**STŘEDNÍ PRŮMYSLOVÁ ŠKOLA STROJNÍ A ELEKTROTECHNICKÁ**
**A VYŠŠÍ ODBORNÁ ŠKOLA, LIBEREC 1, Masarykova 3**

Masarykova 3, 460 84 Liberec 1, tel. 485 100 113, fax 485 100 063, e-mail sekretariat@pslib.cz, http://www.pslib.cz

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

ROBOT OVLÁDANÝ PŘES WIFI

Jan Trejbal

Liberec 2012

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 10.elektrotechnika, elektronika a telekomunikace

# ROBOT OVLÁDANÝ PŘES WIFI
# Robot controlled over WIFI


Autoři: Jan Trejbal

Škola: Střední průmyslová škola strojní a elektrotechnická a Vyšší odborná škola, Liberec 1, Masarykova 3, příspěvková organizace

Konzultant: Miroslav Mach


Liberec 2012

Prohlašuji, že jsem svou práci vypracoval(a) samostatně, použil(a) jsem pouze podklady (literaturu, SW atd.) citované v práci a uvedené v přiloženém seznamu a postup při zpracování práce je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Liberci dne 26. 3. 2012 podpis:.....

# Anotace

Práce se zabývá projektem s názvem „Tyree“, jehož cílem je ovládat pojízdné multifunkční zařízení přes bezdrátovou síť wifi (802.11 b+g). V dokumentaci jsou předloženy důvody použitých technologií a postupů. Práce vychází z možnosti bezdrátové komunikace přes UART pomocí wifi modulu. Projekt přináší praktické zkušenosti v návrhu strojírenských dílů, návrhu a realizace DPS, návrh SW pro MCU, PC, smartphone.

# Summary

This thesis deals with project name “Tyree”. My goal in this project is to remote a multifunctional device over wireless network wifi (802.11 b+g). In this thesis there are presented reasons for technology and method which were used in this project. The thesis is based on possibility wireless communication over UART (Universal Synchronous Receiver and Transmitter) using wifi module. The project brings practical experience of design engineering components, design and realizes printed circuit boards, design software for microprocessor, personal computer and smartphone.

# Poděkování

Rád bych poděkoval panu Miroslavu Machovi za vedení práce a pomoc při návrhu elektrických schémat. Dále panu Jiřímu Bouzkovi za pomoc při návrhu podvozku a ruky. A v neposlední řadě panu Ing. Jaroslavu Semerádovi za umožnění uskutečnění práce.

# Obsah

Anotace ... i
Summary ... i
Poděkování ... i
Klíčová slova ... 1
Úvod ... 2
1 Vznik ... 3
2 Hardware ... 4
2.1 Seznam modulů, modulových desek a příslušenství ... 4
2.2 Wifi modul ... 4
*2.2.1 Nano Socket iWifi ... 4*
*2.2.2 Mini Socket iWifi ... 5*
2.3 Mikrokontrolér ... 5
*2.3.1 Atmel 89C2051 ... 5*
*2.3.2 ATmega128A ... 6*
*2.3.3 Programátor ... 6*
2.4 Úprava komunikace ... 8
2.5 Řízení motorů ... 8
*2.5.1 Krokové motory ... 8*
*2.5.2 Stejnosměrné motory ... 9*
*2.5.3 Redukce na kola ... 9*
2.6 Baterie ... 10
*2.6.1 Olověný článek ... 10*
*2.6.2 Lithium-polymer akumulátor ... 10*
*2.6.3 Deska pro ošetření kritických stavů Li-pol akumulátorů ... 11*
2.7 Napájecí deska ... 11
2.8 Podvozek ... 12
*2.8.1 Podvozek z cuprextitu ... 12*
*2.8.2 Podvozek z nerezového plechu ... 12*
2.9 Kamera ... 13
2.10 Dodatečné obvody ... 14
3 Software ... 15
3.1 Použité programovací jazyky ... 15
*3.1.1 AT+i ... 15*
*3.1.2 Html a javascript ... 15*
*3.1.3 Asembler ... 15*
*3.1.4 C++ ... 15*
*3.1.5 C ... 15*
*3.1.6 C# ... 16*
*3.1.7 J2ME ... 16*
3.2 Výstavba SW ... 16
*3.2.1 Propojení portů MCU a zbylých obvodů ... 16*
*3.2.2 Program MCU ... 17*
*3.2.3 Program PC aplikace ... 18*
*3.2.4 Program pro smartphone ... 19*
*3.2.5 Protokol komunikace ... 20*
Závěr ... 21
Seznam obrázků ... 22
Použité zdroje ... 23

Přílohy ..... 24
A. Obsah přiloženého CD ..... 25
B. Schéma a DPS ATmega128A ..... 26
C. Schéma a DPS stabilizátoru logických úrovní ..... 27
D. Schéma a DPS ošetření kritických stavů akumulátorů ..... 28
E. Schéma a DPS H-můstků ..... 29
F. Schéma a DPS dodatkových obvodů ..... 30
G. Blokové schéma ..... 31
H. Fotodokumentace ..... 32

# Klíčová slova

| Klíčové slovo | Vyjádření zkratky | Vysvětlení |
|---|---|---|
| AD | Analog Digital | Analogově digitální |
| AVR | | Rodina mirkokontrolérů |
| C# | C Sharp | Objektové C, odnož C |
| COM | Serial port | Sériový port |
| DPS | | Deska plošných spojů |
| GND | Ground | Uzemnění |
| GUI | Graphical User Interface rozhraní | Grafické uživatelské |
| HEX | Hexadecimal | Šestnáctková soustava |
| HW | Hardware | Fyzicky exitující zařízení |
| ISP | In-System Programming | Programování bez nutnosti odpojovat MCU od obvodu |
| J2ME | Java 2 Micro Edition | Java pro mobilní zařízení |
| JTAG | Joint Test Action Group | Rozhraní pro programování FLASH pamětí apod. |
| LED | Light-Emitting Diode | Dioda emitující světlo |
| Li-pol | Lithium Polymer Batteries | Lithium-polymerová baterie |
| MCU | Mikrocontroller unit | Mikrokontrolér |
| Paket | | Blok dat přenášený v počítačové síti |
| PC | Personal Computer | Osobní počítač |
| Pin | | Vývod |
| PWM | Pulse-Width Modulation | Pulzně šířková modulace |
| Smartphone | | Chytrý telefon |
| SMD | Surface Mount Device | Součástky pro plošnou montáž |
| SW | Software | Program |
| UART | Universal Synchronous Receiver and Transmitter | Sériová komunikace |
| USB | Universal Serial Bus | Univerzální sériová linka |
| Vcc | Positive Supply Voltage | Kladné napájecí napětí |
| Wifi | | Označení pro standardy bezdrátové komunikace |

# Úvod

Cílem práce je vytvořit modulovatelného robota, jeho podvozek, pohon podvozku, realizovat řídící jednotku hardwarově i softwarově.

Robot by v praxi měl být nápomocen například ve skladech, díky jeho pohyblivosti, přenášení obrazu a zvuku, umožní obsluze kontrolovat větší prostory, než které jsou schopni fyzicky hlídat, díky možnosti pohybu má náskok oproti běžným montovaným bezpečnostním kamerám. V kombinaci s propracovaným SW a doplněním o vhodné senzory (měřiče vzdálenosti, skenování teploty prostoru, kouřové senzory) může být zcela autonomní jednotkou schopnou získávat a využívat data ze zařízení komunikujících na stejných sítích a protokolech (soubor pravidel) a tím velice efektivně střežit prostor.

Též je možné jej nasadit v prostorách životu nebezpečných, např. neutralizace náloží, kontrola zamořených prostor (prostory hořících chemikálii – kontrola nepřítomnosti lidí, zvířat, věcí)

Díky širokým možnostem ovládání, buď přes vytvořený ovládací SW (pro PC, Smartphone), případně využití zařízení s možností připojení k wifi síti robota a ovládání přes webový prohlížeč (pro zařízení nepodporující vytvořené aplikace), není nutné vytvářet ovládací zařízení specializované k ovládání. To z robota činí mobilní zařízení i co se týče ovládacího zařízení (není závislý na jedné platformě).

# 1 Vznik

Projekt Tyree vznikl z nápadu od základu vytvořit zařízení komunikující s PC. Byl hledán způsob jak propojit vlastní HW s PC, nalezena byla firma Spezial elektronics distribuující mimo jiné bezdrátové moduly wifi a bluetooth. Po důkladné analýze ve výběru vyhrála technologie wifi, díky rozšířenosti a možnosti připojit modul k rozsáhlé wifi síti. Distribuované moduly komunikují především přes UART a tomu byl uzpůsoben výběr MCU.

# 2 Hardware

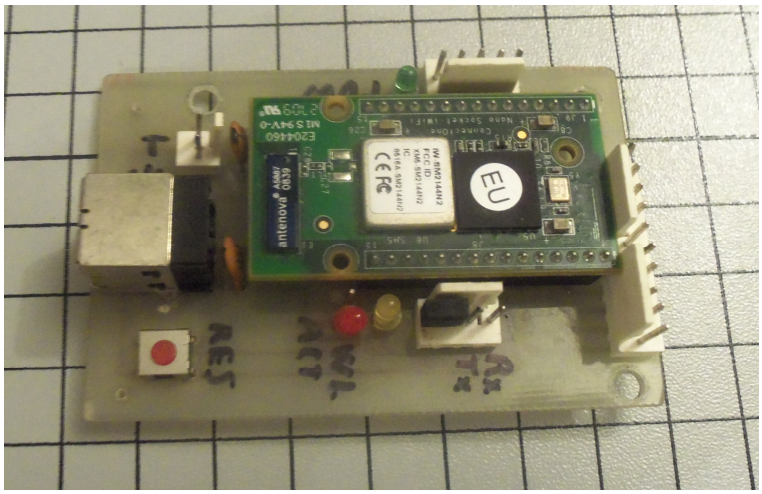
## 2.1 Seznam modulů, modulových desek a příslušenství

- Wifi modul – Mini Socket iWifi
- MCU - ATmega128a
- Programátor – ISP
- Výkonový obvod motorů – L298
- Obraz – IP Kamera WCS-2030
- Akumulátor – Li-pol 7,4 V a 11,1 V
- Stabilizátory – LF50 a LF33 pro logické obvody a LF50 pro výkonové obvody
- Připojení akumulátorů – pojistky, spínání výkonových obvodů
- Doplňující obvody – spínání kamery, akustické obvody

## 2.2 Wifi modul

### 2.2.1 Nano Socket iWifi

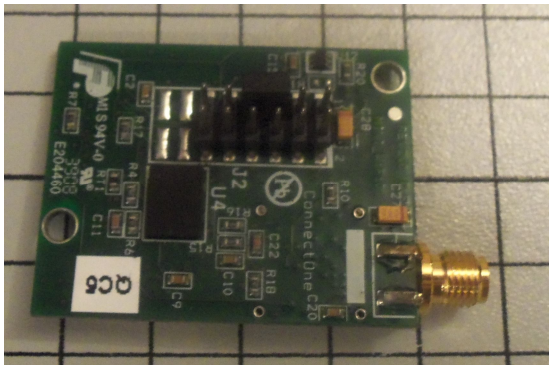
Jako wifi modul byl použit modul „Nano Socket iWifi“. Tento modul byl vybrán pro možnost nastavení přes USB a své široké množství komunikačních portů. Modul se řadí mezi nízkonapěťovou elektroniku (Vcc – 3,3 V). Pro komunikaci je dodáván SW vytvářející virtuální COM port, data odeslaná přes tento port se zkomprimují, přenesou přes wifi do modulu, zde se rozbalí a odešlou na port nastavený v modulu jako výstupní (pro nás UART).



**Obrázek 1 - Nano Socket iWifi**

### 2.2.2 Mini Socket iWifi

Nano Socket iWifi se ukázal jako nevyhovující, z důvodu nevyvedené antény, což se projevilo jako zásadní problém co se týče umístění modulu (modul musel být umístěn v místě vyhovujícím příjmu a vysílání, takovéto místo po vytvoření karoserie již neexistovalo). Nový modul „Mini Socket iWifi“ tento problém řešil a zároveň zjednodušil a zmenšil DPS pro připojení modulu, dalším plusem bylo odstranění přebytečných portů (zůstal pouze UART – což též zmenšilo velikost DPS). Anténu bylo možné připojit přímo k modulu díky konektoru SMA (M), to se ukázalo jako stále nevyhovující, ale řešením bylo použití prodlužovacího kablíku vyvedeného skrz karoserii.

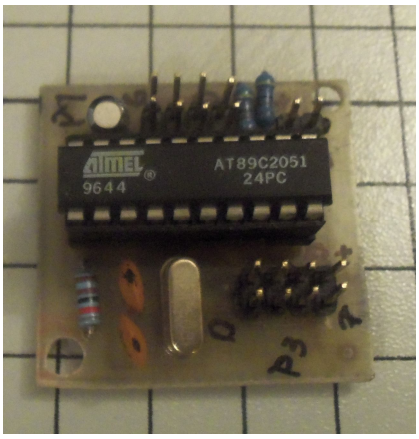


**Obrázek 2 - Mini Socket iWifi**

## 2.3 Mikrokontrolér

### 2.3.1 Atmel 89C2051

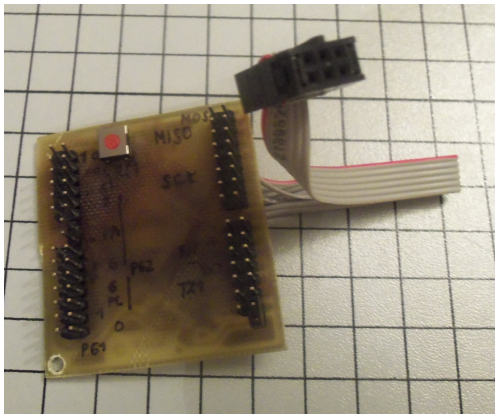
Jako MCU byl vybrán Atmel 89C2051, s možností užití frekvence krystalu až 24MHz, obsaženým HW UART a 15ti užitelnými vstupně výstupními piny. Tento procesor se řadí do rodiny x51 (původní procesor vyvinul Intel a licenci distribuoval dalším firmám, které vytvořily různě sofistikované verze). Pro tento procesor byla sestavena deska pro použití v robotu. Byl sepsán program obsahující vytvořený protokol zprostředkovávající komunikaci po UART (v jednom paketu přenášející pouze 1 Byte, což pro řízení nebylo dostačující).




**Obrázek 3 – Atmel 89C2051**

### 2.3.2 ATmega128A

Atmel 89C2051 se ukázal jako nevyhovující, díky malému množství vstupně výstupních pinů, zbytečné náročnosti řešení některých SW úkonů (které jsou řešitelné HW) a především díky neřešitelnosti později požadovaným úkonům. Pro vyřešení problémů byl zvolen procesor ATmega128A, který byl úmyslně předimenzovaný co se týče vlastností a vstupně výstupních pinů, díky tomuto procesoru není složité připojit další periferie. Tento procesor disponuje již 48-mi programovatelnými vstupně výstupními piny. A mimo jiné dvěma HW UART, několik pinů řízených pomocí HW PWM a A/D převodníkem.



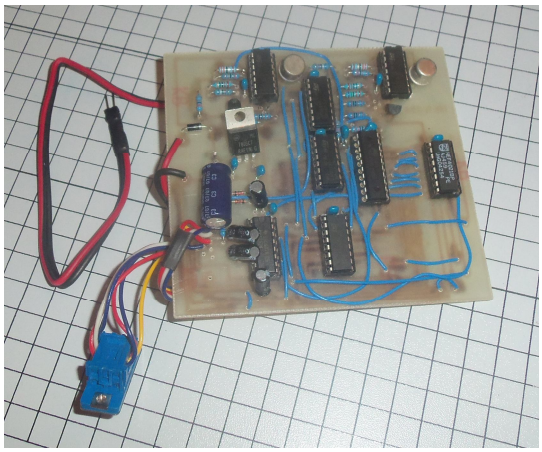
**Obrázek 4 - ATmega128A**

### 2.3.3 Programátor

Pro procesor Atmel89C2051 byl využit programátor uveřejněný v knize Davida Matouška – Práce s mikrokontroléry ATMEL AT89C2051[1], tento programátor byl díky své velikosti poruchový a díky nutnosti využít HW sériový port PC nepraktický

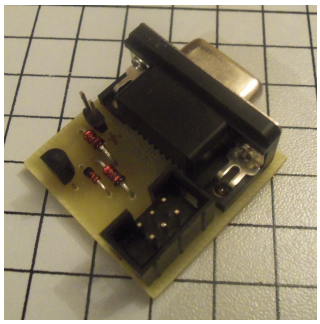
[1] MATOUŠEK, David. Práce s mikrokontroléry Atmel AT89C2051. 2. vydání. BEN – technická literatura, 2002, s. 21-30. ISBN 978-80-7300-276-3.

(v současnosti není již mnoho PC vybavených sériovým portem, využívají se USB převodníky, ale ty nemají identické vlastnosti jako HW port, proto jsou v některých aplikacích nepoužitelné). Též rychlost zápisu dat byla značně pomalá (už při velikosti 300 B byla doba zápisu už 6 minut).



**Obrázek 5 - Programátor pro Atmel 89C2051**

Pro procesor ATmega128A existuje možnost programovat jej přes ISP a JTAG rozhraní, to však z neznámých důvodů (nejspíše prvopočátečním HW chybám DPS) nefungovalo. Proto byl nalezen programátor Simple Serial AVR Programmer[1] (jedná se o ISP programátor), který při své úžasné jednoduchosti fungoval na 100%.



**Obrázek 6 - ISP programátor pro AVR procesory**

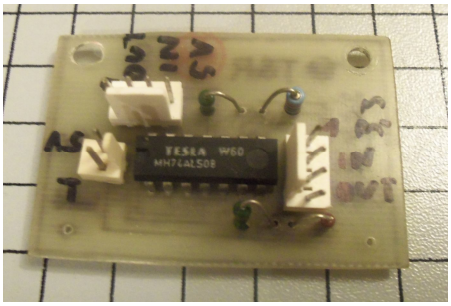
Pro užití ISP programátoru je užit program PonyProg2000[2], do kterého lze vložit HEX soubor z AVR Studia (vývojové prostředí a kompilátor pro AVR procesory). Tento programátor je mnohonásobně rychlejší (doba zápisu je při 8 KB přibližně 15 s).

[1] Simple Serial AVR Programmer. *How Circuits Work* [online]. 2011-04-03 [cit. 2012-03-09]. Dostupné z: http://howcircuits.com/simple-serial-avr-programmer.html

[2] PonyProg: serial device programmer. *Lanconelli Open Systems* [online]. 2008-01-05 [cit. 2012-03-09]. Dostupné z: http://www.lancos.com/prog.html

## 2.4 Úprava komunikace

Při návrhu se objevil problém s rozdílnými logickými úrovněmi užitými pro komunikaci pomocí UART, proto byl navržen obvod logické korekce, který byl při změně procesoru přepracován, značně zjednodušen a integrován do DPS wifi modulu.

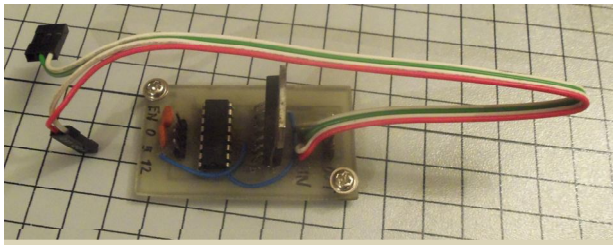


**Obrázek 7 - Deska pro úpravu komunikace**

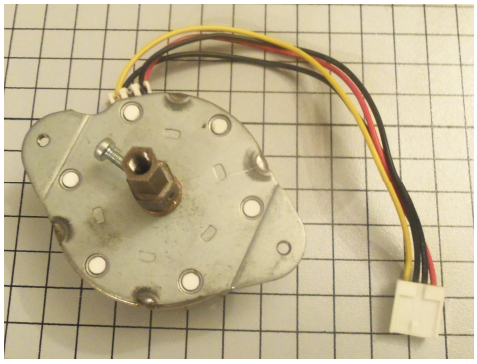
## 2.5 Řízení motorů

### 2.5.1 Krokové motory

Pro pohyb robota byl vybrán krokový motor, díky nemožnosti připojení motoru přímo k MCU (pro procesor příliš velký odběr) byl vybrán obvod L298, který obsahuje 2 výkonové H-můstky, z důvodu vnitřního zapojení obvodu bylo nutné do schématu přidat též ochranné diody, jako vhodné řešení pro zapojení se ukázal obvod L2910 obsahující 2x diodový můstek (jeho zapojení je totožné jako zapojení diod nutné k ochraně obvodu L298).



**Obrázek 8 - Driver L298**



**Obrázek 9 - Krokový motor**

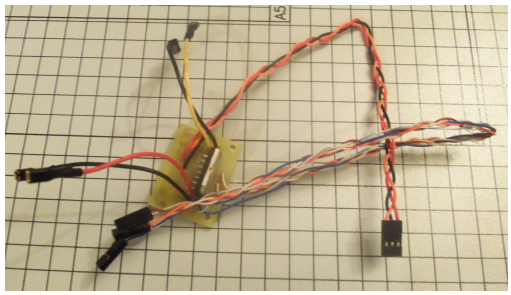
### 2.5.2 Stejnosměrné motory

Návrh s krokovými motory po oživení DPS a vyzkoušení motorů se ukázal jako nedostačující, z důvodu malého kroutícího momentu (nebyly známy specifikace motoru, při realizaci původní DPS, vycházelo se z běžných hodnot napětí a proudu typických pro podobné motory).

Jako řešení bylo vybráno použití stejnosměrných motorů, jejichž ovládání pomocí procesoru se dá řešit obdobným způsobem jako u krokových motorů pomocí obvodu L298, Pro řízení rychlosti bylo použito pulzně šířkové modulace, která se používá například jako regulátor napětí u spínaných zdrojů.

**Obrázek 10 - Stejnosměrný motor**

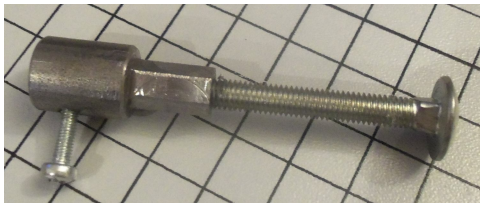
Deska s obvodem L2910 byla přepracována na desku s dvěma oddělenými Grätzovými můstky v SMD provedení pro zmenšení DPS.



**Obrázek 11 - Driver L298 v2**

### 2.5.3 Redukce na kola

Pro přenesení kroutícího momentu z motoru na kolo byla v SW SolidEdge nakreslena redukce, jejíž výkres byl předán do strojních dílen kde byla dle výkresu vyhotovena ve dvou potřebných dílech.

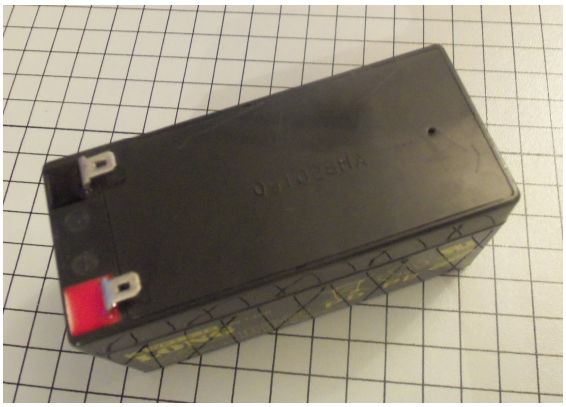


**Obrázek 12 - Redukce z motoru na kolo**

## 2.6 Baterie

### 2.6.1 Olověný článek

Jako původní zdroj byl vybrán olověný (Pb) článek 12V 1300 mA. Tento článek byl vybrán z důvodu velké odolnosti proti zkratu a dalším nepříznivým vlivům, též díky relativně snadné konstrukci nabíječky.



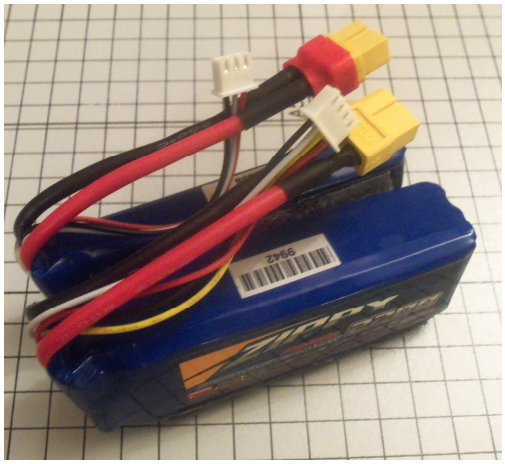
**Obrázek 13 - Olověný článek**

### 2.6.2 Lithium-polymer akumulátor

Olověný článek se v průběhu používání ukázal jako nevyhovující, z důvodu velké váhy, malé kapacity, objemnosti a pomalého nabíjení.

Proto byly vybrány dva Li-pol akumulátory, 7,4 V (1800 mAh) a 11,1 V (2200 mAh), které dohromady zaujímají menší prostor než olověný článek. Díky použití dvou akumulátorů byla získána možnost oddělení výkonové a logické větve napájení.

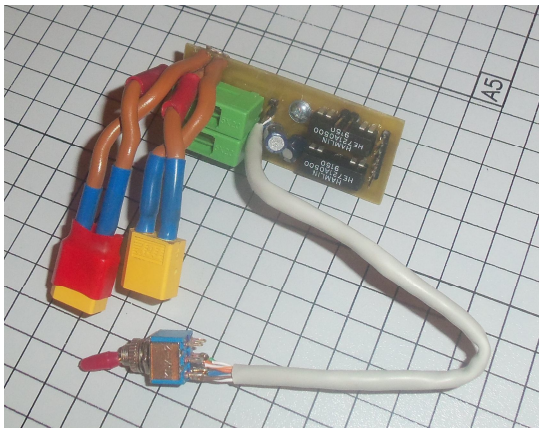
Při nabíjení je nutné nezanedbávat rovnoměrné nabíjení článků akumulátoru ošetřené servisním konektorem. Též je zapotřebí hlídat napětí baterií, které nesmí klesnout pod 3 V na článek, při nižším napětí dochází k nevratným změnám v akumulátoru, za žádných okolností nesmí dojít k úplnému vybití akumulátoru! Nebo zapojení do zkratu!



**Obrázek 14 - Li-pol akumulátory**

### 2.6.3 Deska pro ošetření kritických stavů Li-pol akumulátorů

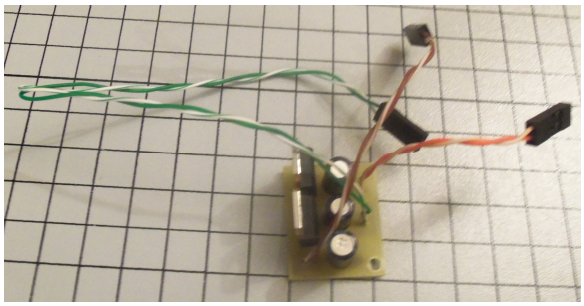
Pro ochranu proti zkratu, stavu úplného vybití, a z toho plynoucí nutnosti měřit napětí byla navržena deska, obsahující pojistky pro ošetření zkratu, nízkoodběrový stabilizátor napájející pouze procesor a relé spínající zbytek elektroniky robota. Též obsahuje děliče napětí použité z důvodu poměrného snížení napětí pro možnost měření napětí procesorem (měřené napětí musí být pod 2.5 V).



**Obrázek 15 - Ošetření kritických stavů akumulátorů**

## 2.7 Napájecí deska

Pro možnost napájení robota z akumulátorů byla vytvořena deska stabilizující napětí pomocí obvodů LF50CV (5 V stabilizátor) a LF33CV (3,3 V stabilizátor) z akumulátorů. Pro logické obvody napětí 3,3 V a 5 V (akumulátor 7,6 V) a pro výkonové obvody 5 V (akumulátor 11,4 V). Tyto stabilizátory byly po přehřívání a následném vypnutí (obsahují tepelnou pojistku) vybaveny chladiči.

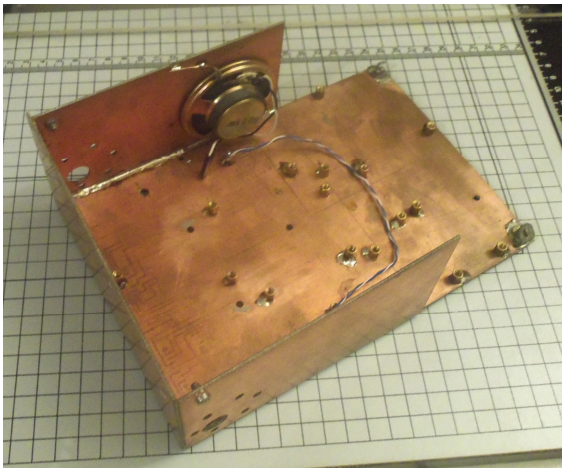


**Obrázek 16 - Stabilizátory pro logickou větev**

## 2.8 Podvozek

### 2.8.1 Podvozek z cuprextitu

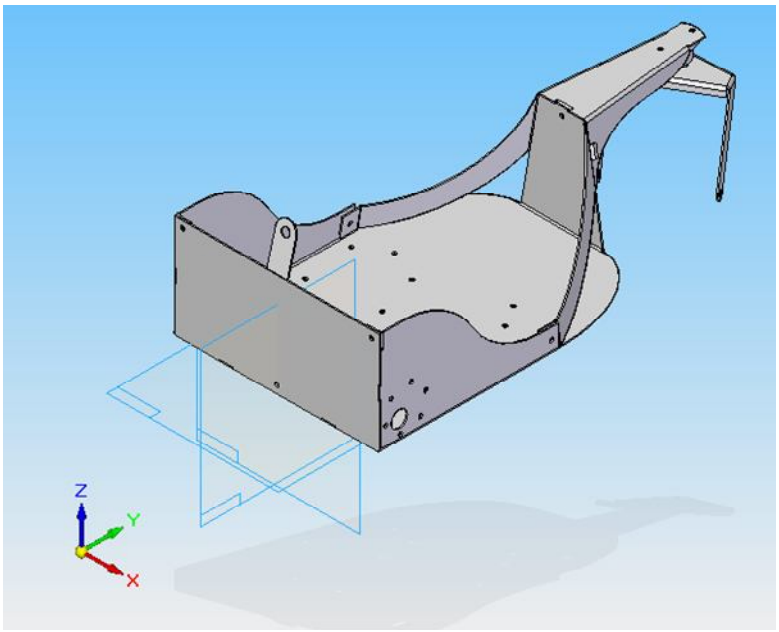
Pro praktické odzkoušení elektroniky byl vytvořen podvozek z cuprextitu (pro elektronika dostupný materiál, snadné spojení jednotlivých dílů v celek pomocí pájení).



**Obrázek 17 – Podvozek**

### 2.8.2 Podvozek z nerezového plechu

Z důvodu nepřesnosti, nestability a nevzhlednosti byl pod odborným dohledem Jiřího Bouzka v SW SolidEdge navržen nový podvozek a ten byl vyroben z nerezového plechu.




**Obrázek 18 - Nákres podvozku v2**

**Obrázek 19 - Podvozek v2**

## 2.9 Kamera

Pro možnost ovládat robota bez přímé viditelnosti byl doplněn o IP kameru, která je nastavena na automatické připojení k wifi síti vytvářené modulem. A tím je dostupná i pro zařízení ovládající robota.

**Obrázek 20 - IP kamera**

## 2.10 Dodatečné obvody

Tyto obvody byly umístěny na jednu společnou desku, jedná se o obvod pro připojení reproduktorů (funkce klaksonu) (obvod obsahuje pouze NPN tranzistor a nutný odpor pro ochranu tranzistoru), relé pro spínání napájení kamery (pro možnost odpojení kamery v případě nepoužívání – úspora akumulátorů) (obsahuje driver určený k ovládání relé), trimr pro nastavení kontrastu displaye, tranzistor pro možnost ovládání posvícení displaye logickými úrovněmi pomocí PWM regulace a tranzistor pro spínání podsvícení podvozku.

# 3 Software

## 3.1 Použité programovací jazyky

### 3.1.1 AT+i

Tento jazyk je použit pro konfiguraci wifi modulů od firmy Connect One, moduly se sice dají konfigurovat pomocí dodávané aplikace (její předností je možnost uložit a obnovit nastavení), ale ta nedosahuje kvalit nastavení modulu pomocí AT+i příkazů.

### 3.1.2 Html a javascript

Tyto jazyky byly použité pro napsání web serveru wifi modulu, pro umožnění jednoduché změny parametrů (IP, maska, Gateway, DNS) a pro snadné zjišťování statusu modulu (síla signálu, kvalita signálu, připojená zařízení, okolní wifi sítě)

### 3.1.3 Asembler

Tento jazyk byl použit pro naprogramování procesoru Atmel89C2051, programování v něm je však uživatelsky nepříliš příjemné a je snadné se v něm ztratit (pokračováním je smazat projekt a začít znovu). Jeho hlavní předností je poměrně malá velikost kódu. Je proto někdy jediným možným řešením jak procesor naprogramovat (u MCU s malou velikostí paměti).

Do tohoto jazyka se pomocí kompilátoru překládá kód v C (a z dalších).

### 3.1.4 C++

Tento jazyk byl vybrán pro jeho užívání při výuce na SPŠSE a VOŠ Liberec. V tomto jazyku bylo napsáno první GUI, které sloužilo jako ovládací SW robota. A byla v něm použita první verze komunikačního protokolu založeného na komunikaci přes COM port (pomocí námi definovaných příkazů protokolu), který virtuálně vytvářela aplikace třetí strany (tato aplikace nebyla vyhovující co se týče její spolehlivosti a komfortu užití).

### 3.1.5 C

Tento jazyk je užit pro naprogramování procesoru ATmega128A, tento procesor se též dá naprogramovat i v asembleru, ale tato možnost řešení je poměrně náročná z pohledu složitosti zdrojového kódu (a v něm užitém komunikační protokolu). U

tohoto procesoru nejsme omezeni velikostí paměti ROM (jak vyplývá z názvu procesoru – 128 KB – což je dostatečná velikost i pro mnohonásobně větší projekty).

Tento jazyk umožnil napsat přehledný a funkční kód, s dokonalejším protokolem. Který je snadno rozšiřitelný o další funkce spojené s dalšími periferiemi.

### 3.1.6 C#

V tomto jazyce byla napsána druhá verze GUI. K tomuto jazyku bylo přistoupeno z důvodu menší SW podpory jazyka C++ (nutnost použít složitější řešení poměrně snadných úkonů a méně komfortní programovací prostředí (oba jazyky psány v Microsoft Visual Studio 2010 Ultimate) nejedná se tedy o nevhodnou volbu prostředí).

V C# bylo využito (po analýze dat odesílaných do wifi modulu, pomocí odchytávání síťové komunikace bylo zjištěno, že se data odesílají nešifrovaná - v ASCII hodnotě) přímé odesílání dat přes TCP/IP (odpadá nutnost využití aplikací třetích stran pro vytváření virtuálních COM portů a zároveň se nabízí možnost napsat aplikaci i pro zařízení pro která neexistuje aplikace vytvářející COM port – např. smartphone a jiné). V této aplikaci byl aplikován dokonalejší protokol, zajišťující kvalitnější komunikaci mezi robotem a aplikací (původní protokol vycházel z bezchybného přenesení všech dat, ten byl díky nezřídka vypadlým datům špatně aplikovatelný).

### 3.1.7 J2ME

Tento jazyk byl vybrán pro jeho snadnou přenositelnost mezi platformami (na světě je přes 1,5 miliardy zařízení používajících Javu). J2ME je odnož Javy – ořezaná pro použití v mobilních telefonech. V tomto jazyce byla napsána aplikace pro smartphone.

## 3.2 Výstavba SW

### 3.2.1 Propojení portů MCU a zbylých obvodů

Pro vhodné napsání SW bylo zapotřebí určit jaký obvod bude připojen k jakému pinu procesoru. Z toho důvodu byla vytvořena tabulka pro snadné zapojení obvodů.

| Přiřazené obvody | PORT.PIN |
|---|---|
| UART | D.2, D.3 |
| Dělič napětí z akumulátorů | F.0, F.1 |
| Spínání periferií | G.1, C.0 |
| Akustická signalizace | B.4 |
| Enable MOT | B.6, B.7 |
| MOT 1 | B.2, B.3 |
| MOT 2 | G.3, G.4 |
| LCD display | C.1 - C.6 |
| Podsvícení LCD | B.5 |
| Podsvícení | C.7 |

**Obrázek 21 - Připojení periferií**

### 3.2.2 Program MCU

Před samotným programováním je nutné nastavit pojistky procesoru (fuse) podle použitého krystalu, nastavení kompatibility s ATmega103, povolení JTAG a dalších pojistek. Pojistky lze vygenerovat například pomocí Engbedded Atmel AVR® Fuse Calculator[1].

Program pro MCU je psán v jazyku C.

Součástí programového vybavení MCU jsou napsané knihovny:

UART (obsahuje funkci pro inicializaci UART1, pro příjem a odesílání dat (data jsou ukládána a čtena z 64 B velkých bufferů, které jsou po odeslání/přečtení dat vyprazdňovány) a obsahuje i funkci „con_not_alive“ (tato funkce je spouštěna pomocí Timeru2 při neexitujícím spojení s PC (smartphone))).

PWM (obsahuje funkci pro inicializaci PWM1, pro nastavení a zastavení motorů).

ADC (obsahuje funkci pro inicializaci ADC, pro zadání požadavku na změření napětí a jeho další zpracování, též i pro odpojení napájení robota v závislosti na úrovní napětí akumulátorů).

TIME (obsahuje funkce pro inicializaci Timeru0, 2 a spínání akustické signalizace).

LCD (obsahuje funkci pro inicializaci displaye a funkcemi pro zápis znaků na display)

[1] Engbedded Atmel AVR® Fuse Calculator. *Engbedded – ensouling circuits* [online]. 2011 [cit. 2012-03-09]. Dostupné z: http://www.engbedded.com/fusecalc

Po startu MCU dojde k inicializaci, která je rozdělena do několika větví:

První větví je nastavení ADC převodníku.

Následuje inicializace rozhraní UART na baud rate 9600.

Další v pořadí je inicializace časovače 0, který má funkci frekvenčního modulování akustické signalizace.

Dále inicializace časovače 2 určeného ke kontrole připojení k PC, při nefungujícím připojení provede činnosti spojené s odpojením od PC (vypnutí motorů, zvukové signalizace, případně vypsání chybové hlášky na display). Zároveň plní funkci automatického spouštění procesů, jako odesílání informací z robota (napětí akumulátorů, …) a kontrolování minimální povolené hranice napětí akumulátorů.

Jako předslední je spuštěna inicializace PWM pro motory.

Na závěr je spuštěna inicializace LCD (po sepnutí logické větve napájení -> pouze v případě dostatečné úrovně napětí akumulátorů)

Po inicializaci se procesor dostane do nekonečné smyčky, v které vyčkává na povely z PC (smartphonu).

```
while (1) {
        recieve_Byte();
        adc_read(); //měření průběžně v průběhu nadefinované
časové konstanty
}
```

### 3.2.3 Program PC aplikace

Z důvodu výrazně složitějšího kódu nebude popisován kód tak detailně jako u MCU, popsány budou pouze nejdůležitější části.

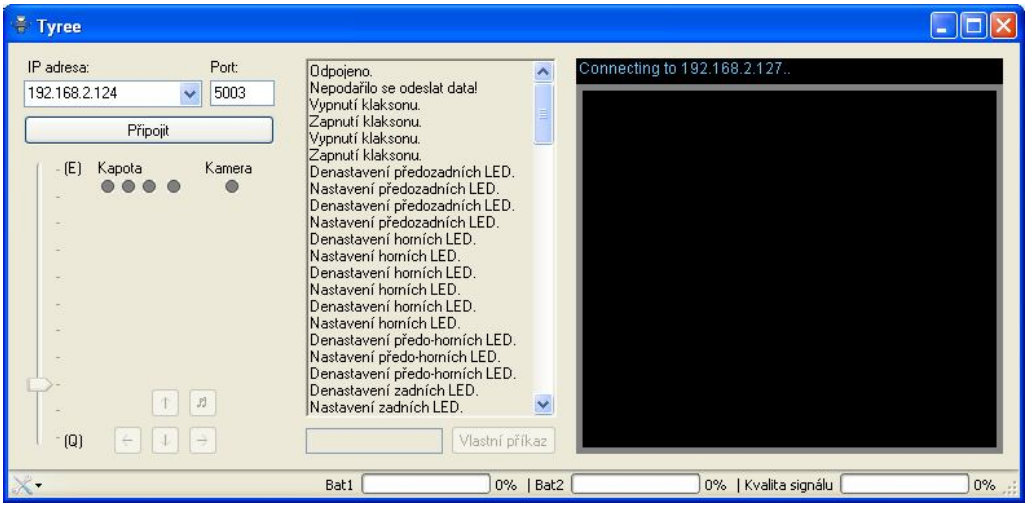
Aplikace je psaná v jazyku C#.

Součástí aplikace jsou napsané knihovny funkcí:

Pro úpravu GUI v závislosti na aktuálním stavu připojení/nastavení/získaných dat z robota.

A TCP IP, která se stará o připojení k robotovi, odesílání dat, odesílání „alive“ paketu, přijímání dat a zjišťování statutu wifi modulu.

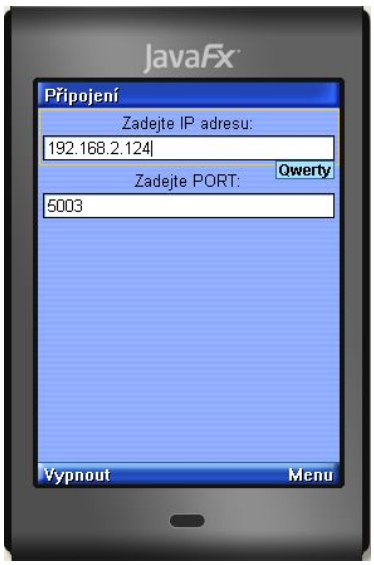
Aplikace využívá několik vláken (threadů) které se starají o GUI, příjímání a odesílání dat odděleně, aby nedocházelo k „zamrzání“ aplikace, které je uživatelsky nepříjemné, přestože nemá hlubší dopad na chod aplikace. Mezi těmito vlákny jsou nadefinovány funkce pro předávání potřebných dat z a do vlákna (IP adresa, port, požadovaná činnost, přijatá data).




**Obrázek 22 - Náhled PC aplikace**

### 3.2.4 Program pro smartphone

Tato aplikace je psaná v prostředí NetBeans v J2ME pro touch display. Obsahuje protokol použitý v GUI aplikaci pro PC a obsahuje přepracované GUI pro malý display. Z důvodu menšího displaye je z aplikace vypuštěn příjem a zobrazení obrazu kamery.




**Obrázek 23 – Náhled mobilní aplikace**

### 3.2.5 Protokol komunikace

Díky nutnosti přenášet data o délce větší než 1 B bylo nutné navrhnout protokol komunikace. První verze, v které nebyla aplikována automatická synchronizace, nebyla stabilní. Proto byl navržen složitější protokol, po testování se ukázal jako stabilní. Pro přenos jedné informace byly vymezeny 3 B, 1. synchronizační o hodnotě 0xFF (HEXA -> 255DEC), 2. s označením „pointer“ (ukazatel) – určuje druh činnosti a 3. nesoucí hodnotu nastavované činnosti.

Např.:

| Činnost | Synchronizace | Pointer | Data |
|---|---|---|---|
| Nastavení motorů, zpětný chod, 15% rychlostí | 11111111 | 00000001 | 10001111 |
| Zastavení motorů | 11111111 | 00000100 | 00000001 |
| Nastavení 1. a 3. LED na kapotě | 11111111 | 00000110 | 00000101 |

**Obrázek 24 - Ukázka struktury protokolu**

# Závěr

V tomto projektu byl vytvořen design aplikací a modelu. Byla navrhnuta schémata pro jednotlivé moduly, podle nich nakresleny a osazeny DPS. Všechny byly úspěšně oživeny.

Díky volným pinům procesoru je stále možné robota rozšiřovat o senzory, manipulační ruku, detektor řeči a jiné periferie.

Projekt začal být vypracováván již před říjnovým schválením v roce 2011, však jeho aktuální podobu (po vizuální i elektronické stránce) získával až v roce 2012. Na projektu bylo velice intenzivně pracováno, odhadovaný čas řešení je přes 1000 hod. Cena dílů použitých v poslední verzi přesahuje 6000 korun, kompletní výdaje užité pro vývoj všech verzí přesahují 11 000 korun.

Dalším z cílů je osadit robota manipulační rukou a displayem pro možnost zobrazení aktuálního stavu hodnot robota, případně komunikovat s okolím (přenos od robota je realizován vizuálním i audio signálem, přenos do robota je zatím realizován pouze akustickým signálem a světelnou signalizací).

# Seznam obrázků

Obrázek 1 - Nano Socket iWifi 4
Obrázek 2 - Mini Socket iWifi 5
Obrázek 3 – Atmel 89C2051 6
Obrázek 4 - ATmega128A 6
Obrázek 5 - Programátor pro Atmel 89C2051 7
Obrázek 6 - ISP programátor pro AVR procesory 7
Obrázek 7 - Deska pro úpravu komunikace 8
Obrázek 8 - Driver L298 8
Obrázek 9 - Krokový motor 8
Obrázek 10 - Stejnosměrný motor 9
Obrázek 11 - Driver L298 v2 9
Obrázek 12 - Redukce z motoru na kolo 10
Obrázek 13 - Olověný článek 10
Obrázek 14 - Li-pol akumulátory 11
Obrázek 15 - Ošetření kritických stavů akumulátorů 11
Obrázek 16 - Stabilizátory pro logickou větev 12
Obrázek 17 – Podvozek 12
Obrázek 18 - Nákres podvozku v2 13
Obrázek 19 - Podvozek v2 13
Obrázek 20 - IP kamera 14
Obrázek 21 - Připojení periferií 17
Obrázek 22 - Náhled PC aplikace 19
Obrázek 23 – Náhled mobilní aplikace 19
Obrázek 24 - Ukázka struktury protokolu 20

# Použité zdroje

[1] ČVUT-FEL. Stručný přehled základní syntaxe jazyka C. Praha [CD-ROM], 1999, 17 s.

[2] ATMEL CORPORATION. *Atmel* [online]. 2012 [cit. 2012-03-11]. Dostupné z: http://www.atmel.com/

[3] Connect One Ltd. *Connect One* [online]. 2012 [cit. 2012-03-11]. Dostupné z: http://www.connectone.com/

[4] MICROSOFT. *MSDN Library* [online]. 2012 [cit. 2012-03-11]. Dostupné z: http://msdn.microsoft.com/en-us/library/ms123401.aspx

[5] ORACLE CORPORATION. *NetBeans* [online]. 2012 [cit. 2012-03-11]. Dostupné z: http://netbeans.org/

[6] ORACLE CORPORATION. *Oracle* [online]. 2012 [cit. 2012-03-11]. Dostupné z: http://www.oracle.com/index.html

[7] SE SPEZIAL-ELECTRONIC AG. *Spezial Electronic* [online]. 2012 [cit. 2012-03-11]. Dostupné z: http://www.spezial.cz/

# Přílohy

A. Obsah přiloženého CD ..... 25
B. Schéma a DPS ATmega128A ..... 26
C. Schéma a DPS stabilizátoru logických úrovní ..... 27
D. Schéma a DPS ošetření kritických stavů akumulátorů ..... 28
E. Schéma a DPS H-můstků ..... 29
F. Schéma a DPS dodatkových obvodů ..... 30
G. Blokové schéma ..... 31
H. Fotodokumentace ..... 32

# A. Obsah přiloženého CD

„AvrCalc.exe“ – výpočet timerů a baud rate pro AVR procesory

„eagle“ – složka se schématy DPS robota

„jazykc.pdf“ – přehled syntaxe C

„SolidEdge“ – složka s výkresy podvozku v SolidEdge

„Tyree - NBeans„ - složka projektu v NetBeans 7.1.1

„Tyree 2 - AVR“ - složka projektu v AVR Studiu 4

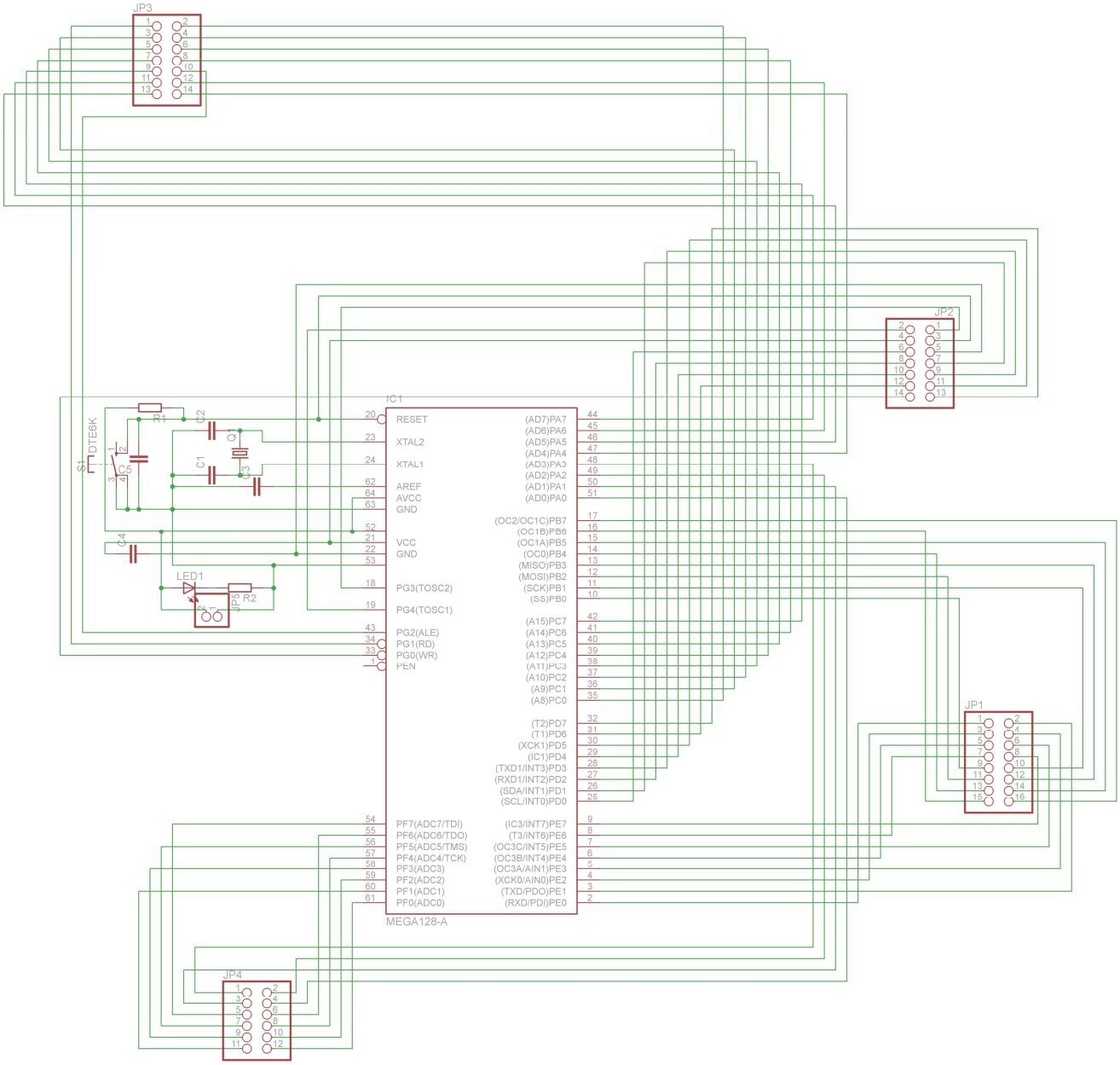
„Tyree 2.3 – MVS2010“ - složka projektu v Microsoft Visual Studiu 2010 Ultimate

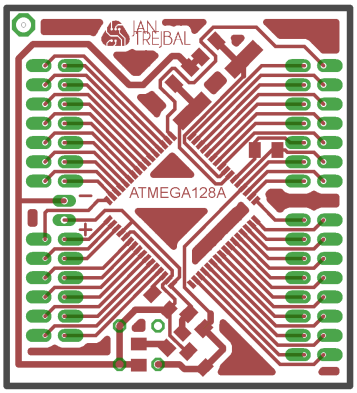
„Tyree - DMP.doc“ – elektronická podoba DMP

„Tyree - SOČ.pdf“ – elektronická podoba SOČ

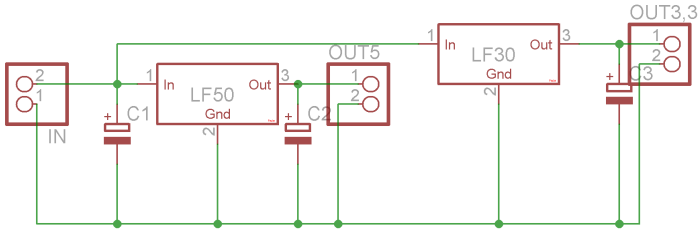
„wifi_settings.rpf“ – soubor s nastavením wifi modulu

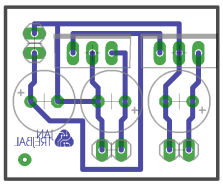
# B.Schéma a DPS ATmega128A



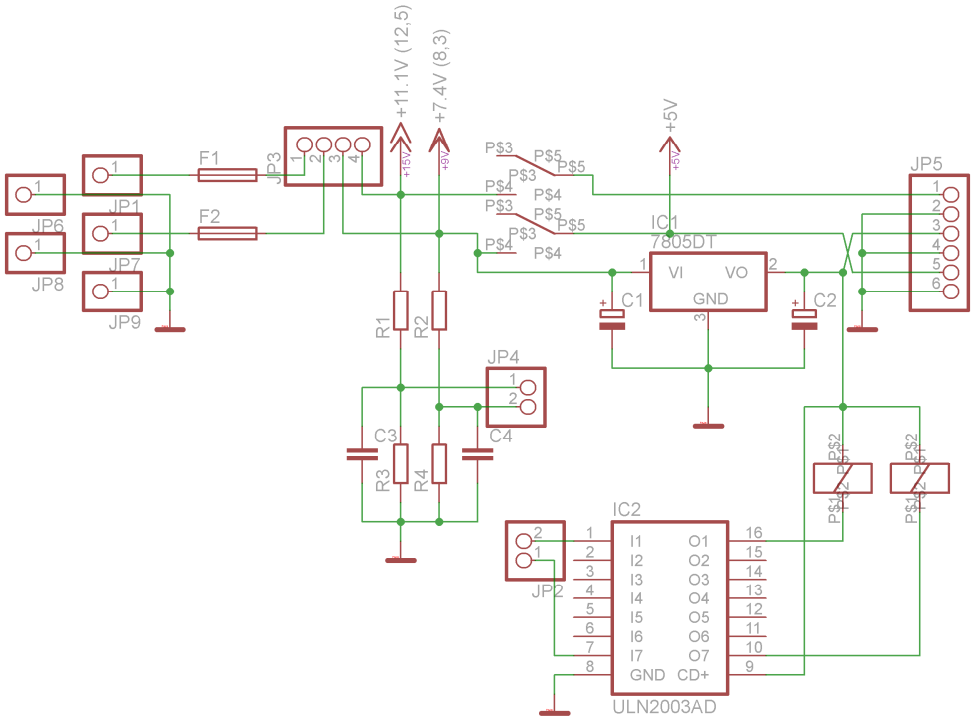



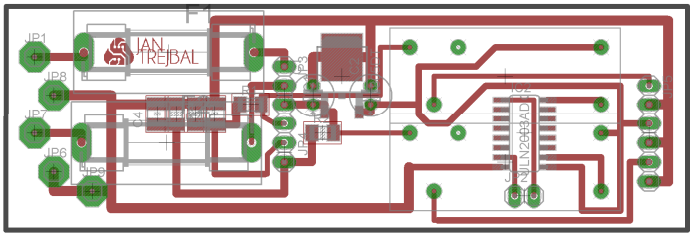
# C. Schéma a DPS stabilizátoru logických úrovní



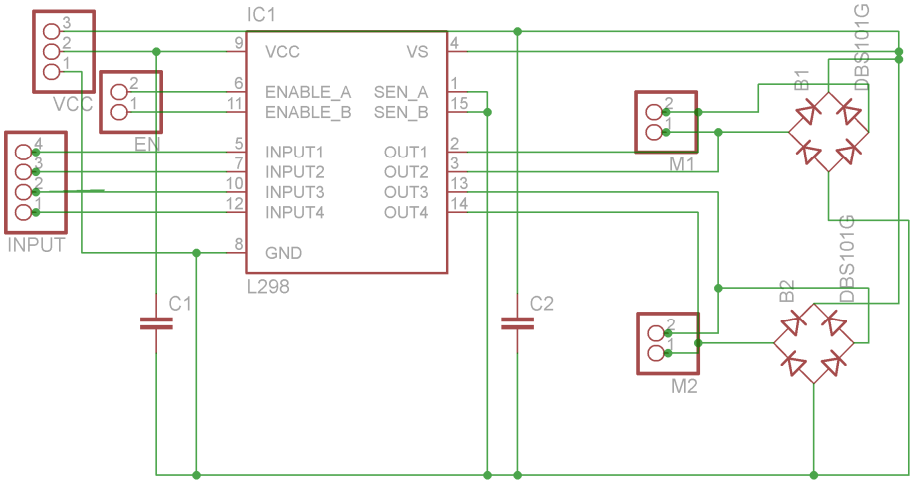



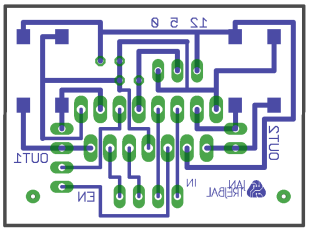
# D.Schéma a DPS ošetření kritických stavů akumulátorů



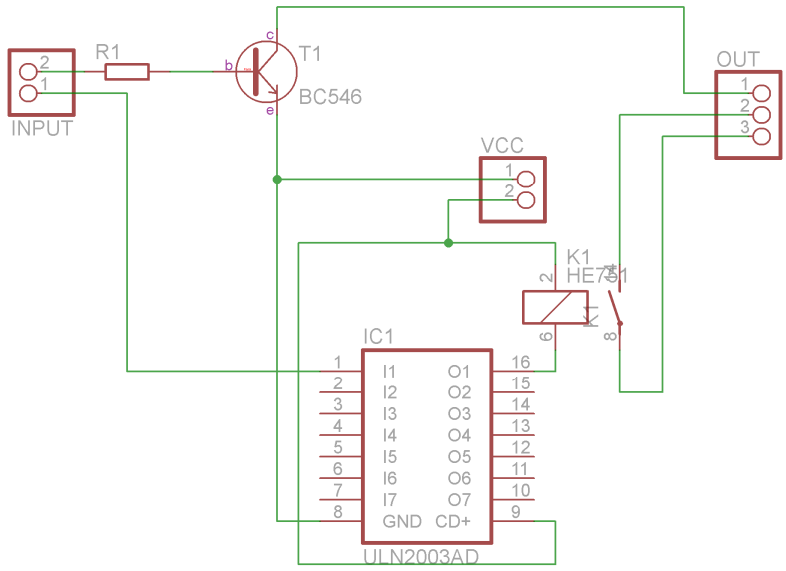



# E. Schéma a DPS H-můstků

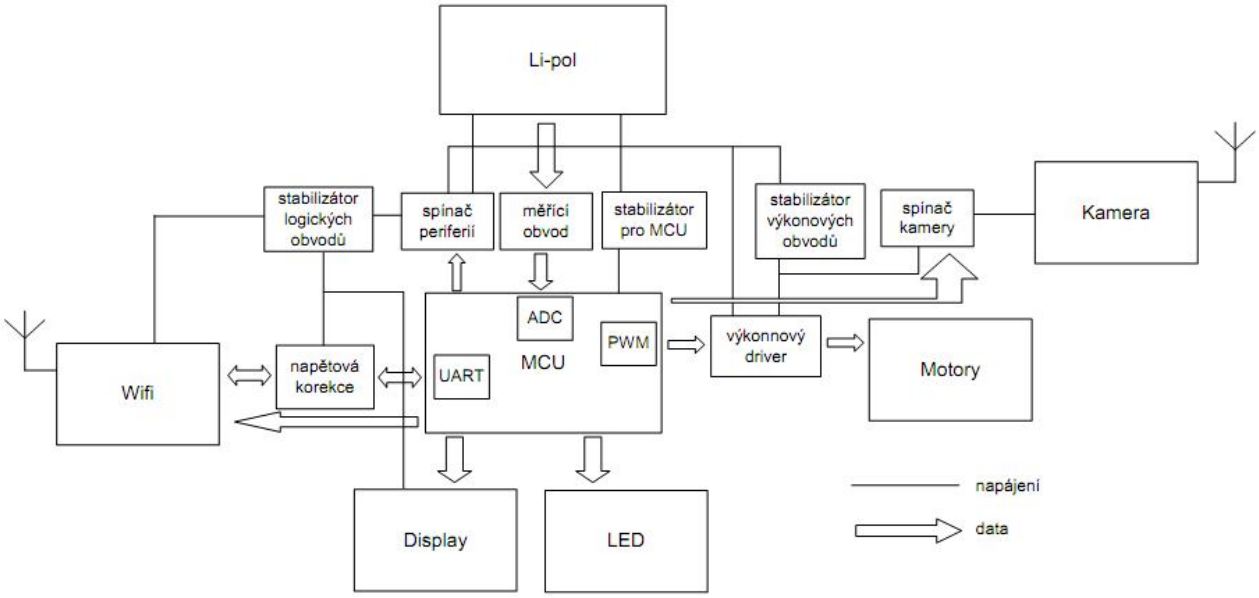
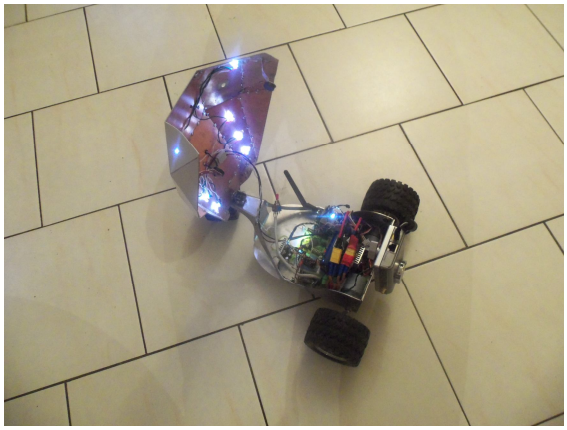
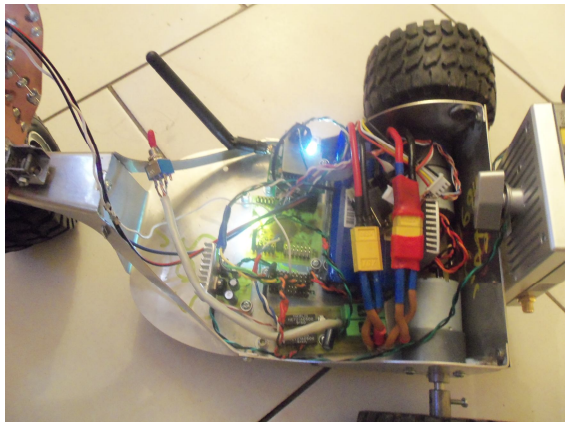
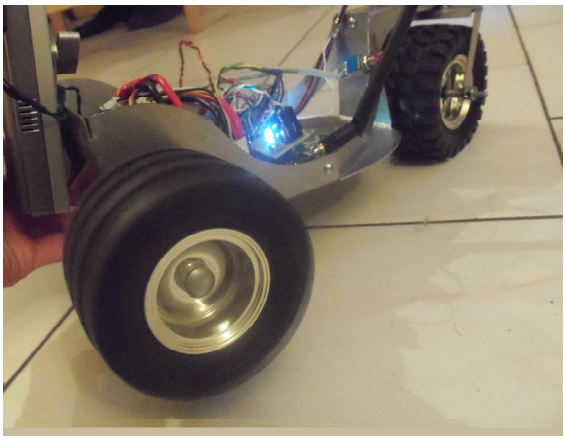


# F. Schéma a DPS dodatkových obvodů



# G. Blokové schéma



# H.Fotodokumentace