, toto balení pak obsahuje pouze desku. Deska na mnou použitém Raspberry Pi 2 Model B kromě procesoru s vestavěnou grafickou kartou disponuje 1GB paměti, čtyřmi USB porty, HDMI výstupem, vstupem i výstupem MIPI neboli Mobile Industry Processor Interface (jedná se o displejový port pro připojení displeje nebo kamery), Ethernet vstupem a audio výstupem 3,5 mm. Navíc má deska ještě 40 GPIO neboli General-Purpose Input/Output pinů, ke kterým se dají připojit další zařízení. Zařízení podporuje mnoho operačních systémů jako je například Noobs, Ubuntu Mate nebo Windows 10 IoT Care. Já osobně použil Raspbian s Pixelem, který je založený na Debianu od Linuxu. Ten je volně dostupný a každý si ho může sám nainstalovat na SD kartu o velikosti alespoň 8 GB. [6]



OBRÁZEK 5 - ZAPOJENÉ RASPBERRY

3.3 Makro pro sběr dat

V této kapitole budu popisovat, jak a proč jsem makro pro zápis hodnot do tabulkového procesoru Calc od Apache LibreOffice udělal, a co jeho jednotlivé části dělají.

Do předem dané tabulky obsahující informace o čísle dne výroby, datu a času zápisu měření, dále proud a napětí z laboratorního zdroje, proudové množství, teplotu okolí, pH a případnou poznámku, kterou by chtěl uživatel napsat.

	A	B	C	D	E	F	G	H
1	Výrobní den	Datum a čas	Napětí [V] ±0.2%	Proud [A] ±0.2%	U [A·hod]	Teplota [°C] ±0.5°	pH [-]	Poznámka
2		124.05.2017 16:16:51	15,0	2,2	0	24	7	
3		124.05.2017 17:26:55	10,0	3,4	4	24	7	
4		225.05.2017 20:32:07	13,5	2,5	72	27	7	
5		326.05.2017 17:28:02	12,0	5,0	176	28	7	

OBRÁZEK 6 - TABULKA V CALCU

Po nezdařených pokusech načítat data z přístrojů přímo makrem jsem se rozhodl pro jinou možnost. Makro příkazem spustí externí skript [12] , který zapíše hodnoty do textového souboru, a z tohoto souboru se po dokončení skriptu data načtou do makra. [11] Takové řešení má i své výhody, protože se data uloží do souboru, odkud je možné dohledat naměřené hodnoty posledního měření. Samotné soubory se vytvářejí při každém spuštění skriptu a není tedy problém, když je uživatel omylem vymaže. Se zápisem data a hodnot z přístrojů do tabulky [2] na první volný řádek [5] nebyl problém. Jak spočítat kolikátý den výroby měření probíhá, vypadalo beznadějně, ale vše se nakonec vyřešilo objevem funkce [4] , která počítá počet dní mezi dvěma daty. Horší to bylo u proudového množství, které počítá, kolik Ampérhodin bylo spotřebováno od prvního měření. Funkci pro počítání počtu hodin mezi dvěma místy v čase jsem už musel vytvořit sám a vkládám ji dynamicky do příslušné buňky.

```
'pro první zápis je výrobní den 1, jinak vypočte rozdíl dnu mezi daty a přičte předchozí výrobní den
if Row = 1 Then
  Cell0.Value = 1
  CellJ.Value = 0
else
  datum = Sheet.GetCellByPosition(1, (Row-1)).GetString()
  Cell0.Value = DateDiff("d", datum, dnes) + Sheet.GetCellByPosition(0, (Row-1)).GetValue()
  'zaroven dosadí vzorec, který vypočítá proudové množství
  CellJ.formula = ("=E"+ (Row) + "+((((("A" + (Row+1) + "-A" + (Row) + ")))*24)+(HOUR(B" +
    (Row+1) + ") - HOUR(B" + (Row) + ")))+(MINUTE(B" + (Row+1) + ") - MINUTE(B" +
    (Row) + ")))/60))*D" + (Row+1) + ")")
end if
```

OBRÁZEK 7 - UKÁZKA FUNKCE PRO VÝPOČET PROUDOVÉHO MNOŽSTVÍ

V posledním kroku má makro ještě navíc změnit aktivní buňku do sloupce s pH, aby uživatel rovnou mohl začít psát bez vybírání příslušné buňky. Problém je ale v tom, že veškeré funkce řeší jaká buňka je aktivní a nikoho už nezajímá, jak by se změnila na jinou.

Po dlouhém hledání řešení se jedno naskytlo: použít příkaz „nahrát makro“ a rozluštit, co které části funkce dělají.

Aplikace ještě navíc automaticky ukládá otevřený soubor každých 30 minut, aby se data při výpadky proudu neztratila.

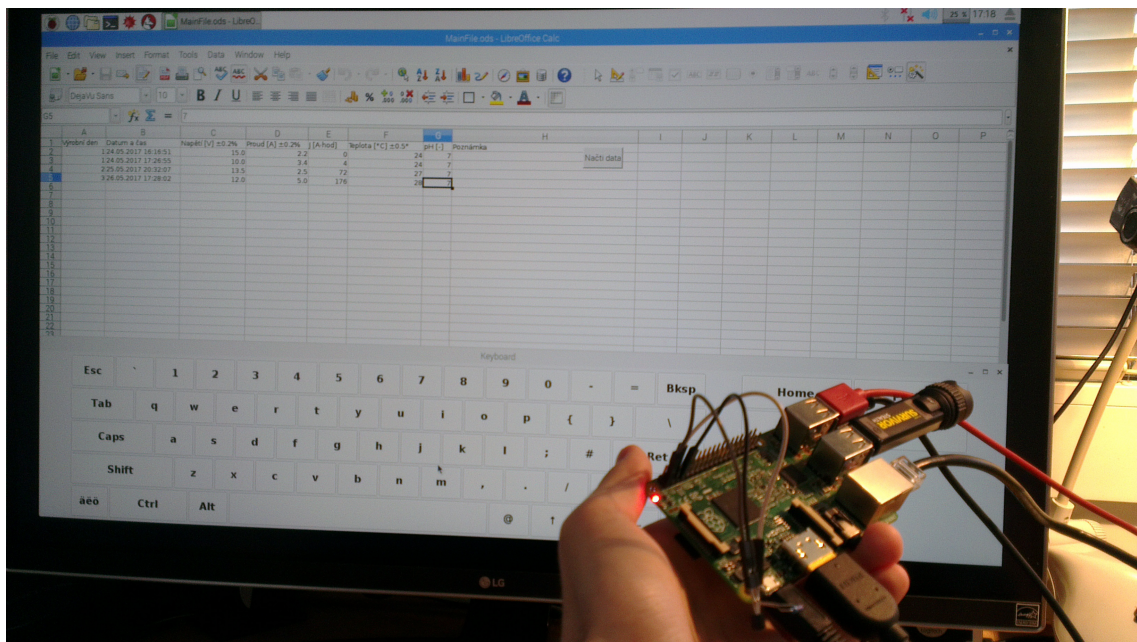
```
'zmena aktivni bunky do sloupce ph
Dim Dispatcher As Object
Dim Document As Object

Document = ThisComponent.CurrentController.Frame
Dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")

'parametry pro zmenu aktivni bunky
dim Args(0) as new com.sun.star.beans.PropertyValue
Args(0).Name = "ToPoint"
Args(0).Value = "§E§" + (Row + 1)

'samotna zmena aktivni bunky
Dispatcher.executeDispatch(Document, ".uno:GoToCell", "", 0, Args())
```

OBRÁZEK 8 - UKÁZKA ZMĚNY AKTIVNÍHO POLE V TABULCE



OBRÁZEK 9 - FINALNI VZHLED MAKRA

3.4 Laboratorní zdroj Manson HCS-3400-USB

Zdroj od firmy Manson, z kterého jsem potřeboval načíst informace zobrazené na displeji, je určený pro širokou škálu aplikací v telekomunikacích, laboratořích nebo průmyslu. Má své vlastní automatické chlazení, jemné nastavování proudu a napětí a tři nastavitelné presety pro snadné přepínání mezi často používanými hodnotami. Zdroj zároveň disponuje USB portem, kterým se dá připojit k počítači, z kterého se dá sériovou komunikací dále programovat, ovládat nebo z něj jen číst data. [16] pro správné fungování se nastavuje bitová rychlost, parita, ukončovací bit a počet přenášených bitů.



OBRÁZEK 10 - ZDROJ MANSON

Samotná komunikace s přístrojem je realizována tzv. AT příkazy, které sepsal v roce 1981 Dennis Hayes pro ovládání modemu [21]. Jde o sérii krátkých příkazů s různými funkcemi, každý příkaz je pak ukončen ASCII (znaková sada používaná v informatice) charakterem CR neboli Carriage Return, který je chápán jako konec zprávy. Podobnou sadu příkazů má i programovatelný zdroj Manson.

Input Command: GETD[CR] Return value: <voltage><current><status>[CR] OK[CR]	Get PS Display values of Voltage, Current and Status of CC/CV <voltage>=???? <current>=???? <status>=0/1 (0=CV, 1=CC)	Input command: GETD[CR] Return value: 150016001[CR] OK[CR] Meaning: The PS Display value is 15V and 16A. It is in CC mode.
--	--	--

OBRÁZEK 11 - AT PŘÍKAZ ZDROJE [16]

Nejvhodnější řešení jak se zdrojem navázat spojení, bylo využít modulu pySerial programovacího jazyku Python. Tento jazyk je již součástí operačního systému Rasbian a je podle mého názoru snadný na pochopení. Modul pySerial je specializován na práci a přístup k sériovému portu.

Pomocí funkcí Serial, ReadLine, Write a Close [14] jsem upravil kód [7] [16] ze stránek firmy Manson [16] aby odpovídal mým potřebám a přidal zápis dat do souboru, z kterého následně hodnoty načítám makrem v Calcu. Po zkoušce nastal problém s oprávněním, ale ten jsem vyřešil zvýšením pravomocí.

Skript je napsán tak, aby bylo možné ho použít pro jakoukoliv sériovou komunikaci po zadání správných parametrů portu a příslušného AT příkazu. Tyto parametry jsou napsány v dokumentaci k danému přístroji a pro zjištění názvu portu, na který je zařízení připojeno jsem připravil skript „DevPath.sh“ [13] , který je k nalezení v domovské složce Raspberry. Skript se proto dá jednoduchým přepsáním AT příkazu změnit ze získávání informací ze zdroje na ovládání zdroje. Když například změním příkaz GETD na VOLT100, tak se na zdroji nastaví napětí 10.0V.

3.5 Teploměr Dallas DS18B20

Je tří drátový (zem, data in/out a napájení) teploměr vracející 9 až 12 bitů, které odpovídají teplotě ve stupních Celsia s přesností 0,5°C a rozlišovací schopností až 0,0625°C pro 12 bitů. Hlavní vlastností teploměru je senzor s přímou konverzí do digitální podoby, který po obdržení signálu zapíše teplotu do vnitřní paměti a z ní je pak možné přečíst teplotu v jiném zařízení po jednom jediném drátě. Zařízení může a nemusí pracovat s externím zdrojem napětí, v případě bez zdroje pak pracuje z impulsy dobíjeného kondenzátoru. Tento způsob napájení ale omezí možnosti, co teploměr umí. Senzor má navíc unikátní 64 bitové sériové číslo, které umožní mít na jednom datovém drátu více zařízení najednou a všechna je mít rozlišena. [10]

Teploměr jsem zvolil díky jeho kompaktnosti a snadné realizaci komunikace s mojí platformou.

Protože GPIO piny na Raspberry Pi mají i možnost napájet externí zařízení, připojil jsem teploměr podle schématu k pinům 3,3V, GPIO 4 a GND, respektive napájení, datový pin a zem. Navíc bylo nutné dodat slabý pullup rezistor mezi napájení a datový pin, aby byla zajištěna vyšší hodnota logické 1.

Následně už stačí z konzole načíst připojená jednodrátová zařízení, a zjistit unikátní identifikační číslo teploměru a v příkazovém řádku (terminal) pomocí příkazu `cat` přečíst obsah našeho senzoru. Konzole zobrazí pár nepodstatných informací a na konci je naše teplota „t=26354“. To znamená, že senzor naměřil teplotu 26,354°C. [9]

Pro snadnější použití využívám opět jednoduchého Python skriptu [9], který načte data uložená na senzoru přes jednodrátové rozhraní GPIO pinů, převede je na desetinné číslo a uloží do souboru, který načítám makrem v LibreOffice.

```
5 #otevře soubor s daty teplomeru - 28-000... je seriove cislo daneho teplomeru
6 tempfile = open("/sys/bus/w1/devices/28-00000883521c/w1_slave")
7 #nacte data do stringu
8 tempdata = tempfile.read()
```

OBRÁZEK 12 - SKRIPT PRO ČTENÍ TEPLOTY

3.6 Dotykový displej

Externí klávesnice a myš mohou na pracovním místě zabírat příliš mnoho místa, proto jsem na doporučení vedoucího práce připojil k Raspberry monitor MicroTouch od firmy 3M s rozlišením 1280 na 1024 pixelů a kapacitním dotykovým rozhraním. [3] Kapacitní displej zároveň umožní ovládat monitor perem Stylus. Video signál je odesílán z HDMI výstupu Raspberry přes HDMI-VGA konvertor do monitoru a dotyky na obrazovce jsou odesílány přes USB rozhraní opačným směrem. [1]

Problém nastal hned po připojení, neboť dotyková oblast nebyla kalibrována, přesněji osy byly prohozené a oblast, kde se kurzor pohyboval, byla menší než rozměry obrazovky. Ovladače a podpora bohužel pro Raspberry není, a proto jsem se musel spolehnout na kalibrační funkce v operačním systému Raspbian. Zde nastal druhý problém, protože od lednové aktualizace jsou tyto funkce změněné. Po dalším hledání jsem našel aktuální znění a stránky s dokumentací a návodem, jak nové funkce použít. Tady nastal třetí problém, dokumentace popisovala syntaxi, ale už neřekla, co která z devíti proměnných změní a návod pro kalibraci byl absolutně nefunkční. Takže jsem se smířil s osudem a postupným zkoušením měnit jednotlivé proměnné jsem zjistil, co která přibližně dělá a nastavit potřebné hodnoty. Finální kalibrační matice pak vypadá takto: (0; -1,36; 1,18; -1,14; 0; 1,23; 0; 0; 1). Finální problém se objevil po restartu Raspberry, kalibrace se resetovala. Tento problém se mi zatím nepodařilo vyřešit. A je tedy pořád pohodlnější používat klávesnici a myš.

Souvisí s ním i moje snaha automaticky spustit po nastartování systému tabulku s makrem. Jak pro kalibraci, tak pro nastartování mám připravené skripty. Ty jsem přidal do souboru, který je určený ke spouštění skriptů při startu, ale ty se z neznámého důvodu nespustí. Když soubor spustím ručně z terminálu, vše proběhne, jak má.

Dotykový displej odstranil potřebu pro myš. Klávesnice je „on display“ dodána aplikací Matchbox Keyboard. [8]

4 Závěr

V mojí práci jsem pohovořil o zabudovaných systémech, a proč jsem si vybral platformu Raspberry Pi. O tomto mikropočítači jsem podal základní informace, a jak a které periferie jsem k němu připojil. Podrobněji jsem se zaměřil na dotykový displej a jeho kalibraci a na dva přístroje, z kterých jsem získával naměřená data. Jedním z nich byl laboratorní zdroj, který pro výměnu dat používá sériovou komunikaci přes USB rozhraní. Druhým přístrojem byl teplotní senzor připojený k jednodrátovému rozhraní Raspberry. U obou přístrojů jsem rozebral, jak jsem provedl získání dat a jak příslušná komunikace funguje. Kromě toho jsem se zaměřil na problém získání aktuální času na zařízení a správného komentování kódu.

Hlavním cílem práce bylo vytvořit open source sběr dat makrem do tabulkového procesoru z laboratorního zdroje a teplotního senzoru. Celý systém shromažďování dat probíhá na open source operačním systému Raspbian na platformě Raspberry. Makro jsem napsal v open source programu Calc od LibreOffice. Veškeré funkce, které makro mělo vykonávat, se mi plně povedlo realizovat, uživatel pak makro spustí stiskem tlačítka přímo v tabulkovém procesoru Calc. Protože se mi nepodařilo získávat data přímo, makro spouští skript a předává si data přes textový soubor. Jak u laboratorního zdroje, tak u teploměru je to realizováno Python programem, který u laboratorního zdroje zahájí sériovou komunikaci a u teploměru načte data přes jednodrátové rozhraní. Zadání jsem podle mého názoru splnil.

Vhodným pokračováním práce by bylo dokončit implementaci dotykového displeje a automatizace zapínání souboru s makrem. Práce by se dala dál rozšiřovat o měření pH nebo dalších veličin díky GPIO pinům na platformě Raspberry Pi. Dále by se zautomatizovalo zálohování na cloud, odkud by si mohl zákazník v reálném čase kontrolovat průběh měření. Posledním a nejpracnějším pokračováním této práce by asi bylo vzdálené ovládání Raspberry z vnější sítě. K desce by se navíc připojila kamera k rozhraní MIPI, aby měl pracovník video přenos z laboratoře a skript k laboratornímu zdroji by se obohatil o nové příkazy, které by zdroj umožnili i ovládat.

4.1 Seznam použité literatury

- |1| 3M *MicroTouch Display*. (2009). Získáno 25. 5 2017, z 3M:
<http://multimedia.3m.com/mws/media/249468O/3mtm-microtouch-display-m1700ss-product-sheet.pdf>
- |2| *Cells and Ranges*. (15. 3 2010). Získáno 19. 5 2017, z OpenOffice Wiki:
https://wiki.openoffice.org/wiki/Documentation/BASIC_Guide/Cells_and_Ranges
- |3| *Commenting*. (nedatováno). Získáno 21. 5 2017, z The University of Utah:
<http://www.cs.utah.edu/~germain/PPS/Topics/commenting.html>
- |4| *Date and Time Functions*. (9. 2 2012). Získáno 24. 5 2017, z LibreOffice Help:
https://help.libreoffice.org/3.4/Calc/Date_and_Time_Functions
- |5| FJCC. (1. 9 2014). *Find next blank row in wb*. Získáno 19. 5 2017, z OpenOffice Forum: <https://forum.openoffice.org/en/forum/viewtopic.php?f=20&t=71838>
- |6| FOUNDATION, R. P. (nedatováno). *Dokumentace a často kladené dotazy*. Získáno 5. 3 2017, z <https://www.raspberrypi.org/>
- |7| Kevin. (2. 12 2016). *Asking the user for input until they give a valid response*. Získáno 19. 5 2017, z stackoverflow: <http://stackoverflow.com/questions/23294658/asking-the-user-for-input-until-they-give-a-valid-response>
- |8| *Matchbox Keyboard - Raspberry Pi Touchscreen Keyboard*. (11. 9 2015). Získáno 24. 5 2017, z Mod my Pi: <https://www.modmypi.com/blog/matchbox-keyboard-raspberry-pi-touchscreen-keyboard>
- |9| Matt. (15. 3 2013). *Raspberry Pi 1- Wire Digital Thermometer Sensor (DS18B20)*. Získáno 20. 5 2017, z Raspberry Pi Spy: <http://www.raspberrypi-spy.co.uk/2013/03/raspberry-pi-1-wire-digital-thermometer-sensor/#prettyPhoto>
- |10| MAXIM, D. S. (nedatováno). *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. Získáno 20. 5 2017, z GM Electronic:
<https://www.gme.cz/data/attachments/dsh.530-067.1.pdf>
- |11| *Open Statement [Runtime]*. (18. 6 2016). Získáno 19. 5 2017, z LibreOffice help:
https://help.libreoffice.org/Basic/Open_Statement_Runtime
- |12| *Other Functions*. (22. 11 2009). Získáno 15. 5 2017, z OpenOffice Wiki:
[https://wiki.openoffice.org/wiki/Documentation/BASIC_Guide/Other_Functions_\(Runtime_Library\)](https://wiki.openoffice.org/wiki/Documentation/BASIC_Guide/Other_Functions_(Runtime_Library))

- [13] Patrick. (14. 7 2014). *command to determine port of device*. Získáno 19. 5 2017, z Unix and Linux: <https://unix.stackexchange.com/questions/144029/command-to-determine-ports-of-a-device-like-dev-ttyusb0>
- [14] *pySerial*. (nedatováno). Získáno 19. 5 2017, z PythonHosted: <http://pythonhosted.org/pyserial/shortintro.html>
- [15] *Real-Time Clocks*. (2017). Získáno 19. 5 2017, z Maxim Integrated: <https://www.maximintegrated.com/en/products/digital/real-time-clocks.html>
- [16] *Remote Programming Lab.Grade Switching Mode Power Supply*. (nedatováno). Získáno 19. 5 2017, z Manson: <http://www.manson.com.hk/products/detail/158>
- [17] *Role of Supercapacitor*. (21. 4 2017). Získáno 19. 5 2017, z Battery University: http://batteryuniversity.com/learn/article/whats_the_role_of_the_supercapacitor
- [18] Rouse, M. (12 2016). *embedded system*. Získáno 2017. 5 19, z IoT Agenda: <http://internetofthingsagenda.techtarget.com/definition/embedded-system>
- [19] *Serial Communications*. (nedatováno). Získáno 19. 5 2017, z Swarthmore College: [https://www.swarthmore.edu/NatSci/echeeve1/Class/e91/Lectures/E91\(10\)Serial.pdf](https://www.swarthmore.edu/NatSci/echeeve1/Class/e91/Lectures/E91(10)Serial.pdf)
- [20] Spreadsheet. (17. 5 2017). Získáno 19. 5 2017, z Wikipedia: <https://en.wikipedia.org/wiki/Spreadsheet>
- [21] *The Modem of Dennis Hayes and Dale Heatherington*. (nedatováno). Získáno 19. 5 2017, z History Computer: <http://history-computer.com/ModernComputer/Basis/modem.html>
- [22] *W example*. (nedatováno). Získáno 26. 5 2017, z Learning MSP: https://learningmsp430.files.wordpress.com/2014/01/w_example.png
- [23] *Embedded Platforms: Trends and Products of 2017 by Michael Parks*. (nedatováno). Získáno 26. 5 2017, z Mouser Electronics: <http://eu.mouser.com/applications/embedded-platform-trend-products/>
- [24] *1-Wire*. (6. 1 2017). Získáno 26. 5 2017, z Wikipedia: <https://en.wikipedia.org/wiki/1-Wire>

4.2 Seznam obrázků

OBRÁZEK 1 - RASPBERRY PI 2 MODEL B	13
OBRÁZEK 2 - LIBREOFFICE CALC	15
OBRÁZEK 3 - PŘENOS CHARAKTERU PO LINCE 22 	16
OBRÁZEK 4 - BLOKOVÉ SCHÉMA.....	20
OBRÁZEK 5 - ZAPOJENÉ RASPBERRY.....	21
OBRÁZEK 6 - TABULKA V CALCU	22
OBRÁZEK 7 - UKÁZKA FUNKCE PRO VÝPOČET PROUDOVÉHO MNOŽSTVÍ	22
OBRÁZEK 8 - UKÁZKA ZMĚNY AKTIVNÍHO POLE V TABULCE	23
OBRÁZEK 9 - FINALNI VZHLED MAKRA	23
OBRÁZEK 10 - ZDROJ MANSON.....	24
OBRÁZEK 11 - AT PŘÍKAZ ZDROJE 16 	24
OBRÁZEK 12 - SKRIPT PRO ČTENÍ TEPLoty	26