

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

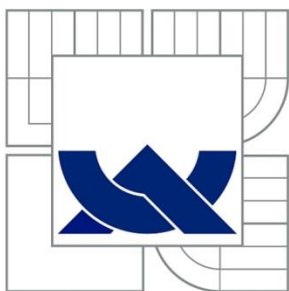
SÍŤOVÝ DATALOGGER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

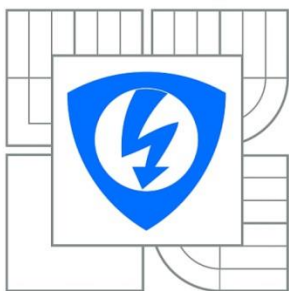
DANIEL MAŠTÁLKA

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

SÍŤOVÝ DATALOGGER

NETWORK DATALOGGER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL MAŠTÁLKA

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PAVEL ŠTEFFAN, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav mikroelektroniky

Bakalářská práce

bakalářský studijní obor
Mikroelektronika a technologie

Student: Daniel Maštálka
Ročník: 3

ID: 146056
Akademický rok: 2013/2014

NÁZEV TÉMATU:

Sít'ový datalogger

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte sít'ový teplotní datalogger, který bude zaznamenávat vnitřní a venkovní teplotu a vlhkost vzduchu. Návrh realizujte na procesoru s architekturou ARM a svou volbu zdůvodněte. Měřené teploty zobrazujte na displeji a ukládejte na webový server, tak aby je bylo možné vzdáleně sledovat a vyhodnocovat.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 10.2.2014

Termín odevzdání: 5.6.2014

Vedoucí práce: doc. Ing. Pavel Šteffan, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Háze, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

MAŠTÁLKA, D. *Síťový datalogger* . Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 40 s. Vedoucí bakalářské práce doc. Ing. Pavel Šteffan, Ph.D..

Abstrakt

Práce pojednává o teoretickém návrhu a realizaci zařízení pro záznam vnitřních a vnějších teplot a vlhkostí na vzdálený server. Jako mikrokontrolér je použit vývojový kit STM32 Value Line Discovery a pro komunikaci se serverem využívá protokol Ethernet. Primární zobrazení je pomocí vzdáleného přístupu k serveru, sekundární pak na alfanumerickém displeji.

Abstract

Work is about theoretical proposal of network datalogger, which logs indoor and outdoor temperature and humidity. As microcontroller is used development kit STM32 Value Line Discovery and for communication with server is used Ethernet protocol. As primary display is used remote access to server, as secondary alphanumeric liquid crystal display.

Klíčová slova

ARM, STM32, DHT-22, Měření teploty, Měření vlhkosti, Ethernet

Keywords

ARM, STM32, DHT-22, Temperature measurement, Humidity measurement, Ethernet

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma „**Sítový datalogger**“ jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne 13. prosince 2012

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Pavlu Šteffanovi, Ph.D. za pomoc při návrhu zařízení, odbornou pomoc a za mnoho cenných rad při zpracování projektu.

V Brně dne 27. Května 2014

.....
podpis

Obsah

Abstrakt	5
Abstract	5
Klíčová slova.....	5
Keywords	5
Prohlášení	6
Poděkování.....	6
Úvod.....	9
1 Popis komponent síťových dataloggerů.....	10
1.1 Mikrokontrolér.....	10
1.1.1 Architektura ARM.....	10
1.1.2 Cortex-M3	10
1.1.3 Vývojový kit.....	12
1.2 Periferie	13
1.2.1 Teplotní a vlhkostní čidla	13
1.2.2 Digitální čidla	15
2 Vlastní vylepšení a realizace	17
2.1 Teoretická funkce	17
2.1.1 Komunikační rozhraní.....	18
2.2 Teoretická realizace	19
2.2.1 Návrh schéma a DPS.....	19
2.2.2 Návrh serveru	22
3 Praktická realizace.....	24
3.1 Programové vybavení	24
3.1.1 Maple Leaf Labs.....	24
3.1.2 Programátor STM32 ST-LINK Utility.....	25
3.2 Programování.....	26
3.2.1 STM-32 Value Line Discovery	26
3.2.2 Webový server.....	28
3.2.3 Databázový server	31

3.3	Výroba záznamníku	32
3.3.1	Výroba desek plošných spojů	32
3.3.2	Výroba krabičky	34
3.4	Seznam součástek	36
4	Závěr	37
	Seznam obrázků	38
	Seznam tabulek	38
	Seznam zkratek	39
	Seznam použitých zdrojů	40

Úvod

Měření teploty je v dnešní době velice důležité a na přesnost měření jsou kladeny stále vyšší nároky. S rostoucími nároky na přesnost měření však roste cena. A přesné přístroje pro měření v amatérských laboratořích či domácnostech, jsou velice nákladné. Proto vzniká potřeba pro vytvoření levné, ale přesné měřicí techniky, která by byla dostačující i pro malé laboratoře.

Vysoká cena komerčních měřících přístrojů a záznamníků je dána především vysokým množstvím periférií. Je tedy nutné eliminovat možnosti připojení pouze na ty, které jsou moderní a často používané. Z výčtu periférií se nejvýhodnějším připojením zdá být Ethernet, jež má ve většině budov již velmi dobrou infrastrukturu. Proto je připojení přes toto rozhraní velmi výhodné a vyžaduje jen minimální zásahy. Dále je nutné zařídit zobrazování informací. Pro základní informace o aktuálních teplotách slouží vestavěný displej. Pro pokročilé informace je již nutné využití počítače. Zde komerční záznamníky využívají jednoduchých vestavěných serverů, které mají řadu nevýhod. Jsou to především rychlost a jednoduchost zobrazení. Rychlost zobrazení je nevýhodou, pokud k záznamům přistupuje více uživatelů zároveň a kompenzovat ji lze lepšími procesory, které zvyšují cenu. Jednoduchost zobrazení je nevýhodou převážně při zobrazování velkého množství naměřených hodnot, kdy na základních vestavěných serverech nelze spouštět pokročilé skripty. Tyto nevýhody lze eliminovat i bez nároků na cenu. A to přesunutím serveru mimo záznamník. Na záznamník tedy nikdo nepřistupuje a veškerá data se nacházejí na vzdáleném serveru, který může být buď v LAN nebo někde v celé síti internetu. A veškerá záloha probíhá na vzdálený server. Díky tomu lze ušetřit i na velikosti lokálního uložení.

1 Popis komponent síťových dataloggerů

1.1 Mikrokontrolér

1.1.1 Architektura ARM

ARM je rodina 32-bitových nebo 64-bitových RISC procesorů. První procesory vznikly v Bristolské společnosti Acorn Computers, která byla založena roku 1978. Původní představa byla vytvořit osobní počítač, s procesorem využívajícím redukovanou instrukční sadu. Rozmach však přišel až roku 1983, kdy architekturu začala vyvíjet firma Advanced RISC Machines, dnes ARM Holding. Již v roce 2007 se procesory ARM nacházeli v 98% prodaných mobilních telefonů.

Tato firma v dnešní době nevyrabí žádné procesory. Vyvíjí však nové technologie, které poté licencuje různým výrobcům. Z těch známých například Apple, Samsung, Texas Instruments nebo STMicroelectronics.

Procesory ARM však neobsahují pouze CPU, nýbrž celý funkční systém a jsou tedy nazývány jako SOC (Systém on chip). V jednom pouzdru je tedy umístěno jak CPU tak i GPU, RAM a další vstupně výstupní periferie, které se liší podle určení a výrobce.

Procesory jsou pak rozděleny podle použité verze jádra například dnes nejvíce používaný ARMv7. Dále je dělení pro různé aplikace, kdy jádra s přívlaskem R jsou určeny pro náročné Real-Time aplikace a přívlaskem M určuje jádro pro levné časově nenáročné aplikace.

1.1.2 Cortex-M3

Procesory řady Cortex-M3 jsou založeny na ARMv7 architektuře. Je to 32-bitový SOC. Systém ke kterému se vztahují všechna data je STM32F100RB.

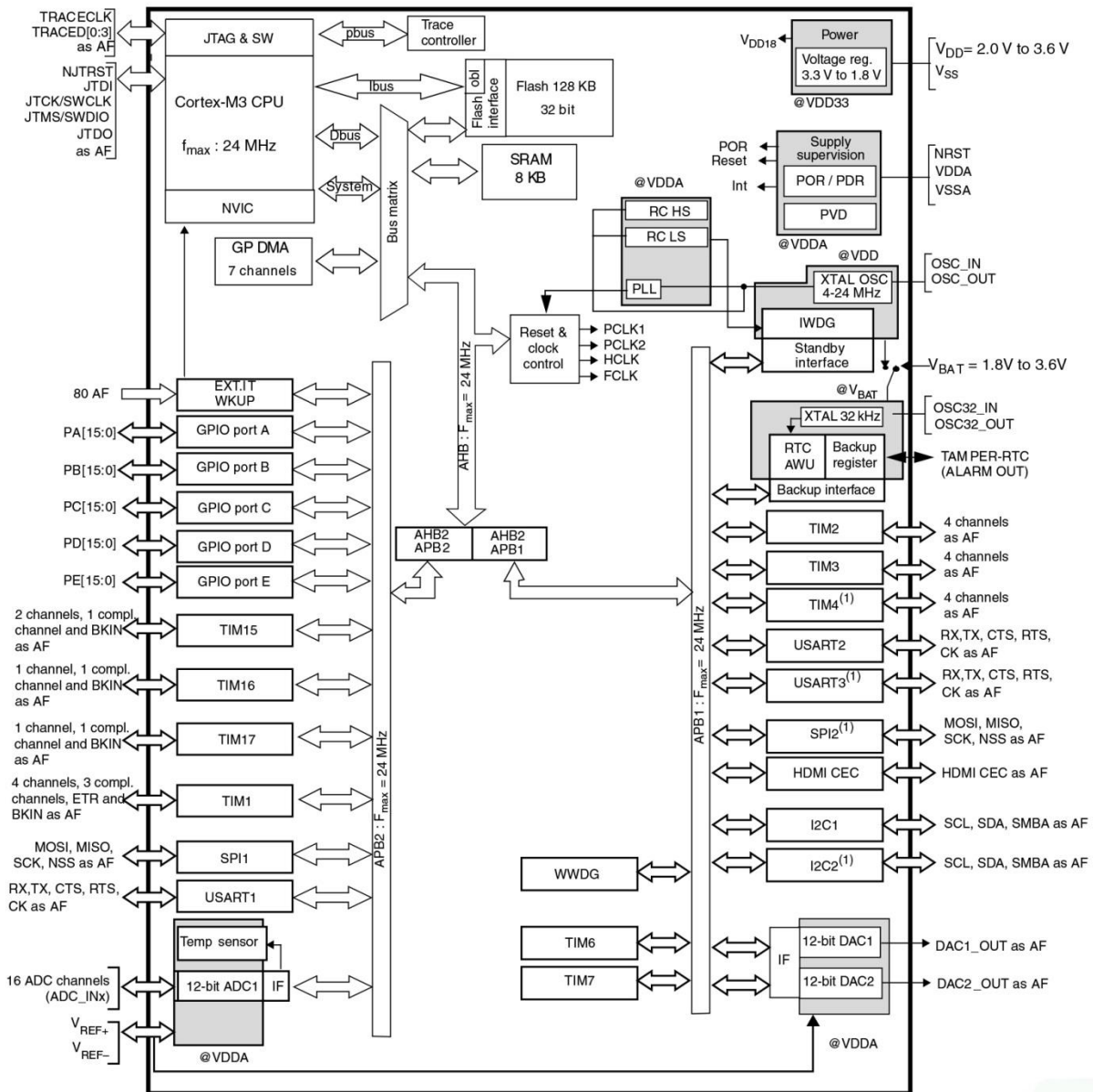
Tab. 1 - Parametry procesoru STM32F100RB [10]

Architektura	Přerušení	Výkon [DMIPS/MHz]	Výrobní proces	Spotřeba
ARMv7-M	1-240 fyzických přerušení	1,25	90 nm	32 μW/MHz

Tento systém obsahuje následující vlastnosti:

- Dvakrát SPI s maximální rychlostí 12 Mbit/s
- Dvakrát I²C
- 12-bitový A/D převodník, dvakrát 12-bitový D/A převodník
- Obsahuje zabudovaný obvod reálného času
- Třikrát USART
- Pracovní frekvence 24MHz

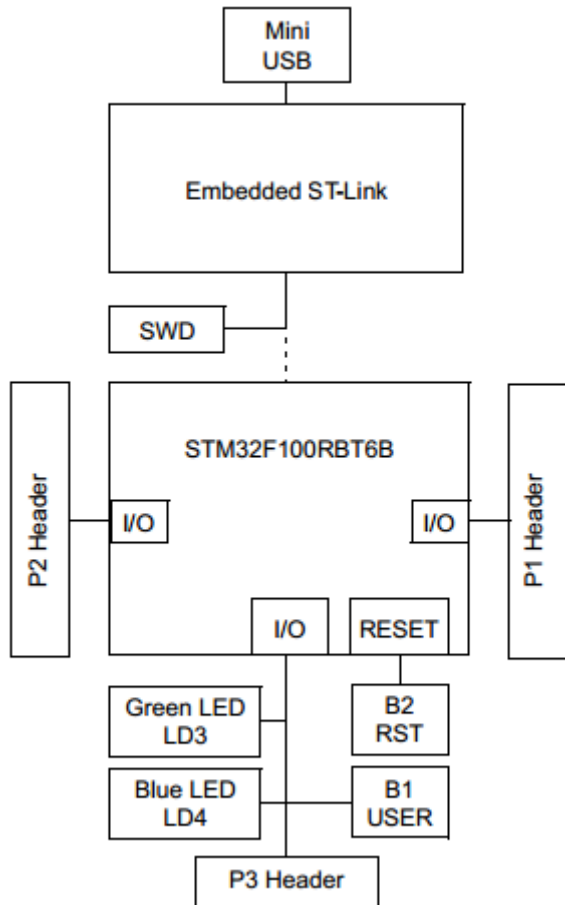
Mikrokontrolér je dostačující pro jednoduché operace a může být rychlejší a stabilnější náhradou za AVR mikrokontrolér. Obsahuje veliké množství periférií a portů to je možné vidět na Obr. 1



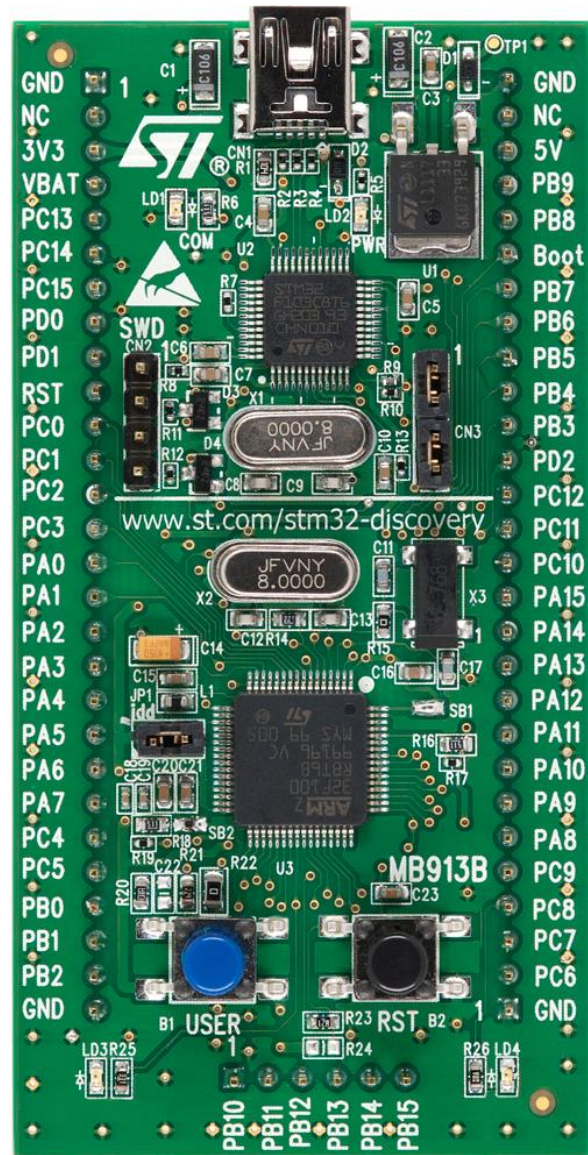
Obr. 1- Blokové schéma mikrokontroléru STM32F100RB6B [10]

1.1.3 Vývojový kit

Pro aplikaci v síťovém záznamníku teploty a vlhkosti byl pro cenovou dostupnost použit vývojový kit STM32 Value Line Discovery (viz Obr. 3). Tento vývojový kit je nejlevnější varianta jak začít s architekturou ARM. Kit byl vybrán zejména z důvodu problémové výroby desky plošných spojů pro puzdra LQFP64 v amatérských podmínkách. Další výhodou tohoto kitu je programování pomocí USB s ST-Link převodníkem integrovaným na desce (viz Obr. 2).



Obr. 2 - Blokové schéma STM32 VL Discovery [11]



Obr. 3 - Vývojový kit STM 32 VL Discovery [11]

Velkou výhodou tohoto kitu je tedy maximální integrace periférií na desce. Tím se minimalizuje cena výsledného produktu a není nutné rozšiřovat desku například o programátor a debugger.

Pro připojení periférií, které nejsou obsaženy na desce, je možné využít portů zmíněných v kapitole Cortex-M3. Tyto porty jsou vyvedeny na postranní piny. Nevýhodou tohoto kitu je nemožnost ho umístit do kontaktního pole, jelikož je zde vyveden krátký port P3. Kit lze tedy umístit pouze za použití dvou kontaktních polí, v tomto případě nelze použít

port P3 na který je vyveden jeden SPI port a jeden I²C port. V testovacím zapojení je tedy nutné počítat s tímto omezením.

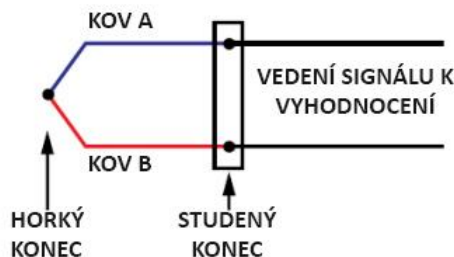
1.2 Periferie

1.2.1 Teplotní a vlhkostní čidla

Teplotní čidla

Teplotní čidla se globálně dělí na analogová a na digitální. Každé z těchto dvou kategorií mají svoje výhody. U analogových je to především široký pracovní rozsah, který je mnohdy až do 2300°C. Avšak je zde nutnost jejich kalibrace přímo na zařízení. U digitálních čidel se toto dá vyřešit kalibrací přímo v čidle, jelikož čidlo obsahuje analogový snímač teploty, A/D převodník a mikroprocesor, který obstarává vyhodnocení a odeslání dat.

Analogová čidla lze dále rozdělit na termočlánky a odporová teplotní čidla. Termočlánek vznikne spojením dvou kovů do série, kdy vznikne horký a studený konec (viz Obr. 4). Horký konec je ten, který je v kontaktu s měřeným objektem.



Obr. 4 - Popis termočlánku



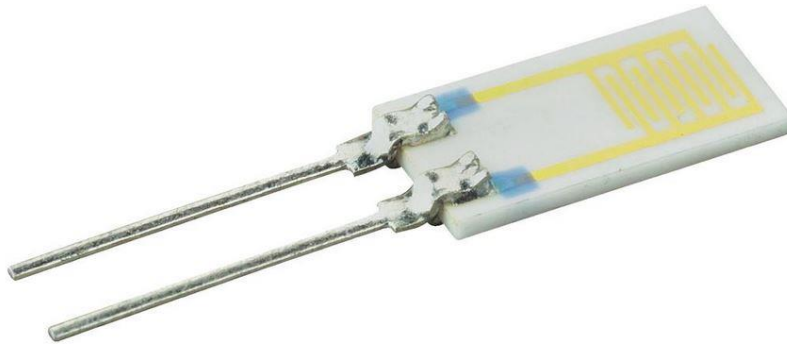
Obr. 5 - Závislost odporu na teplotě u čidla PT1000 [2]

Tímto spojením vznikne na studeném konci rozdíl potenciálu na jednotlivých vodičích. Napěťový potenciál je v řádu desítek $\mu\text{V/K}$. Díky tomu se v koncových zařízeních využívá většího počtu termočlánků zapojených do série, čímž vznikne větší napěťový potenciál. Ten se poté vede do A/D převodníku a do MCU pro vyhodnocení.

Další zástupce jsou odporová teplotní čidla. Nejznámější teplotní čidlo je platinové. Vyrábí se více verzí podle odporu při 0 °C. Označení je například PT1000, kdy PT označuje platinu a 1000 označuje, že při 0 °C má teplotní čidlo odpor 1000 Ω . Největším problémem u těchto čidel je jejich kalibrace, průběh závislosti odporu na teplotě vykazuje nelinearitu, tu je možné pozorovat na Obr. 5. Při kalibraci však musíme počítat s odporem vedení a nesmí být překročen maximální proud teplotním čidlem. Ten by ohříval teplotní čidlo a měření by nebylo přesné.

Vlhkostní čidla

V praxi se nejčastěji používá odporových čidel. Jedná se o tenkou vrstvu hyroskopického materiálu naprášeného na keramickém substrátu opatřeného vývody (viz Obr. 6). Se změnou vlhkosti se mění výsledný odpor nanesené odporové vrstvy. Tato čidla jsou velice přesná řádově desetiny procent relativní vlhkosti. Avšak na velmi úzkém spektru rozsahu. Pro celkový rozsah 0 – 100 %RH se přesnost snižuje a výsledná přesnost jsou jednotky %RH.



Obr. 6 - Odporové vlhkostní čidlo EFS 10, 0 - 60%RH

Pro kalibraci těchto čidel a pro použití v laboratorních podmínkách se využívá gravimetrie (viz Obr. 7). Kdy sušidlo absorbuje vodní páru ze známého objemu vzduchu a zjišťuje se celková hmotnost. Toto měření je velice zdlouhavá, zvláště ve spodní oblasti měřeného rozsahu. Dále je tento přístroj rozměrný.



Obr. 7 - Příklad měření pomocí gravimetrie - Sušící váhy [8]

1.2.2 Digitální čidla

V dnešní době se nejčastěji používá digitálních senzorů pro měření. Digitální čidlo se skládá z čidla analogového, A/D převodníku a MCU pro odesílání a příjem dat. Ve své podstatě jsou tedy tato čidla omezena na vlastnosti analogových čidel. Přidávají však jistou možnost kalibrace přímo v čidle, zapsáním kalibračního koeficientu do paměti MCU. Nespornou výhodou pak je teoretická neomezenost délky vedení a u mnohých také možnost umístit více senzorů na jedno vedení. Při délce vedení závisí především na její kapacitanci, která zaobluje jednotlivé pulsy posílané zařízením a ty jsou poté nečitelné.

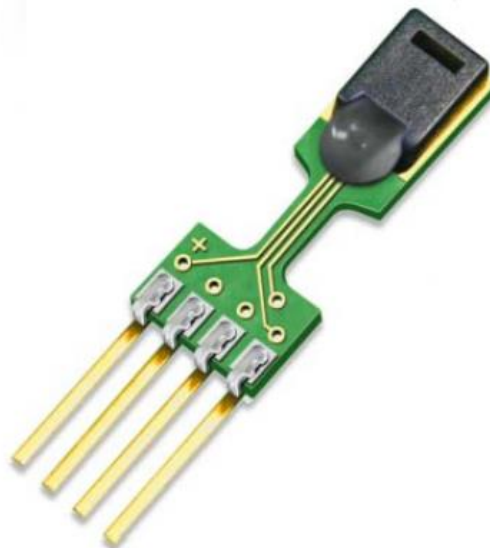
Další výhodou je umístění více senzorů do jednoho pouzdra. V tomto případě pak lze s jedním čidlem v jednom čtení zjistit jaká je například teplota i relativní vlhkost, z toho lze spočítat rosný bod a jedno čidlo je tak schopno obstarat tři veličiny. Velkou nevýhodou takovýchto teplotních čidel je jejich rozsah, jelikož obsahují řídicí elektroniku jejíž maximální teplota je okolo 125 °C, dle typu elektroniky. Je horní rozsah omezen právě na tuto hodnotu. S nejnižší teplotou měření je to stejné.

Tab. 2 - Srovnání digitálních čidel [1] [6] [9]

	DS18B20	DHT-22	SHT-75
Rozsah teplot [°C]	-55 až +125	-40 až +125	-40 až +123,8
Přesnost teploty[°C]	0,5	0,2	0,4
Rozlišení teploty[°C]	0,0002	0,1	0,01
Rozsah vlhkosti[%RH]	-	0-100	0-100
Hystereze vlhkosti[%RH]	-	0,3	1
Přesnost vlhkosti[%RH]	-	2	2
Dlouhodobá stabilita vlhkosti[%RH/Rok]	-	0,5	0,5
Doba pro získání veličiny[s]	0,75	2	-
Spotřeba[mA]	4	2,5	0,55
Cena[Kč]	55,-	233,-	1510,-

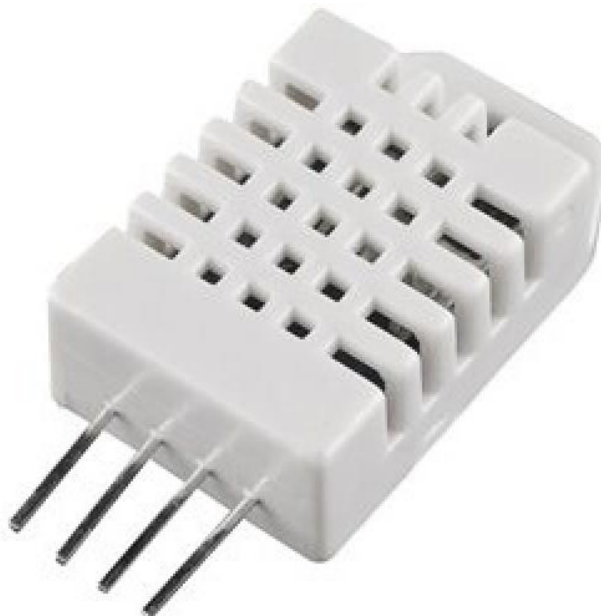
Z této tabulky je vidět, že digitální čidla jsou každé na jiné úrovni jak parametry tak i cenové. Nejlépe je to vidět na srovnání čidel DHT-22 a SHT-75, obě dokáží měřit stejné veličiny a mají přibližně stejné parametry. Čidlo SHT-75 umožňuje kontinuální měření teploty i vlhkosti avšak není to vhodné vzhledem k zahřívání čidla a výsledné nepřesnosti měření. V praxi se tak tato dvě čidla vyrovnají i v této hodnotě. Rozdíl v ceně je však markantní.

Čidlo SHT-75 (viz), jehož výrobcem je Sensirion, obsahuje jejich vlastní elektroniku a komunikace probíhá podobně jako I²C, kontinuální měření tento senzor ohřeje i o 5 °C. Je tedy nutno ošetřit aby měření probíhalo v intervalu přibližně 5s, který je pro toto čidlo bezpečný.

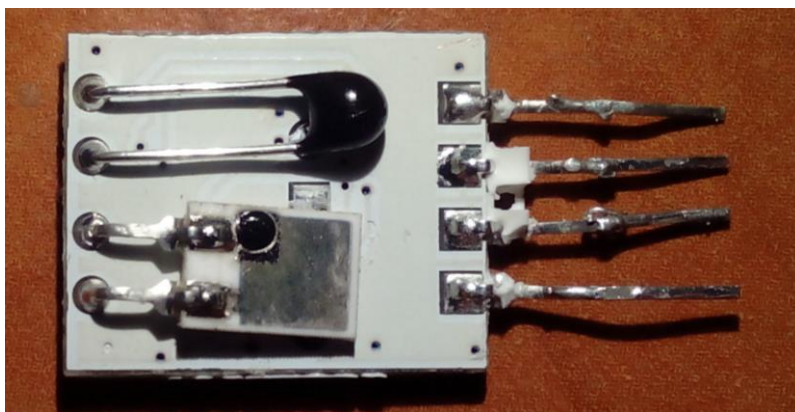


Obr. 8 - Senzor SHT-75 [9]

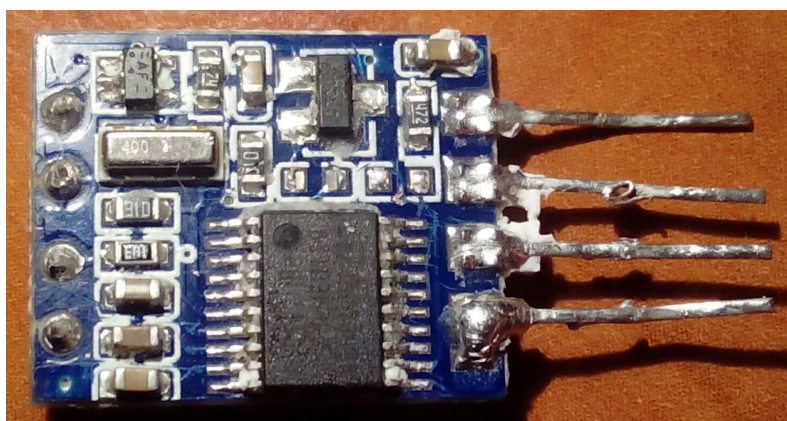
Další z řady čidel je DHT-22. Je to digitální čidlo, které v sobě ukrývá další digitální čidlo a to již zmíněné DS18B20, které samostatně má nevýhodu zdržování programu, jelikož ten musí čekat 700ms než bude čidlo připraveno. Tento neduh je u DHT-22 vyřešen a při čtení z čidla program může pokračovat, informace však je na čidle připravena až za dvě vteřiny. Je zde také minimalizována nepřesnost teplotního čidla a to kalibračním koeficientem, který je nahrán v ROM paměti procesoru. Dále toto čidlo obsahuje odporový senzor vlhkosti



Obr. 9 - Digitální teplotní a vlhkostní čidlo DHT-22 [1]



Obr. 10 Teplotní čidlo DHT22 strana čidel



Obr. 11 Teplotní čidlo DHT22 strana elektroniky

2 Vlastní vylepšení a realizace

2.1 Teoretická funkce

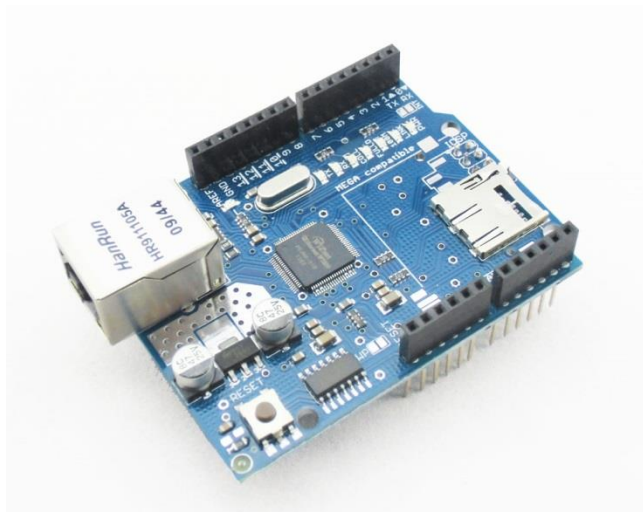
Síťový záznamník postavený na kitu STM32 Value Line Discovery. Záznam probíhá primárně na vzdálený server, který slouží zároveň pro zobrazení hodnot a vykreslení grafů. To je z důvodu odlehčení procesoru při používání v podmínkách, kdy je možné, že na síťový záznamník bude přistupovat více uživatelů zároveň. Díky tomu lze ušetřit spoustu peněz za rychlý mikrokontrolér a rychlou síťovou kartu. Toto řešení je velice ojedinělé a díky němu je cena výsledného produktu přibližně pětina funkčně adekvátních zařízení.

Ke správné funkci zařízení je však nutný webový server s PHP a SQL databází. Ten je buď možné umístit v LAN i WAN. Pokud se tedy v budově, kde bude záznamník aplikován, nenachází webový server. Je zde možnost založit webový server zdarma někde v internetu. Zde je však omezení v podobě maximální velikosti datového uložště.

2.1.1 Komunikační rozhraní

Rozhraní pro komunikaci v síti je zvolen Ethernet. Tato volba má hned několik důvodů. První z nich je cena, která je v porovnání s WiFi modulem přibližně pětinová. Druhá výhoda je, že drtivá většina firem používá Ethernet jako základní síťovou infrastrukturu, WiFi již tak běžná není. Třetí výhodou je odolnost metalického vedení vůči rušení a tím jeho vysoká stabilita. Navíc zde není kladen důraz na rychlost, jelikož se do sítě posílají pouze hodnoty naměřených veličin.

Jako síťová karta je tedy použit čip W5100 (viz Obr. 12), který je na desce jako Arduino Shield. Ten je původem neoriginální příslušenství pro Arduino. Jeho cena se pohybuje okolo 200 Kč. Je to tedy velmi levný způsob jak připojit MCU do sítě ethernet. Komunikace probíhá pomocí rychlého sériového rozhraní SPI. I když se jedná o příslušenství pro Arduino je použitelné s jakýmkoli mikrokontrolerem.



Obr. 12 - Síťová karta W5100

2.2 Teoretická realizace

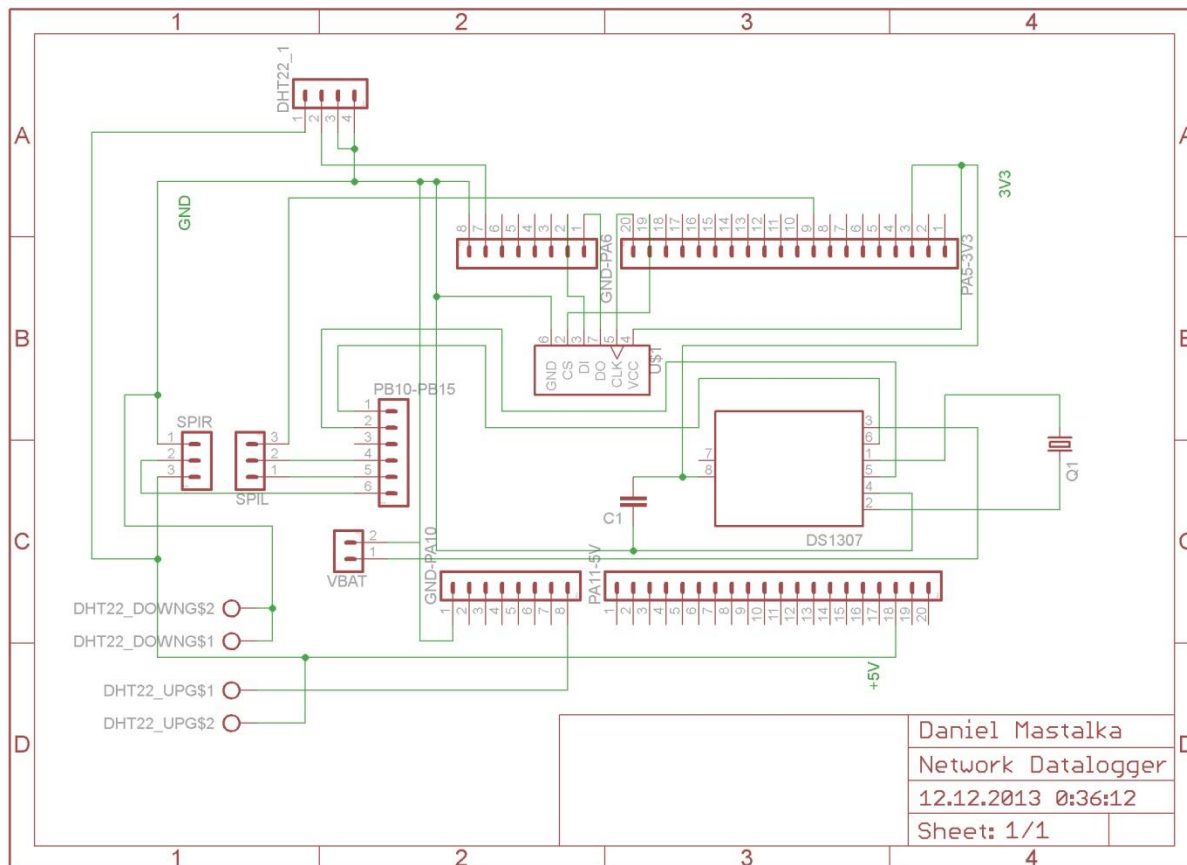
2.2.1 Návrh schéma a DPS

Jelikož je použit kit a síťový adaptér je již na desce, schéma se zužuje pouze na připojení SD karty a RTC do kitu a vytvoření redukce mezi STM32 VL Discovery a síťovým adaptérem standardu Arduino Shield.

Pro připojení síťového adaptéru je využit port SPI2 nacházející se na portech dle Tab. 3. Čidel je možné k vývojovému kitu připojit téměř tolik co je volných portů. Je zde tedy možnost určitého rozšíření a měřit tak teplotu například ve více místnostech zároveň.

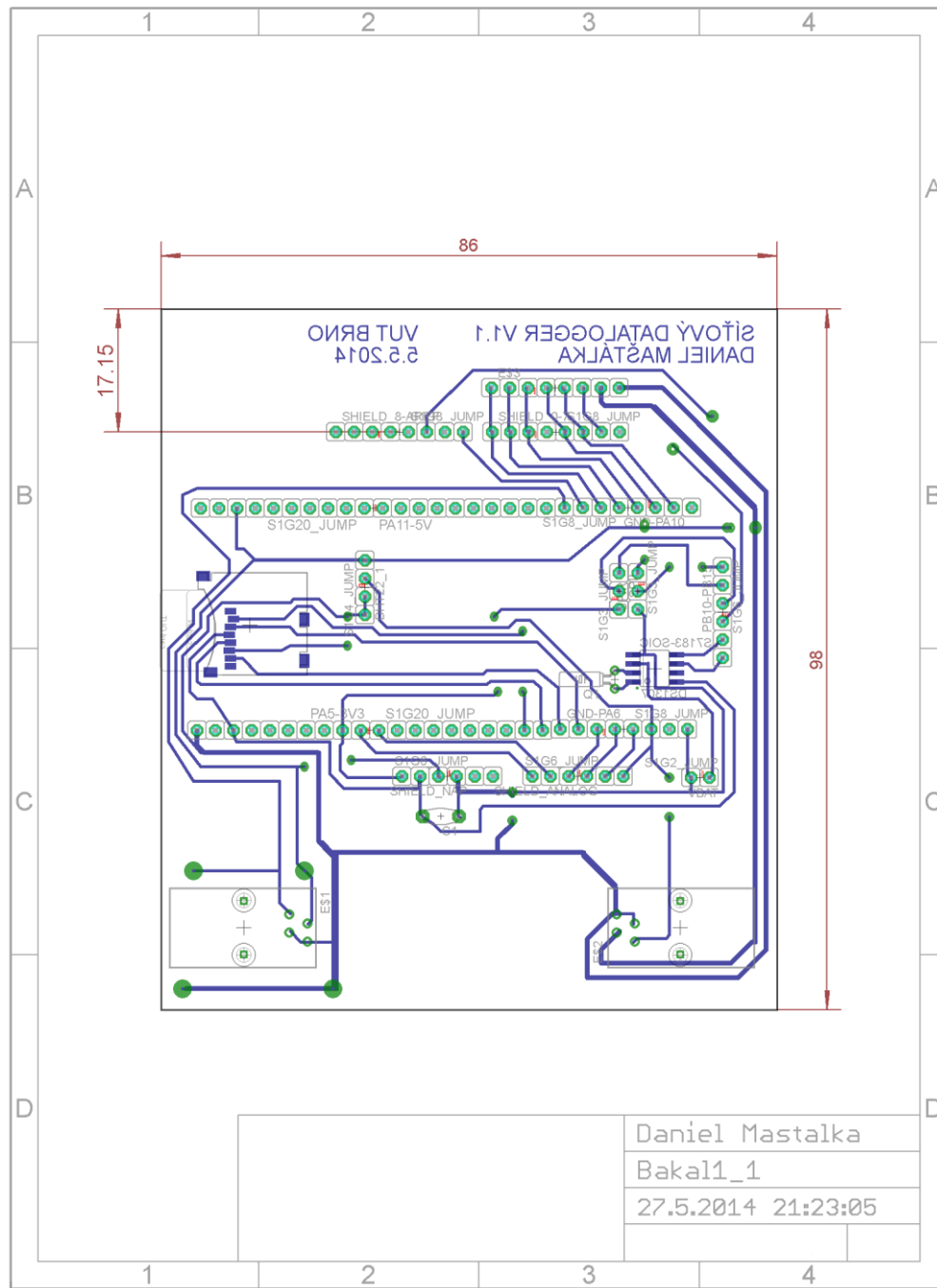
Tab. 3 - Popis rozhraní [11]

Periferie	Rozhraní	Porty rozhraní	Fyzické porty
Ethernet	SPI2	SPI2-SS	PB12
		SPI2-Clock	PB13
		SPI2-MISO	PB14
		SPI2-MOSI	PB15
RTC	I2C2	I2C2-SCL	PB10
		I2C2-SDA	PB11
DHT22_Indoor	One-Wire	-	PB1
DHT22_Outdoor	One-Wire	-	PA10
LCD	Paralelní komunikace s řadičem HD44780	RS	PC6
		E	PC7
		D4	PC8
		D5	PC9
		D6	PA8
		D7	PA9



Obr. 13 - Schéma síťového záznamníku

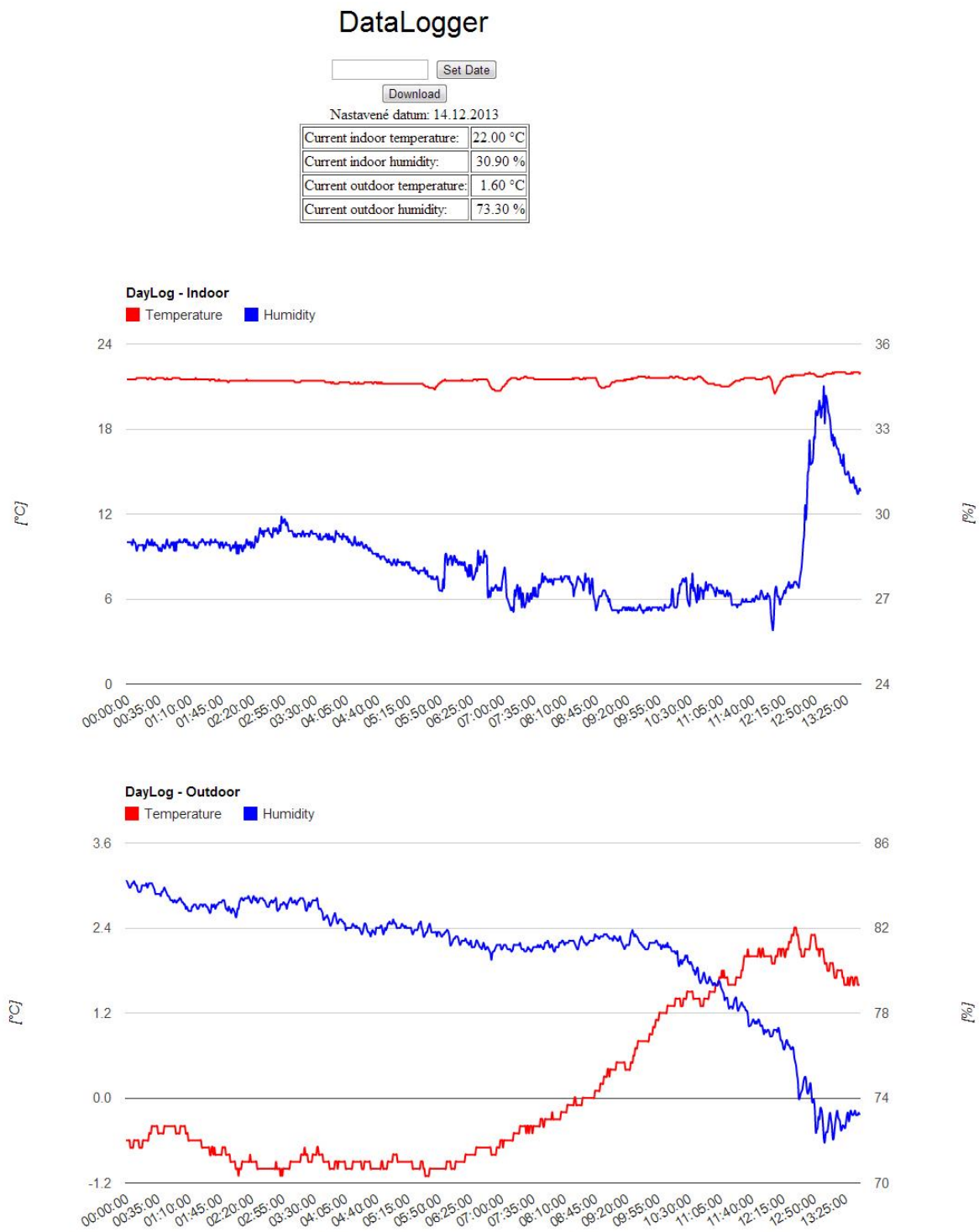
Jak je vidět na Obr. 13 schéma slouží k připojení periférií jako jsou digitální čidla, obvod reálného času a čtečky MicroSD karet. Další funkcí je konektor SPI pro připojení síťové karty. Další funkcí výsledné desky, pro kterou toto schéma bylo zdrojem, je propojení mezi kitem STM32VLD a standardem Arduino Shield.



Obr. 14 - Návrh DPS pro připojení periférií

2.2.2 Návrh serveru

Jelikož jsou všechna data uložena na serveru a ten slouží pro primární zobrazení výstupních dat. Je nutné vytvořit softwarové vybavení pro server. Je zároveň nutné požit více programovacích jazyků. Jejich výčet je HTML, PHP a MySQL. Každý z nich má ve své opodstatnění a používá se pro odlišné účely.



Obr. 15 - Vizualice serveru

Rovněž je vymyslet systém ukládání dat do databáze. Jelikož se jedná o dlouhodobé ukládání, mohl by po čase nastat zmatek v datech. Ukládání uvnitř databáze probíhá dle struktury, která je vidět v Tab. 4. Ukládání jednotlivých databází pak probíhá v systému Měsíc_Rok. Jednotlivé databáze tak jsou např. 11_2013, 12_2013. Tím je zajištěna přehlednost celkové databáze.

Tab. 4 - Struktura SQL databáze

Název	Typ	Null	Extra
ID	int(11)	Ne	AUTO_INCREMENT
Date	date	Ano	
Time	time	Ano	
Temp_IN	float(3,1)	Ano	
Hum_IN	float(3,1)	Ano	
Temp_OUT	float(3,1)	Ano	
Hum_OUT	float(3,1)	Ano	

3 Praktická realizace

3.1 Programové vybavení

Bylo nutné najít vhodný programátor pro naprogramování vývojového kitu STM-32. Software dodaný přímo od výrobce sice zaručuje maximální rychlost a vysokou úroveň podpory. Je však z části placený, je možné naprogramovat pouze 32 kB velký strojový kód. Již od začátku však bylo jasné, že program pro tuto práci bude větší než 32 kB. Proto bylo nutné najít jiné řešení.

3.1.1 Maple Leaf Labs

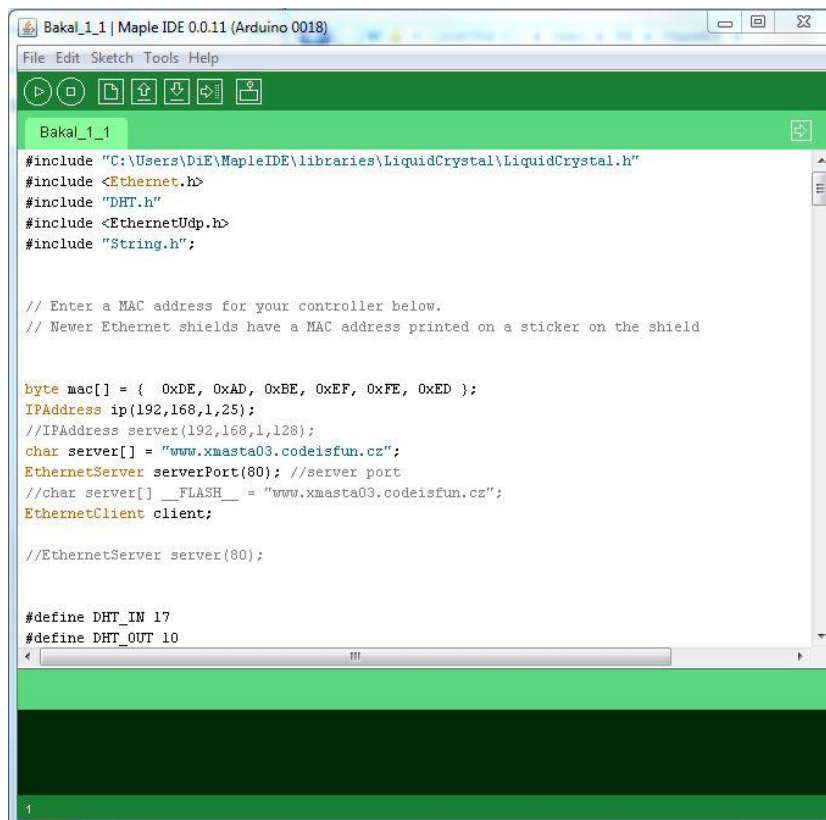
Tento překladač vypadal jako velmi vhodný pro tuto aplikaci. Maximální velikost kódu je omezena na 512 kB. Plně tedy dostává požadavku na 128 kB paměti, který má použitý procesor.

Instalace překladače byla složitější. Jelikož se jedná o desku podporovanou neoficiálně, bylo nutné do původní instalace vložit balík dalších dat, ve kterém se nacházelo označení desky, vývodů desky, naprogramování časovačů a rychlosti procesoru, aby doba v programovém kódu byla přesná.

Prostředí a kód je na první pohled téměř shodný s velmi známým Arduinem viz. Obr. 16. Toto je však pouze vizuální dojem. A jelikož všechny vnitřní knihovny, které jsou implementovány již v překladači, obsahují funkce odlišné od těch použitých v Arduinu, je programování více komplexní. Je nutné knihovnu staženou z Arduino Playground přepsat a modifikovat pro Maple. Některé základní knihovny jsou již implementovány a jsou plně funkční např. LiquidCrystal(pro obsluhu LCD), DHT(pro obsluhu čidel). Některé bylo nutné více či méně upravit pro chod nejprve pod překladačem Maple a poté i pro funkci na desce STM-32 VL Discovery.

Celkový potenciál Maple Leaf Labs je bohužel nevyužitý. Vývojáři doufali ve velkou podporu komunity jako má Arduino. To se však nestalo a tak je překladač někde na půl cesty. Funkční překladač ve velmi známém a přívětivém kódu, který však trpí minimální podporou a minimem profesionálně převedených knihoven.

Na konci roku 2013 představilo STM projekt Nucleo, který se zdá být vhodný pro programování procesorů STM. Je v něm vysoká podpora knihoven jako na Arduinu a je tedy možné plně využívat příslušenství k Arduinu, včetně standardu Arduino Shield. Tento překladač však vyšel pozdě a migrace kódu již nebyla z časových důvodů možná.



```
Bakal_1_1 | Maple IDE 0.0.11 (Arduino 0018)
File Edit Sketch Tools Help
Bakal_1_1
#include "C:\Users\DiE\MapleIDE\libraries\LiquidCrystal\LiquidCrystal.h"
#include <Ethernet.h>
#include "DHT.h"
#include <EthernetUdp.h>
#include "String.h";

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,25);
//IPAddress server(192,168,1,128);
char server[] = "www.xmasta03.codeisfun.cz";
EthernetServer serverPort(80); //server port
//char server[] __FLASH__ = "www.xmasta03.codeisfun.cz";
EthernetClient client;

//EthernetServer server(80);

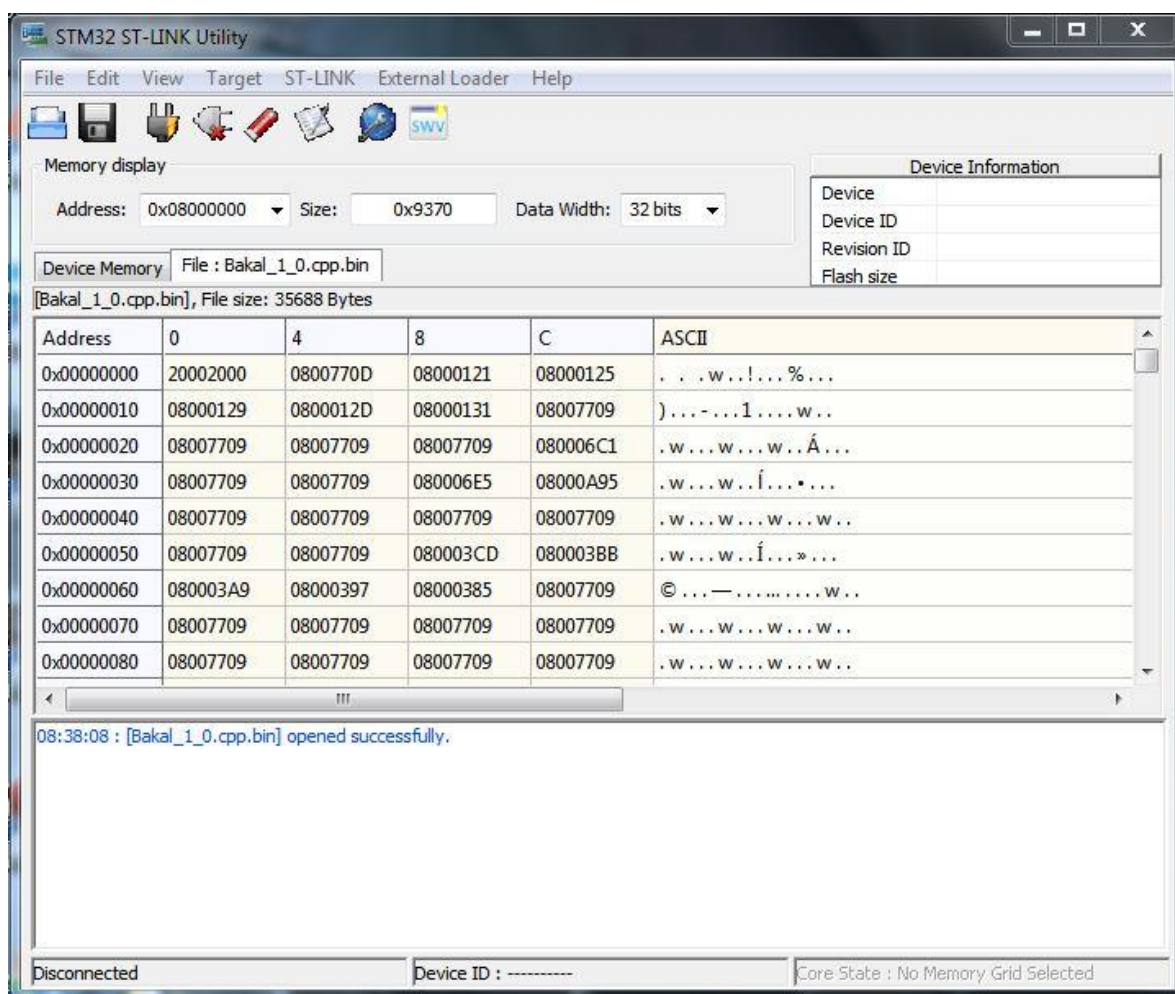
#define DHT_IN 17
#define DHT_OUT 10
```

Obr. 16 Překladač Maple Leaf Labs

3.1.2 Programátor STM32 ST-LINK Utility

Tento programátor je plně komerční řešení od STM. Je však zdarma pro nahrávání neomezeného množství dat. Obsahuje i jednoduchý debugger, ten však nefunguje ve spojení s překladačem Maple. Program umožňuje zapisovat jako do paměti pro firmware a tím je možné ho aktualizovat, tak i do zbytku paměti pro nahrání 128 kB programového kódu.

Programový kód je ve formátu *.bin a pokud je použit překladač Maple tak se nachází ve složce C:\Users\UserName\AppData\Local\Temp\Build(číslo sestavení)\Název sezení.bin. Nahrávání je tedy celkem zdouhavé, zvláště když nejde odkazovat do jedné složky a při každém zapnutí programu se číslo sestavení změní. Po nahrání souboru do programátoru již stačí zkontrolovat a zapsat, poté proběhne další kontrola a restart desky.



Obr. 17 STM32 ST-LINK Utility

3.2 Programování

Bylo nutné programovat ve více jazycích. Jednak se musel programovat vývojový kit, dále pak bylo nutné naprogramovat stránky pro uživatelské rozhraní.

3.2.1 STM-32 Value Line Discovery

K programování tohoto vývojového kitu bylo využito již zmíněného překladače Maple Leaf Labs. Zde hlavní kód shodný s tím pro Arduino. Nejvíce práce však zabrala úprava knihoven. Nejprve byly staženy knihovny z Arduino Playground nebo z podobných stránek a poté byla provedena úprava, aby knihovna, alespoň v základním režimu, pracovala s deskou STM-32 VLD.

Nejsložitější na úpravu byla knihovna pro Ethernet, která i po úpravě je funkční pouze při frekvenci SPI 3 MHz a není možné ji provozovat rychleji. Hlavní úprava tedy spočívala ve zjištění funkční frekvence SPI a synchronizace frekvencí v celé knihovně. Poté je u každé knihovny nutné zapsat funkci, která zjišťuje o jaký procesor se jedná a nastaví vše nutné pro běh na této desce.

Po všech těchto úpravách jsou knihovny z většiny funkční. Bohužel jen z většiny a plné funkčnosti všech knihoven se nepovedlo dosáhnout. Pro provoz zařízení jsou však tyto funkce dostačující.

Poté je nutné již jen naprogramovat procesor v kódu Arduino, který je široce podporovaný. Zde jsou vysvětleny některé funkční celky.

```
//Vložení všech knihoven -----
#include "C:\Users\DiE\MapleIDE\libraries\LiquidCrystal\LiquidCrystal.h"
#include <Ethernet.h>
#include "DHT.h"
#include <EthernetUdp.h>
#include "String.h";
//END Vložení všech knihoven -----
//Inicializace síťové karty-----
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };//Nastavení MAC adresy
IPAddress ip(192,168,1,25);//Nastavení defaultní IP adresy
//IPAddress server(192,168,1,128);
char server[] = "www.xmasta03.codeisfun.cz";//Nastavení serveru pro odesílání dat
EthernetServer serverPort(80); //Nastavení portu webového serveru
//char server[] __FLASH__ = "www.xmasta03.codeisfun.cz";
EthernetClient client; //Inicializace klienta síťové karty
//END Inicializace síťové karty-----
//Definice a proměnné pro čidla-----
#define DHT_IN 17 //Vnitřní čidlo
#define DHT_OUT 10 //Venkovní čidlo
#define DHTTYPE DHT22 //Definice typu čidla pro knihovnu
DHT dht_in(DHT_IN, DHTTYPE); //Inicializace vnitřního čidla
DHT dht_out(DHT_OUT, DHTTYPE); //Inicializace venkovního čidla
float tIn,hIn,tOut,hOut; //Globální proměnné pro zápis hodnot
unsigned long wait=0; //Proměnná pro funkci odloženého zápisu
bool connected = false;
//END Definice a proměnné pro čidla-----
void loop()
{
  hIn = dht_in.readHumidity(); //Volání funkce knihovny pro čtení vlhkosti
  tIn = dht_in.readTemperature(); //Volání funkce knihovny pro čtení teploty

  hOut = dht_out.readHumidity();
  tOut = dht_out.readTemperature();

  if (!isnan(tIn) || !isnan(hIn) || !isnan(tOut) || !isnan(hOut))
  {
    if (wait != (millis()/10000)){ //Funkce pro odložený zápis(Snižuje zahlcení sítě)
      EthSendPHP(tIn,hIn,tOut,hOut); //Volání funkce pro odeslání dat na server
      lcd.clear(); //Preventivní vyčištění displeje
      wait = millis()/10000;
    }
    LCDPrint(); //Výpis hodnot na LCD
  }
  EthServer(); //Volání funkce pro vytvoření webového serveru
}
}
```

Obr. 18 Důležité části kódu

```

void EthSendPHP(float TempIN,float HumIN,float TempOUT, float HumOUT)
{
  if(client.connect(server,80)){ //vytvoření spojení na portu 80
    client.print("GET /add.php?"); //Požadavek na GET příslušné stránky

    client.print("tIn"); //Označení vnitřní teploty
    client.print("=");
    client.print(TempIN); //výpis vnitřní teploty
    client.print("&&");

    client.print("hIn"); //Označení vnitřní vlhkosti
    client.print("=");
    client.print(HumIN); //výpis vnitřní vlhkosti
    client.print("&&");

    client.print("tOut"); //označení venkovní teploty
    client.print("=");
    client.print(TempOUT); //výpis venkovní teploty
    client.print("&&");

    client.print("hOut"); //označení venkovní vlhkosti
    client.print("=");
    client.print(HumOUT); //výpis venkovní vlhkosti

    client.println(" HTTP/1.1"); //specifikace protokolu
    client.println("Host: www.xmasta03.codeisfun.cz");//specifikace serveru
    client.println("Connection: close");//oznámení ukončení spojení se serverem
    client.println();
    client.stop();//ukončení spojení se serverem
  }
}

```

Obr. 19 Arduino kód pro odeslání dat

Na Obr. 18 je vidět hlavní část inicializace a deklarace proměnných. Jako první je nutné nahrát všechny potřebné knihovny. Poté je vhodné provést inicializaci síťové karty, té trvá načtení nejdéle. Poté je již možné definovat proměnné pro čidla. Následuje část kódu, kde se provádí nastavení všech periférií. Poté je již možné použít hlavní část kódu, smyčku pro funkci celého záznamníku. V té jsou nejdříve načtena data z obou čidel, poté je provedena kontrola, zda proměnné obsahují číslo. Prvních několik volání této funkce má za následek pouze nečíselnou hodnotu. Dále již nic nebrání přiřazení hodnot do funkce pro odeslání dat, ta je volána každých deset vteřin.

Funkce EthSendPHP jejíž vstupní proměnné jsou čtyři proměnné typu číslo s desetinnou čárkou. Provádí odesílání dat na PHP server. Zde je nutné se nejdříve připojit k serveru a označit, že data budou odesílána pomocí příkazu GET na stránku „add.php“. Poté lze odesílat jednotlivé proměnné ve formátu „tIn=xx.x“. Odeslání příkazu se provede ukončením spojení.

3.2.2 Webový server

Dále bylo důležité napsat webový server pro ukládání a prezentaci dat. Celý server je v PHP a HTML. Ukládání probíhá do databáze pomocí MySQL.

Vzhled stránky je v HTML a byla snaha o co nejjednodušší vzhled pro zachování celkové přehlednosti. Proto je web po vizuální stránce velmi strohý avšak naprosto funkční. Stránka obsahuje oznámení o umístění venkovního čidla, výběr dne ze zálohy, tabulku obsahující aktuální hodnoty obou čidel. Ta pokud jsou data v databázi více jak dvě minuty

staré vypíše místo hodnot prázdnou tabulku a uživateli oznámí, že je záznamník mimo provoz. Následuje již jen přehledný denní graf pro vnitřní a venkovní čidla.

K plné funkci webového serveru je nutných několik souborů. Některé části kódu důležitých komponent jsou vysvětleny v dalších obrázcích.

```
$Last = mysql_query("SELECT (Time) FROM ".$stabilka."
WHERE ID=(SELECT MAX(ID) FROM ".$stabilka.""); //Funkce pro vypsání posledního zápisu
$LastResult = mysql_fetch_array($Last);
$LastTime = substr($LastResult[0],-5,2); //Zjištění času z posledního zápisu

if ($mins != $LastTime && ($h1 != 0 && $h2 != 0 )){ //Zapiše data každou minut
$CreateTable = "CREATE TABLE ".$stabilka."(ID int auto_increment, Date date ,Time time,Temp_IN float(3,1),
Hum_IN float(3,1),Temp_OUT float(3,1),Hum_OUT float(3,1),
PRIMARY KEY (ID)); //Funkce pro vytvoření nové tabulky pokud nastane nový měsíc
$Vysledek = mysql_query($CreateTable); //Volání funkce pro vytvoření tabulky
$query = "INSERT INTO ".$stabilka." (Date,Time,Temp_IN, Hum_IN, Temp_OUT, Hum_OUT) VALUES
('".$datum . "','" . $cas . "','" . $_GET['tIn'] . "','" . $_GET['hIn'] . "','" .
$_GET['tOut'] . "','" . $_GET['hOut'] . "')"; //Funkce pro vložení dat poslaných ze záznamníku
$result = mysql_query($query) or die("Query chyba: " . mysql_error()); //Volání funkce pro odeslání dat
}
```

Obr. 20 PHP kód pro vložení dat do databáze

Práce s SQL v kódu PHP spočívá v přípravě SQL dotazu do proměnné a jeho následného volání. V prvních dvou řádcích je vidět uložení do proměnné `$Last`, kde výsledkem je čas a datum posledního zápisu, který je zjištěn podle nejvyššího čísla ID, které je unikátní v každé tabulce. Po vyvolání příkazu je důležité osamostatnit pouze čas, to udělá příkaz `substr` s parametrem. Jeho výsledkem je minuta, ve které byl záznam proveden. V následujícím bloku je zkontrolováno, zda již uběhla minuta od posledního zápisu. Pokud ano, je provedena kontrola, zda se vlhkosti nerovnájí 0%RH. To je chyba, která vznikne po zapnutí záznamníku, kdy je při úvodní inicializaci čidel vlhkost na obou čidlech rovna 0%RH. Pokud jsou tyto dvě podmínky splněny, vytvoří se nová tabulka s potřebnou strukturou a názvem. Pokud tabulka existuje, požadavek skončí chybou, která se nevypisuje. Poté již nic nebrání zapsání záznamu s datem, časem, teplotami a vlhkostmi do tabulky. Jako volání příkazu je proměnná `$result` a jejím výsledkem je buď provedený příkaz nebo ukončení příkazu na chybě, která se vypíše.

```

$stable = array();
$stable['cols'] = array( //Vytvoření 2D pole

    array('label' => 'Time', 'type' => 'string'), //První sloupec hlavička: čas typ: String
    array('label' => 'Temperature', 'type' => 'number'), //První sloupec hlavička: Teplota typ: Číslo
    array('label' => 'Humidity', 'type' => 'number') //První sloupec hlavička: Vlhkost typ: Číslo

);

$rows = array();
while($r = mysql_fetch_assoc($query)) { //Volání funkce pro načtená dat z databáze
    $temp = array();
    $temp[] = array('v' => $r['Time']); //Zapsání času do proměnné
    $temp[] = array('v' => (float) $r['Temp_IN']); //Převedení teploty do datového typu Float a zapsání do proměnné
    $temp[] = array('v' => (float) $r['Hum_IN']); //Převedení vlhkosti do datového typu Float a zapsání do proměnné

    $rows[] = array('c' => $temp); //Zapsání celé proměnné do pole
}

$stable['rows'] = $rows;

$jsonTable = json_encode($stable); //Přeformátování pole to formátu JSON

header('Cache-Control: no-cache, must-revalidate');//Nastavení hlaviček, nejsou důležité
header('Expires: Mon, 20 Jul 2000 05:00:00 GMT');
header('Content-type: application/json');

echo $jsonTable; //Vypsání celé JSON tabulky
?>

```

Obr. 21 PHP kód pro převod dat do formátu JSON

Jelikož je vykreslování grafů Google Chart v JavaScriptu je nutné data převést do přehledného JSON formátu. Tento formát je jednoduše čitelný jak strojevě tak i člověkem. Vzor tohoto formátu je vidět na Obr. 22. Pro převod dat je nejprve nutné si připravit proměnnou typu dvourozměrné pole. Do ní je nutné zapsat hlavičky jednotlivých položek. Poté již můžeme naplnit příslušné buňky daty. Pro převod do JSON je také nutné definovat datový typ, který se v PHP standardně nedefinuje. Po naplnění celého pole je již možné provést příkaz `json_encode()`, ten převede celé pole do JSON formátu. Zbývá již jen vypsát celý výsledek.

```

[
  {Time: '12:00:00', Temperature: '19.7', Humidity: '28,3'},
  {Time: '12:01:00', Temperature: '19.7', Humidity: '28,3'},
  {Time: '12:02:00', Temperature: '19.7', Humidity: '28,3'},
  {Time: '12:03:00', Temperature: '19.7', Humidity: '28,3'},
  {Time: '12:04:00', Temperature: '19.7', Humidity: '28,3'},
]

```

Obr. 22 Data ve formátu JSON

```

google.setOnLoadCallback(drawChart);

function drawChart() {
    var json = $.ajax({
        url: 'GetOutdoor.php', // Označení stránky která převádí data do JSON
        dataType: 'json',      //Určení datového typu
        async: false
    }).responseText;

    // Vytvoří tabulku z JSON dat
    var data = new google.visualization.DataTable(json);
    var options = {
        title: 'DayLog - Outdoor', //Název grafu
        width: 1200,               //Šířka grafu
        height: 600,              //Výška grafu
        curveType: 'function',    //Nastavení zaoblení průběhu
        focusTarget: 'category',  //Při označení bodu se označí ve všech průbězích
        selectionMode: 'multiple', //Možnost označit více bodů
        colors:['red','blue'],    //Nastavení barev průběhů
        vAxes: {0: {logScale: false,title: '[°C]'}, //Nastavení osy y pro teplotu
                1: {logScale: false,title: '[%]'}}, //Nastavení osy y pro vlhkost
        series:{
            0:{targetAxisIndex:0}, //Přiřazení průběhů k osám
            1:{targetAxisIndex:1}}
    };
    var chart = new google.visualization.LineChart(document.getElementById('chart_div2'));
    chart.draw(data, options);
}

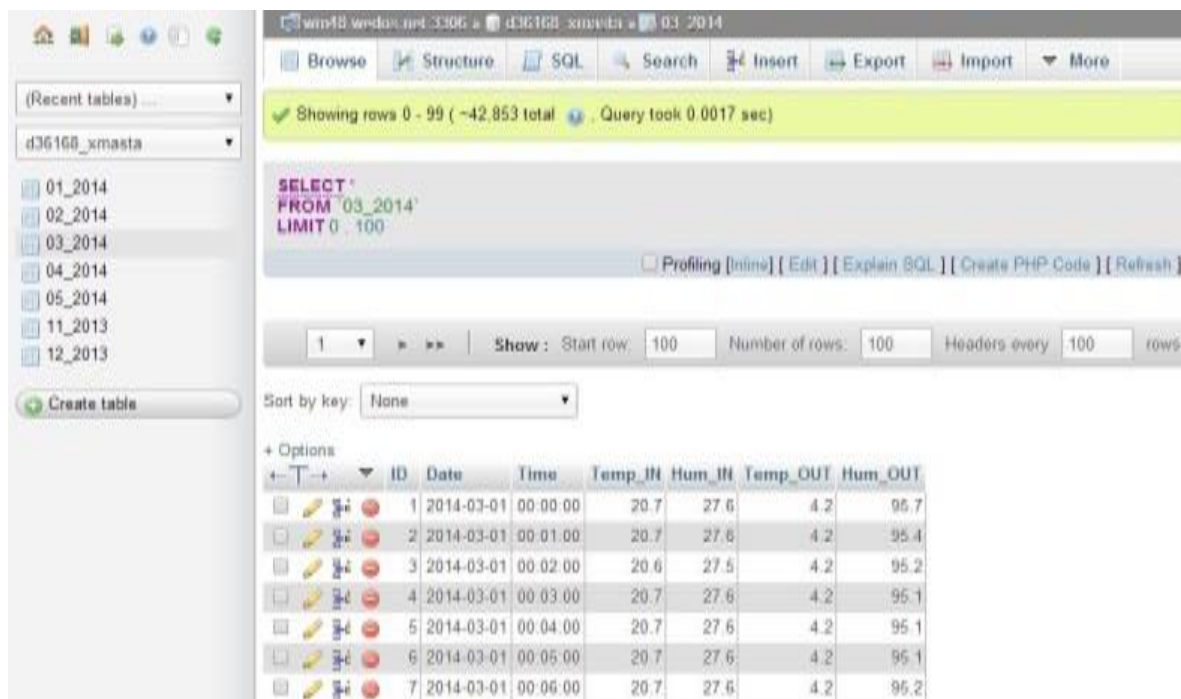
```

Obr. 23 Vykreslení grafu Google Chart

Vykreslení grafů je řešeno přes Google Chart. Zpracování probíhá na serveru Google a řešení je kompletně zdarma. Data již jsou ve formátu JSON, stačí je tedy jen vložit do příkladového kódu. Zde je ještě možnost nastavit některé možnosti grafů, jak je možné vidět na Obr. 23. Všechna nastavení jsou popsána na stránkách Google a je zde ještě možnost si předem graf vyzkoušet přímo na stránkách Google. Řešení je to tedy velice elegantní, stabilní a vše je zdarma.

3.2.3 Databázový server

Je nutné všechna data ukládat do databáze. Nejvhodnější je řešení přes MySQL, toto řešení je hojně využíváno v profesionálních aplikacích. Má skvělou přehlednost díky uživatelskému rozhraní PHPMyAdmin a vysokou úroveň podpory. Všechny úkony v databázi se provádí přes SQL příkazy. V rozhraní PHPMyAdmin, je pak možné dělat jednoduché příkazy i bez nutnosti znalosti SQL jazyka. Uživatel tedy může například promazat záznamy nebo smazat celou tabulku pro daný měsíc.



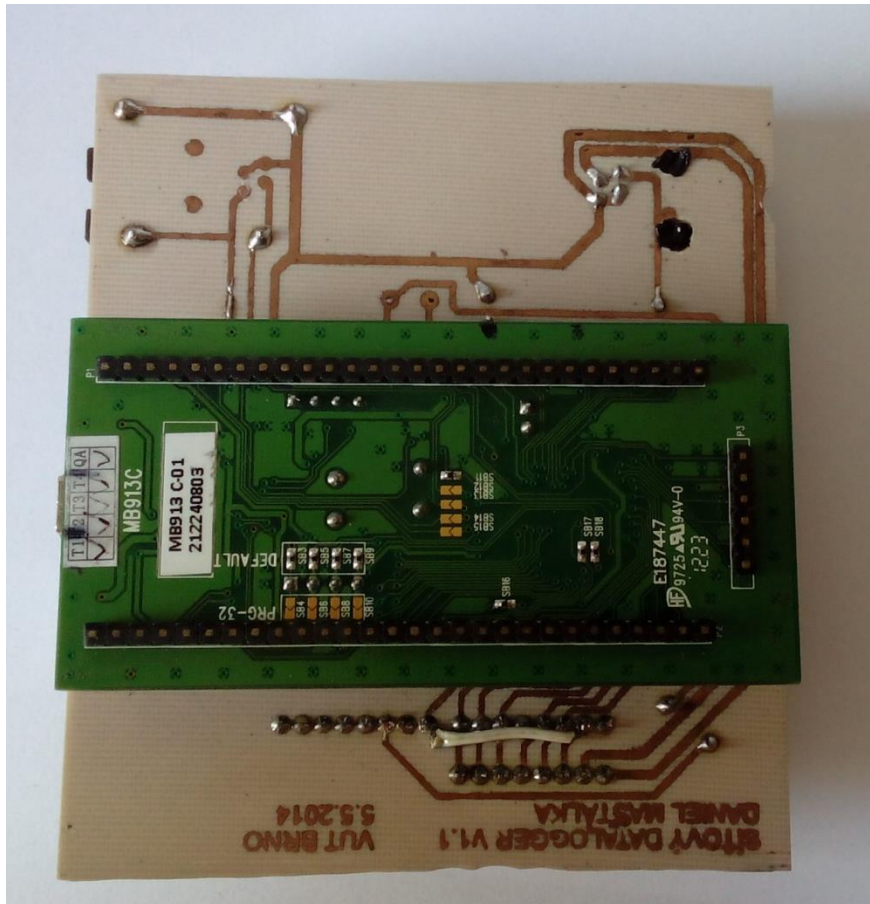
Obr. 24 Rozhraní PHPMYAdmin

3.3 Výroba záznamníku

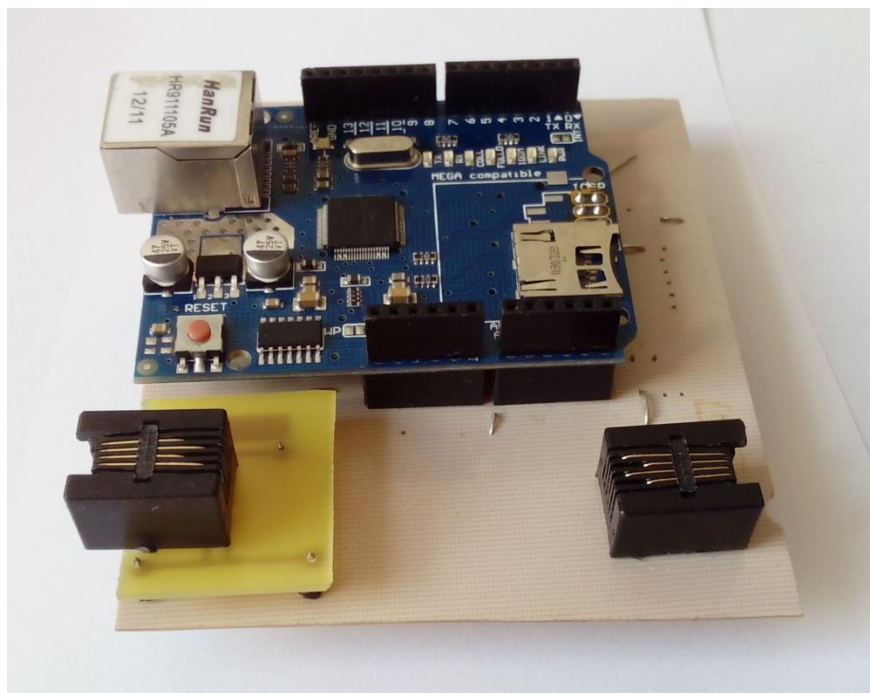
3.3.1 Výroba desek plošných spojů

Podle návrhu na Obr. 14 byla vyrobena deska plošných spojů. Použitá metoda pro výrobu byla nažehlováním toneru a následné vyleptání v chloridu železitém. Tato metoda je sice méně přesná, než použití fotorezistu. Je však levnější a dostupnější. Jako základní materiál pro hlavní desku je použit CEM-1, není zde takový nárok na povětrnostní podmínky jako u čidel, kde byl použit materiál FR4. Po vyleptání následuje osazení potřebnými konektory. Osazená deska je vidět na Obr. 27. Jak bylo původně zamýšleno, deska slouží pouze pro propojení všech potřebných periférií.

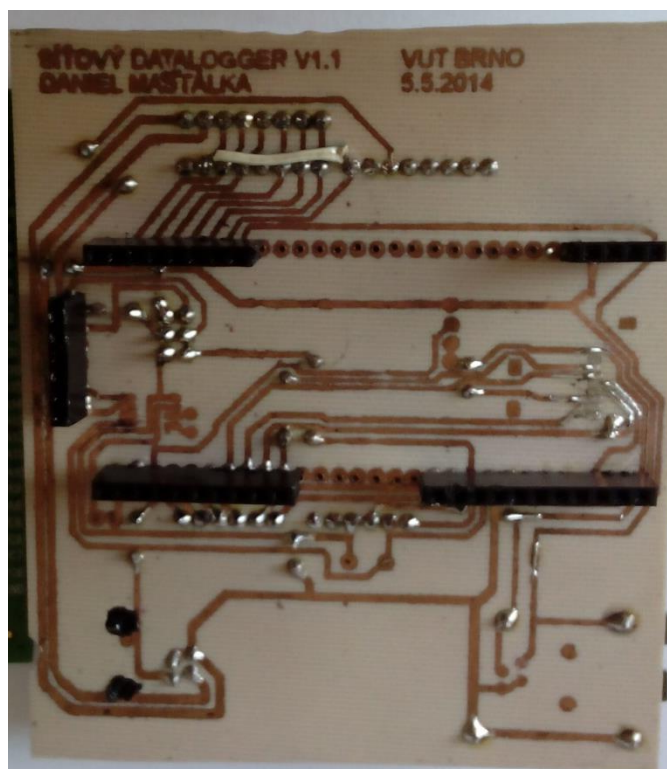
Pro vyrovnání konektorů RJ-45 a RJ-11 bylo dále nutné vyrobit jednu malou desku, na které je pouze jeden konektor. Tím je tento konektor vyvýšen a boční panel je tak opticky sladěn. Pro čidla je pak také vyrobena malá DPS, na které se nachází pouze čidlo a konektor RJ-11. Po zasunutí všech desek dohromady vznikne sestava, jež je vidět na Obr. 25 a Obr. 26.



Obr. 25 Sestava desek - Pohled zdola



Obr. 26 Sestava desek - pohled zhora



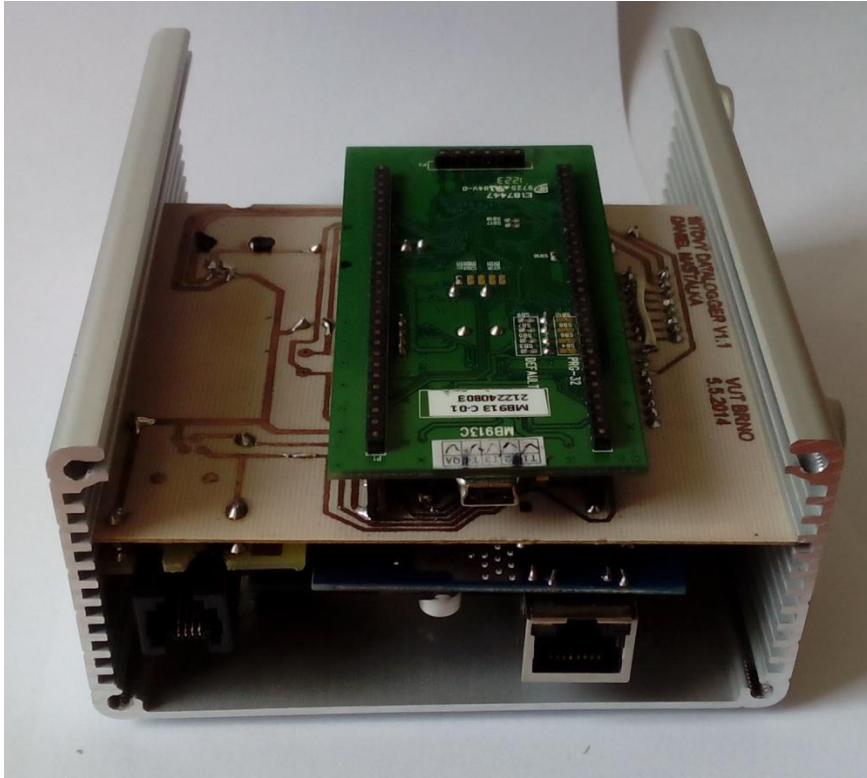
Obr. 27 Osazená deska plošných spojů

3.3.2 Výroba krabičky

Je nutné desky náležitě chránit uložením do krabičky. Byla zvolena hliníková krabička, která je z vizuálního hlediska vhodnější než plastová. Avšak boční panely jsou plastové z důvodu snadnějšího opracování. Na vrchní straně krabičky bylo nutné vyřezat otvor pro LCD, to bylo provedeno ručně pomocí přímé brusky. Mnohem vhodnější by bylo nechat tento otvor vyříznout strojně.

Na pravém bočním panelu jsou vyříznuty otvory pro konektory RJ-11 pro připojení čidla, RJ-45 pro připojení sítě ethernet a MiniUSB pro napájení celého systému. Na levém panelu je umístěno vnitřní čidlo. To musí být umístěné vně krabičky, aby nebylo ovlivněno teplem, které uvolňuje především řadič síťové karty.

Pro umístění DPS do krabičky bylo využito drážek, které jsou v krabičce připraveny.



Obr. 28 Umístění DPS v krabičce



Obr. 29 Rozmístění konektorů na panelu



Obr. 30 Výsledný vzhled záznamníku

3.4 Seznam součástek

Celkový seznam součástek je vidět v Tab. 5. Cena celkového výrobku se pohybuje okolo jednoho tisíce korun. Součástky jsou objednávány ze zahraničí. V ceně nejsou započteny náklady na výrobu desek plošných spojů.

Tab. 5 Seznam součástek a celkové náklady

Název	Počet	Cena
Síťová karta W5100	1	155 Kč
Vývojový kit STM32-VLD	1	252 Kč
Čidlo DHT-22	2	80 Kč
Krabička venkovního čidla	1	38 Kč
Krabička vnitřního čidla	1	30 Kč
Hliníková krabička	1	370 Kč
Spojovací materiál		50 Kč
Celkem		1055 Kč

4 Závěr

V práci je proveden teoretický návrh mikrokontroleru a jeho periférií. Jako mikrokontroler byl vybrán vývojový kit STM 32 VL Discovery a to zejména z důvodu ceny. Jako periferie byl vybrán vhodný ethernet řadič, pro připojení záznamníku do sítě ethernet. Pro sběr dat byla vybrána čidla DHT-22. Jsou to velmi levná teplotní a vlhkostní čidla, avšak velmi přesná.

Dále byl navržen server pro sběr dat, vyhodnocení dat a interaktivní spojení uživatele s daty. Grafické prostředí bylo navrženo s ohledem na přehlednost a efektivitu používání. K vykreslování grafů bylo využito aplikace Google Chart. Data jsou ukládána do přehledné databáze na server MySQL. Vzniká tak konzistentní datová řada, ve které je velmi snadná orientace a vyhodnocení dat. Jako vhodná doba mezi záznamy byla vybrána jedna minuta. Nevzniká tím datová náročnost a data se jeví jako spojitá.

Poté byla navržena a zhotovena deska pro spojení všech periférií. Vše bylo umístěno do hliníkové krabičky, která byla upravena otvory pro umístění konektorů a jedním velkým pro LCD, které slouží pro primární zobrazení aktuálních dat a IP adresy v síti. Krabička je opatřena silikonovými nožičkami a je možno přístroj provozovat ve svislé i ležaté poloze. Čidla jsou k záznamníku připojena konektorem RJ-11 a čtyřvodičovým kabelem. Ten je vhodný zejména z důvodu, že je plochý a může projít rámem plastového okna.

Těmito kroky bylo zadání bakalářské práce splněno. Síťový záznamník funguje naprosto spolehlivě. Požadavek na cenově dostupný produkt byl také splněn a cena nepřesahuje 1100 Kč. Záznamník byl několik měsíců testován a vykazuje velmi malou chybovost. Je tedy vhodný i pro malé laboratoře. Data lze jednoduše exportovat a dále s nimi pracovat.

Pro budoucí rozšiřování by bylo vhodné použít jiný překladač a celý projekt navrhnout na platformě Mbed, což je platforma přímo od STMicroelectronics. Záznamník by se mohl rozšířit o vnitřní uložení pro lokální zálohu dat a bezdrátová čidla. Bylo by však nutné přenos opatřit kódováním a kontrolou, aby byla zachována spolehlivost celého přístroje. Pro tato vylepšení je však nutné změnit platformu. Překladač Maple Leaf Labs není vhodný z důvodu nulové podpory.

Seznam obrázků

Obr. 1 - Blokové schéma mikrokontroléru STM32F100RBT6B [10]	11
Obr. 2 - Blokové schéma STM32 VL Discovery [11]	12
Obr. 3 - Vývojový kit STM 32 VL Discovery [11]	12
Obr. 4 - Popis termočlánku	13
Obr. 5 - Závislost odporu na teplotě u čidla PT1000 [2]	13
Obr. 6 - Odporové vlhkostní čidlo EFS 10, 0 - 60%RH	14
Obr. 7 - Příklad měření pomocí gravimetrie - Sušící váhy [8]	14
Obr. 8 - Senzor SHT-75 [9]	16
Obr. 9 - Digitální teplotní a vlhkostní čidlo DHT-22 [1]	16
Obr. 10 Teplotní čidlo DHT22 strana čidel	17
Obr. 11 Teplotní čidlo DHT22 strana elektroniky	17
Obr. 12 - Síťová karta W5100	18
Obr. 13 - Schéma síťového záznamníku	20
Obr. 14 - Návrh DPS pro připojení periférií	21
Obr. 15 - Vizualice serveru	22
Obr. 16 Překladač Maple Leaf Labs	25
Obr. 17 STM32 ST-LINK Utility	26
Obr. 18 Důležité části kódu	27
Obr. 19 Arduino kód pro odeslání dat	28
Obr. 20 PHP kód pro vložení dat do databáze	29
Obr. 21 PHP kód pro převod dat do formátu JSON	30
Obr. 22 Data ve formátu JSON	30
Obr. 23 Vykreslení grafu Google Chart	31
Obr. 24 Rozhraní PHPMyAdmin	32
Obr. 25 Sestava desek - Pohled zdola	33
Obr. 26 Sestava desek - pohled zhora	33
Obr. 27 Osazená deska plošných spojů	34
Obr. 28 Umístění DPS v krabici	35
Obr. 29 Rozmístění konektorů na panelu	35
Obr. 30 Výsledný vzhled záznamníku	36

Seznam tabulek

Tab. 1 - Parametry procesoru STM32F100RB [10]	10
Tab. 2 - Srovnání digitálních čidel [1] [6] [9]	15
Tab. 3 - Popis rozhraní [11]	19
Tab. 4 - Struktura SQL databáze	23
Tab. 5 Seznam součástí a celkové náklady	36

Seznam zkratek

ARM	
Advanced RISC Machines	5, 10, 12
CEM	
Composite Epoxy Material.....	32
CPU	
Central Processing Unit.....	10
DPS	
Deska Plošných Spojů.....	19, 21
FR	
Flame Retardant	32
GPU	
Graphic Processing Unit.....	10
I ² C	
Inter-Integrated Circuit.....	10, 13, 15
LAN	
Local Area Network	9, 17
LCD	
Liquid Crystal Display	24, 34, 37
LQFP64	
Low Profile Quad Flat Package	12
MCU	
Microcontroller Unit	13, 15, 18
RAM	
Random Access Memory	10
RISC	
Reduced Instruction Set Computing	10
ROM	
Read Only Memory.....	16
RTC	
Real Time Clock.....	19
SOC	
System On Chip	10
SPI	
Serial Peripheral Interface.....	10, 13, 18, 20
USART	
Universal Synchronous / Asynchronous Receiver and Transmitter.....	10
WAN	
Wide Area Network	17

Seznam použitých zdrojů

- [1] Aosong(Guangzhou) Electronics Co.,Ltd, „Digital-output relative humidity & temperature sensor/module,“ DHT-22(AM2303) datasheet
- [2] CYNTEC CO., LTD., „Specification of Pt Thermal Sensor,“ PT1000 datasheet
- [3] Reading an SD card with an ATMEGA168. In: Elastic Sheep: Because elasticdog was already taken [online]. 2010, 12.1.2010 [cit. 2013-12-15]. Dostupné z: <http://elasticsheep.com/2010/01/reading-an-sd-card-with-an-atmega168/>
- [4] Procesorty ARM: Základ nové éry. In: KARCHŇÁK, Daniel. Živě.cz [online]. Vyd. 1. Praha: Mladá fronta, 2012, 8.6.2012 [cit. 2013-12-15]. Dostupné z: <http://www.zive.cz/clanky/procesory-arm-zaklad-nove-ery/sc-3-a-164061/default.aspx>
- [5] Maxim Integrated Products Inc, „64 x 8, Serial, I2C Real-Time Clock,“ DS1307 datasheet, July 2005
- [6] Maxim Integrated Products Inc, „Programmable Resolution 1-Wire Digital Thermometer,“ DS18B20 datasheet, May 2002
- [7] Úvod do architektury Cortex-M3: díl. 1. In: Pandatron.cz: Elektrotechnický magazín [online]. 2010, 10. února 2010 [cit. 2013-12-15]. ISSN 1803-6007. Dostupné z: http://pandatron.cz/?1252&uvod_do_architektury_cortex-m3_-_dil._1
- [8] Analyzátory vlhkosti Sartorius (sušící váhy). SARTALEX, spol. s r.o. Sartalex: Velmi přesné přístroje [online]. 2013 [cit. 2013-12-15]. Dostupné z: <http://sartalex.cz/produkty/analyzatory-vlhkosti/>
- [9] Sensirion AG, „Humidity & Temperature Sensor,“ SHT75 datasheet, March 2007
- [10] STMicroelectronics, „Low & medium-density value line, advanced ARM-based 32-bit MCU with 16 to 128 KB Flash, 12 timers, ADC, DAC & 8 comm interfaces,“ STM32F100RBT6B datasheet, April 2011
- [11] STMicroelectronics, „STM32 value line Discovery,“ STM32VLDISCOVERY datasheet, June 2011
- [12] FURBER, Stephen B. ARM system-on-chip architecture. New York: Addison-Wesley, 2000, xii, 417 s. ISBN 978-0-201-67519-1.
- [13] マイコン風雲録: Maple/libmapleでEthernetライブラリを使う。Wiznet w5100/w5200. マイコン風雲録 [online]. 2012, 18.2.2012 [cit. 2014-05-25]. Dostupné z: <http://avr.paslog.jp/article/2379948.html>