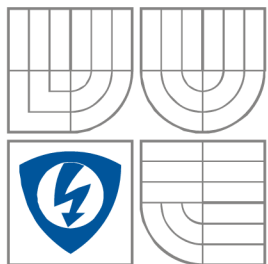


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

IMPLEMENTACE KOMUNIKAČNÍHO ROZHRANÍ USB/LAN/WIFI

IMPLEMENTATION OF THE COMMUNICATION INTERFACE USB/LAN/WIFI

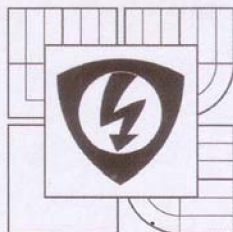
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV VONEŠ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. MILOSLAV STEINBAUER, Ph.D.



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ
Fakulta elektrotechniky
a komunikačních technologií
Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Jaroslav Voneš

Ročník: 3

ID: 155657

Akademický rok: 2015/16

NÁZEV TÉMATU:

Implementace komunikačního rozhraní USB/LAN/WIFI

POKYNY PRO VYPRACOVÁNÍ:

Obsahem práce je implementace komunikačního rozhraní pro ovládání inkubátoru počítačem pomocí rozhraní USB/LAN/WIFI. Seznamte se s definicemi rozhraní USB, LAN TCP/IP a WiFi, prostudujte možnosti jejich obvodové realizace. Navrhněte zapojení komunikačního rozhraní pro připojení zařízení s procesorem Atmel AVR k počítači. Zařízení realizujte a vytvořte program pro PC, sloužící k nastavování a čtení zadaných parametrů inkubátoru.

DOPORUČENÁ LITERATURA:

[1] ATmega8. 8-bit AVR Microcontroller with 8K In-System Programmable Flash. Data Sheet [online]. Atmel Corp., 2005. Dostupné na: http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf.

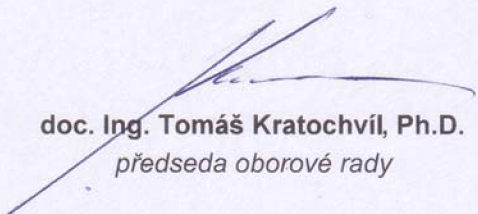
[2] XPort Data Sheet 910-815E [online]. Lantronix Inc., 2005. Dostupné na [www](http://www.lantronix.com/pdf/XPort_DS.pdf): http://www.lantronix.com/pdf/XPort_DS.pdf.

Termín zadání: 8. 2. 2016

Termín odevzdání: 26.5.2016

Vedoucí práce: doc. Ing. Miloslav Steinbauer, Ph.D.

Konzultanti bakalářské práce:


doc. Ing. Tomáš Kratochvíl, 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.

ABSTRAKT

Tato bakalářská práce se zabývá implementací nejrozšířenějších moderních komunikačních rozhraní USB/LAN/WIFI pro ovládání inkubátoru přímo počítačem a také přes webové rozhraní. V bakalářské práci je popsáno celkové řešení řízení inkubátoru za pomoci mikrokontroléru ATmega8 s možností výměny komunikačních rozhraní. Pro převodník USB byl použit obvod FT232 BM firmy FTDI, od firmy LANTRONIX pro převodník Ethernet byl použit převodník XPORT a nakonec pro komunikaci WIFI byl zvolen obvod ESP 8266.

KLÍČOVÁ SLOVA

Mikrokontrolér, ATmega8, převodník, USB, LAN, WiFi, FT232, Xport, Nano Socket iWiFi, ESP 8266.

ABSTRACT

This thesis deals with the implementation of the most modern communication interfaces USB/LAN/WIFI to control the computer itself and also the incubator via a Web interface. In Bachelor thesis described the overall solution to the management of the incubator for using microcontroller ATmega8 with the possibility of the exchange of communication interfaces. For the USB converter was used the circuit BM FTDI FT232 company, from LANTRONIX Ethernet was used for Converter converter XPORT and eventually to communicate the WIFI was elected circuit ESP 8266.

KEYWORDS

Microcontroller, ATmega8, convertor, USB, LAN, WiFi, FT232, Xport, Nano Socket iWiFi, ESP 8266.

VONEŠ, J. *Implementace komunikačního rozhraní USB/LAN/WIFI*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2016. 36 s. , 9 s. příloh. Bakalářská práce. Vedoucí práce: doc. Ing. Miloslav Steinbauer, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou na téma Implementace komunikačního rozhraní USB/LAN/WIFI jsem vypracoval samostatně pod vedením vedoucího bakalářské práce 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 uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 23.5.2016

.....

(podpis autora)

OBSAH

Seznam obrázků	viii
Seznam tabulek	x
Úvod	1
1 Definice jednotlivých rozhraní	2
1.1 Rozhraní USB	2
1.2 Rozhraní LAN a TCP/IP protokol	3
1.3 Rozhraní WiFi.....	3
1.4 Sériové rozhraní USART	4
1.5 Standart RS 232	5
2 Popis použitých jednotlivých prvků	6
2.1 XPort – Lantronix	6
2.2 Převodník USB	8
2.3 Převodník WiFi.....	10
2.4 Mikrokontrolér ATmega8.....	11
3 Zapojení Hardware	12
3.1 Napájení zařízení	12
3.2 Zapojení USB převodníku	13
3.3 Zapojení TCP/IP převodníku	14
3.4 Zapojení WIFI převodníku	15
3.5 Zapojení mikroprocesoru ATmega8	19
4 Software	21
4.1 Program pro procesor ATmega8.....	21
4.2 Program pro PC	28
5 Závěr	32
Literatura	33
Seznam symbolů, veličin a zkratk	35

A	Návrh zařízení	37
A.1	Obvodové zapojení základní desky	37
A.2	Deska plošného spoje základní desky – top (strana součástek).....	38
A.3	Deska plošného spoje základní desky – bottom (strana spojů)	38
A.4	Osazovací plán základní desky – top (strana součástek)	39
A.5	Deska plošného spoje převodníku LAN – bottom (strana spojů).....	39
A.6	Osazovací plán převodníku LAN – top (strana součástek).....	39
A.7	Deska plošného spoje USB převodníku – bottom (strana spojů)	40
A.8	Osazovací plán USB převodníku – bottom (strana spojů).....	40
A.9	Deska plošného spoje WiFi převodníku – bottom (strana spojů).....	40
A.10	Osazovací plán WiFi převodníku – top (strana součástek).....	41
A.11	Deska plošného spoje WiFi převodníku pro KME control board ATmega16 – top (strana součástek).....	41
A.12	Deska plošného spoje WiFi převodníku pro KME control board ATmega16 – bottom (strana spojů)	41
A.13	Osazovací plán WiFi převodníku pro KME control board ATmega16 – top (strana součástek).....	42
B	Seznam součástek	43

SEZNAM OBRÁZKŮ

Obrázek 1.1 Zapojení konektoru USB do verze 2.0	2
Obrázek 1.2 Příklad odeslání dat po sériovém rozhraní	4
Obrázek 2.1 Detail převodníku sériové linky	6
Obrázek 2.2 Detaily nastavení XPortu	7
Obrázek 2.3 Výpis konfigurace XPortu za pomoci Telnetu	7
Obrázek 2.4 Příklad nastavení komunikačního protokolu zařízení	8
Obrázek 2.5 Doporučené zapojení obvodu FT232RL	9
Obrázek 2.6 Ukázka správně nainstalovaného sériového portu	9
Obrázek 2.7 Modul Nano Socket iWiFi™	10
Obrázek 2.8 WiFi modul pro Arduino	10
Obrázek 2.9 Zapojení vývodů ATmega8	11
Obrázek 3.1 Problém Windows® s načítáním zařízení	12
Obrázek 3.2 Zapojení zdroje	12
Obrázek 3.3 Zapojení modulu USB převodníku	13
Obrázek 3.4 Realizace modulu USB převodníku	14
Obrázek 3.5 Zapojení modulu TCP/IP převodníku	14
Obrázek 3.6 Připojení TCP/IP převodníku pro vyzkoušení komunikace	15
Obrázek 3.7 Rozložení vývodů ESP8266	15
Obrázek 3.8 Zapojení WiFi modulu pro KME control board ATmega16	16
Obrázek 3.9 Zapojení WiFi modulu pro inkubátor	16
Obrázek 3.10 Připojení WiFi modulu v obvodu inkubátoru	16
Obrázek 3.11 Příklad komunikace s modulem v programu Termite	17
Obrázek 3.12 Úspěšně přehrané firmware	18
Obrázek 3.13 Nastavení konfigurace flasheru	18
Obrázek 3.13 Zapojení mikroprocesoru v obvodu inkubátoru	19
Obrázek 3.14 Realizace hotového inkubátoru	20
Obrázek 3.15 Realizace hotového inkubátoru s USB modulem při komunikaci	20
Obrázek 4.1 Vývojový diagram hlavního programu	22
Obrázek 4.2 Vývojový diagram podprogramu komunikace UART	23
Obrázek 4.3 Program pro PC po spuštění	28

Obrázek 4.4 Formulář pro výběr portu	28
Obrázek 4.5 Program pro PC po úspěšném připojení.....	29
Obrázek 4.6 Změna nastavení intervalu	29
Obrázek 4.7 Změna nastavení vlhkosti.....	30
Obrázek 4.8 Výběr WiFi sítě.....	30
Obrázek 4.9 Nastavení sítě LAN	31

SEZNAM TABULEK

Tabulka 1.1 Přehled nejpoužívanějších standardů od roku 1997	4
Tabulka 4.1 Přiřazení zástupných symbolů komunikace.....	21

ÚVOD

Integrace moderních komunikačních rozhraní do aplikací a následné ovládání za pomoci osobního počítače (PC), je v dnešní uspěchané době již standardem. Dálkové řízení, sběr dat, kontrola a následné zpracování hodnot lze tak soustředit do jednoho místa a odtud následně ovládat jedním uživatelem i v aplikacích, ve kterých bylo dříve potřeba nepřetržité a neustálé kontroly více pracovníky. Díky rozvoji mikroprocesorové techniky a komunikačních modulů lze takové zařízení lehce vytvořit a zabudovat do jakéhokoliv zařízení na trhu a to i do zařízení dříve vyrobených.

Tato práce se zaměřuje na implementaci komunikačního rozhraní USB, LAN TCP/IP a WiFi pro ovládání inkubátoru pro chov exotických zvířat za pomoci PC. Efekt projektu spočívá v mikroprocesorem řízené regulaci teploty a vlhkosti a dále posílána přes komunikační prostředí dále do PC. Ovládací program pak umožňuje zobrazovat a nastavovat přímo hodnoty pro řízení ventilátoru a topení připojené k mikroprocesoru.

Obsahem práce je seznámení s použitým komunikačním prostředím i s použitými komunikačními prvky. Celkovým návrhem vlastního řešení komunikace mezi PC a mikrokontrolérem a zapojením hardware včetně návrhu plošného spoje, vnitřního programu do mikrokontroléru a vlastního ovládacího programu pro PC.

Hlavním cílem je sestavení uceleného řešení vlastního projektu, které vychází ze závěrů předchozí semestrální práce. Zhotovení obslužného programu a celkového odladění komunikace tak, aby co nejméně zatěžovala mikrokontrolér i samotné PC při komunikaci, a to především při neaktivním spojení.

1 DEFINICE JEDNOTLIVÝCH ROZHRAŇÍ

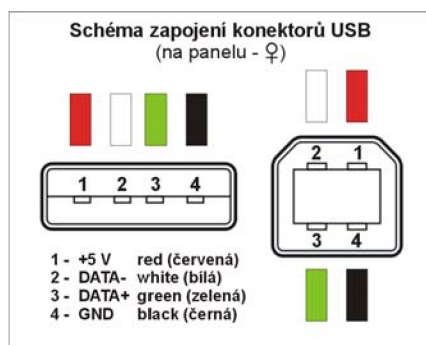
S rozvojem a modernizací komunikačních technologií se na trhu objevuje čím dál více různých rozhraní a protokolů pro komunikaci mezi jednotlivými zařízeními. V následujících kapitolách jsou popsány jednotlivé definice stávajících nejrozšířenějších rozhraní.

1.1 Rozhraní USB

USB – Universal Serial Bus, je univerzální sériová sběrnice, nahrazující všechny dříve používané periférie v počítačích. Výhodou rozhraní USB je možnost připojování a odpojování přístrojů za chodu počítače bez nutnosti restartování. Další výhodou tohoto rozhraní je možnost poskytnout zařízením napájecí napětí 5V o proudu až 500 mA.

Celkově se můžeme setkat se třemi protokoly USB. Verze USB 1.1, v nichž existují pomalá (Low-Speed) zařízení s přenosovou rychlostí 1,5 Mbit/s (187,5 kB/s) a rychlá zařízení (Full-Speed) s rychlostí 12 Mbit/s (1,5 MB/s). USB 1.1 však nebylo schopno konkurovat vysokorychlostním rozhraním proto, byla vytvořena verze USB 2.0, která nabídla maximální rychlost 480 Mbit/s (60MB/s) v režimu Hi-Speed. Třetí verze USB 3.0 (Superspeed USB) disponuje více než 10× větší rychlostí, přenosová rychlost je 5 Gbit/s (625 MB/s). Nová technologie má ale 9 vodičů namísto původních 4, přesto zpětně podporuje USB 2.0. Vylepšená verze (USB 3.1 Gen 2) má přenosovou rychlost 10 Gbit/s, která používá efektivnější kódování dat oproti verzi 3.0.

Maximální garantovaná délka kabelu mezi sousedními zařízeními je 5 m, při použití delšího kabelu nemusí zařízení pracovat správně. Kromě verzí USB 3.0 a vyšší obsahuje kabel 4 vodiče. Dva jsou pro napájení, další je kroucený pár, který slouží pro přenos dat. Sběrnice USB přináší tu výhodu, že při připojení přídavného rozdělovače sběrnice (hub) jsou k dispozici tři nové porty a celkem je tedy možno na USB připojit až 127 zařízení. Nevýhodou pro amatérského vývojáře je velká složitost USB. Na straně přístroje je třeba použít převodník na USB nebo softwarovou knihovnu. Dále je nutné na straně PC instalovat ovladač. Každé zařízení USB má interní číslo dodavatele (vendor ID), které je oficiálně udělováno organizací USB. Zařízení je možno dodávat na trh jen s platným VID.



Obrázek 1.1 Zapojení konektoru USB do verze 2.0

1.2 Rozhraní LAN a TCP/IP protokol

LAN – Local Area Network, znamená lokální síť nebo též místní síť a označuje počítačovou síť. Používá se nejen k propojení počítačů, ale i zařízení spotřební elektroniky jako jsou televizní přijímače, herní konzole a také jako drátové rozhraní pro přístupové body WiFi a zařízení pro přístup k Internetu. Jedná o síť uvnitř místností, budov nebo malých areálů, ve firmách i v domácnostech. Je charakterizována vysokou přenosovou rychlostí (až desítky Gbps).

TCP – Transmission Control Protocol, je primární přenosový protokol. Síťová komunikace je rozdělena do tzv. vrstev, které znázorňují hierarchii činností. Výměna informací mezi vrstvami je přesně definována. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší. Data se přenášejí za pomoci tzv. paketů. Ty kromě dat obsahují navíc další doplňkové informace, jako je informace o odesílateli, příjemci, pořadové číslo paketu nebo kontrolní součet. Tyto data slouží ke správnému složení celé poslané informace a zjištění zda informace přišla nepoškozená.

IP protokol – protokol síťové vrstvy neboli internetový protokol, který provádí vysílání datagramů na základě síťových IP adres obsažených v jejich záhlaví. Datagramy putují sítí nezávisle na sobě a pořadí jejich doručení nemusí odpovídat pořadí ve zprávě. Tento protokol se stará o segmentaci a znovu sestavení datagramů do a z rámců podle protokolu nižší vrstvy. Nejrozšířenější protokol je verze 4 (IPv4), která má několik nedostatků. Ty byly vyřešeny ve verzi 6 (IPv6), která má dostatek IP (128 bit adresy oproti 32 bit u verze 4), dále také vyšší podporu bezpečnosti, ale není zpětně kompatibilní s předchozí verzí.

1.3 Rozhraní WiFi

WiFi – je označení pro bezdrátovou komunikaci v sítích LAN. Technologie využívá bezlicenčního frekvenčního pásma 2,4 GHz. Účelem je zajišťovat propojení jednotlivých přenosných zařízení mezi sebou a dále propojení těchto zařízení do sítí LAN. Je ideální pro budování levných a přitom výkonných sítí bez nutnosti pokládky kabelů.

Bezdrátová síť může být vybudována různými způsoby v závislosti na požadované funkci. Ve všech případech hraje klíčovou roli identifikátor SSID, což je řetězec až 32 ASCII znaků, kterými se jednotlivé sítě rozlišují. SSID identifikátor je v pravidelných intervalech vysílán jako broadcast, takže všichni potenciální klienti si mohou snadno zobrazit dostupné bezdrátové sítě, ke kterým je možné se připojit. [1]

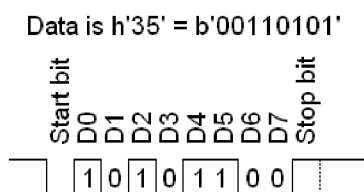
Standard	Pásmo	Maximální rychlost
	[GHz]	[Mbit/s]
původní IEEE 802.11	2,4	2
IEEE 802.11a	5	54
IEEE 802.11b	2,4	11
IEEE 802.11g	2,4	54
IEEE 802.11n	2,4 nebo 5	600
IEEE 802.11y	3,7	54
IEEE 802.11ac	2,4 a 5	1000
IEEE 802.11ad	2,4 , 5 a 60	7000

Tabulka 1.1 Přehled nejpoužívanějších standardů od roku 1997

1.4 Sériové rozhraní USART

Jde o zařízení pro sériovou komunikaci, které lze nastavit buď pro asynchronní režim nebo synchronní režim. Jako jeden ze sériových I/O modulů je USART implementován ve většině MCU firmy Atmel.

Nejčastěji se používá asynchronní přenos, kde se vysílají data na pinu označovaném obvykle jako TX (transmit) a přijímají na pinu RX (receive). Klidová úroveň signálu je log. 1. Vysílání je zahájeno změnou hodnoty signálu na log. 0 po dobu jednoho bitu (tzv. start-bit). Následovně se posílá nejnížší datový bit, poslední nejvýznamnější datový bit je následován stop bitem, který má opět úroveň log. 1. Po odvysílání stop-bitu může začít přenos dalšího bajtu. Na obrázku je to vyznačeno tečkovaně. [2]



Obrázek 1.2 Příklad odeslání dat po sériovém rozhraní

1.5 Standart RS 232

RS 232 – sériový port, komunikační rozhraní počítačů a ostatních zařízení pro přenos informací vytvořené původně pro komunikaci dvou zařízení do vzdálenosti 20 m. Umožňuje vzájemnou sériovou asynchronní komunikaci pomocí dvou párů vodičů se společnou zemí. V sekvenci datových bitů předchází jeden start bit, kterým se logická hodnota na lince přepne z původně klidového stavu do stavu opačného. Po datových bitech následuje paritní bit a za ním jeden nebo více stop bitů, během kterých je linka opět v klidovém stavu. Pořadí přenosu datových bitů je od nejméně významného bitu, po bit nejvýznamnější. Počet datových bitů je volitelný, obvykle se používá 8 bitů, lze se také setkat se 7 nebo 9 bity.

2 POPIS POUŽITÝCH JEDNOTLIVÝCH PRVKŮ

2.1 XPort – Lantronix

Je to převodník sériových linek na 10/100 Base-TX, výkonný webový server a procesor s externí pamětí a obvodem reálného času. Zabudovaný přímo do konektoru RJ45, po přidání do aplikace s minimálním technickým úsilím vytvoří převodník LAN <-> UART.

Bez šifrování pro zabezpečení komunikace s operačním systémem CoBos a 16-bit procesorem pracujícím na 48 MHz nebo 88 MHz. Obsahuje plné sériové rozhraní. Maximální rychlost 921 600 bps. Napájení 3,3 V s maximálním příkonem 0,77 W.



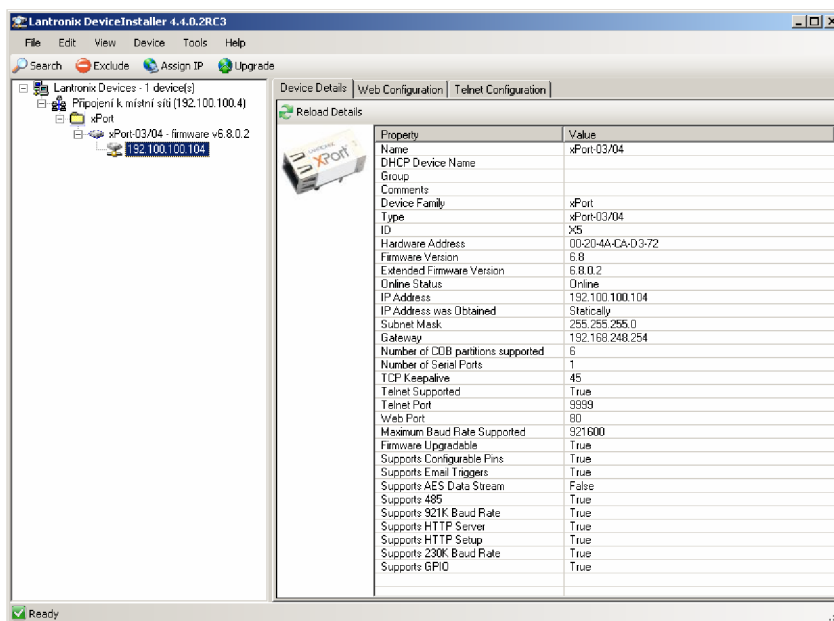
Obrázek 2.1 Detail převodníku sériové linky

2.1.1 Nastavení a připojení ve vývojovém kitu

Převodník po připojení na napájecí napětí a síťový kabel si vyžádá adresu s DHCP serveru, pokud není adresa nastavena pevně. Síťová adresa lze zjistit pomocí nástroje firmy Lantronix – DeviceInstaller.

Pomocí tohoto software lze měnit nejen síťovou adresu, ale také obnovit a zálohovat/aktualizovat firmware převodníku, případně nahrávat vlastní webové stránky. Pokud se převodník připojuje do vlastní sítě, je nutné nastavit IP adresu v rozmezí, která je povolena DHCP serverem tohoto routeru, jinak nebude převodník fungovat správně.

Po úspěšném připojení software vypíše další detaily nastavení:



Obrázek 2.2 Detaily nastavení XPortu

V software lze dále provést konfiguraci převodníku jednak za pomoci webového rozhraní nebo telnetu na portu: 9999, po připojení se vypíše aktuální konfigurace se seznamem možností:

```

IP Address: 192.100.100.104  Port: 9999  Disconnect  Clear  *** E-mail
Mail server: 0.0.0.0
Unit :
Domain :
Recipient 1:
Recipient 2:

- Trigger 1
Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

- Trigger 2
Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

- Trigger 3
Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

Change Setup:
0 Server
1 Channel 1
3 E-mail
5 Expert
6 Security
7 Defaults
8 Exit without save
9 Save and exit  Your choice ?

*** basic parameters
Hardware: Ethernet TPI
IP addr 192.100.100.104, gateway 192.168.248.254, netmask 255.255.255.0
DNS Server 192.168.248.254
DHCP FQDN option: Disabled

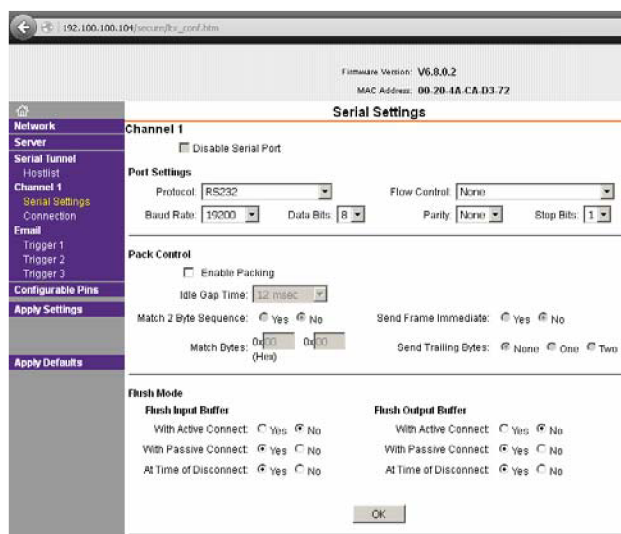
*** Security
SNMP is enabled
SNMP Community Name: public
Telnet Setup is enabled
TFTP Download is enabled
Port 77FEh is enabled
Web Server is enabled
Web Setup is enabled
ECHO is disabled
Enhanced Password is disabled
Port 77F0h is enabled

*** Channel 1
Baudrate 19200, L/F Mode 4C, Flow 00
Port 10001
Connect Mode : 00
Send '+++' in Modem Mode disabled
Show IP addr after 'RMC' disabled
auto increment source port disabled
Remote IP Id: --- none ---, Port 00000
Disconn Mode : 00
Flush Mode : 66

*** Expert
TCP Keepalive : 45s
ARP cache timeout: 600s
CPU performance: Regular
Monitor Mode @ bootup: enabled
RS485 tx enable : active low
HTTP Port Number : 80
SMTP Port Number : 25
MTU Size: 1400
TCP Re-transmission timeout: 500 ms
Alternate MAC: disabled
Ethernet connection type: auto-negotiate
  
```

Obrázek 2.3 Výpis konfigurace XPortu za pomoci Telnetu

Konfiguraci portu pomocí webového rozhraní můžeme provést i přes obyčejný internetový prohlížeč. Po zadání platné IP adresy portu se po zadání přihlašovacího jména a hesla se načte správce zařízení konfigurace serveru.



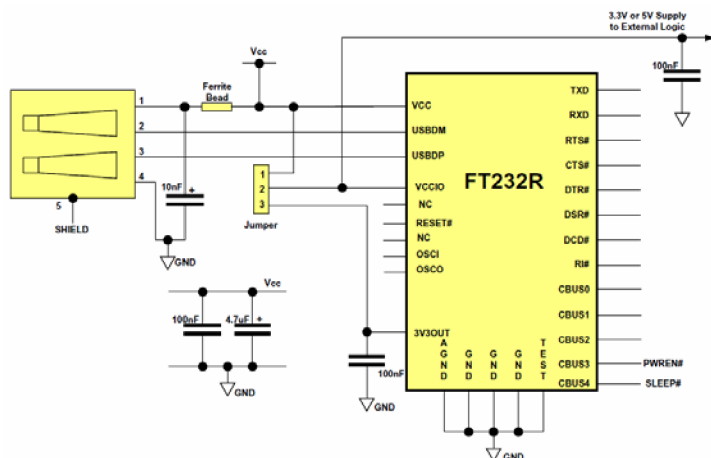
Obrázek 2.4 Příklad nastavení komunikačního protokolu zařízení

2.2 Převodník USB

Základem tohoto převodníku je známý obvod FT232RL od firmy FTDI, která se specializuje na konverzi klasických periférií PC na USB. Tento obvod slouží pro konverzi rozhraní USB \leftrightarrow UART s přenosovou rychlostí 300 Bd až 3 MBd (dle typu). V obvodu je zabudována dvouportová vyrovnávací paměť o velikosti 128 B ve směru od PC k aplikaci a 384 B ve směru k PC. Dále je na čipu paměť EEPROM obsahující základní nastavení obvodu:

- typ obvodu
- USB VID a PID
- USB sériové číslo
- identifikace výrobce
- popis produktu
- oblast vyhrazenou pro uživatelská data

Obvod FT232RL lze napájet přímo z portu USB pro nenáročné aplikace nebo externím napájením v rozmezí 3,3V – 5V. Proudová spotřeba je max. 50 mA při normálním provozu a max. 200 μ A v režimu USB.

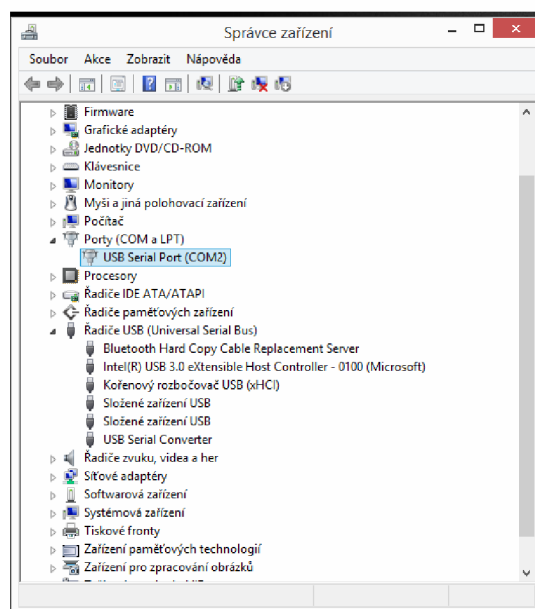


Obrázek 2.5 Doporučené zapojení obvodu FT232RL

2.2.1 Nastavení a připojení do PC

Po doporučeném zapojení obvodu dle výrobce čipu a připojení zařízení do PC, si operační systém vyhledá příslušné ovladače. USB Serial Converter a dále virtuální USB sériový Port.

Pokud operační systém přiřadí zařízení vysoké číslo portu a aplikace umožňuje nastavení portu pouze do COM4 je potřeba ve Správci zařízení tento port upravit na nižší neobsazené číslo.



Obrázek 2.6 Ukázka správně nainstalovaného sériového portu

2.3 Převodník WiFi

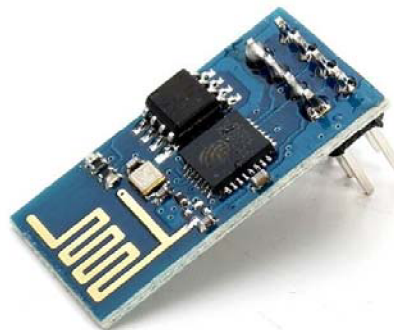
Na internetu se nabízí velice mnoho profesionálních modulů, které obsahují už kompletní řešení. Příkladem je Izraelská firma Connect One, která tyto bezdrátové WiFi moduly má vybavené integrovaným web serverem. Mají bohatou nabídku rozhraní (UART, SPI a USB). Miniaturní moduly této firmy dovolují nasazení bezdrátové WiFi konektivity bez jakékoli znalosti IP stacku. Po prozkoumání všech modulů od této firmy se pro montáž v amatérských podmínkách jeví jako nejvýhodnější produkt Nano Socket iWiFi™, protože má vhodně rozmístněné kolíky do PCB. Ostatní moduly mají buď speciální konektor nebo montáž SMT [3]. Nevýhodou těchto modulů je však jejich špatná dostupnost na trhu a vysoká cena [4].



Obrázek 2.7 Modul Nano Socket iWiFi™

Při hledání jiného snadnějšího a dostupnějšího řešení se nabízí WIFI modul ESP8266 pro platformu Arduino, který nabízí kompletní a ucelené Wi-Fi síťové řešení. Mezi některé parametry tohoto modulu patří například: SDIO 2.0, SPI, UART, integrovaný RF přepínač, balun, 24dBm PA, DCXO, a PMU, integrovaný RISC procesor, on-chip paměť, externí paměťové rozhraní, integrované šifrování a zabezpečení WEP, TKIP, AES, a WAPI s podporou APSD pro VoIP aplikace.

Co se týká parametrů sítě pracuje v pásmu 802.11 b / g / n, s podporou 3 režimů (AP, STA, AP + STA). Výstupní výkon je v režimu 802.11 b + 19,5 dBm, spotřeba v pohotovostním režimu je menší jak 1 mW. Integrovaný 32-bit procesor může být použit jako procesor pro vlastní aplikaci, dále obsahuje vestavěný teplotní senzor. Napájení je 3,3 V a rozměry vlastní desky jsou 25 mm × 14 mm.

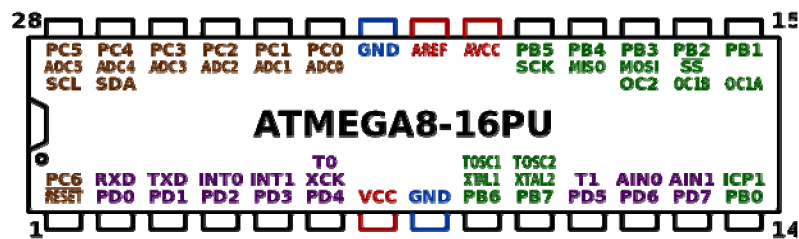


Obrázek 2.8 WiFi modul pro Arduino

2.4 Mikrokontrolér ATmega8

Mikrokontrolér AVR je z produkce firmy Atmel. Jedná se o obvod s architekturou RISC, u které není oddělena vnitřní krystalová časovací frekvence. Předností tohoto mikrokontroléru je, že při relativně nízké pracovní frekvenci zůstává odběr proudu velmi malý.

Instrukční sada obsahuje 120 instrukcí. Typ programovatelné paměti je FLASH o velikosti 8 KB. Dále obsahuje datovou paměť typu EEPROM o velikosti 512×8, velikost jádra 8 Bit, vzorkovací frekvence 16 MHz. Má 23 vstupů/výstupů a obsahuje sběrnice I²C, SPI, UART/USART a A/D převodník 6×10b. Rozsah napájecích napětí je 4,5V – 5,5V.



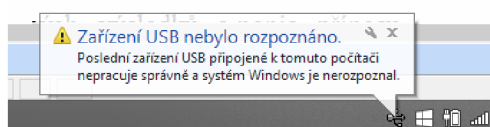
Obrázek 2.9 Zapojení vývodů ATmega8

3 ZAPOJENÍ HARDWARE

Realizace celého zařízení vychází ze zapůjčených přípravků a komponentů vedoucím. Jednalo se o vývojovou desku s označením KME control board ATmega16, vývojový kit firmy Lantronix s modulem Xport a převodník USB – RS232.

Jak KME control board ATmega16, tak i převodník USB – RS232 nebyl pod operačním systémem Windows 8.1[®] i Windows 10[®] rozpoznán. Nebyl načten správný VID a PID zařízení a nastavení ovladače přímo způsobilo, že zařízení nešlo spustit.

S největší pravděpodobností je to způsobeno tím, že novější verze Windows[®] nedokáží spolupracovat s obvodem FT232RL, které jsou osazené v obou přípravcích. Problémem může také být jejich obvodovým zapojením. Dále to může být způsobeno, že přípravky jsou osazené padělanými obvody FT232RL, kdy originální ovladač přepíše PID zařízení na 0000 a tím znemožní operačnímu systému rozeznat a přiřadit správný ovladač. [5] [6] To jsou důvody, proč bylo při řešení použito zapojení s obvodem FT232BM, které fungovalo bez problémů.

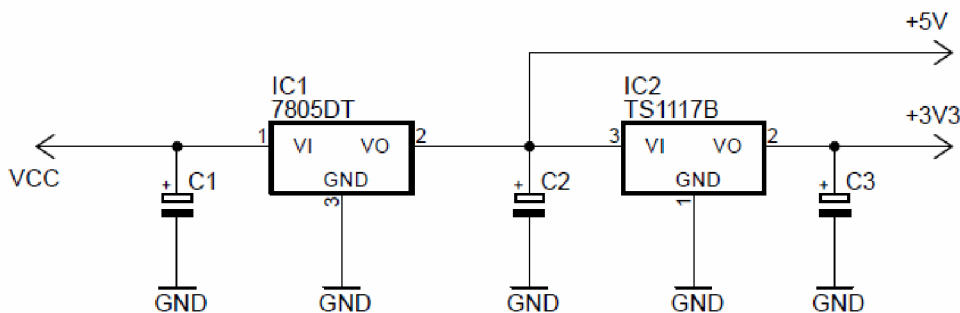


Obrázek 3.1 Problém Windows[®] s načítáním zařízení

Celkové zapojení je řešeno modulárně. Mikroprocesor v podstatě jako ovládací zařízení inkubátoru se zdrojem na jedné základní desce plošných spojů, do kterých se modulárně připojí vždy rozhraní, které bude využito pro přenos dat. (USB, LAN, WIFI)

3.1 Napájení zařízení

Popsané použité obvody a zařízení mají rozdílné napájecí napětí. Zejména pak Xport nebo WiFi modul se svými 3,3 V a ostatní zařízení 5 V. Použity byly lineární stabilizátory, které zaručí udržení výkonu při plném zatížení. Napájení je v klasickém zapojení.



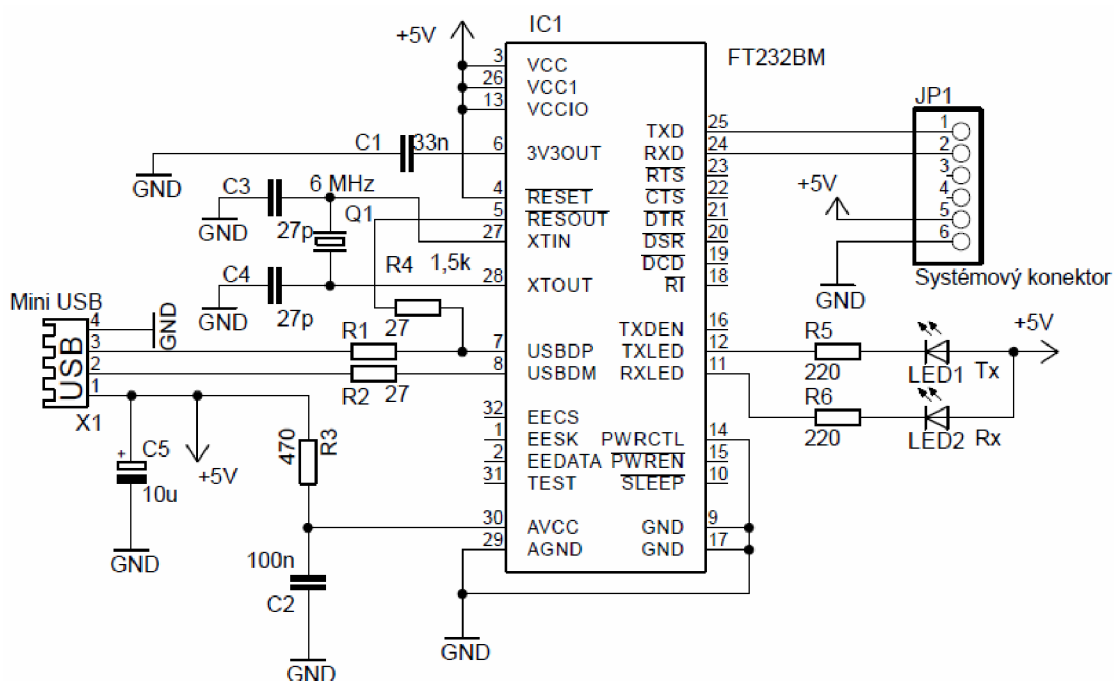
Obrázek 3.2 Zapojení zdroje

3.2 Zapojení USB převodníku

Jeho základ tvoří katalogové zapojení firmy FTDI pro FT232BM. To je však upravené pro spolehlivé načtení ve všech operačních systémech Windows[®]. U obvodu FT232BM nehrozí přepsání PID a VIP ovladačem, protože neobsahuje interní EEPROM.

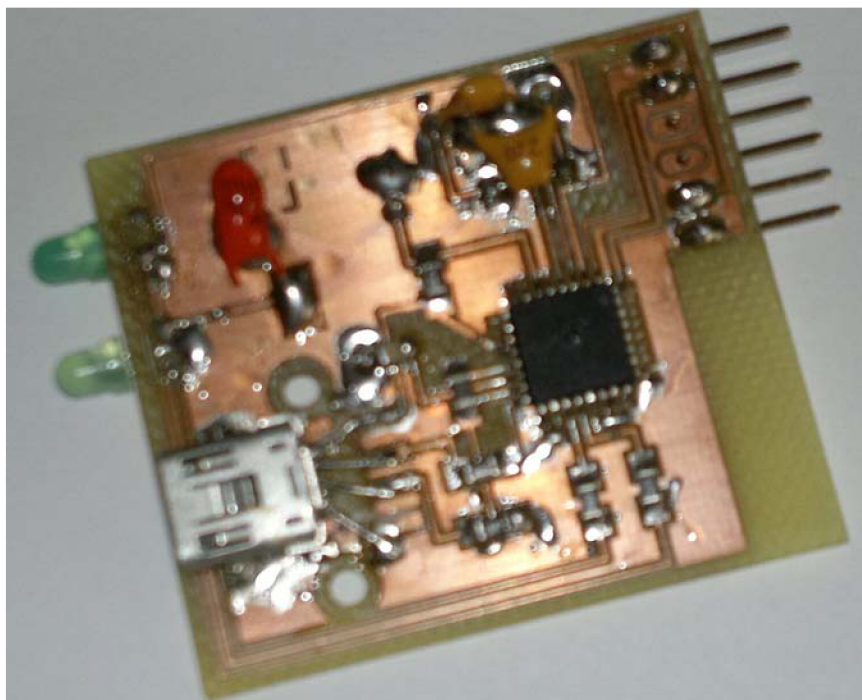
Externí EEPROM lze v případě potřeby připojit na vývody 1, 2 a 32 a pak lze po naprogramování k tomuto obvodu přistupovat jako k vlastnímu USB přípravku. Je však nutné použít jiné ovladače nebo napsat ovladače vlastní. Výhodou tohoto zapojení je, že po připojení tohoto zařízení do PC, které obsahuje již příslušné upravené ovladače si ovládací program toto zařízení sám najde a umí je obsluhovat.

Při zapojení obvodu podle obrázku 3.3 zařízení funguje v PC jako virtuální COM port a není nutné žádné ovladače upravovat. Rovnou lze použít originální ovladače od firmy FTDI, které si novější verze Windows[®] zvládnou automaticky stáhnout.



Obrázek 3.3 Zapojení modulu USB převodníku

Převodník byl realizován na jednostranné desce plošných spojů. Systémový konektor, krystal a LED diody jsou umístěny ze strany součástek. Ty jsou navíc vyhnuté před konektor USB na čelní stranu modulu.



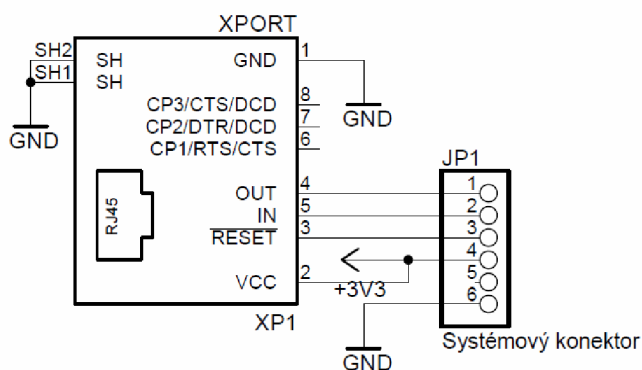
Obrázek 3.4 Realizace modulu USB převodníku

3.3 Zapojení TCP/IP převodníku

Jelikož je vše integrováno v konektoru RJ45, není potřeba řešit zbylé součástky a pouze vyvést použité vývody tohoto převodníku na konektor modulu.

Důležitý je pro správnou funkci XPortu odvod tepla. Plošný spoj by měl mít přibližně 2,5 dm² mědi připojeno na stínění. Kryt Xportu je významným zdrojem tepla a mohlo by tak dojít ke zničení zařízení. [7]

Po delší době zapojení ve vývojové desce se Xport začíná významně zahřívat a dochází k výpadkům konektivity. Navrhnutá deska v příloze A4, tento problém dostatečně eliminuje.



Obrázek 3.5 Zapojení modulu TCP/IP převodníku

Navrhnutá deska je určena pro přímé zapájení Xportu na desku plošných spojů. Poskytnutý převodník lze, ale připojit pro vyzkoušení komunikace přes vývojovou desku firmy Lantronix. Na vývojové desce jsou vyvedeny potřebné vývody Tx, Rx, GND i 3,3V, proto stačí tyto vývody propojit s vlastní deskou inkubátoru.

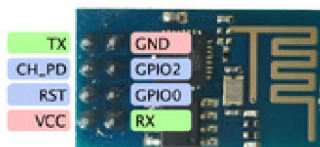


Obrázek 3.6 Připojení TCP/IP převodníku pro vyzkoušení komunikace

3.4 Zapojení WIFI převodníku

Použitý modul má na sobě všechny součástky a vyveden PCB konektor jedná se zde pouze o redukci zapojení konektorů. Navíc s propojením vývodu CH_PD na VCC jinak se modul vůbec nepřihlásí.

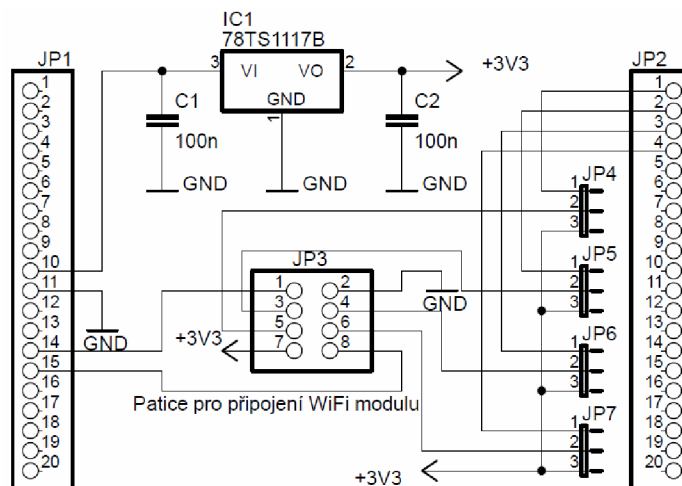
Vývody tohoto modulu dokáží komunikovat v napěťových úrovních 3,3V – 5V a tak pro připojení do aplikace inkubátoru není potřeba žádný napěťový převodník. Napájení tohoto převodníku však musí být přesně 3,3V, protože při dlouhodobém namáhání napájecím napětím 5V se modul začne přehřívat a hrozí jeho zničení.



Obrázek 3.7 Rozložení vývodů ESP8266

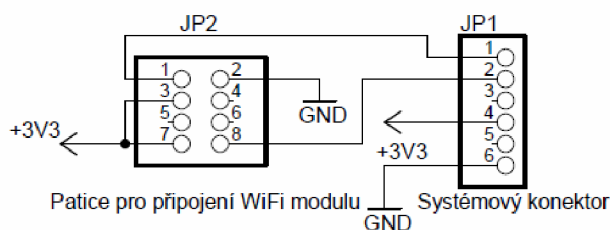
Na trhu je dostupných mnoho těchto modulů z nichž některé nefungují pokud zbývající vývody nejsou připojeny na log.1. [8] Je v návrhu pro KME kontrol board ATmega16 umožněno propojení těchto vývodů na 3,3V.

Redukce jsou navrženy dvě. První pro použití v KME kontrol board ATmega16, která nebyla realizována, protože KME kontrol board ATmega16 nefunguje pod operačním systémem Windows 10[®]. KME kontrol board ATmega16 je napájen zdrojem 5V, v zapojení je tedy přidán stabilizátor napětí na 3,3V. Přes propojky jsou na port A (PA0 – PA4) vyvedeny piny z WiFi modulu pro budoucí použití v aplikacích s tímto zařízením.

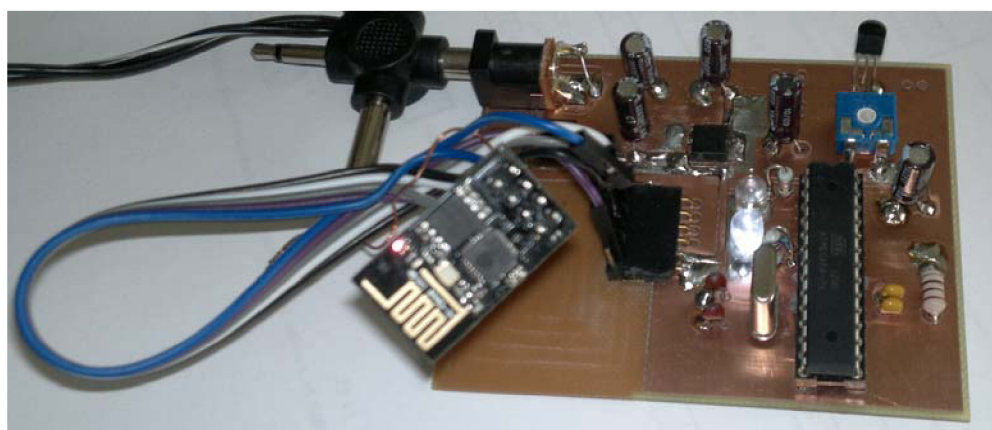


Obrázek 3.8 Zapojení WiFi modulu pro KME control board ATmega16

Druhá redukce je pro zde popisovaný inkubátor.



Obrázek 3.9 Zapojení WiFi modulu pro inkubátor



Obrázek 3.10 Připojení WiFi modulu v obvodu inkubátoru

U modulu ESP8266 je, ale nutné před použitím ve vlastní aplikaci vyzkoušet jeho funkčnost a rychlost komunikace. V případě, že modul komunikuje pouze na rychlosti 115 200 baudů, je nutné pro využití v aplikaci inkubátoru provést přehrání vnitřního firmware starším.

Pro komunikaci s modulem ESP8266 je potřeba použít převodník USB <-> UART, který poskytuje dostatečné napájení 3,3V. Také lze využít převodník USB <-> UART popsany v kapitole 3.2, u kterého zapojíme piny TX a RX do kříže a pin CH_PD připojit na VCC.

V jakémkoliv sériovém terminálu nejčastěji na rychlosti 115200 baudů otestujeme za pomoci AT příkazů, zda je modul funkční a komunikuje. [9]

Ukázka a vysvětlení komunikace v programu Termite 3.2 [10]

- Otestování, zda modul komunikuje:

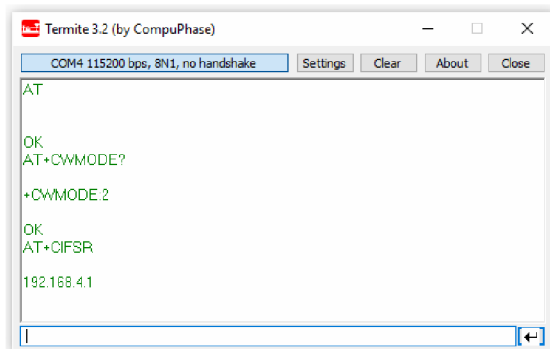
AT >>> OK

- V jakém módu se nachází modul (1 = klient, 2 = AP, 3 = kombinovaný):

AT+CWMODE? >>> +CWMODE:2 >>> OK

- Jakou modul obdržel IP adresu:

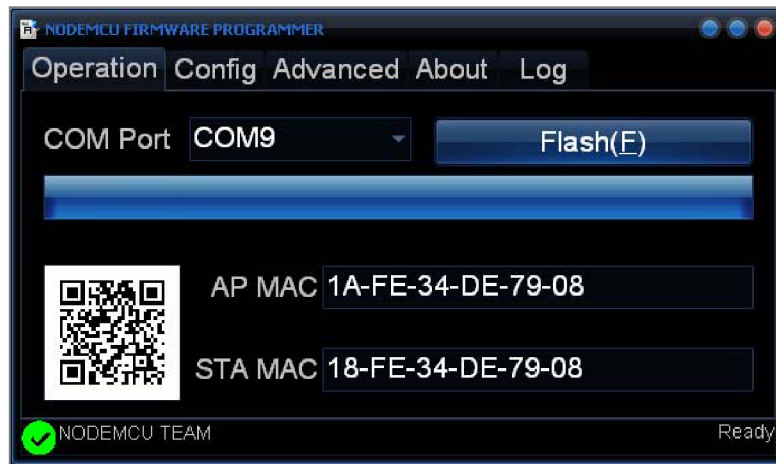
AT+CIFSR >>> 192.168.4.1 >>> OK



Obrázek 3.11 Příklad komunikace s modulem v programu Termite

Pokud modul nevypisuje výše zmíněné, ale pouze nesmyslné znaky, komunikuje na jiné rychlosti, která lze v tomto programu nastavit v rozmezí 1 200 – 230 400 baudu. K dosažení požadované komunikace 9 600 baudu, změníme firmware za pomoci Nodemcu flasheru [11] a staršího firmware v0.9.5.2 [12].

Je nutné správně nastavit sériový port a cestu k souboru. Na modulu ESP8266 kromě propojení CH_PD na VCC, propojit vývod GPIO0 na GND. Po odpojení napájení a rozpojení vývodů GPIO0 a GND a znovu připojení na napájecí napětí včetně pinu CH_PD je již modul ESP8266 připraven na použití v aplikaci spolehlivě fungující na rychlosti 9 600 baudu.



Obrázek 3.12 Úspěšně přehrané firmware



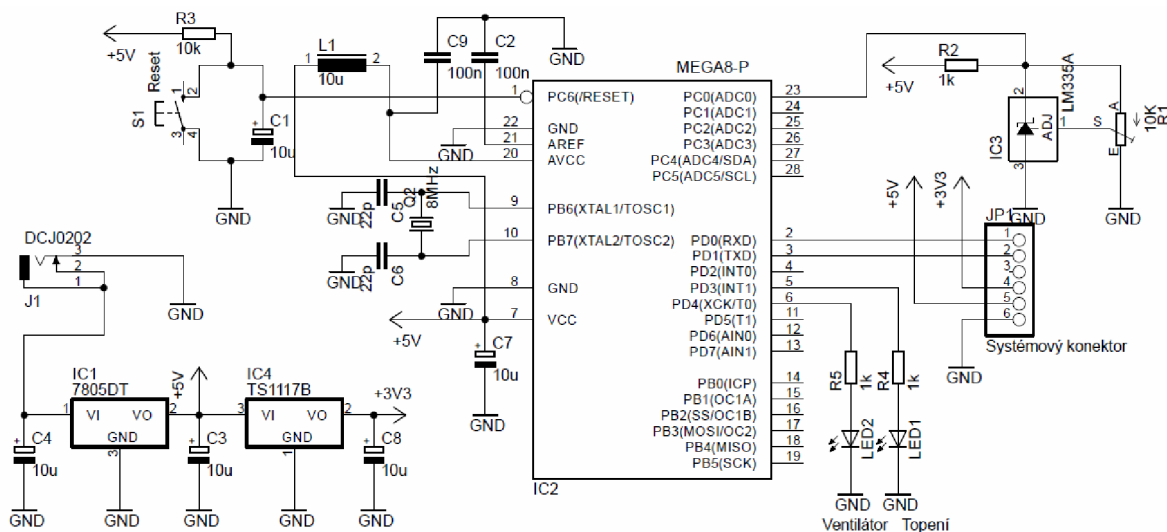
Obrázek 3.13 Nastavení konfigurace flasheru

3.5 Zapojení mikroprocesoru ATmega8

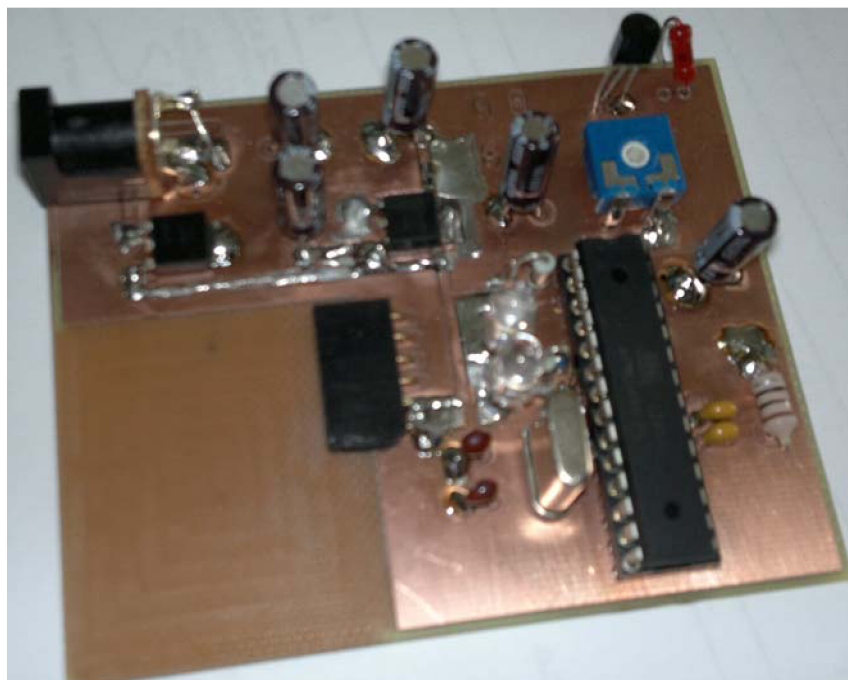
Tato práce se zabývá konstrukcí komunikace mikroprocesoru a PC, aby mohla být využita pro řízení inkubátoru pro odchov exotických zvířat, kde se bude nastavovat teplota a vlhkost, která se dále bude měřit a posílat do PC, s minimálním zásahem uživatele.

Při první myšlence, by data z PC stačilo zapisovat a číst z vnitřní paměti, ale nelze na první pohled zjistit, jestli komunikace opravdu funguje a jestli byla na správné místo do paměti zapsána hodnota. Proto je obvod doplněn o teplotní čidlo společně s dvěma LED simulující jak sepnutí relé ventilátoru, tak sepnutí topení pro udržení správného klimata v inkubátoru.

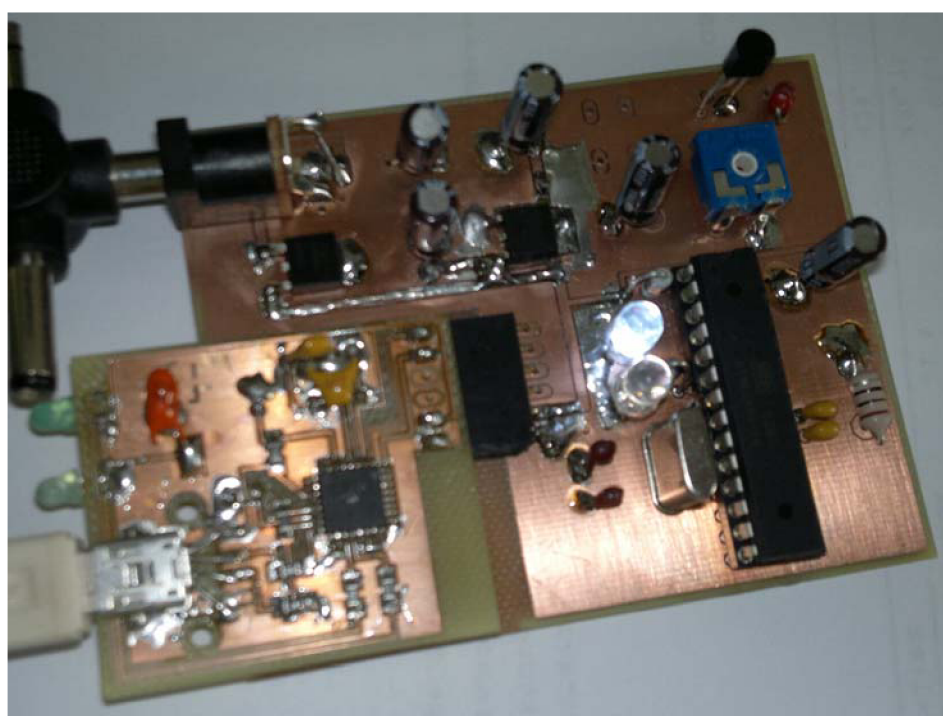
Zapojení lze tedy rozdělit do tří částí: podpůrné obvody (LED a teplotní čidlo), konektor pro připojení modulu převodníku uvedeného v předchozích kapitolách pro komunikaci z PC a vlastní mikroprocesor, který zařizuje pravidelné měření analogových hodnot, příjem povelů z PC a odesílání naměřených hodnot zpět do PC.



Obrázek 3.13 Zapojení mikroprocesoru v obvodu inkubátoru



Obrázek 3.14 Realizace hotového inkubátoru



Obrázek 3.15 Realizace hotového inkubátoru s USB modulem při komunikaci

4 SOFTWARE

Při návrhu software bylo důležité uvědomit si, co je pro inkubátor důležité. Je to hlavně hlídání teploty a vlhkosti a od toho se odvíjející spínání topení a větrání. Program by měl ventilaci a topení ovládat samočinně na základě nastavených dat. Tyto data, je potřeba číst a nastavovat dálkově za pomoci PC.

4.1 Program pro procesor ATmega8

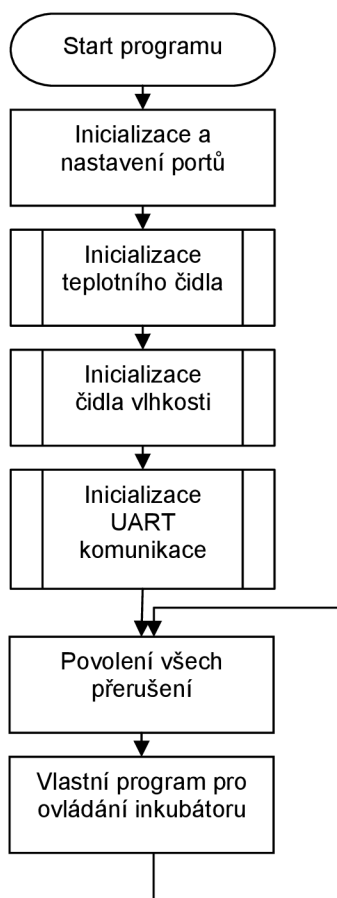
U procesoru bylo nutné nastavit komunikaci tak, aby nebyla zbytečně zatěžována komunikační linka. O celou obsluhu inkubátoru se stará vnitřní program, kterému jsou předána vstupní data a zároveň vyčtena, vždy když o to PC požádá.

Aby byla komunikace co nejjednodušší a nedesílalo se velké množství dat (dlouhých řetězců), bylo jednoduší znakům přiřadit jednotlivé zástupné symboly, jak je uvedeno v tabulce 4.1.

Povel procesu	Zástupný znak
Začátek komunikace	<
Ventilátor zapnutí	V
Ventilátor vypnutí	v
Topení zapnutí	T
Topení vypnutí	t
Teplota uvnitř inkubátoru	C
Nastavení teploty v inkubátoru	c
Vlhkost uvnitř inkubátoru	H
Nastavení vlhkosti v inkubátoru	h
Dotaz o stavu zařízení	D
Konec komunikace	>

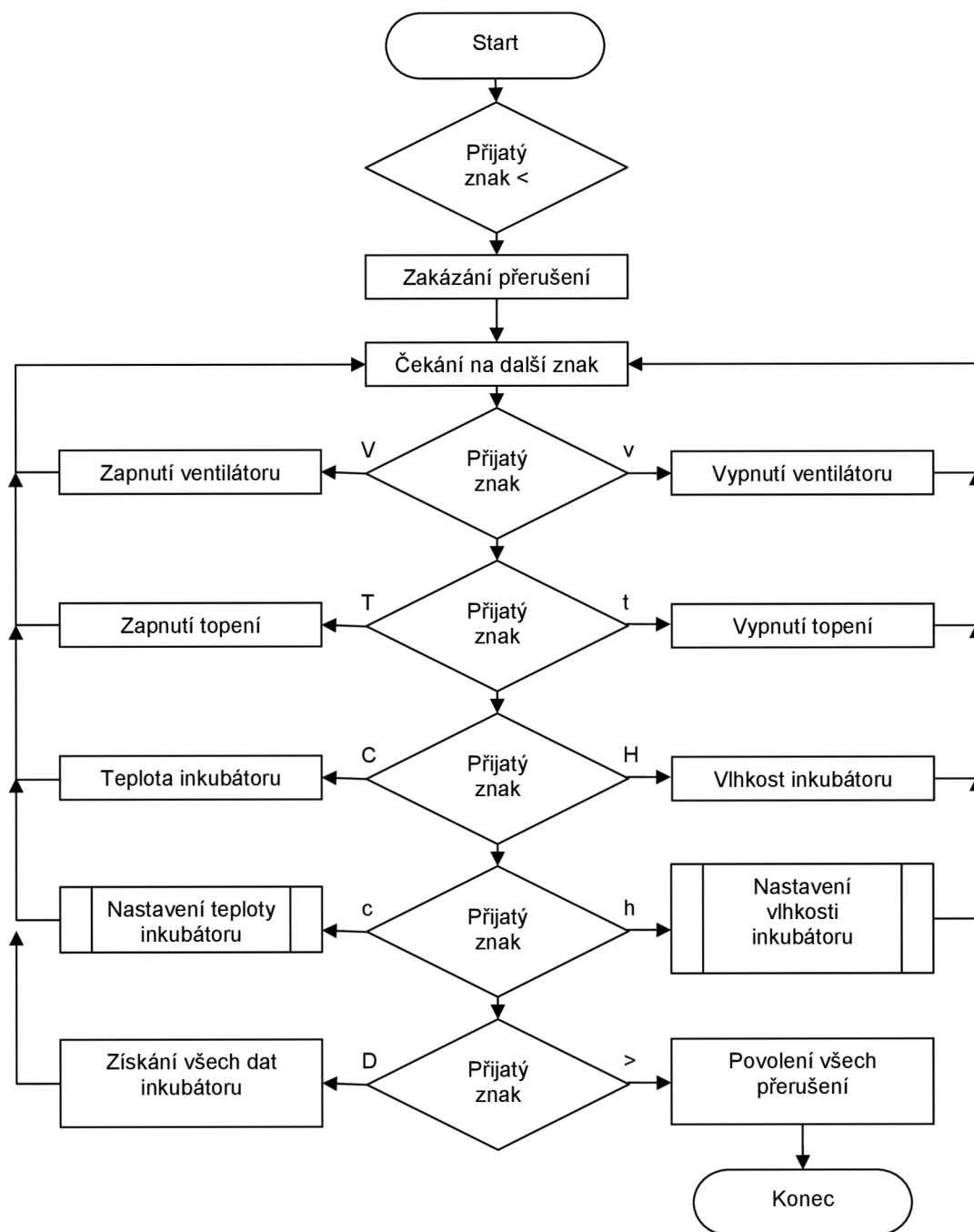
Tabulka 4.1 Přiřazení zástupných symbolů komunikace

Pro vysvětlení komunikace je uveden následující příklad, pokud dojde k vyslání sekvence 3 znaků: <T> znamená to pro mikroprocesor, že okamžitě přeruší právě probíhající program a zároveň dojde k sepnutí ventilátoru při současném vypnutí topení. Pro úpravu teploty v programu inkubátoru se vyšle sekvence znaků <c24>, program nastaví přímo v programu přes vnitřní paměť EEPROM udržování teploty na hodnotu 24 °C. Dále je doplněn dotaz o stavu zařízení <D>, mikroprocesor by měl odeslat zpět do počítače sekvenci znaků: např. <vTC24H89>, tím obslužnému programu oznámí ventilátor je vypnut, topení je zapnuto, vnitřní teplota je 24°C a vlhkost 89 %. Tuto sekvenci vysílá obslužný program na straně PC tak často, jak si uživatel nastaví dobu pro vykonávání tohoto dotazu.



Obrázek 4.1 Vývojový diagram hlavního programu

Na obrázku 4.2 je grafické znázornění podprogramu UART komunikace, kde další dva podprogramy „nastavení teploty a vlhkosti inkubátoru“ fungují podobně. Po přijatém znaku „c“ nebo „h“ program čeká na další dva číselné znaky a uloží je do předem definované paměti vlastního řídicího programu inkubátoru. Jakmile obdrží znak „>“ opět povolí všechna přerušení a podprogram se ukončí.



Obrázek 4.2 Vývojový diagram podprogramu komunikace UART

Výsledný program byl vytvořen za pomoci Atmel Studia 7.0 a dále pomoci stránek [13] [14] a vypadá takto:

```
/*
 * Komunikace USART.c
 *
 * Vytvořeno: 17. 04. 2016 14:50:20
 * Autor : Jaroslav Voneš
 */

#define F_CPU 8000000 // krystal 8 MHz
#include <avr/io.h> // knihovna AVR pro Input/output
#include <util/delay.h> // hlavičkový soubor pro zpoždění
#include <avr/interrupt.h> // hlavičkový soubor pro přerušení
#include <avr/eeprom.h> // hlavičkový soubor pro zápis vnitřní paměti EEPROM

#define BAUD 9600 // rychlost 9600 Baud
#define MYUBRR F_CPU/16/BAUD-1 // vypočtení přenosové rychlosti

volatile int analogova_hodnota; // teplota pro čtení z A/D převodníku
volatile int analogova_hodnota_h; // vlhkost pro čtení z A/D převodníku
volatile int nastav_hodnota; // teplota pro nastavení do paměti první znak
volatile int nastav_hodnota2; // teplota pro nastavení do paměti druhý znak
volatile int nastav_hodnota_h; // vlhkost pro nastavení do paměti první znak
volatile int nastav_hodnota_h2; // vlhkost pro nastavení do paměti druhý znak

void USART_Init( unsigned int ubrr) // inicializace komunikace USART
{
    /* nastavení rychlosti (Baud) */
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char)ubrr;

    /* Povolení vysílače a přijímače */
    UCSRB = (1<<RXIE)|(1<<RXEN)|(1<<TXEN);

    /* Nastavení formátu: asynchronní, žádná parita, data 8-bit, 1 stop bit */
    UCSRC = (1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);
}

int main(void)
{
    /* nastavení portů */
    DDRC = 0x00; // Port C jako vstup
    DDRD = 0xFF; // Port D jako výstup
    ADMUX = 0x60; // Nastavení analogového vstupu na PC0 - měření teploty
    ADCSRA = 0x8F; // Nastavení A/D převodníku
    SFIOR = 0x00; // Zdroj spouštění A/D převodníku

    USART_Init ( MYUBRR ); // inicializace komunikace USART
    sei(); // povolení všech přerušení

    while (1)
    {
        ADCSRA |= 0b01000000; // spuštění A/D převodu
        _delay_ms(50);

        nastav_hodnota_h = eeprom_read_dword (0x00);
        // Načte do proměnné hodnotu z EEPROM
        nastav_hodnota_h2 = eeprom_read_dword (0x20);
        // Načte do proměnné hodnotu z EEPROM
    }
}
```

```

        nastav_hodnota = eeprom_read_dword (0x30);
// Načte do proměnné hodnotu z EEPROM
        nastav_hodnota2 = eeprom_read_dword (0x40);
// Načte do proměnné hodnotu z EEPROM
        _delay_ms(50);
    }
    return(1);
}

ISR(ADC_vect)                // přerušení po ukončení A/D převodu
{
    analogova_hodnota = ADCH;
}

ISR(USART_RXC_vect)         // Přerušení po přijetí znaku
{
    if(UDR=='<')           // Pokud přijde znak <
    {
        cli();             // Zakázání všech přerušení
nacti_dalsi_znak:
        while (!(UCSRA & (1<<RXC)));    // Přijetí dalšího znaku
        switch UDR
// Podle druhu znaku se provede výběr podle přijatého znaku
        {
            case 'V':        // V - Příkaz k zapnutí ventilátoru
                PORTD = (1<<PIND4);      // Sepnutí ventilátoru
                break;

            case 'v':        // v - Příkaz k vypnutí ventilátoru
                PORTD = (0<<PIND4);      // Vypnutí ventilátoru
                break;

            case 'T':        // T - Příkaz k zapnutí topení
                PORTD = (1<<PIND3);      // Sepnutí ventilátoru
                break;

            case 't':        // t - Příkaz k vypnutí topení
                PORTD = (0<<PIND3);      // Vypnutí topení
                break;

            case 'C':        // C - Dotaz na teplotu inkubátoru
                UDR='<';      // Zapiše první znak < pro odesílání
                while (!(UCSRA & (1<<UDRE)));
                UDR='C';      // Zapiše druhý znak C pro odesílání
                while (!(UCSRA & (1<<UDRE)));
                UDR=analogova_hodnota;
// Zapiše se aktuální hodnota teploty odečtená z A/D převodníku
                while (!(UCSRA & (1<<UDRE)));
                UDR='>';      // Zapiše poslední znak > pro odesílání
                break;

            case 'c':
// c - Příkaz k nastavení teploty inkubátoru
                while (!(UCSRA & (1<<RXC)));
// Čekání na příjem dalšího znaku (teploty)
                nastav_hodnota = UDR;
// Hodnota registru se se uloží do proměnné

```

```

        while (!(UCSRA & (1<<RXC)));
// Čekání na příjem dalšího znaku
        nastav_hodnota2 = UDR;
// Hodnota registru se se uloží do 2 proměné
        while (!(UCSRA & (1<<RXC)));
// Čekání na příjem dalšího znaku
        if(UDR=='>')
// Pokud přijme znak > provede se následující instrukce
        eeprom_write_dword (0x30, nastav_hodnota);
// Obsah je zapsán do paměti EEPROM na adresu 0x30
        eeprom_write_dword (0x40, nastav_hodnota2);
// Obsah je zapsán do paměti EEPROM na adresu 0x40
        break;

        case 'H':          // H - Dotaz na vlhkost inkubátoru
            UDR='<';      // Zapiše první znak < pro odesílání
            while (!(UCSRA & (1<<UDRE)));
            UDR='H';      // Zapiše druhý znak H pro odesílání
            while (!(UCSRA & (1<<UDRE)));
/*
v případě, že by byl vzorek osazen i čidlem vlhkosti by se použila následující
část kódu:

        UDR=analogova_hodnota;

pro demonstraci předávání nastavovacích hodnot do paměti a z paměti je použit
následující kód:
*/
        UDR=nastav_hodnota_h;
// Zapiše aktuální hodnota vlhkosti odečtená z EEPROM
        while (!(UCSRA & (1<<UDRE)));
        UDR=nastav_hodnota_h2;
// Zapiše aktuální hodnota vlhkosti odečtená z EEPROM
        while (!(UCSRA & (1<<UDRE)));
/*
Je to vhodnější skrze kolize vlastního programu inkubátoru, který si vykonává sám
měření a regulaci a vždy aktuální teplotu či vlhkost uloží do paměti. Teplotu i
vlhkost, kterou má docílit si zase z jiné části paměti vyčte. Tím nebude narušeno
zasahování PC do běhu vlastního programu inkubátoru.
*/
        UDR='>';        // Zapiše poslední znak > pro odesílání
        break;

        case 'h':
// h - Příkaz k nastavení vlhkosti inkubátoru
        while (!(UCSRA & (1<<RXC)));
// Čekání na příjem dalšího znaku (vlhkosti)
        nastav_hodnota_h = UDR;
// Hodnota registru se se uloží do proměné
        while (!(UCSRA & (1<<RXC)));
// Čekání na příjem dalšího znaku
        nastav_hodnota_h2 = UDR;
// Hodnota registru se se uloží do proměné
        while (!(UCSRA & (1<<RXC)));
// Čekání na příjem dalšího znaku
        if(UDR=='>')
// Pokud přijme znak > provede se následující instrukce

```

```

        eeprom_write_dword (0x00 ,nastav_hodnota_h);
// Obsah je zapsán do paměti EEPROM na adresu 0x00
        eeprom_write_dword (0x20 ,nastav_hodnota_h2);
// Obsah je zapsán do paměti EEPROM na adresu 0x20
        break;

        case 'D':          // D - Dotaz na stav inkubátoru
            UDR='<';      // Zapiše první znak < pro odesílání
            while (!(UCSRA & (1<<UDRE)));
                if( bit_is_clear (PIND, 4)) UDR='v';
// Pokud je vypnutý ventilátor nastaví druhý znak v pro odesílání
                else UDR='V';      // Jinak druhý znak je V
            while (!(UCSRA & (1<<UDRE)));
                if( bit_is_clear (PIND, 3)) UDR='t';
// Pokud je vypnutý ventilátor nastaví třetí znak t pro odesílání
                else UDR='T';      // Jinak druhý znak je T
            while (!(UCSRA & (1<<UDRE)));
                UDR='C';
// Zapiše další znak C pro odesílání
            while (!(UCSRA & (1<<UDRE)));
/*
Pro demonstraci se načítá hodnota přímo z PORTu C0
*/
                UDR=analogova_hodnota;
// Zapiše aktuální hodnota teploty odečtená z A/D převodníku
            while (!(UCSRA & (1<<UDRE)));
                UDR='H';
// Zapiše druhý znak H pro odesílání
            while (!(UCSRA & (1<<UDRE)));
                UDR=nastav_hodnota_h;
// Zapiše aktuální hodnotu vlhkosti načtenou z EEPROM
            while (!(UCSRA & (1<<UDRE)));
                UDR=nastav_hodnota_h2;
// Zapiše aktuální hodnotu vlhkosti načtenou z EEPROM
            while (!(UCSRA & (1<<UDRE)));
                UDR='>';      // Zapiše poslední znak > pro odesílání
            break;

        case '>':          // > - ukončení komunikace
            goto posledni_znak; // skočí za smyčku načítání znaků
            break;

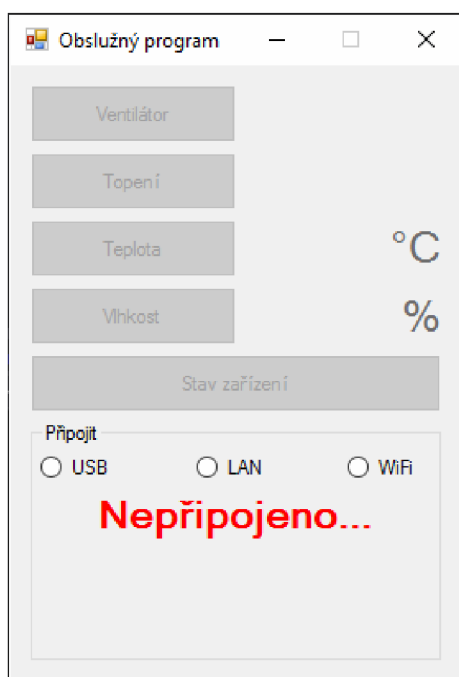
    }
    goto nacti_dalsi_znak;

}
posledni_znak:
    sei();                // povolení všech prerušení
}

```

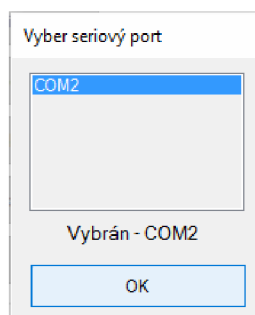
4.2 Program pro PC

Při návrhu programu byla preferována uživatelská vstřícnost a jednoduchost obsluhy zařízení. Po zapnutí programu uživatel nejprve vybere typ připojení USB, LAN nebo WiFi. Při úspěšném zapojení inkubátoru (vlození správného modulu pro připojení) a úspěšném připojení vypíše program „Připojeno na...“ případně vypíše červeně chybovou hlášku.



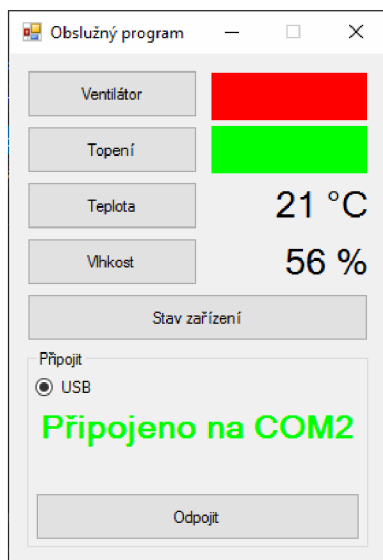
Obrázek 4.3 Program pro PC po spuštění

Nejprve uživatel musí vybrat příslušné připojení. Při vložení modulu USB klikne na USB. Otevře se dialogové okno se všemi dostupnými sériovými porty v PC. Uživatel vybere příslušný port a potvrdí.



Obrázek 4.4 Formulář pro výběr portu

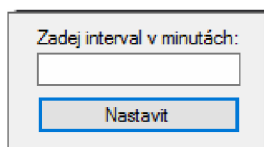
Při úspěšném připojení vypíše připojení na příslušný port a povolí příslušná tlačítka, současně provede dotaz na stav zařízení. V obslužném programu se rozsvítí příslušný stav. Červená – vypnuto, zelená – zapnuto. Zároveň program ukazuje hodnoty aktuální teploty a vlhkosti.



Obrázek 4.5 Program pro PC po úspěšném připojení

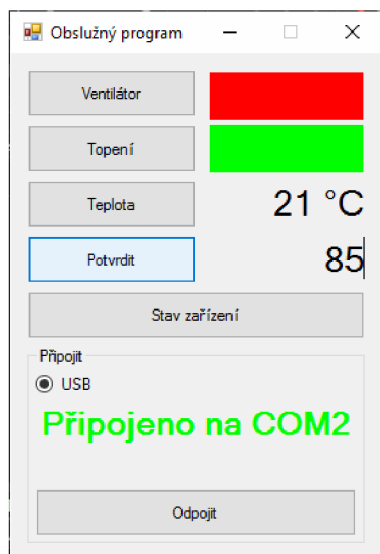
Pro změnu v nastavení ventilátoru uživatel klikne na tlačítko Ventilátor, klikne na toto tlačítko a program vyšle příslušnou sekvenci znaků. Na straně inkubátoru se provede změna a vyšle sekvenci znaků zpět. Na straně PC obslužný program změní příslušný stav. Obdobně se změny provedou při stisku tlačítka Topení.

Při stisknutí tlačítka Stav zařízení se zobrazí formulář pro nastavení doby po jaké má opět počítač vyslat sekvenci na dotaz o stavu inkubátoru.



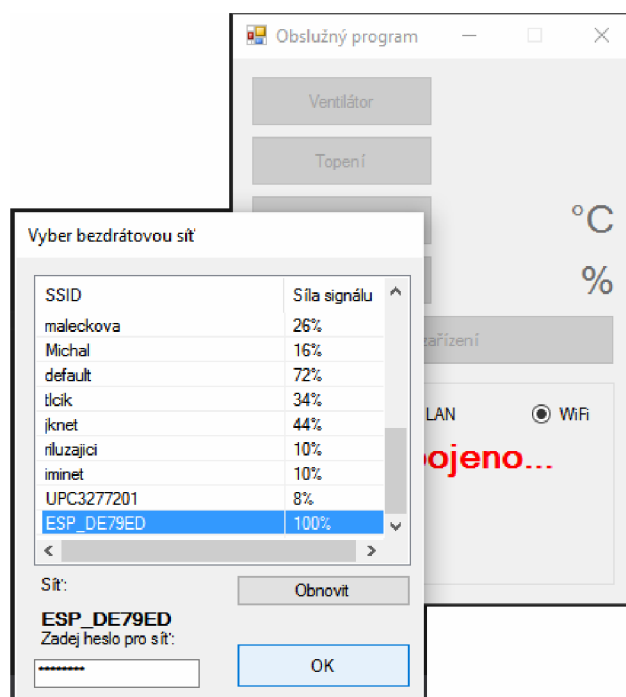
Obrázek 4.6 Změna nastavení intervalu

Pro změnu v nastavení teploty nebo vlhkosti, klikne na příslušné tlačítko a vepíše novou hodnotu, kterou potvrdí. Tato hodnota se přenesení do paměti EEPROM mikrokontroléru ATmega8. Strana inkubátoru opět pošle zpět sekvenci znaků s příslušným stavem.



Obrázek 4.7 Změna nastavení vlhkosti

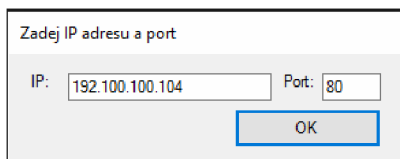
Po stisknutí tlačítka Odpojit přejde program opět do výchozího stavu tak, jak je zobrazeno na obrázku 4.3. Nyní má uživatel možnost vybrat jiné rozhraní. Při zasunutí modulu WiFi a výběru WiFi, se zobrazí formulář s vypsáním dostupnými sítěmi. Uživatel po vybrání příslušné sítě musí zadat ještě přístupové heslo pro vybranou síť v seznamu, tlačítkem Obnovit dojde znovu k načtení všech dostupných WiFi sítí.



Obrázek 4.8 Výběr WiFi sítě

Po vybrání WiFi sítě inkubátoru a zadání správného hesla si po připojení program z dané sítě načte přidělenou IP adresu ESP8266 a dále funguje obdobně jako po připojení přes síť LAN za pomoci TCP/IP protokolu.

U výběru připojení prostřednictvím modulu LAN se příslušná IP adresa i používaný port zadává ručně, dle nastavení převodníku Xport zapojeného ve vlastní síti.



The image shows a small dialog box with a light gray background and a black border. The title bar at the top reads "Zadej IP adresu a port". Below the title bar, there are two input fields. The first is labeled "IP:" and contains the text "192.100.100.104". The second is labeled "Port:" and contains the text "80". Below these two fields, centered, is a button with the text "OK".

Obrázek 4.9 Nastavení sítě LAN

Celá aplikace je řízena výhradně uživatelem. Uživatel musí pouze vybrat převodník, který chce pro komunikaci s inkubátorem používat a potom kliknout na příslušné připojení. Dále nastaví četnost vyčítání dat z inkubátoru, která je implicitně nastavena na 1 minutu.

5 ZÁVĚR

Výsledkem této bakalářské práce je návrh a výroba 4 modulů, ve kterých je implementováno rozhraní mezi PC a inkubátorem včetně vlastní základní desky inkubátoru. Modul pro KME control board ATmega16 nemohl být realizován z důvodu nefunkčnosti této vývojové desky pod systémem Windows 8.1[®] a vyšší.

V práci je dále popsán a objasněn vnitřní program pro mikrokontrolér AVR ATmega8, napsaný v jazyce C ve vývojovém studiu firmy Atmel, AtmelStudio 7.0, kde v nápovědě je naznačeno možnost rozšíření o další čidla (např. čidlo vlhkosti), se schopností ukládání a načítání hodnot pro nastavení přes vnitřní paměť EEPROM. Nahrávání a odzkoušení programu bylo prováděno za pomoci programátoru BiProg [15].

Hlavním cílem projektu byla minimalizace celého zařízení tak, aby po instalaci vlastního inkubátoru zabíralo co možná nejméně místa. Dalším důležitým sledovaným aspektem byla snadná výměna jednotlivých modulů celého zařízení.

Použité hardwarové komponenty jsou funkční na všech operačních systémech s Windows XP[®] a vyšších. Obslužný program pro ovládání byl však vyvinut již v jazyce C# z prostředí Microsoft Visual Studia 2015 a díky použitým komponentům z platformy .NET Framework 4.6 byl přizpůsoben pouze pro Windows 10[®] a nebyl odzkoušen na nižších verzích Windows[®].

V obslužném programu je použita podpora pro komunikaci přes USB a TCP/IP přímo z komponent obsažené v .NET Framework 4.6. Pro obsluhu WiFi modulu jsou použity komponenty NativeWifi [16], [17]. Z důvodu jednoduchosti, přehlednosti a snadné modifikovatelnosti není program doplněn o další ochranné podprogramy, které by z nezpřehlednili základní kód.

Výsledkem práce je funkční řešení hardwaru inkubátoru, software pro mikrokontrolér i počítač dle zadání bakalářské práce.

LITERATURA

- [1] Wi-Fi. In: *Wikipedia: the free encyclopedia* [\[online\]](#). San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-29]. Dostupné z: <https://cs.wikipedia.org/wiki/Wi-Fi>

- [2] USART. *Wikipedia: the free encyclopedia* [\[online\]](#). San Francisco (CA): Wikimedia Foundation, 2015. Dostupné také z: <https://cs.wikipedia.org/wiki/USART>

- [3] Bezdrátové WiFi/UART/SPI moduly s integrovaným web serverem. *Spezial Electronic* [\[online\]](#). Spezial Electronic: Spezial Electronic, 2014 [cit. 2016-03-29]. Dostupné z: <http://www.spezial.cz/connectone/bezdratove-seriove-rs232-wifi-moduly-iwifi-mini-secure-socket-iwifi-connectone.html>

- [4] Nano Socket iWiFi. *Pandatron* [\[online\]](#). web: Pandatron, 2015 [cit. 2016-03-29]. Dostupné z: http://pandatron.sk/?shop&sla=12&pn=90017&tx=nano_socket_iwifi

- [5] FTDI FT232RL: real vs fake. *Zeptobars* [\[online\]](#). Internet: Zeptobars, 2014 [cit. 2016-02-23]. Dostupné z: <http://zeptobars.com/en/read/FTDI-FT232RL-real-vs-fake-supereal>

- [6] Windows Update drivers bricking USB serial chips beloved of hardware hackers. *ARS Technika* [\[online\]](#). ARS Technika: ARS Technika, 2014 [cit. 2016-03-29]. Dostupné z: <http://arstechnica.com/information-technology/2014/10/windows-update-drivers-bricking-usb-serial-chips-beloved-of-hardware-hackers>

- [7] Datasheet XPort. In: *Lantronix.com* [\[online\]](#). United States: Lantronix, 2014 [cit. 2016-04-26]. Dostupné z: http://www.lantronix.com/wp-content/uploads/pdf/XPort_DS.pdf

- [8] Pár poznámek k ESP8266. *Retročip* [\[online\]](#). online: Retročip, 2014 [cit. 2016-03-29]. Dostupné z: <http://retrocip.cz/par-poznamek-k-esp8266/>

- [9] Datasheet ESP8266: Kapitola 7 - AT příkazy. In: *Https://nurdspace.nl/ESP8266* [\[online\]](#). NURDS: Site4U, 2013 [cit. 2016-04-26]. Dostupné z: <https://nurdspace.nl/ESP8266>

- [10] Seriový terminál TERMITE 3.2. In: *Http://www.compuphase.com/software_termite.htm* [\[online\]](#). CompuPhase: CompuPhase, 2015 [cit. 2016-04-26]. Dostupné z: http://www.compuphase.com/software_termite.htm

- [11] Nodemcu flasher. In: *Https://github.com/nodemcu/nodemcu-flasher* [\[online\]](#). GitHub: GitHub, 2015 [cit. 2016-04-26]. Dostupné z: <https://github.com/nodemcu/nodemcu-flasher>

- [12] Firmware ESP8266 AT. In: <https://drive.google.com/file/d/0B3dUKfzZnlwdUJUc2hkZDUyVjA/view> [online]. GitHub: GitHub, 2015 [cit. 2016-04-26]. Dostupné z: <https://drive.google.com/file/d/0B3dUKfzZnlwdUJUc2hkZDUyVjA/view>
- [13] Datasheet ATmega8(L). In: *Atmel.com* [online]. San Jose: Atmel, 2013 [cit. 2016-03-29]. Dostupné z: http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf
- [14] Základy ovládání mikrokontroléru ATmega(16/32) 18.díl – Jednotka USART (1. část – Popis). *Tajned* [online]. online: Tajned, 2016 [cit. 2016-03-29]. Dostupné z: <http://www.tajned.cz/2015/05/zaklady-ovladani-mikrokontroleru-atmega1632-18-dil-jednotka-usart-1-cast-popis/>
- [15] POVALAČ. Konstrukce programátoru procesorů AVR. In: *Ústav Radioelektroniky* [online]. Brno: VUT Brno, 2013 [cit. 2016-04-25]. Dostupné z: http://www.urel.feec.vutbr.cz/index.php?page=study_all
- [16] WlanApi.cs. In: [Http://managedwifi.codeplex.com/SourceControl/changeset/view/41381#WlanApi.cs](http://managedwifi.codeplex.com/SourceControl/changeset/view/41381#WlanApi.cs) [online]. CodePlex: CodePlex, 2015 [cit. 2016-04-26]. Dostupné z: <http://managedwifi.codeplex.com/SourceControl/changeset/view/41381#WlanApi.cs>
- [17] Interop.cs. In: [Http://managedwifi.codeplex.com/SourceControl/changeset/view/41381#Interop.cs](http://managedwifi.codeplex.com/SourceControl/changeset/view/41381#Interop.cs) [online]. CodePlex: CodePlex, 2015 [cit. 2016-04-26]. Dostupné z: <http://managedwifi.codeplex.com/SourceControl/changeset/view/41381#Interop.cs>

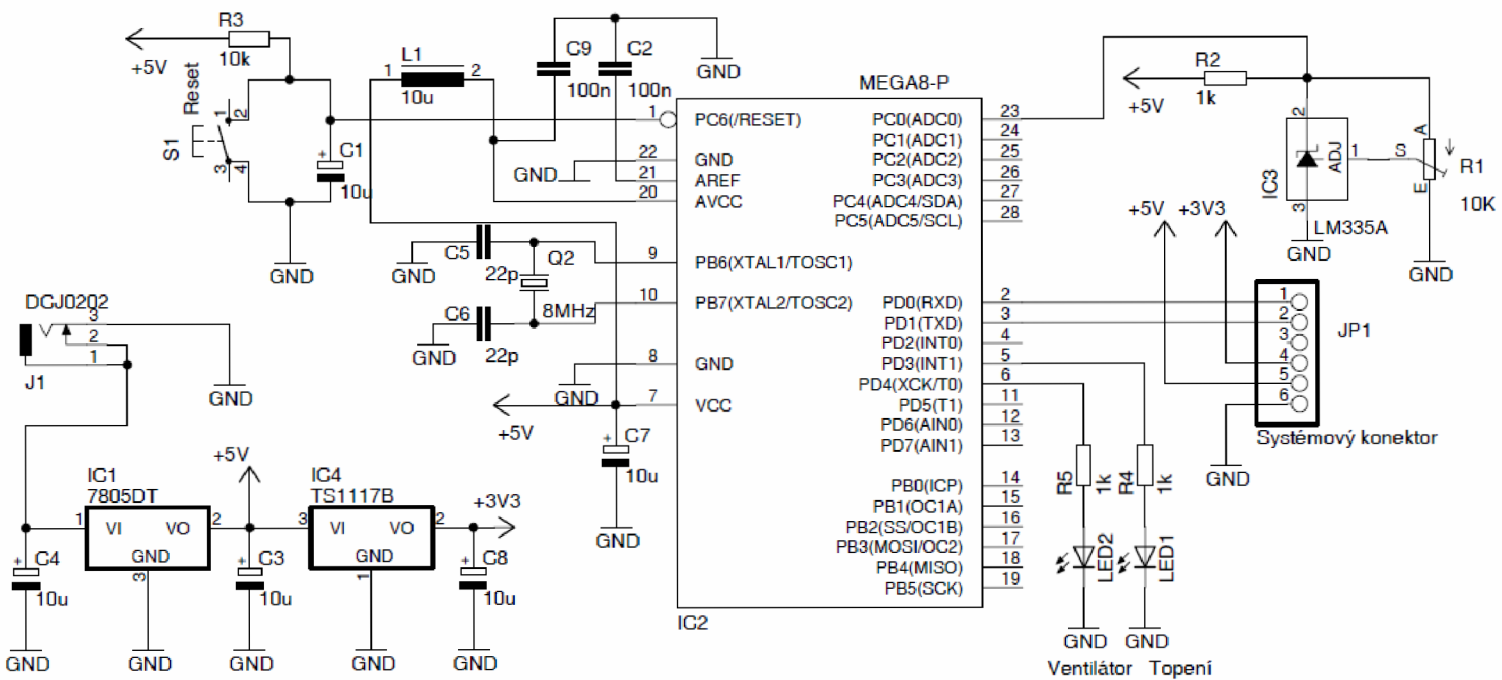
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

PC	Personal computer, osobní počítač.
USART	Universal Synchronous / Asynchronous Receiver and Transmitter, Univerzální synchronní a asynchronní přijímač vysílač.
EMC	Electromagnetic compatibility, elektromagnetická kompatibilita.
USB	Universal Serial Bus, univerzální sériová sběrnice.
VID	Vendor ID, číslo dodavatele USB zařízení.
LAN	Local Area Network, lokální počítačová síť nebo místní počítačová síť.
TCP	Transmission Control Protocol, protokolem transportní vrstvy
IP	Jednoznačná identifikace zařízení v počítačové síti,
WiFi	Wireless LAN, bezdrátová komunikace v počítačových sítích.
SSID	Service Set Identifier, jedinečný identifikátor každé bezdrátové sítě.
ASCII	American Standard Code for Information Interchange, americký standardní kód pro výměnu informací.
TX	transit, odeslat.
RX	receive, přijmout.
RS 232	sériová linka.
RJ45	Konektor pro zapojení síťových kabelů UTP.
DHCP	Dynamic Host Configuration Protocol, automatická konfigurace počítačů připojených do počítačové sítě.
Telnet	Telecommunication Network, protokolu který umožňuje uživateli připojení ke vzdálenému počítači.
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektricky mazatelná paměť typu ROM.
COM	sériový port.
SPI	Serial Peripheral Interface, sériové periferní rozhraní.
SMT	Surface mount technology, systém povrchové montáže.
RISC	Reduced Instruction Set Computing, procesor s redukovanou instrukční sadou.
WEP	Wired Equivalent Privacy, zabezpečení počítačových sítí.
TKIP	Temporal Key Integrity Protocol, zabezpečení počítačových sítí.
AES	Advanced Encryption Standard, standard pokročilého šifrování.
APSD	Automatic Power Save Delivery, automatická úspora energie.

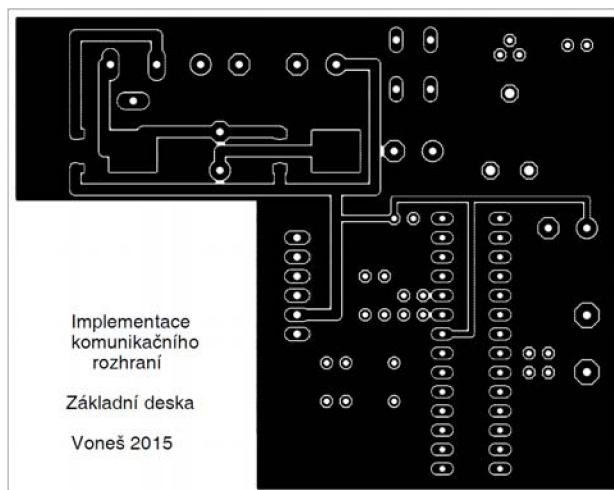
VoIP	Voice over Internet Protocol, přenos digitalizovaného hlasu prostřednictvím počítačové sítě.
AP	Access point, přístupový bod.
I ² C	Inter-Integrated Circuit, multi-masterová počítačová sériová sběrnice.
A/D	Analogově/Digitální převodník
Windows	Operační systém od firmy Microsoft.
Firmware	Software, který slouží pro řízení vestavěného systému.
Software	Programové vybavení
Hardware	Technické vybavení přístroje.

A NÁVRH ZAŘÍZENÍ

A.1 Obvodové zapojení základní desky

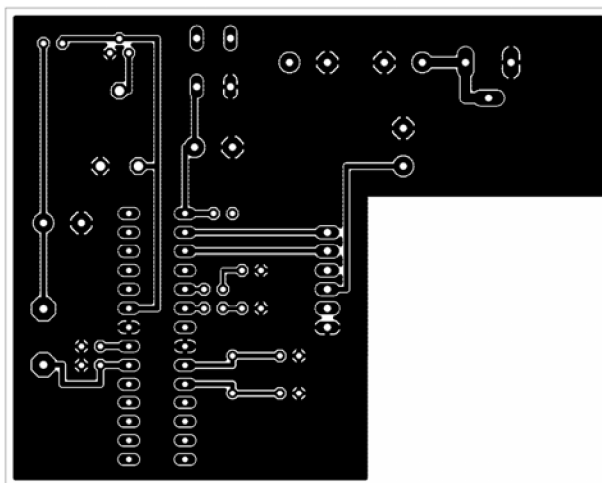


A.2 Deska plošného spoje základní desky – top (strana součástek)



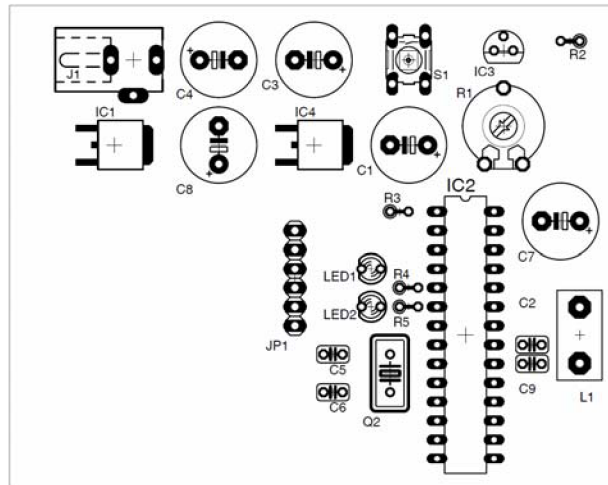
Rozměr desky 81×65 [mm], měřítko M1:1

A.3 Deska plošného spoje základní desky – bottom (strana spojů)

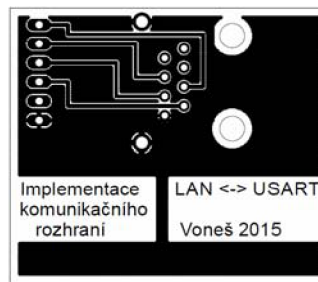


Rozměr desky 81×65 [mm], měřítko M1:1

A.4 Osazovací plán základní desky – top (strana součástek)

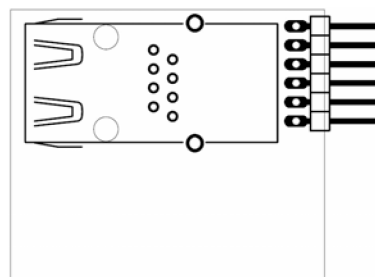


A.5 Deska plošného spoje převodníku LAN – bottom (strana spojů)

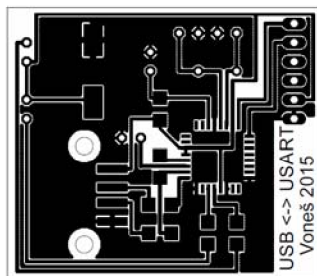


Rozměr desky 37 × 42 [mm], měřítko M1:1

A.6 Osazovací plán převodníku LAN – top (strana součástek)

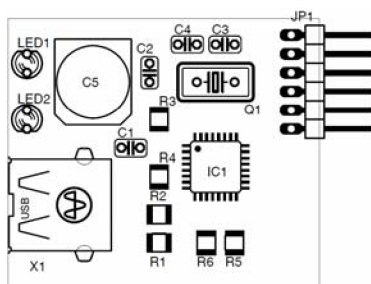


A.7 Deska plošného spoje USB převodníku – bottom (strana spojů)

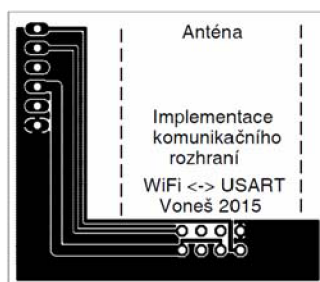


Rozměr desky 37×42 [mm], měřítko M1:1

A.8 Osazovací plán USB převodníku – bottom (strana spojů)

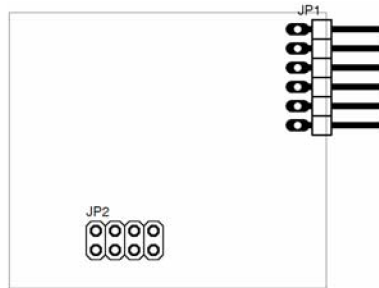


A.9 Deska plošného spoje WiFi převodníku – bottom (strana spojů)

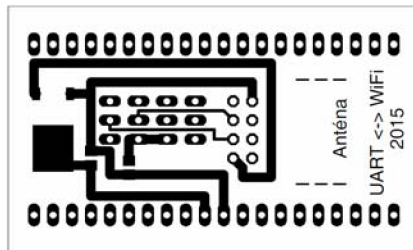


Rozměr desky 37×42 [mm], měřítko M1:1

A.10 Osazovací plán WiFi převodníku – top (strana součástek)

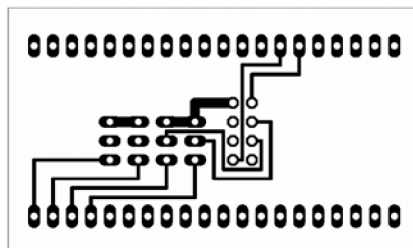


A.11 Deska plošného spoje WiFi převodníku pro KME control board ATmega16 – top (strana součástek)



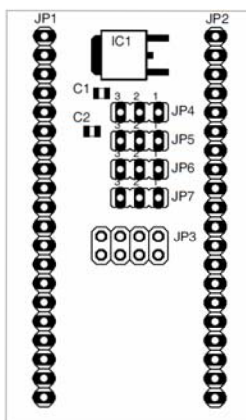
Rozměr desky 33×55 [mm], měřítko M1:1

A.12 Deska plošného spoje WiFi převodníku pro KME control board ATmega16 – bottom (strana spojů)



Rozměr desky 33×55 [mm], měřítko M1:1

A.13 Osazovací plán WiFi převodníku pro KME control board ATmega16 – top (strana součástek)



B SEZNAM SOUČÁSTEK

Seznam pro základní desku:

Označení	Hodnota	Pouzdro	Popis
IC1	7805	TO 252 (DPAK)	Stabilizátor 5V
IC2	ATmega8A	DIL28	AVR procesor
IC3	LM 335A	TO92	Teplotní čidlo
IC4	TS1117B	TO 252 (DPAK)	Stabilizátor 3,3V
L1	10 μ H	Axiální 7,5 mm	Indukčnost
S1		TC-0103	Mikrospínač do DPS
R1	10 k Ω	RM 5 \times 10 mm	O odporový trimr
R2	1 k Ω	0207	Drátový rezistor
R3	10 k Ω	0207	Drátový rezistor
R4	1 k Ω	0207	Drátový rezistor
R5	1 k Ω	0207	Drátový rezistor
Q2	8 MHz	HC49U	Krystal
C1	10 μ F/6V	RM 5 mm	Elektrolytický kondenzátor
C2	100 nF	RM 2,5 mm	Keramický kondenzátor
C3	10 μ F/6V	RM 5 mm	Elektrolytický kondenzátor
C4	10 μ F/25V	RM 5 mm	Elektrolytický kondenzátor
C5	22 pF	RM 2,5 mm	Keramický kondenzátor
C6	22 pF	RM 2,5 mm	Keramický kondenzátor
C7	10 μ F/6V	RM 5 mm	Elektrolytický kondenzátor
C8	10 μ F/6V	RM 5 mm	Elektrolytický kondenzátor
C9	100 nF	RM 2,5 mm	Keramický kondenzátor
J1	DCJ0202		Napájecí konektor
JP1		1 \times 6	Zásuvková lámací lišta 90°
	Patice	DIL 28	Pro procesor
LED 1	Bílá	5 mm	LED dioda
LED 2	Bílá	5 mm	LED dioda

Seznam pro převodník USB:

Označení	Hodnota	Pouzdro	Popis
IC1	FT232BM	TQFP32	Převodník USB
X1	Mini USB	B F SMD	USB konektor SMD
Q1	6 MHz	HC49S	Krystal
LED 1	zelená	3 mm	Led dioda
LED 2	zelená	3 mm	Led dioda
R1	27 Ω	0805	SMD rezistor
R2	27 Ω	0805	SMD rezistor
R3	470 Ω	0805	SMD rezistor
R4	1,5 k Ω	0805	SMD rezistor
R5	220 Ω	0805	SMD rezistor
R6	220 Ω	0805	SMD rezistor
C1	33 nF	RM 2,5 mm	Keramický kondenzátor
C2	100 nF	RM 2,5 mm	Keramický kondenzátor
C3	27 pF	RM 2,5 mm	Keramický kondenzátor
C4	27 pF	RM 2,5 mm	Keramický kondenzátor
C5	10 μ F/6V	RM 2,5 mm	Tantalový kondenzátor
JP1		1 \times 6	Kolíková lámací lišta 90°

Seznam pro převodník LAN:

Označení	Hodnota	Pouzdro	Popis
IC1	Xport		Převodník Xport
JP1		1 \times 6	Kolíková lámací lišta 90°

Seznam pro WiFi převodník:

Označení	Hodnota	Pouzdro	Popis
IC1	ESP 8266		Převodník WiFi
JP1		1 \times 6	Kolíková lámací lišta 90°
JP2		4 \times 2	Kolíková lámací lišta

Seznam pro WiFi převodník pro KME control board ATmega16:

Označení	Hodnota	Pouzdro	Popis
IC1	TS1117B	TO 252 (DPAK)	Stabilizátor 3,3V
C1	100 nF	0805	SMD kondenzátor
C2	100 nF	0805	SMD kondenzátor
	ESP 8266		Převodník WiFi
JP1		1×20	Kolíková lámací lišta
JP2		1×20	Kolíková lámací lišta
JP3		4×2	Kolíková lámací lišta
JP4	Jumper	1×3	Kolíková lámací lišta
JP5	Jumper	1×3	Kolíková lámací lišta
JP6	Jumper	1×3	Kolíková lámací lišta
JP7	Jumper	1×3	Kolíková lámací lišta