



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra aplikované fyziky a techniky

Diplomová práce

Modulární systém s jednočipovým  
mikroprocesorem Atmel AVR pro měření  
při hrách v přírodě.

Vypracoval: Bc. Bohumil Kroupa

Vedoucí práce: Ing. Michal Šerý Ph.D.

České Budějovice 2016

## **Prohlášení**

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě - v úpravě vzniklé vypuštěním vyznačených částí archivovaných Pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Prachaticích dne 19. prosince 2016

.....

**Anotace:**

Tato diplomová práce se zabývá návrhem, konstrukcí a programováním univerzálního modulárního systému časomíry pro sportovní hry dětí v přírodě. Základem systému jsou jednočipové mikrokontroléry AVR Atmel ATmega 32 a ATmega 328. Přenos dat mezi moduly je realizován pomocí bezdrátového sériového portu HC-12. Systém je použitelný pro více druhů soutěží s různými typy čidel pro měření času a není závislý na nepřetržité dodávce elektrické energie.

**Klíčová slova:**

Měření času při soutěžích v přírodě, bezdrátový sériový port, AVR Atmel ATmega 32, ATmega 328, BasCom, KiCad, Microsoft Visual Studio.

**Annotation:**

This diploma thesis deals with the design, construction and programming of universal modular system of timekeeping for children's outdoor sports games. The basis of this system are single-chip AVR microcontrollers Atmel ATmega 32 and ATmega 328. Data transmission between the modules is realized by a wireless serial port HC-12th. The system is applicable to multiple types of competitions with different types of sensors for time measuring, and it is independent of the continuous supply of electricity.

**Keywords:**

Time measuring at the competitions in nature, wireless serial port, Atmel AVR ATmega 32, ATmega 328, BasCom, KiCad, Microsoft Visual Studio.

## **Poděkování**

Chtěl bych touto cestou poděkovat panu Ing. Michalu Šerému Ph.D. za cenné rady a odborné vedení diplomové práce a panu Tomáši Balcerovi za pomoc při výrobě.

# Obsah

1	Úvod.....	7
2	Popis soutěží.....	7
2.1	Rýžování zlata .....	7
2.2	Soutěže mladých požárníků.....	10
2.3	Startování.....	11
3	Použitý SW pro vývoj a konstrukci.....	12
3.1	Program KiCad .....	12
3.2	Bascom AVR.....	16
3.3	Microsoft Visual Studio .....	19
4	Konstrukce .....	20
4.1	Atmel ATmega328.....	21
4.2	Bezdrátový sériový port HC-12.....	22
4.3	Atmel ATmega32L .....	24
4.4	LCD 16x2 .....	25
4.5	Stanice (stopky) .....	26
4.5.1	Deska plošných spojů s mikrokontrolérem AT Mega 328.....	28
4.5.2	Deska plošných spojů s LCD.....	29
4.6	Základna .....	32
4.6.1	Propojovací deska pro ATmega32 .....	34
5	Software .....	36
5.1	Program pro stanici – stopky.....	36
5.2	Program pro základnu .....	39
5.2.1	Identifikace (IDENT) .....	40
5.2.2	Nulování (NULL) .....	40
5.2.3	Vyčítání dat ze stanic (COUNT).....	40
5.3	Program pro připojený notebook. ....	42

6	Závěr .....	44
7	Přílohy:.....	46
7.1	Stupně krytí (ČSN EN 60 529) [13] .....	46
7.2	Bloková schémata použitých mikrokontrolérů.....	47
7.3	Seznam obrázků.....	49
7.4	Seznam tabulek.....	51
8	Použité prameny a literatura.....	52

# 1 Úvod

Úkolem práce bylo sestrojít elektronickou bezdrátovou časomíru pro soutěže v rýžování zlata mladých zlatokopů. Stopky musí být odolné jak mechanicky, tak proti prachu, zvýšené vlhkosti i možnému polití. Celý systém se skládá z jedné centrální „základny“ a z několika „stanic“. Stanice jsou samostatné stopky umístěné vedle každého soutěžícího, ovládané základnou. Protože jsem chtěl, aby časomíra byla univerzální a různě konfigurovatelná, postavil jsem celý systém na mikrokontrolérech Atmel. Velká variabilita mikrokontrolérů mi dovolí měřit čas s pomocí široké škály čidel, ať už analogových nebo digitálních. Díky tomu je možné systém rozšiřovat, modulárně upravovat, přeprogramovat a používat ho i pro jiné druhy soutěží pro děti i dospělé. Už během práce se mi ozvali vedoucí mladých hasičů, že by měli o podobnou časomíru zájem. Při hromadných soutěžích obecně jsou vždy problémy s rychlým a spolehlivým vyhodnocováním časových údajů. Použité systémy jsou buď nedokonalé, pomalé a nespolehlivé nebo velmi drahé, což si většina klubů pracujících s dětmi nemůže dovolit. Proto jsem chtěl vyvinout časomíru dostatečně robustní, spolehlivou, s rychlým zpracováním výsledků a také finančně přijatelnou. Desku s ATmega328, která je použita ve stanicích, lze dnes koupit za přibližně 60,- Kč a modul ATmega 32 Mini, který jsem použil v základně, za 210,- Kč. Největší položkou na stanici byla nakonec montážní krabice s cenou 220,- Kč. Díky takto nízkým cenám nejdůležitějších částí se celý systém stal dostupný pro mnohem větší počet dětských sportovních klubů. To byl vlastně účel celé mé práce - aby děti mohly provozovat svůj oblíbený sport s vybavením podobným, jako je na vrcholových mezinárodních soutěžích. V rýžování zlata naši sportovci patří mezi světovou špičku a šampionátů se pravidelně účastní.

## 2 Popis soutěží

### 2.1 Rýžování zlata

Princip soutěže v rýžování zlata je v hledání několika zlatěnek – maličkých šupinek zlata o velikosti 1 až 2 mm v 10 kg říčního písku smíchaného s vodou. Zlatěnky se hledají pomocí plechové rýžovací pánve, děti mají lehčí plastovou. Každý soutěžící má před sebou plastový „bazének“ - rýžoviště o rozměrech 130 x 180 x 50 cm, ve kterém je přibližně 20 cm vody a vědro s pískem, s již připravenými zlatěnkami. Přesný počet zlatěnek umístěných

v písku závodníci neznají. Vedle bazénku je stojan s tlačítkem pro zastavení běžícího času. Těchto stanovišť pro závodníky je na závodišti dvacet až třicet, podle typu soutěže. Jakmile rozhodčí odstartuje závod, spustí časomíru, závodníci vysypou písek z vědra do rýžoviště a začnou rýžovat. Pokud si myslí, že už našli všechny zlatěnky, stisknou tlačítko vedle rýžoviště a tím zastaví svůj čas, který se následně vyhodnocuje. Z předešlého vyplývá, že závody v rýžování jsou poměrně outdoorová aktivita a tomu musí odpovídat i konstrukce systému časomíry. Bezdrátová konstrukce byla zvolena z důvodu odstranění propojovacích vodičů proplétajících se na sportovišti. Tím jsem chtěl zvýšit bezpečnost sportovců při pohybu po ploše a zároveň snížit poruchovost zařízení možným utržením (ukopnutím) těchto vodičů.



Obr. č. 1, Závodnice chce právě stisknout tlačítko.

Na Obr. č. 1 je vidět, jak mladá závodnice, která právě ukončila rýžování, chce stisknout tlačítko připevněné na sloupku vedle ní. Před ní i v pozadí jsou vidět kabely, které vedou k hodinám umístěným na začátku sportovního pole. Na Obr. č. 2 jsou na zemi vidět svazky kabelů, které se sbíhají v zařízení časomíry, pro přehlednost označeném šipkou. Tato



konkrétní časomíra funguje tak, že při zmáčknutí tlačítka se zobrazení času na chvíli zastaví a u něj na displeji se ukáže číslo závodního místa. Rozhodčí si oba údaje opíše a časomíra pokračuje dál. Z tohoto jednoduchého popisu je jasné, že může dojít k chybě a často dochází k reklamacím zaznamenaného času od závodníků. Tedy hlavně od dospělých závodníků.



Obr. č. 2, Zařízení kabelové časomíry.

Mnou vytvořený systém zaznamenává jednotlivé naměřené časy automaticky do paměti základny a ta je pak předá počítači (notebooku), který je automaticky vyhodnotí. Kromě toho se čas každého závodníka zobrazuje na displeji umístěném na každé jednotlivé stanici. Tento výsledný čas zůstane na displeji až do té doby, než rozhodčí všechny stanice povelom vynuluje. Tím je prostor pro případnou kontrolu času rozhodčím přímo na místě sportovce a protesty závodníků se minimalizují. K hodnocení nestačí jen dosažený čas, důležitý je i počet nalezených zlatěnek. Pokud závodník nenajde všechny zlatěny umístěné v písku, připočítají se mu „trestné body“, které mu rozhodčí připíše k času. To je už ale věcí programu na počítači a nesouvisí přímo se systémem základny a stanic.

## 2.2 Soutěže mladých požárníků

Při soutěži požárníků – požárním útoku se používají dva terče v určitém rozestupu od sebe. Terče jsou buď nástřikové, dotykové nebo sklopné. U dotykových terčů, které se zpravidla používají při dětských soutěžích, elektronika vyhodnocuje pokropení terče vodou z proudnice. Sklopné terče jsou jednodušší variantou dotykových terčů. U nich je uprostřed terče za otvorem umístěn válec, který se po zásahu vodním proudem překlápí a stiskne kontakt, nepotřebuje tedy elektroniku pro vyhodnocení zásahu. Nástřikový terč má uprostřed také otvor, ale za ním je nádrž o objemu 5 až 10 litrů se snímačem hladiny, většinou plovákem. Při použití nástřikových terčů je úkolem soutěžících udržet vodní proud na středu terče tak dlouho, než se nádoba naplní do určité hladiny a plovák sepne kontakt. Při požárním útoku se soutěžícím počítá čas od startu do zásahu druhého terče v pořadí. Pro tento typ soutěží bude systém časomíry poněkud jednodušší. Kromě základny se použijí jen dvě stanice, ke kterým se připojí spínače od jednotlivých terčů. Do stanic se nebudou instalovat LCD, naopak k základně se připojí velký LED panel zobrazující ubíhající čas, který se po zasažení terčů zastaví. Na vyhodnocení obou terčů by stačila stanice jedna, ale terče by musely být spojeny kabelem a tím bychom přišli o jednu z důležitých vlastností časomíry – nezávislost na vodičích.

Na Obr. č. 3 a Obr. č. 4 jsou dva druhy terčů před a po úspěšném útoku.



Obr. č. 3, Nástřikové terče (vpravo naplněn vodou), převzato a upraveno z [1].



Obr. č. 4, Mechanismus sklopných terčů, převzato a upraveno z [2].

Mladí požárníci soutěží i v jiných disciplínách, jako jsou například štafety 4 x 60 m, CTIF a štafeta požárních dvojic, které se startují stejným způsobem jako požární útok, ale výsledný čas se vyhodnocuje pomocí fotobuňky. Není problém do jedné ze stanic místo tlačítka nebo paralelně k němu připojit fotobuňku a jinak lze stanici ponechat tak jak je včetně LCD, který může zobrazovat cílový čas souběžně s velkým LED panelem. Dokonce není nutné ani nijak zásadně upravovat software celého systému, jelikož je velmi univerzální.

Z předešlého textu je patrné, že díky zvoleným mikrokontrolérům Atmel a jejich variabilitě lze systém časomíry použít v celé řadě dalších soutěží pro děti i dospělé.

## 2.3 Startování

Soutěže zlatokopů i požárníků se obvykle startují výstřelem, píšťalkou nebo jiným hlasitým zvukovým signálem. Původně jsem chtěl do systému zapracovat startování pistolí. Při pokusech s různými mikrofony a piezoelektrickými snímači, které jsem prováděl v různých prostředích, jsem zjistil, že snímání výstřelu akusticky je celkem složité a nespolehlivé. Často systém nepozná, že bylo vystřeleno, nebo naopak vyhodnotí jako výstřel zvuk, který výstřelem není, např. zvuk motoru startujícího vodního čerpadla apod. V literatuře [3] jsem našel úplně jinak řešené startování časomíry výstřelem. Autor sestrojil nástavec na revolver, který mechanicky snímá tlak spálených plynů vzniklých při výstřelu „slepé“ munice. Informace o výstřelu se pak přenáší do systému bezdrátově pomocí IQRF



modulu. Toto řešení se mi jeví jako velmi elegantní, ale je závislé na funkci sestavy jednoho konkrétního upraveného revolveru s elektronikou. Pokud by se tato sestava porouchala nebo z nějakého důvodu nebyla k dispozici, velmi těžko bychom jí nahradili něčím jiným a startování by nebylo možné. Kromě toho nákup revolveru, jeho mechanická úprava a výroba bezdrátového snímače výstřelu by celý systém prodražilo. Také se domnívám, že mechanický snímač výstřelu se bude časem zanášet povýstřelovými zplodinami a tím se bude zhoršovat spolehlivost a funkčnost tohoto řešení. Tomu jsem se chtěl vyhnout. Pro startování jsem tedy zvolil sirénu, která po zmáčknutí startovacího tlačítka spouštějícího časomíru vydá hlasitý zvuk – signál pro start soutěže. Tato konkrétní siréna byla původně součástí zabezpečovacího zařízení, je tedy dostatečně hlasitá, mechanicky i klimaticky odolná, což je při vyjmenovaných typech soutěží velmi žádoucí.

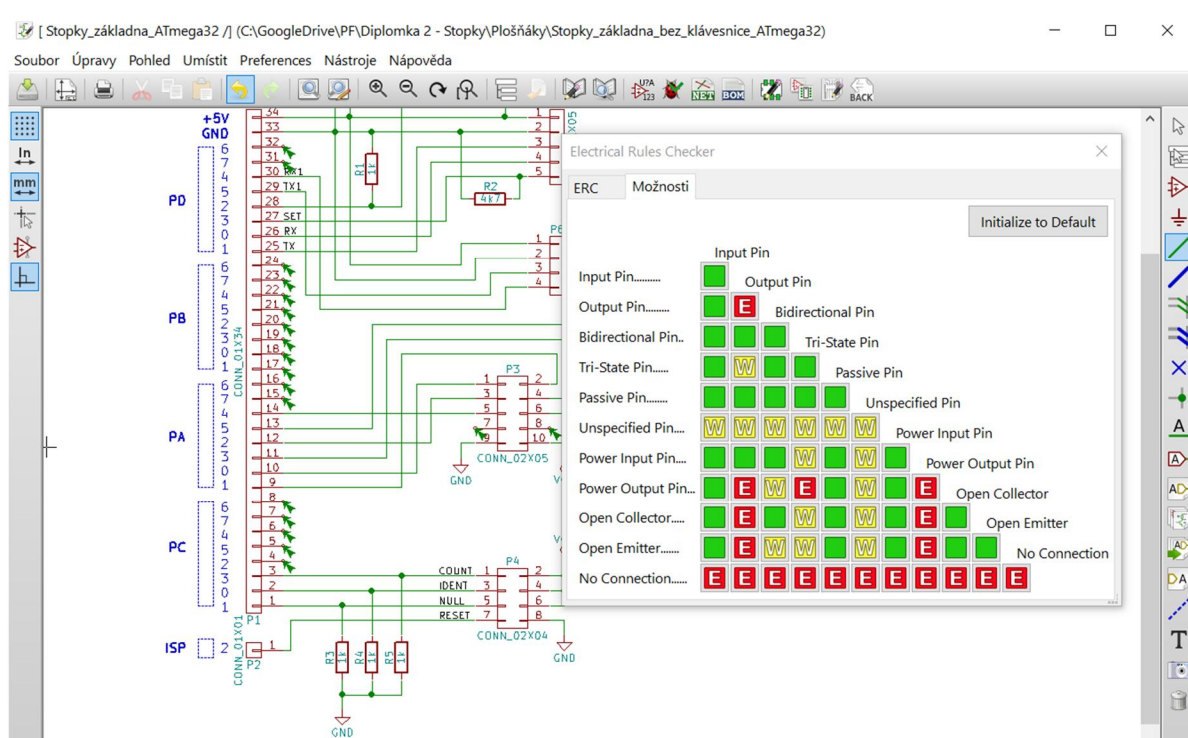
### 3 Software použitý pro vývoj a konstrukci

#### 3.1 Program KiCad

Pro tvorbu schémat a návrh všech desek plošných spojů (dále jen DPS) jsem použil program KiCad. Je to open source software pro návrh DPS a automatizaci designu elektronických obvodů EDA (electronic design automation). Je produktem francouzských autorů a existuje od roku 1992, kdy byla uvolněna první release verze, jejímž autorem byl Jean-Pierre Charras. Dnes jej vyvíjí KiCad Developers Team. Program je volně šiřitelný pod licencí GNU General Public License (GPL) version 3 nebo vyšší. Dříve jsem pro návrhy DPS používal CAD soft EAGLE, který lze za určitých podmínek také používat bezplatně, ale jsem toho názoru, že KiCad je přinejmenším stejně dobrý a navíc podporuje funkce, které EAGLE neumí, např. výměna pouzdra součástky ve finálním návrhu DPS, trojrozměrné zobrazení navrhované desky apod. KiCad jako open source se neustále vyvíjí, má poměrně širokou komunitu uživatelů, kteří zpřístupňují knihovny nových součástek na Internetu, a tím dělají tento software velmi lehce použitelným. K dispozici jsou zdrojové kódy, program pracuje pod operačními systémy Linux, Microsoft Windows, Apple a dalšími. [4]

Hlavní okno programu je vlastně jen správcem projektů a z něj se spouští jednotlivé části: **Eeschema** - editor elektrického schématu. Slouží pro tvorbu a úpravy schématu elektrického obvodu. Lze v něm provádět umístování součástek, propojení, smazání celých nebo jen částí spojů, pojmenování sítě, globální označení, sběrnice, kopírování součástky

nebo bloků, kopírování bloků mezi schémata, vkládání textů, grafiky, obrázků a automatické očíslování součástek (anotace). Je v něm možno vytvářet jedno i více stránková schémata a hierarchická schémata. Pro kontrolu schématu je k dispozici ERC (elektrická kontrola schématu), kde lze pomocí matice nastavit chyby nebo varování při spojení různých typů vývodů. Na obrázku (Obr. č. 5) je hlavní okno editoru schémat s ovládacími panely okolo. Uvnitř je právě otevřeno okno kontroly elektrického schématu (ERC), na kterém je vidět, jak jsou aktuálně nastavena hlášení varování a chyb při probíhající kontrole. Na schématu jsou vidět zelené šipky, které ukazují výsledek proběhlé kontroly - nezapojené piny jednotlivých konektorů.

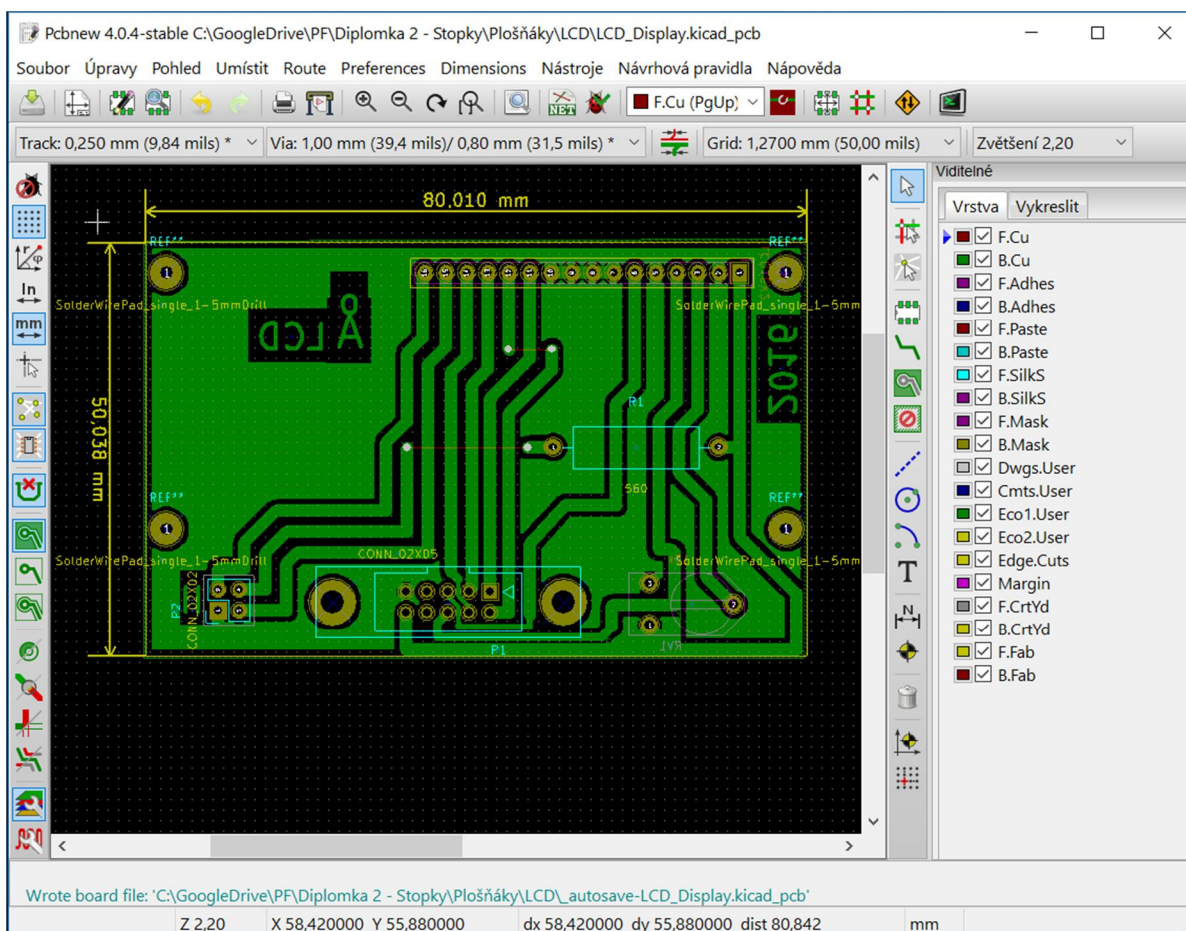


Obr. č. 5, Editor elektrických schémat.

Na ovládacím panelu v pravé části okna jsou prvky pro vlastní kreslení schémat, jako je vkládání součástek, spojů, sběrnic, uzlů (junction), textu apod. Schematické symboly součástek se vkládají z již existující knihovny. Pokud žádná schematická značka nevyhovuje, můžeme si pomocí vestavěného **Schematic library editoru** nějakou značku upravit nebo si vytvořit zcela novou, kterou pak do vybrané knihovny schematických značek zařadíme. Po nakreslení schématu se provede přiřazení pouzder jednotlivým součástkám. Pouzdra lze vybrat z připojených knihoven obsahujících pouzdra nebo si lze nakreslit vlastní pomocí vestavěného **Footprint editoru**. Tvorba pouzder tímto způsobem je velmi intuitivní.

Jako předlohu lze použít již existující pouzdro z libovolné knihovny a potom ho umístit do vlastní nebo jakékoliv jiné již připravené knihovny. Po přiřazení pouzder je třeba vygenerovat NETlist, se kterým pak pracuje editor plošného spoje.

Když je schéma hotové, můžeme rovnou přepnout do editoru plošných spojů **Pcbnew**. Po načtení NETlistu umístíme součástky na desku. Jejich rozměry a tvar jsme zvolili přiřazením v editoru schémat. V případě, že se nám nějaké pouzdro nehodí, můžeme ho hned nebo kdykoliv později změnit. Tohle je velká výhoda proti Eaglu, v němž je výměna pouzdra poměrně složitou operací a ve finální fázi návrhu zabere dost času. Kromě toho nevybíráme pouzdro ze stejné knihovny, ve které je symbol součástky, takže můžeme použít libovolné dostupné pouzdro. Jen je vhodné dodržet počet a zapojení vývodů součástky. Na Obr. č. 6 je vidět uspořádání okna **Pcbnew** při navrhování DPS, které je velmi podobné s oknem **Eeschema**.

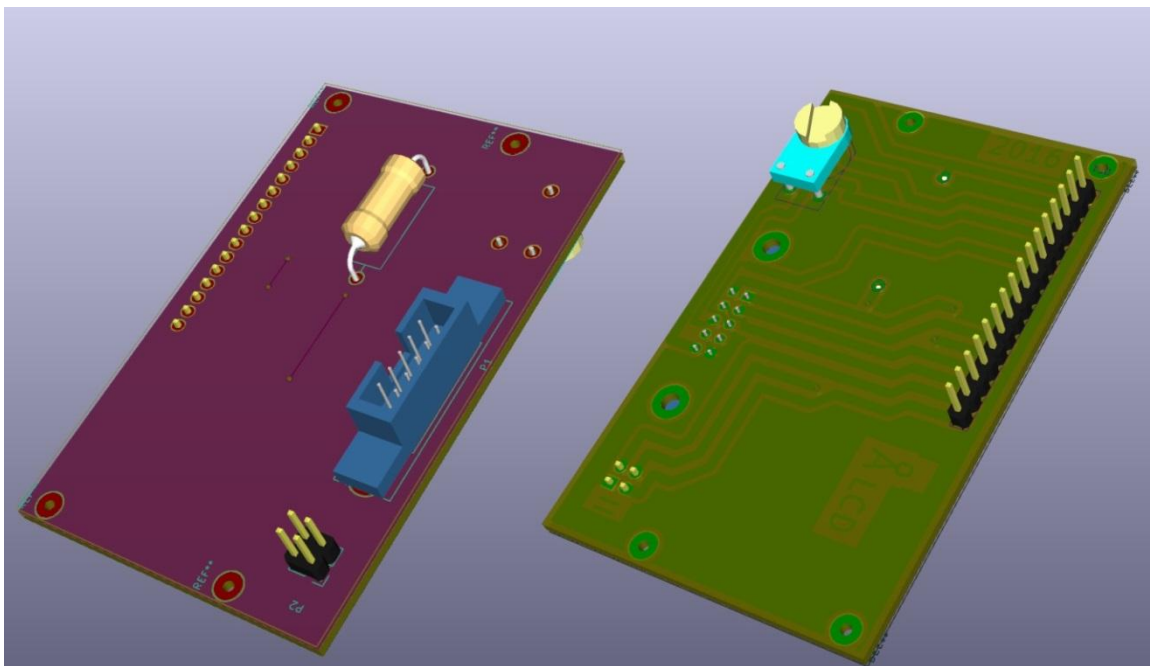


Obr. č. 6, Editor plošného spoje.

V levém ovládacím panelu jsou volby určující zobrazení pracovní plochy. Zobrazení/skrytí cest a průchodek, vyplněných ploch, druhy kurzoru, rastru atd. Vpravo je

panel s ovládacími prvky pro samotnou tvorbu DPS jako je vkládání a úprava pouzder součástek, spojů, výplňových ploch, textu a různých tvarů. Těsně vedle je panel vrstev výkresu. Na něm je možné kteroukoliv vrstvu zapnout nebo vypnout, změnit jí barvu, pojmenovat a určit právě aktivní vrstvu, se kterou se bude pracovat. V horní liště jsou kromě roletových menu ostatní ovládací prvky pro nastavení stránky, změnu a úpravu pouzder součástek, tisk, zvětšení a úpravu pohledu, načtení NETlistu, určení a výběr aktivních vrstev a kontrolu desky podle návrhových pravidel. Návrhová pravidla umožňují nastavení velkého množství parametrů návrhu jako je šířka spojů, izolačních mezer, průchodek pro různé vodiče a definování různých tříd sítí např. pro napájení. Tato pravidla si můžeme libovolně přizpůsobovat hlavně s ohledem na výrobní technologii desek, kterou budeme používat. Jestli např. budeme schopni zvoleným postupem spolehlivě vyrobit velmi tenké spoje nebo úzké izolační mezery. Důležité parametry pro nastavení návrhových pravidel se týkají také velikosti proudů tekoucích měděnými spoji, hodnoty nejvyššího napětí připojovaného k desce, atd.

Pokud použítá pouzdra mají v knihovně součástek vytvořený trojrozměrný model, můžeme si v editoru **Pcbnew** zobrazit 3D pohled, jak bude deska vypadat po osazení. To je výhodné například pro porovnání výšky součástek a jejich případné překrývání. S 3D modelem si můžeme libovolně otáčet a zvětšovat ho. Pokud 3D model u některé součástky chybí, můžeme ho jednoduše vytvořit pomocí programu Wings3D, který je také volně k použití.



Obr. č. 7, 3D pohled na desku pro LCD.

Předchozí obrázek je složený ze dvou pohledů na desku pro LCD. Je na něm vidět, že součástky jsou umístěny z obou stran plošného spoje a na spodní zelené straně můžeme rozeznat i měděné spoje a izolační mezery.

Součástí KiCadu je pomůcka nazvaná **PCB Calculator**. Ten kromě tabulky tříd DPS a barevného kódu značení odporů umožňuje vypočítat různé parametry desek. Např. šířku izolačních mezer podle připojeného napětí, šířku spoje podle procházejícího proudu a teploty desky, parametry útlumových R můstků (T, II a dalších). Funkce kalkulátoru se postupně rozšiřují spolu s vývojem KiCadu.

Program umí kromě svého formátu vytvářet a načítat soubory typu Gerber, SVG, Postscript, DXF, HPGL, PDF, DRL, XML, SpectraDSN atd. Pro převod již hotových schémat a návrhů DPS z jiných programů (např. Eagle, OrCAD, PCB123, Tango a jiné) existují různé konvertory, což usnadňuje návrhářům přechod na KiCad.

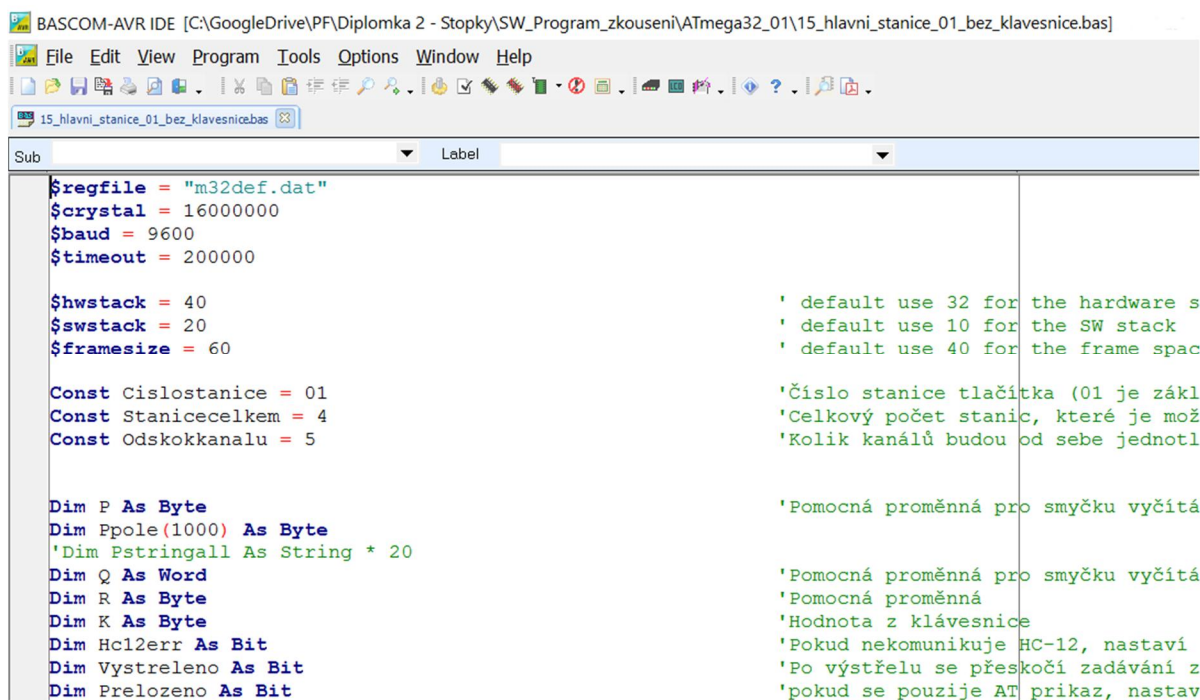
Ovládání programu je velmi intuitivní, na všechny operace lze používat myš a kontextová menu. Pokud si ale osvojíte alespoň některé klávesové zkratky, které jsou uvedené u každého povelu v menu, práce s programem se podstatně zrychlí.

## 3.2 Bascom AVR

Bascom (**BASicCOM**piler) je jednoduchý programovací jazyk k vytváření programů pro mikrokontroléry. V roce 1995 ho začal tvořit Mark Alberts, aby umožnil i začátečníkům psát programy pro jednočipové mikrokontroléry. Tento vývojový nástroj vydává firma MCS electronic ve dvou řadách. Bascom 51 pro mikrokontroléry rodiny I8051 a Bascom AVR pro RISCové mikrokontroléry Atmel AVR. Firma MCS electronic má na svých stránkách kompletní dokumentaci včetně nápovědy se všemi použitými příkazy a syntaxí. [5]

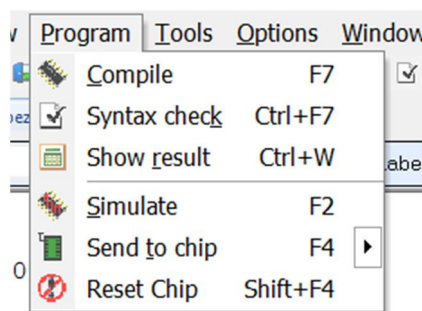
Vývojové prostředí (IDE) je windowsovským prostředím pro editaci programového kódu, překlad, debugging, simulaci atd. Jeho ovládání je vcelku intuitivní a většina akcí odpovídá zaběhlým postupům při práci s MS Windows. Editor zdrojového kódu automaticky zvýrazňuje syntaxi a upravuje text do sloupců pro lepší orientaci v napsaném kódu. Převážnou většinu ovládacích a nastavovacích prvků najdeme v nástrojové liště a v roletových menu v základním okně, viz následující obrázek.





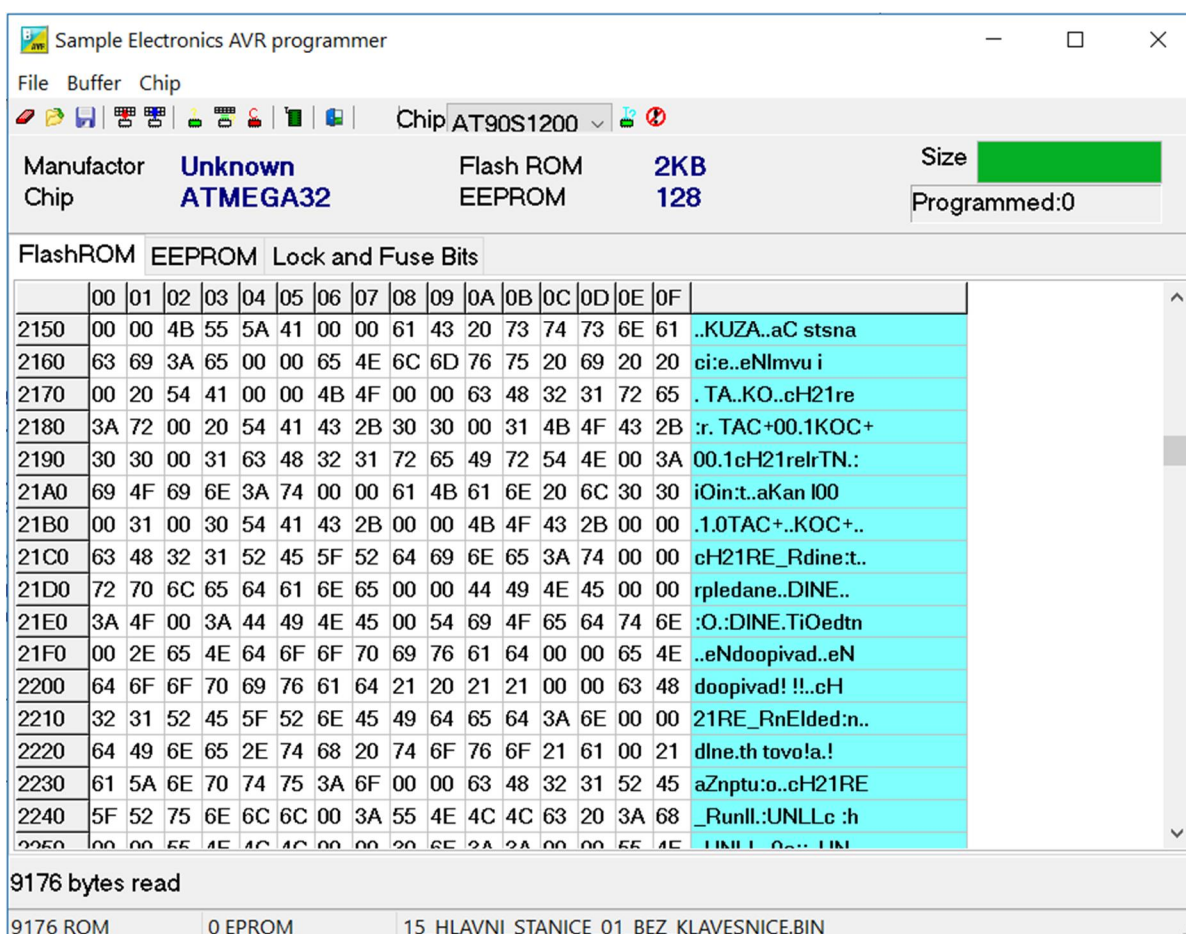
Obr. č. 8, Základní okno programového prostředí Bascom.

Kromě vcelku obvyklých menu *File*, *Edit* a *View*, je v roletce *Program* (Obr. č. 9) několik samostatných funkcí, které usnadňují vytváření programů v prostředí Bascomu. Položka *Compile* spouští překladač zdrojového kódu. Výsledkem je několik souborů s příponami .BIN a .HEX, sloužících pro naprogramování mikrokontroléru AVR a další, které se používají pro simulaci běhu, včetně soboru .RPT s protokolem o překladu. Příkaz *Syntax check* umožňuje kontrolu kódu a snadné opravení případných syntaktických chyb. *Show result* zobrazí již zmíněný protokol o překladu – soubor .RPT. Příkaz *Simulate* spouští vestavěný simulátor. Ten obsahuje okna zobrazující vlastní zdrojový kód, obsah všech registrů, příznaky, obsah paměti, stavy portů i periférií. Umí simulovat LEDky a klávesnice připojené k portu MCU i víceřádkový LCD. Při simulaci můžeme nastavovat breakpointy, měnit obsahy proměnných i registrů a krokovat vlastní program. *Send to chip* spouští AVR programmer (Obr. č. 10), kterým můžeme mazat kompletní paměť čipu, načítat a kontrolovat uložená data, ale hlavně programovat mikrokontrolér. Před samotným nahráním programu do čipu si programátor zkontroluje, zda připojený čip odpovídá napsanému programu, pak Flash paměť smaže a nahraje nový zkompileovaný program.



Obr. č. 9, Menu v Bascomu.

V okně programmeru můžeme vidět obsah Flash i EEPROM paměti, typ mikrokontroléru, který je právě připojen, a při přenášení dat z nebo do čipu průběh operace.



Obr. č. 10, Okno AVR programmeru.

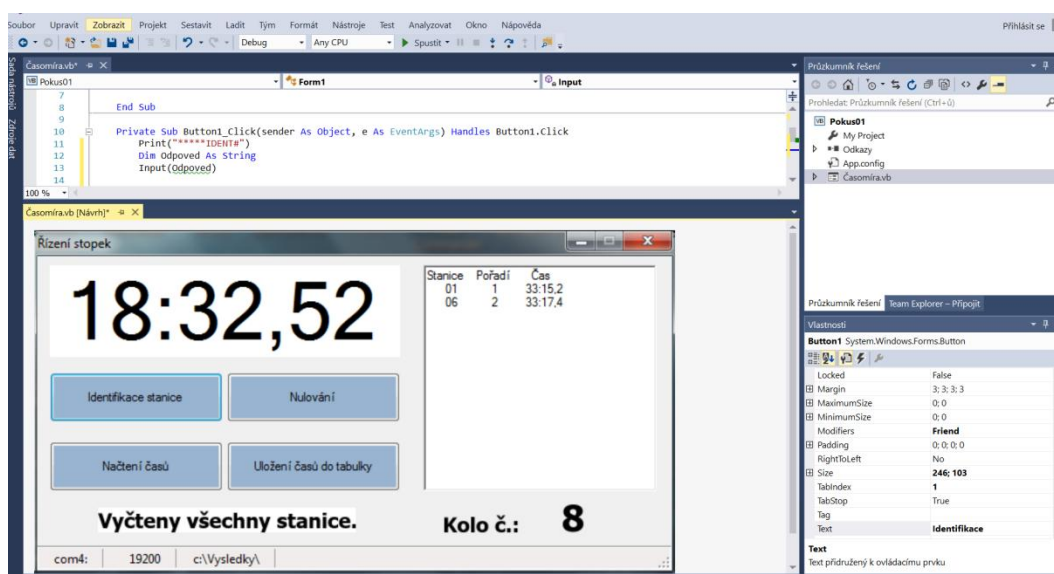
Roletové menu **Tools** v hlavním okně Bascomu slouží ke spouštění pomocných nástrojů. Obsahuje **Terminal emulátor**, který otevře terminálové okno pro zachycování a odesílání sériové komunikace na portu RS 232. Dále **LCD designer**, kterým můžeme tvořit uživatelské znaky na displeji. **LIB Manager** umožňuje do Bascomu přidávat uživatelské knihovny a také podprogramy napsané v Asembleru, které pak můžeme v kódu velmi jednoduše volat jako procedury. V menu **Tools** jsou dále nástroje pro export zdrojového kódu do RTF, PDF a HTML formátu, dávkové spouštění kompilace, konvertor grafických symbolů do jiného rozlišení a příkazy na správu zásuvných modulů.

Programovací jazyk Bascom je poměrně nový, vznikl v roce 1995. Je to vlastně obdoba jazyka Basic používaného pro tvorbu software pro PC. Inspirace tímto jazykem je na první pohled patrná v podobné syntaxi. Díky tomu umožňuje Bascom napsat program ve vyšším jazyce i začínajícím a méně zkušeným programátorům.

### 3.3 Microsoft Visual Studio

Při návrhu systému jsem předpokládal, že připojený notebook (PC) bude mít jako operační systém nějakou z posledních a podporovaných verzí Microsoft Windows, proto jsem pro tvorbu programu pro něj použil Microsoft Visual studio community 2015. Jde o vývojové prostředí (IDE), které se dá použít pro vývoj konzolových aplikací i aplikací s grafickým rozhraním, Windows Forms, webových aplikací atd. Pro tvorbu software si můžeme vybrat z více programovacích jazyků, jako je Visual Basic, C++, C#, F#, Python, JavaScript a SQL. Vytvořené programy lze provozovat na platformách MS Windows, Windows Mobile, Windows CE, .NET, .NET Compact Framework a Microsoft Silverlight.

Visual Studio obsahuje editor kódu s mnoha pokročilými funkcemi od zvýraznění syntaxe přes kontextovou nápovědu i navigaci, pomocníka pro jednoduchý refaktoring až po správce ideálního rozložení oken na pracovní ploše s více monitory. Součástí editoru je také modul pro analýzu kódu a rychlou opravu chyb. Při práci v editoru vidíme okno návrhu formulářů a dalších grafických prvků a zároveň v dalším okně vidíme tvořící se kód, do kterého můžeme vpisovat přímo.



Obr. č. 11, Hlavní okno Visual Studia při návrhu.

Další částí Visual Studia je ladící program, kterým můžeme snadno vyhledat a odladit chyby v napsaném kódu a dokonce i v jeho historii. Ladění máme k dispozici bez ohledu na použitý programovací jazyk a funguje i napříč mezi nimi v režimu interoperability. Kód můžeme ladit bez ohledu na umístění. Může běžet v lokální aplikaci, na ploše, v emulátoru Androidu, ve webovém prohlížeči atd. Při ladění můžeme procesy

pozastavovat, krokovat s přeskočením nebo s vnořením funkcí, můžeme nastavovat zarážky a využívat mnoho dalších pokročilých funkcí.

Chyby v kódu a neočekávané situace se při normálním běhu programu projevují jako výjimky. Výjimky způsobí, že se aplikace zhroutlí nebo se z nich jednoduše stanou chyby, které je těžké vystopovat. Při ladění se sadou Visual Studio můžeme nakonfigurovat oznamování výjimek, ke kterým dojde, a dokonce si zvolit konkrétní výjimky, které nás zajímají. Dostaneme upozornění a z něj pak můžeme kód ladit stejně snadno, jako kdybychom se dostali na zarážku.

Pro ladění více vláknového kódu obsahuje Visual Studio grafické zobrazení, ve kterém jsou přehledně umístěny informace o všech vláknech, a vyhodnocuje výrazy napříč více vlákny, abychom mohli porovnávat hodnoty. Snadno si dokážeme zobrazit informace o vláknech a úkolech, označovat je příznakem a zmrazit je. Můžeme přepínat mezi kontexty provádění a díky značkám vláken vidíme, které řádky kódu v nich se v daném okamžiku provádějí.

Po odladění kódu máme k dispozici nástroje pro jeho testování. Bohatá je sada integrovaných šablon projektů a testovacích architektur podporujících více platforem. Můžeme testovat vyvíjené uživatelské rozhraní i zatížení a výkon webu. Plně integrovaná analýza rozsahu testování kódu pomůže s vizualizací a vytvářením sestav o rozsahu testování kódu na úrovni řádků a bloků, a to pro spravované i nativní aplikace.

Visual Studio kromě již popsaných vlastností obsahuje i modul pro týmovou práci na projektu a funkce pro rozšíření uživatelského rozhraní o doplňky a vylepšení, které pomohou produktivitu vývoje vlastních aplikací. [6]

## 4 Konstrukce

Základem celého systému časomíry jsou jednočipové mikrokontroléry AVR od firmy Atmel. Mikrokontrolér je spojením mikroprocesoru s vnitřní pamětí a obvody rozhraní, což je pro praktické použití výhodnější řešení, než samotný mikroprocesor. Architekturu AVR vymysleli dva studenti Norského technického institutu a první mikropočítač AVR byl vyvinut v centru ASIC v Trondheimu, kde tito studenti pracovali. Jedná se o nízko příkonové osmibitové mikrokontroléry, založené na AVR RISC architektuře. Ta je kombinací dvou architektur – Harvardské architektury, díky níž je možné mít oddělenou paměť pro program (typu Flash) a paměť pro data (typu SRAM), a taktéž architektury RISC,

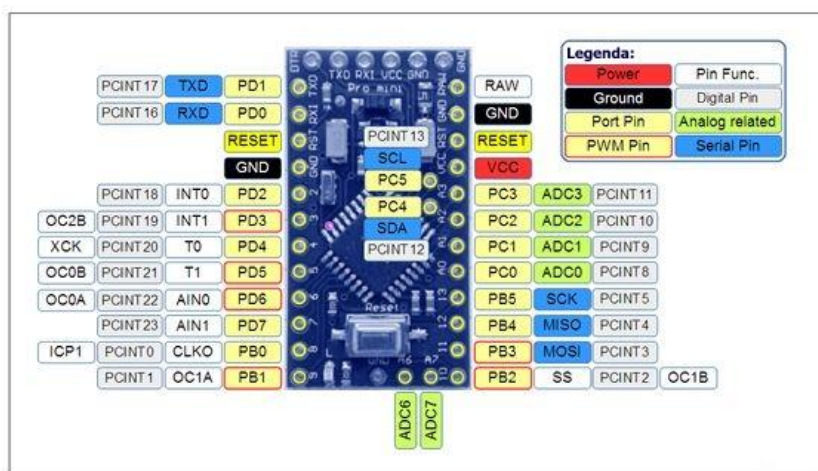
z níž si konstruktéři mikrořadičů AVR vzali především ideu jednoduchých strojových instrukcí prováděných v jednom strojovém cyklu (což mj. zjednodušuje i obvodové řešení reakce na vznik přerušení) a taktéž vysoký počet pracovních registrů (16 či 32), s nimiž mohou být instrukce prováděny. [7]

Díky těmto vlastnostem se mikrořadiče AVR staly krátce po svém uvedení na trh velmi oblíbenými, o čemž mj. svědčí i jejich časté používání v různých kitech, jako jsou například Arduino, Atmel Xplained, Genuino, německé Funduino atd.

## 4.1 Atmel ATmega328

Uvnitř každé stanice je mikrokontrolér ATmega328. Obsahuje programovou FLASH paměť o velikosti 32 kB a 2 kB paměť SRAM pro data. Pro komunikaci má tři osmibitové porty, dva digitální (brány PBx a PDx) a jeden A/D (brána PCx). Komunikace se základnou probíhá přes programovatelný sériový USART (Universal Synchronous / Asynchronous Receiver and Transmitter). Instrukční soubor má 131 výkonných instrukcí, které dokáže zpracovávat rychlostí až 20 MIPS (při taktovacím kmitočtu 20 MHz). Kromě dvou vnitřních nezávislých osmibitových časovačů má integrovaný také jeden 16bitový časovač, který jsem použil pro měření času ve stopkách. [8]

Pro tuto konkrétní aplikaci byl vybrán modul ATMEGA328p 5V 16MHz Pro Mini Module Board. Obsahuje všechny potřebné obvody pro činnost mikrokontroléru včetně krystalu určujícího kmitočet a resetovacího tlačítka. Pomocí jednořadých konektorových lišt je deska modulu osazena na desku stanice.

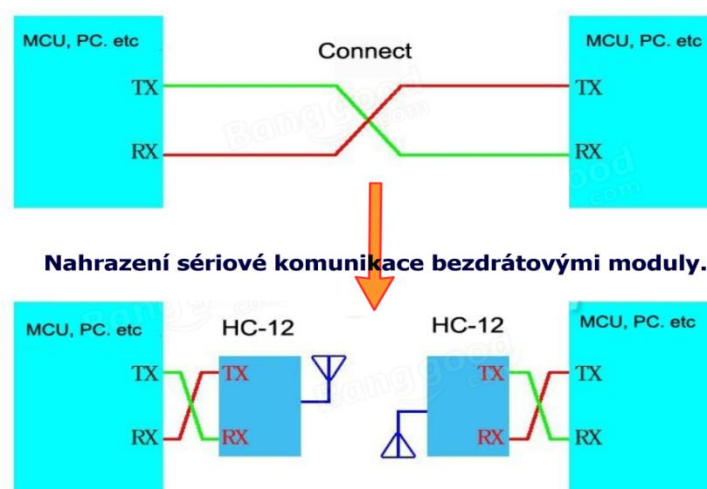


Obr. č. 12, Modul 328P Pro Mini s popisem pinů, převzato a upraveno z [9].

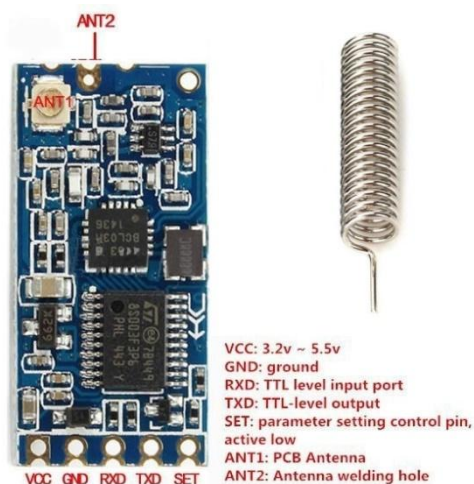


## 4.2 Bezdrátový sériový port HC-12

Komunikaci mezi stanicemi a základnou zajišťuje sériový modul HC-12 SI4463. Bezdrátová část pracuje na kmitočtovém pásmu 433 MHz určeném k provozování zařízení krátkého dosahu podle všeobecného oprávnění. Komunikační vzdálenost do 300 m, kterou požadujeme na volném prostranství, nám HC-12 i s rezervou zajistí při přenosové rychlosti 5000 bps.



Obr. č. 13, Nahrazení sériové komunikace, převzato a upraveno z [10].



Obr. č. 14, Deska modulu HC-12 s anténou, převzato a upraveno z [10].

Modul HC-12 lze nastavovat pomocí AT příkazů. Můžeme jimi nastavit přenosovou rychlost sériového portu (1 200 bps – 115 200 bps), výstupní výkon bezdrátového vysílače (v osmi krocích), vysílací kanál, délku datového slova včetně stop bitu a režim provozu,

kterým lze ovlivnit spotřebu bezdrátového modulu. Pro přepnutí HC-12 do nastavovacího módu se na pin SET připojí low level (0 V). Potom je možno na vstup Rx poslat příslušné AT příkazy. Po připojení pinu SET na high level (+5 V) pracuje HC-12 opět jako sériový port. Při provozu stanice používám AT příkazy především ke změně komunikačního kanálu. [11]

Tab. č. 1, Seznam AT příkazů HC-12. [11]

	Příkaz	Odpověď	Význam	Poznámka
1.	AT	OK	Testovací instrukce	
2.	AT+Byyyy	OK+Byyyy	Příkaz pro změnu přenosové rychlosti sériového portu.	1200; 2400; 4800; 9600; 19,200; 38,400; 57,600; 115,200 (bps)
3.	AT+Cyyy	OK+Cyyy	Změna kanálu bezdrátové komunikace.	Čísla kálů 001 až 127
4.	AT+FUy	OK+FUy	Změna přenosového módu sériového portu.	Přenosové módy: FU1, FU2, FU3 a FU4 (více manuál)
5.	AT+Py	OK+Py	Změna vysílacího výkonu VF modulu.	Hodnoty y v rozmezí 1 až 8 (-1 dBm až 20 dBm po 3 dBm)
6.	AT+Ry	OK+Ryy....	Vrátí nastavení některého z parametrů HC-12.	Za y dosadit některé z písmen: B (rychlost v baudech), C (kanál), F (přenosový mód), P (vysílací výkon)
7.	AT+RX		Vrátí nastavené hodnoty přenosového módu, rychlosti, kanálu a výkonu.	Např.: OK+FU3 OK+B9600 OK+C001 OK+RP:+20dBm
8.	AT+Udps	OK+Udps	Nastavení délky datového slova ( <i>d</i> ), parity ( <i>p</i> ) a stop bitu ( <i>s</i> ).	Např.: "AT+U801"
9.	AT+V		Vrátí verzi firmware.	Např.: "HC-12_V2.3"
10.	AT+SLEEP	OK+SLEEP	Převede HC-12 do režimu spánku s odběrem pouhých 22μA.	V tomto režimu není povolena komunikace. Spánek ukončí další AT příkaz.
11.	AT+DEFAULT	OK+DEFAULT	Nastavení HC-12 do továrních hodnot.	9600bps, 8 data bits, no parity, 1 stop bit, kanál 001, vysílací výkon 20dBm, přenosový mód sériového portu FU3
12.	AT+UPDATE		Nastavení modulu do stavu čekání na update firmware.	

Při prvních pokusech s HC-12 jsem narazil na limity, které jsou daní za nízkou cenu. Modul je velmi jednoduchý sériový port, a proto jsem musel některé nedostatky ošetřit v programu, kterým se modul obsluhuje. Jedná se především o inicializaci modulu po zapnutí. Na začátku programu jak v základně, tak ve všech stanicích se spustí procedura inicializace, která přepne HC-12 do nastavovacího módu, pošle mu testovací instrukci „AT“

a čeká na odpověď, po jejím obdržení přepne modul zpět do normálního módu. Pokud odpověď není „OK“, celý proces se zopakuje. Většinou stačí jedno posláni testovací instrukce a modul pracuje správně. Bez této inicializace se mi stávalo, že jeden nebo i několik prvních poslaných Bytů nebylo odesláno, nebo byly „zkomoleny“ a druhá strana je nedokázala přečíst. Jiná trochu nepříjemná vlastnost je, že při přenosu se na přijímané straně „objeví“ vždy jeden až dva znaky navíc oproti znakům poslaným. To jsem ošetřil celkem snadno - oříznutím přijatého řetězce zprava na správný počet znaků.

Další z vlastností je neschopnost obousměrné komunikování více modulů na jednom kanále. V odstavci o programování jsem popsal způsob řešení tohoto problému.

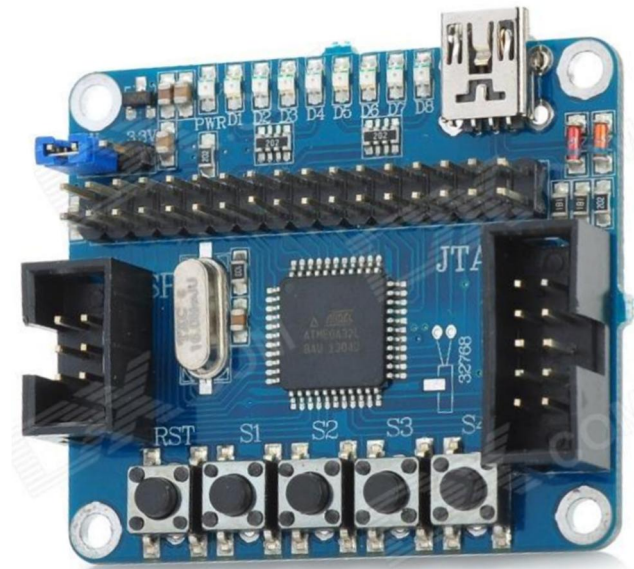
Pro testování přenosových vlastností sériového modulu jsem pro základnu a jednu stanici napsal program, který na jedné straně vysílal řetězce znaků, na druhé straně probíhalo vyhodnocování, jestli příjem proběhl v pořádku. Během tohoto testování jsem zvětšoval vzdálenost mezi základnou a stanicí na více než požadovaných 300 m ve volném prostoru a zkoušel přenos i uvnitř budov. Při zkouškách, které probíhaly několik dní, nedocházelo k žádným ztrátám dat a přenos probíhal spolehlivě. Proto jsem i přes zmíněné nedostatky zvolil modul HC-12 jako součást systému časomíry.

### 4.3 Atmel ATmega32L

Hlavní součástí základny je mikrokontrolér ATMEGA32L-8AU pracující na taktovací frekvenci 16 MHz. Stejně jako ATmega328 obsahuje programovou flash paměť o velikosti 32 kB a 2 kB paměť SRAM pro data. I další důležité vlastnosti jsou shodné. Jediným důvodem, pro který jsem zvolil pro základnu tuto desku, je větší počet I/O portů kvůli možnosti připojení více periférií.

Deska je osazena rozhraním ISP, JTAG a 34 pinovým dvouřadým konektorem, na kterém jsou vyvedeny 4 osmibitové porty (PA, PB, PC, PD) a napájení +5 V. Port PA je určený pro připojení LCD, port PC pro externí funkční klávesy, k portu PD je připojen bezdrátový sériový komunikátor HC-12, startovací tlačítko a modul sériového portu, komunikujícího s notebookem. Port PB je zatím v rezervě pro velkoformátový LED displej, který by měl být použitý při soutěžích mladých požárníků, ale tento vývoj ještě není dokončen.





Obr. č. 15, Modul s ATmega 32, převzato a upraveno z [10].

Při porovnání blokových schémat mikrokontrolérů, která jsou uvedena v přílohách, je vidět, že oproti mikrokontroléru ATmega 328, který jsem použil ve stanicích, má ATmega 32 umístěná v základně širší výbavu.

#### 4.4 LCD 16x2

Pro zobrazení hodnot a povelů na stanicích i základně jsem použil LCD displej 1602A s modrým podsvícením a bílými znaky. Zobrazuje až 16 znaků ve dvou řádkách, každý znak se zobrazuje na matici 5x8 bodů. Je vybaven velmi jednoduchým řadičem HD44780 s paralelním rozhraním. Pro jeho obsluhu je tedy třeba 8žilový propojovací kabel pro spojení s deskou mikrokontroléru – 6 vodičů pro datovou komunikaci a 2 pro napájení.



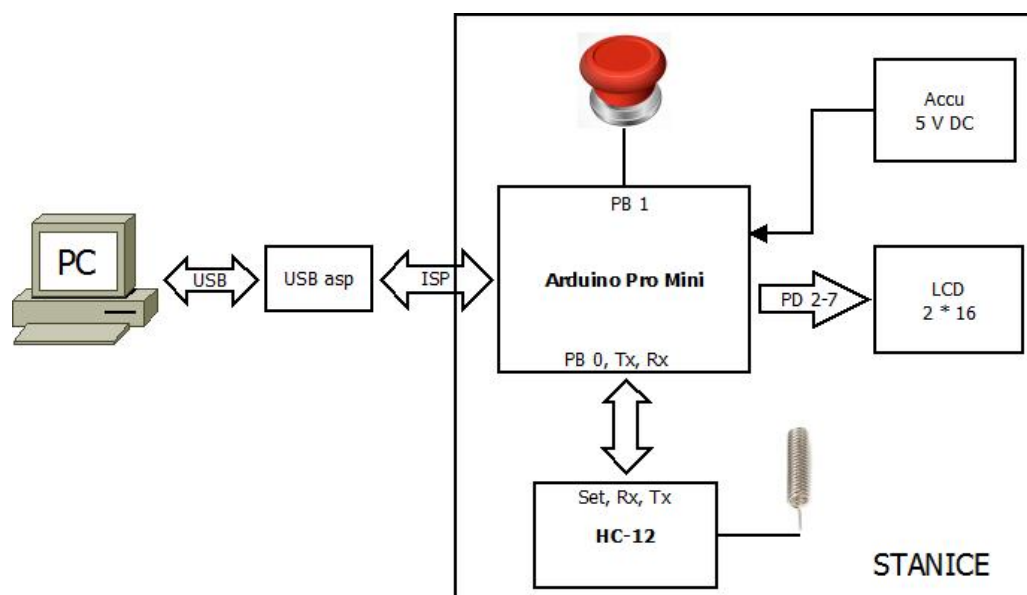
Obr. č. 16, LCD 1602, převzato a upraveno z [10].

## 4.5 Stanice (stopky)

Všechny části stanice jsou zabudovány do krabice s označením „TK PS 1809-6-to“. Jde o montážní krabici pro elektrotechniku vyrobenou z polyesteru s průhledným víkem z polykarbonátu. Tyto materiály by měly zaručit dostatečnou odolnost při méně šetrném zacházení v polních podmínkách a stupeň krytí IP 66, který udává výrobce, s rezervou dostačuje pro ochranu veškeré elektroniky proti prachu a vlhkosti. Stupně ochrany krytem jsou včetně popisu uvedeny v příloze.



Obr. č. 17, Krabice s hřibovým tlačítkem, převzato a upraveno z [12].



Obr. č. 18, Blokové schéma stanice.

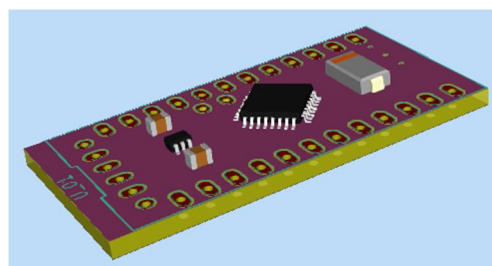


Obr. č. 19, Foto sestavené stanice.

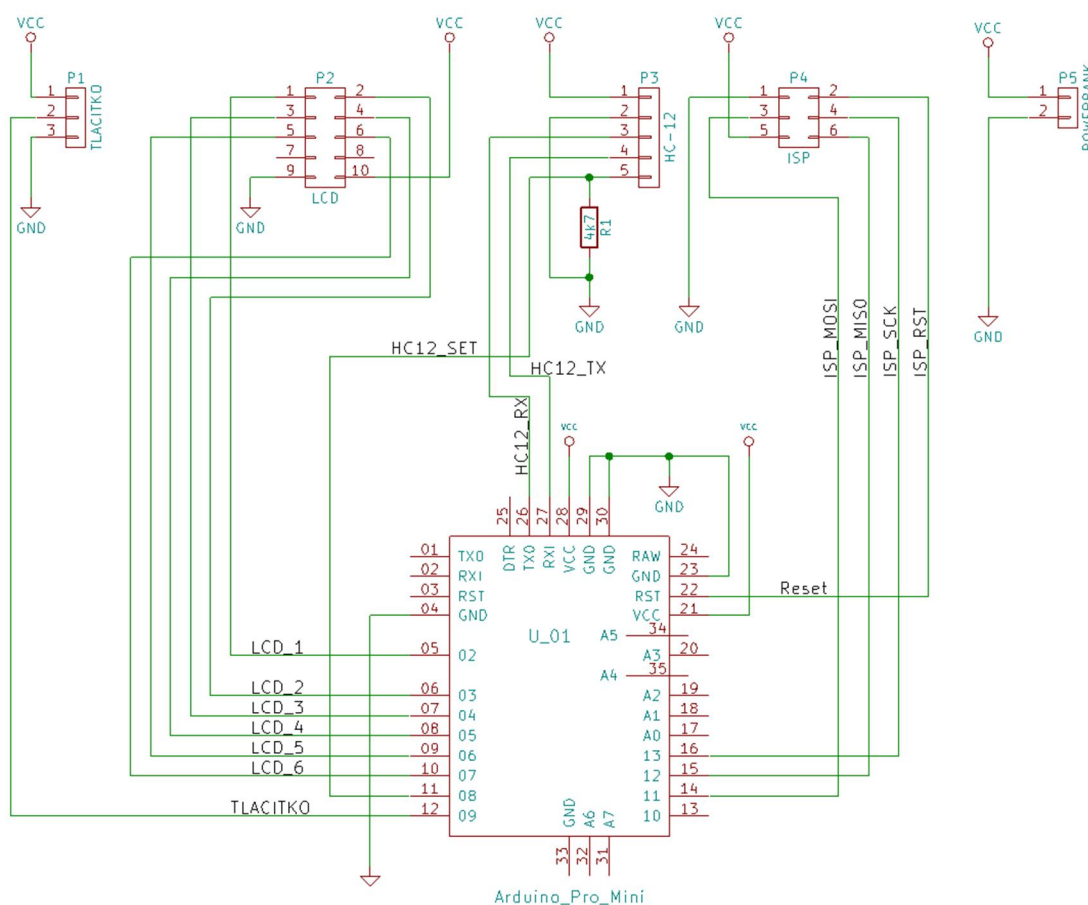
Z blokového schématu (Obr. č. 18) vyplývá, že komunikace mikrokontroléru probíhá buď uvnitř stanice, nebo bezdrátově, tudíž se stanice nemusí během provozu vůbec otvírat. Jedinou výjimkou je přenos programu pomocí USB-ISP programátoru, kdy se krabice musí otevřít. Nepředpokládám, že k tomu by mělo docházet často, takže krabice zůstane obvykle zavřená.

#### 4.5.1 Deska plošných spojů s mikrokontrolérem AT Mega 328

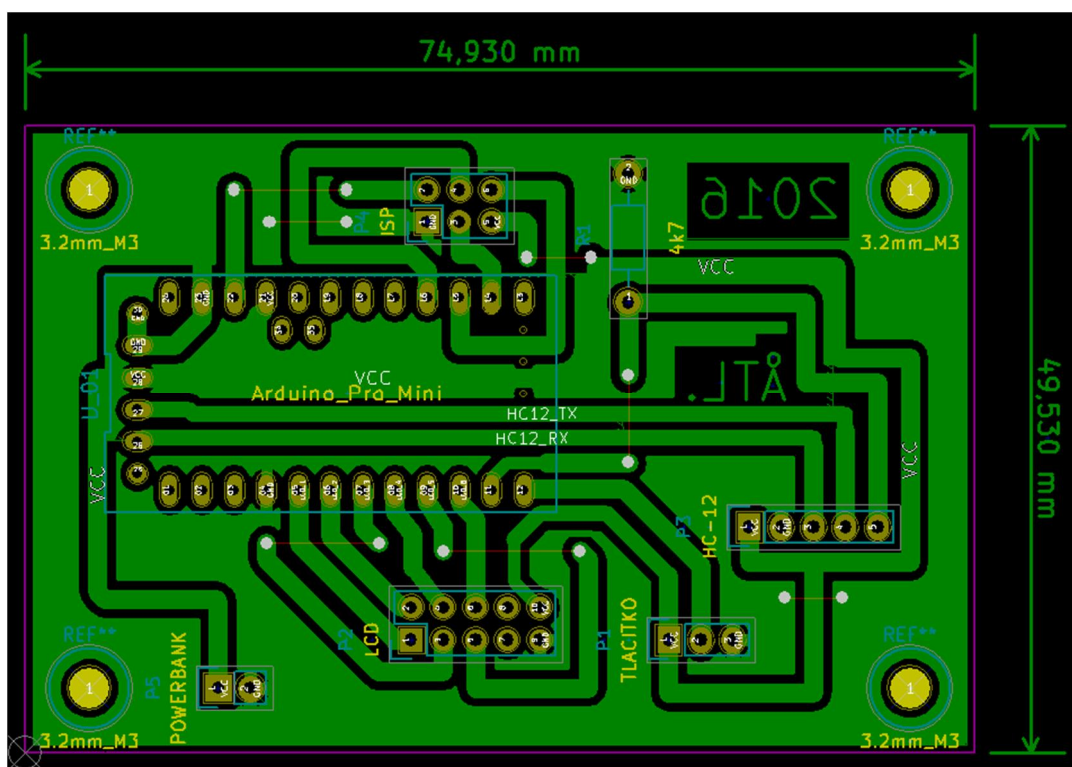
Tato deska je umístěna ve střední části krabice blízko dna těsně nad powerbankou. Kromě AT Megy je osazena třemi konektory pro připojení desky LCD, programátoru ISP, červeného hříbového tlačítka, a je spojena kablíkem s powerbankou, která všechny obvody stopek napájí stabilizovaným napětím 5 V. Dále se zde nachází obvod bezdrátového sériového portu HC-12. Modul mikrokontroléru a HC-12 jsou k DPS připojeny a zároveň připevněny jednořadými konektorovými lištami kvůli snadné výměně při případné poruše. Pro propojení s deskou LCD je použit plochý kablík stejného typu, jaký se dříve používal na připojení IDE HDD v počítačích. Na koncích je osazen dvouřadými samozářezávacími 10pínovými konektory. Na Obr. č. 22 je deska plošných spojů pro stanici. Při jejím návrhu jsem musel vytvořit zcela nový model pouzdra pro modul mikrokontroléru AT Mega 328 nazvaný „Arduino Pro Mini“.



Obr. č. 20, 3D model pouzdra pro Arduino.



Obr. č. 21, Schéma desky s AT mega 328.



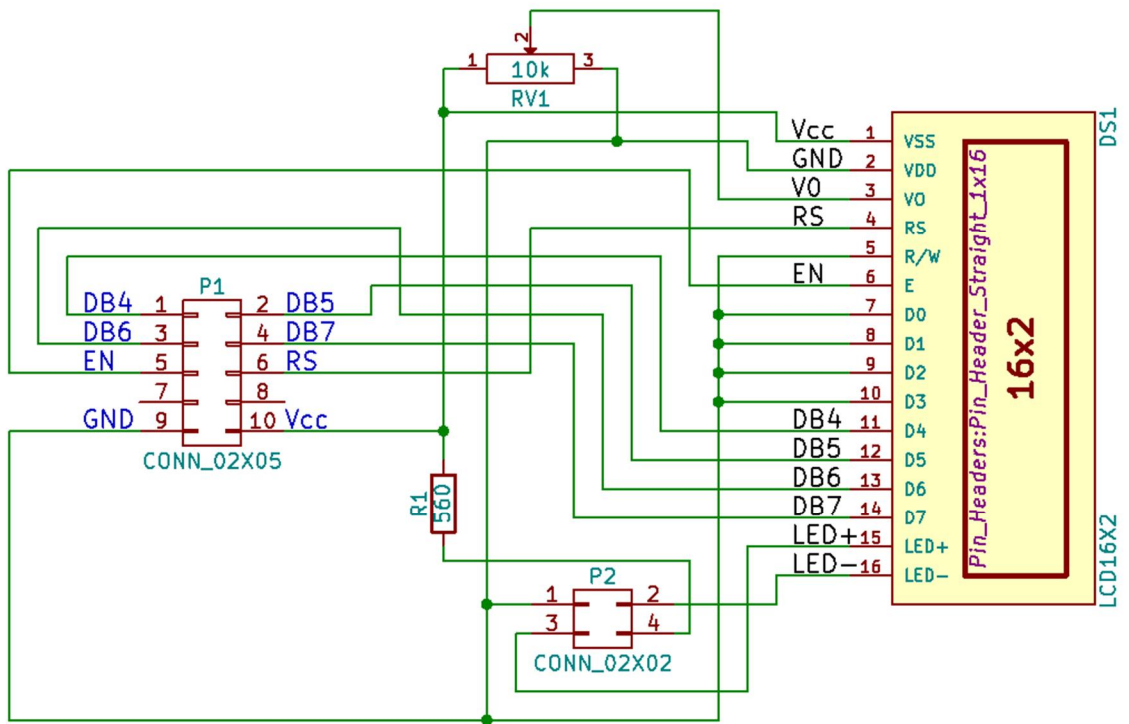
Obr. č. 22, Návrh DPS stanice ze strany součástek.

Jak se zmiňuji v kapitole 3.1, nakreslit vlastní pouzdro pomocí Footprint editoru je opravdu snadné. Kromě pouzdra jsem vytvořil i velmi jednoduchý 3D model uvedený na Obr. č. 20.

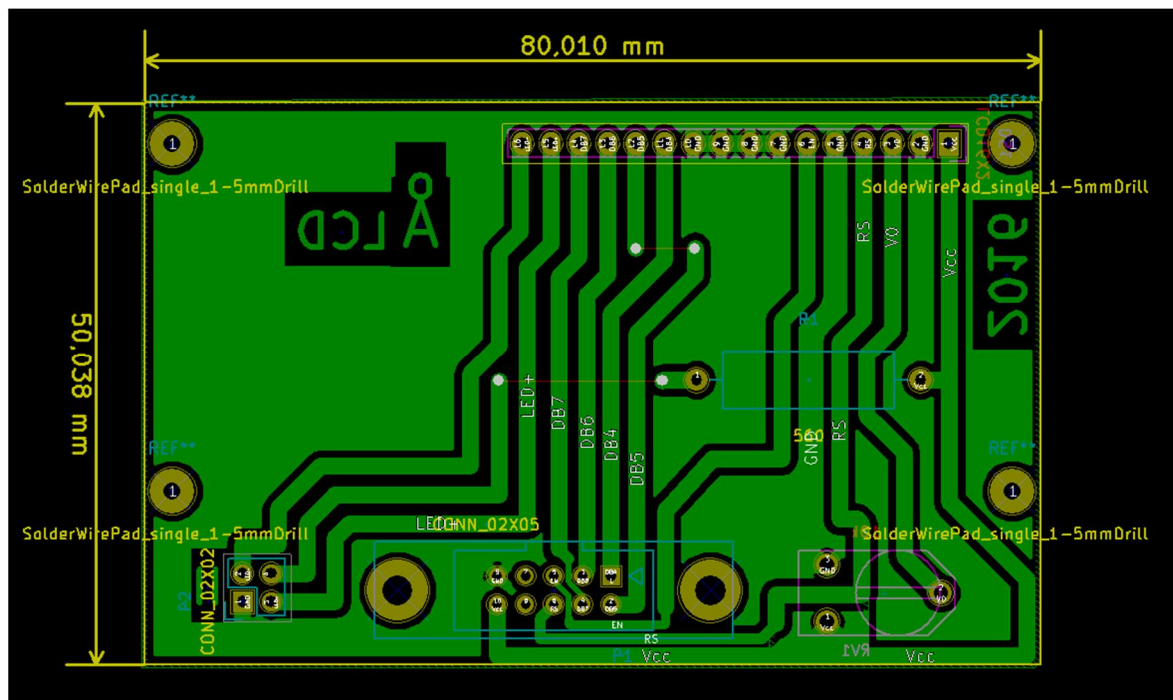
#### 4.5.2 Deska plošných spojů s LCD

Na desce je připojen displej 16 x 2 (viz výše), dále je na DPS odporový trimr, kterým se nastavuje jas displeje, a konektor pro propojení s deskou mikrokontroléru zajišťující zároveň napájení LCD. Pomocí čtveřice pinů označených ve schématu jako součástka **P2** lze změnit polaritu napájení podsvícení LCD. Některé moduly displejů, přestože vypadají stejně, mají prohozené vývody 15 a 16. Abych nemusel měnit návrh DPS a znovu ji vyrábět, zapracoval jsem do schématu tento jednoduchý „přepínač“. Prostým otočením propojek se napájení podsvícení přepóluje a stejnou DPS mohou použít pro oba typy výrobního provedení LCD. Deska je v krabici připevněna pomocí distančních tyček, které zaručují její správné umístění těsně pod průhledné víko, aby byl údaj na displeji dobře čitelný. Za normálních světelných podmínek není třeba kvůli krytu nastavovat nejvyšší jas displeje, naopak při vysokých hodnotách jasu se zobrazený údaj začíná ztrácet.

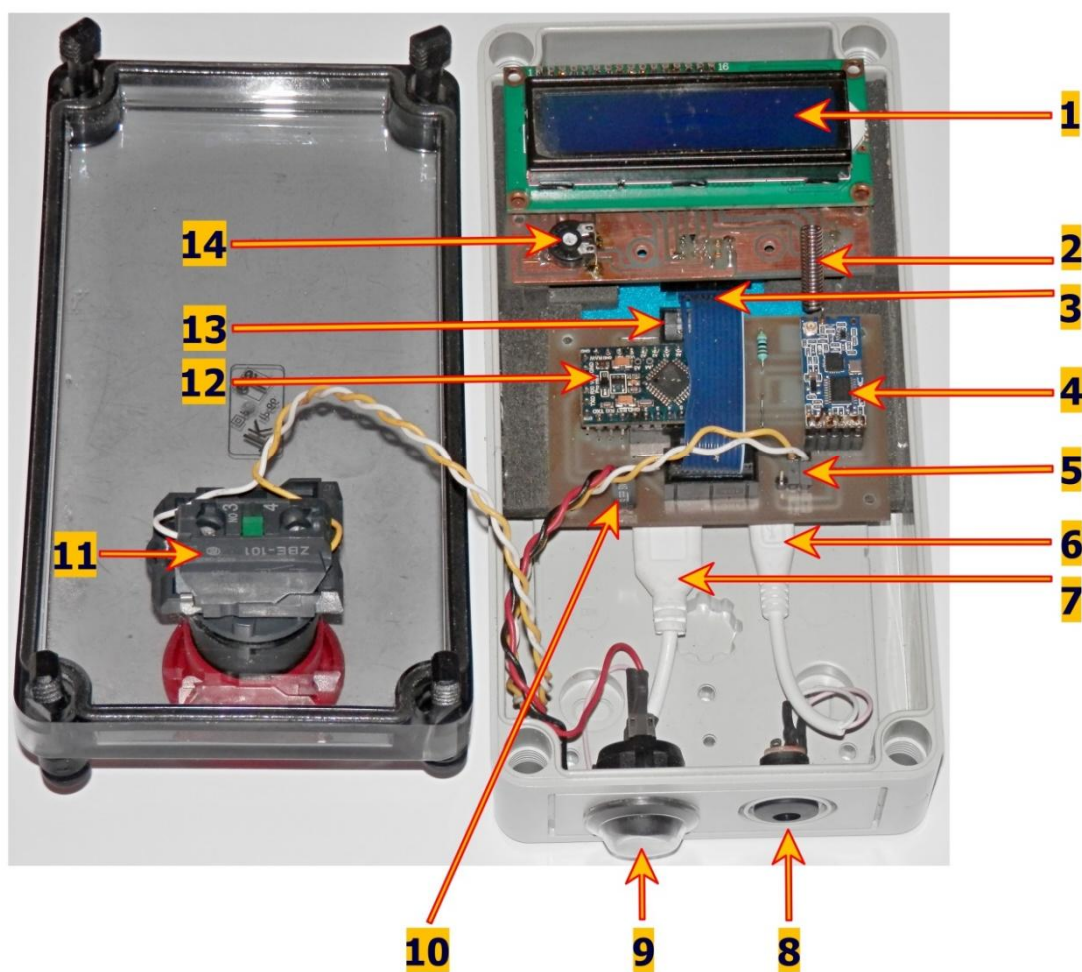




Obr. č. 23, Schéma desky LCD.



Obr. č. 24, Návrh DPS s LCD ze strany součástek.



Obr. č. 25, Foto otevřené stanice s popisem.

- 1 – LCD
- 2 – Anténa modulu HC-12.
- 3 – Propojovací kabel mezi deskou LCD a deskou mikrokontroléru.
- 4 – Bezdrátový modul HC-12.
- 5 – Konektor pro připojení stop tlačítka.
- 6 – Kabel nabíjení powerbanky.
- 7 – Napájecí kabel stanice z powerbanky.
- 8 – Nabíjecí konektor.
- 9 – Vypínač.
- 10 – Konektor pro připojení napájení na desku mikrokontroléru.
- 11 – Stop tlačítko.
- 12 – Modul s mikrokontrolérem ATmega328.
- 13 – Programovací konektor ISP (jen částečně viditelný).
- 14 – Trimr pro regulaci jasu LCD.

Pro napájení stanic jsem zvolil powerbanku „Ultrathin 5600mAh“ a to hlavně pro její nízký profil, který se snadno zabuduje do instalační krabice. Kapacita Li-Ion akumulátoru plně dostačuje. Během mých pokusů powerbanka udržela zařízení nepřetržitě v provozu tři dny. Celé stopky jsou uzavřeny ve voděodolné krabici a díky nabíjecímu konektoru, který je umístěn na vnější stěně krabice, se nemusí za normálního provozu vůbec otevírat.

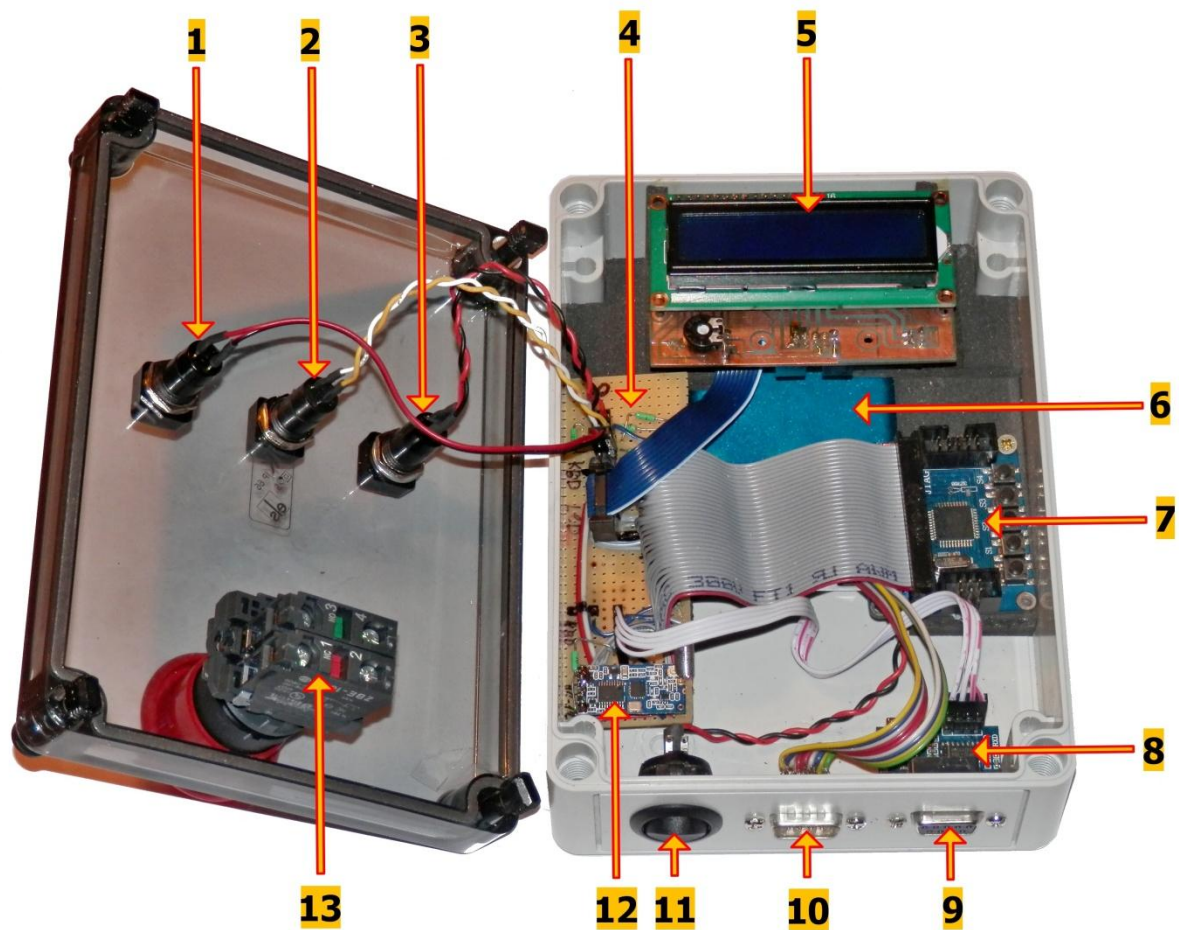


Obr. č. 26, Powerbanka, převzato a upraveno z [10].

## 4.6 Základna

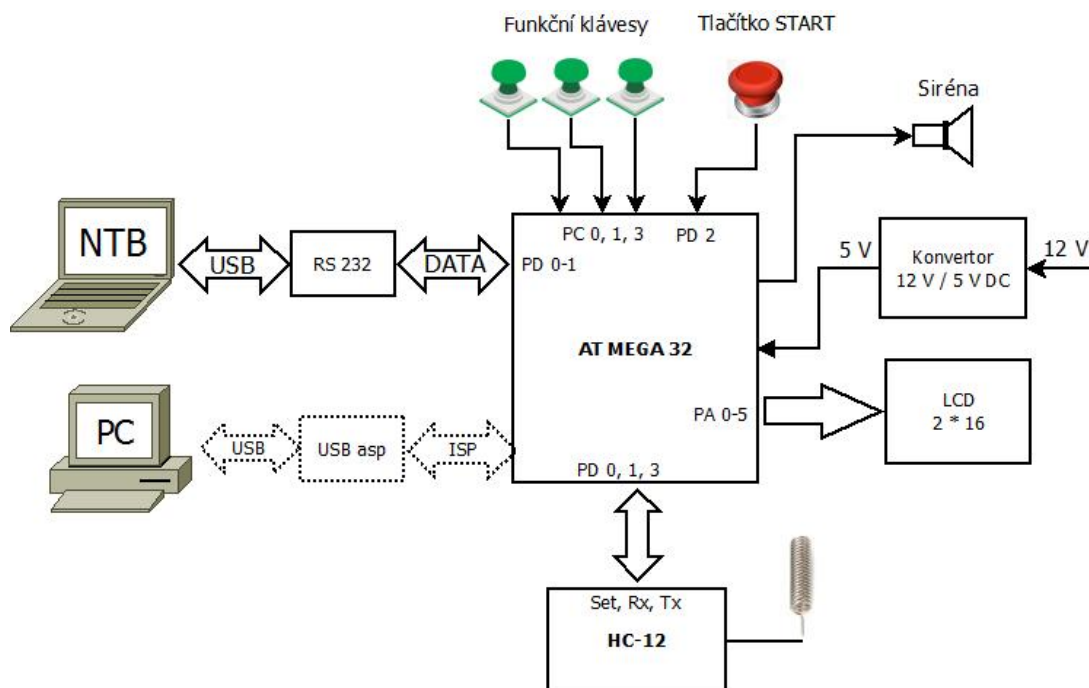
Celá základna je zabudována do montážní krabice „TG PC 2516-12-to“ s průsvitným víkem. Uvnitř krabice jsou umístěny všechny komponenty základny, kromě funkčních tlačítek, startovacího tlačítka a sirény. Siréna je instalována v samostatné krabici, je připojena dlouhým kabelem, aby ji obsluha mohla umístit na nejvhodnější místo. Kvůli použité siréně, která vyžaduje napájení 12 Vss, se základna napájí z autobaterie a uvnitř krabice je konvertor 12V/5V. Ten slouží pro nabíjení powerbanky i pro napájení obvodů mikrokontroléru a LCD. Autobaterie není součástí systému, počítá se s tím, že na závodech bude minimálně jeden automobil, který napájení poskytne. Proto je u základny kabel, který má na jednom konci zástrčku do automobilové zásuvky (zapalovače). Tlačítka jsou na vrchní straně krabice. Nejsou sice vodotěsná, ale základna bude vždy při nepřízni počasí schovaná pod nějaký přístřešek spolu s rozhodčími. Naopak siréna je v krabici, která umožňuje venkovní montáž, proto jí špatné počasí nevádí. Bude se umisťovat poblíž stanovišť závodníků.





Obr. č. 27, Foto otevřené základny s popisem.

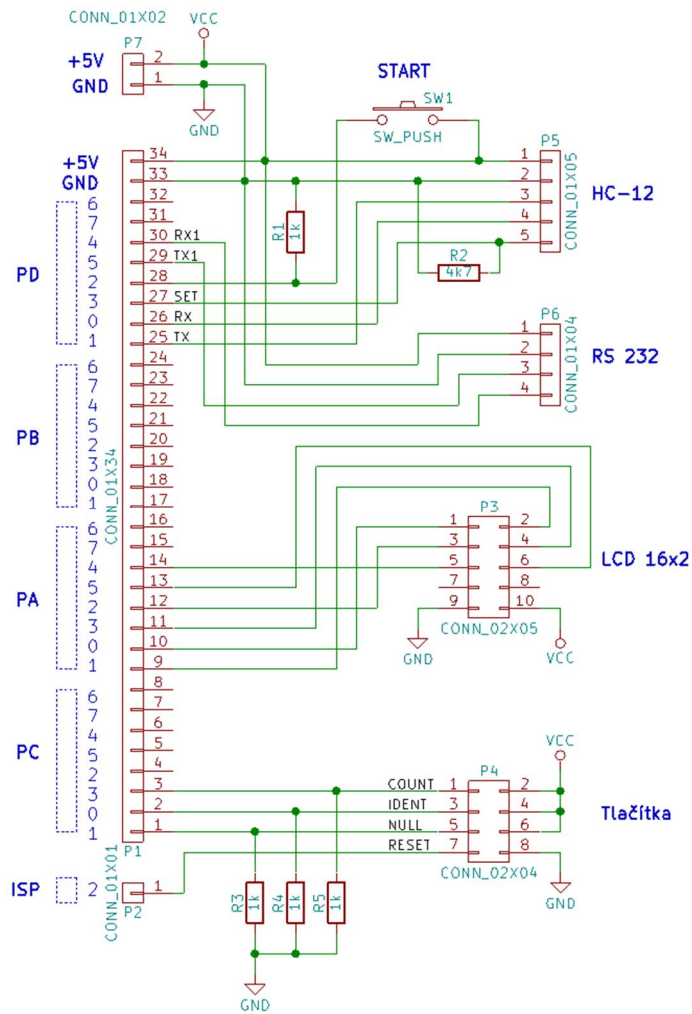
- 1 – Funkční tlačítko vyčítání výsledků.
- 2 – Funkční tlačítko nulování.
- 3 – Funkční tlačítko identifikace stanic.
- 4 – Propojovací deska.
- 5 – LCD
- 6 – Powerbanka.
- 7 – Modul s ATmega 32.
- 8 – Modul RS 232.
- 9 – Sériový port.
- 10 – Konektor napájení a ovládání sirény.
- 11 – Vypínač.
- 12 – Modul bezdrátového sériového portu HC-12
- 13 – Startovací tlačítko.



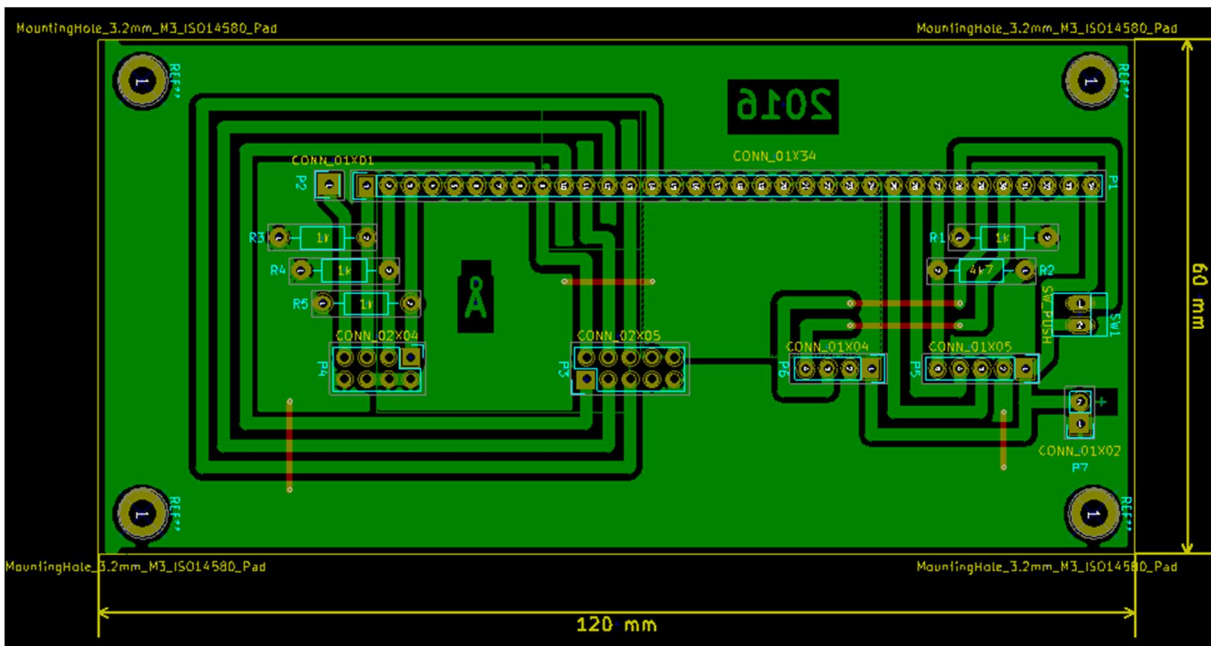
Obr. č. 28, Blokové schéma základny.

#### 4.6.1 Propojovací deska pro ATmega32

Kvůli spojení modulu ATmega32 s ostatními částmi základny jsem navrhnul propojovací desku. Modul ATmegy je s touto deskou propojen 34 žilovým plochým kabelem, stejným jaký se dříve používal pro připojení disketové mechaniky k PC včetně původního konektoru. Dále jsou na desce konektory pro tlačítka, LCD, HC-12 a napájení ATmegy32 i ostatních modulů. Na desce jsou osazeny i všechny odpory, které k propojení periférií patří. Celý systém je navržen tak, aby byly jednotlivé komponenty snadno vyměnitelné v případě poruchy nebo upgrade zařízení. Startovací tlačítko jsem použil se dvěma galvanicky oddělenými kontakty, kde druhý kontakt (není na následujícím schématu) spíná sirénu. Původně jsem chtěl, aby sirénu spouštěla ATmega32 jedním volným pinem pomocí zesilovacího relátka, ale zvolená konstrukce je mnohem jednodušší a pro požadované použití zcela dostačující. Funkční tlačítka posílají některé z povelů (NULL, IDENT, COUNT – viz kapitola 5.2 o software). Stejně povely lze poslat i z aplikace v notebooku, ale pro zjednodušení ovládání ve venkovních podmínkách v atmosféře soutěže jsou opakující se povely realizovány i speciálními tlačítky.



Obr. č. 29, Schéma propojovací desky.



Obr. č. 30, DPS propojovací desky.

## 5 Software

### 5.1 Program pro stanici – stopky

Tento program má dva hlavní úkoly – „poslouchat“ na kanále pomocí modulu HC-12, čekat na povely od základny a vykonat je. Druhým úkolem je měření času a předání jeho konečné hodnoty opět základně. Skládá se z hlavní části programu, která rozpoznává přijaté povely, podle nichž řídí běh mikrokontroléru, a z několika podprogramů zpracovávajících často opakované procedury.

Jeden z podprogramů zabezpečuje přeladění HC-12 na jiný kanál („soukromý“ kanál stanice) a po vykonání příslušné akce následuje přeladění zpět na kanál č. 001 (základní „poslouchací“ kanál). Toto přeladění je důležité kvůli spolehlivosti komunikace mezi základnou a stanicí. HC-12 s čipem Si4463 je velmi jednoduchý sériový port a pokud by na jednom kanále chtělo komunikovat více modulů HC-12, mohlo by docházet, a většinou i dochází, ke kolizím a zarušení probíhající komunikace. Tento modul totiž nemá žádné pokročilé funkce pro řízení komunikace a řešení kolizí, jako je tomu například v sítích Wi-Fi. Proto jsem určil kanál č. 001, na kterém stanice po zapnutí poslouchají a pokud obdrží od základny příkaz, přeladí se. Další komunikace se základnou už probíhá izolovaně na soukromém kanále. Soukromý kanál konkrétní stanice se určuje pomocí konstanty v programu s názvem *Cislostanice*. Tuto konstantu má každá stanice jedinečnou a pro snazší identifikaci je napsána uvnitř krabice stanice na desce mikrokontroléru. Číslo stanice se vynásobí další konstantou *Odskokkanalu*, která je pro všechny stejná a má v tuto chvíli hodnotu 5. Tím se získá číslo kanálu, na kterém ta která stanice komunikuje. Odladění o 5 kanálů jsem zvolil ze dvou důvodů. Zaprvé odstup soukromých kanálů je takto 2MHz (5 x 400kHz na 1 kanál), to s rezervou stačí na to, aby se předešlo případným přeslechům a mezikanálovému rušení. Zadruhé tímto získám 20 soukromých kanálů, to v současné konfiguraci celého zařízení vyhovuje. Pokud by vznikla potřeba vyššího počtu soukromých kanálů, dá se konstanta *Odskokkanalu* nastavit na hodnotu 4 nebo 3, tím by vzniklo 25, resp. 33 použitelných soukromých kanálů se stále dostatečným kmitočtovým odstupem. Přeladění se provádí pomocí AT příkazů. Na následujícím podprogramu je vidět, jakým způsobem se z proměnných *Cislostanice* a *Odskokkanalu* sestavuje AT příkaz pro přeladění stanice na soukromý kanál.

Během vývoje jsem zjistil, že použití konstanty *Cislostanice* je nepraktické. Musí totiž existovat tolik verzí zkompilevaného programu pro stanice, kolik je stanic. Do příští

verze systému zapracuji DIP přepínač, kterým se bude hardwarově nastavovat číslo stanice, a postačí jedna verze programu pro všechny.

```

Sub Preladeni:

    Preladenizacatek:

    Canal = Cislostanice - 1
    Canal = Canal * Odskokkanalu
    Canalstring = Str(Canal)
    Canalformat = Format(Canalstring, "000")
    Odpoved2 = "AT+C" + Canalformat

    Reset PORTB.0
    Waitms 200
    Print Odpoved2
    Input Odpoved2

    Odpoved = Right(odpoved2, 7)
    Odpoved2 = Left(odpoved, 4)

    If Odpoved2 <> "OK+C" Then
        Hc12err = 1
        Cls
        Lcd "Hc12ERR " ; Operace
        Wait 4
        Cls
        Waitms 500
        Goto Preladenizacatek
    Else
        Set PORTB.0
        Waitms 200
        Hc12err = 0
    End If

End Sub

```

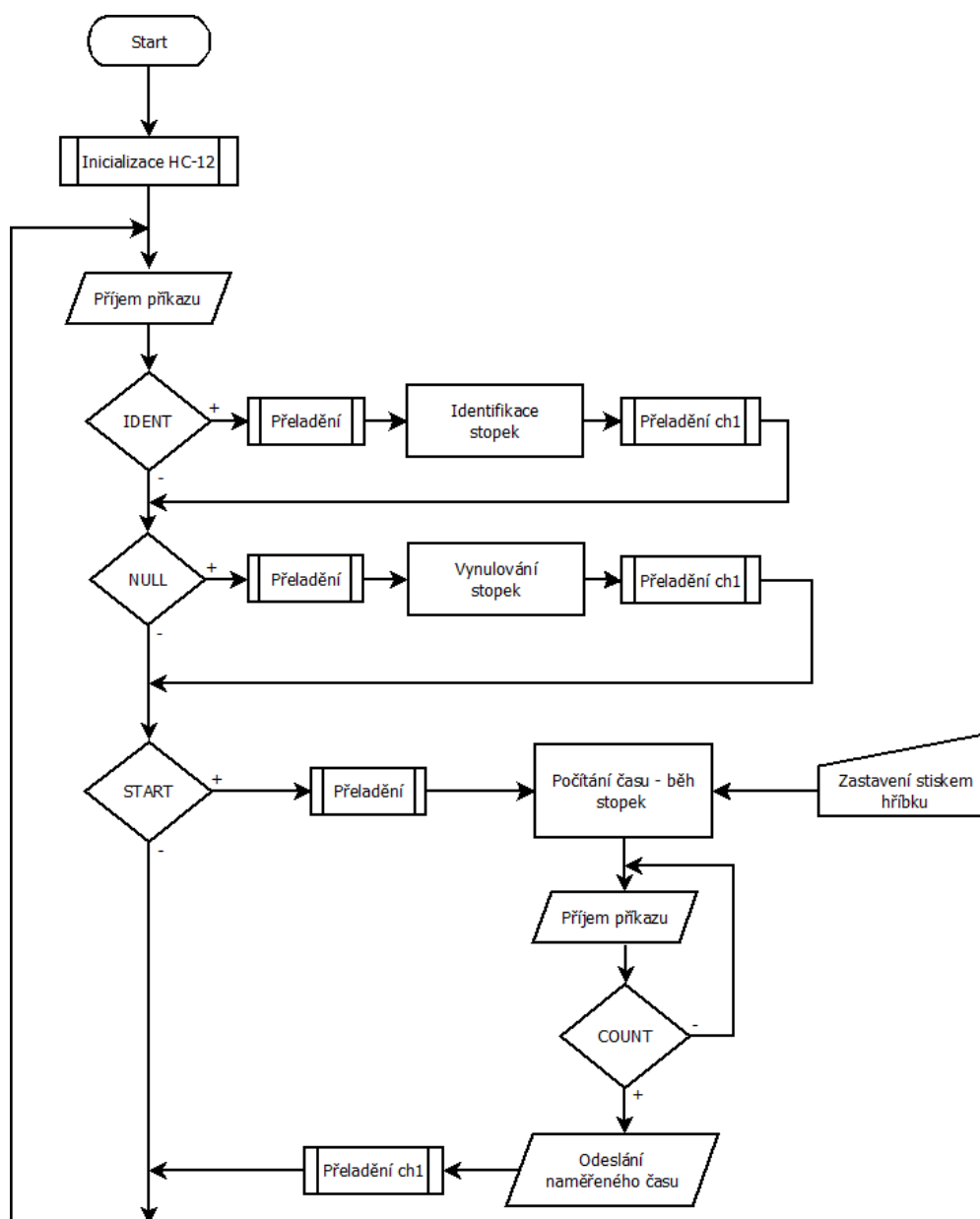
'Výpočet soukromého kanálu  
'Převod na formát string  
'Vytvoření kompletního AT příkazu  
'Nastavení HC-12 do programovacího módu  
'Ověření správné odpovědi HC-12  
'Pokud se přeladění nepovede, zkusí to znovu.  
'Nastavení HC-12 zpět do komunikačního módu

Obr. č. 31, Procedura přeladění na soukromý kanál.

Následující vývojový diagram (Obr. č. 32) schematicky ukazuje celý program uložený v mikrokontroléru ATmega328. Chybí na něm symbol konce. Sám program totiž funkci ukončení běhu nemá, vše skončí stiskem vypínače a odpojením napájení.

Podprogram *Inicializace* je určen pro „nastartování“ a kontrolu funkce HC-12 pomocí AT příkazů. Bez použití této procedury se někdy stalo, že bezdrátový modul ztratil několik prvních povelů a komunikace se základnou neprobíhala úplně od začátku bezchybně. Součástí inicializace je také přeladění na kanál č. 001.

Podprogram *Přeladění* je použitý při identifikaci stanice, při vynulování stopek i při předávání naměřených časů. Při prvních dvou jmenovaných procedurách se kvůli kontrole provádí komunikace se základnou, která si při těchto operacích ukládá jejich výsledky, a v případě, že některá stanice přestane komunikovat, oznámí chybu stanice na svém displeji. Takto má obsluha základny (rozhodčí) kontrolu správné činnosti všech použitých stanic.



Obr. č. 32, Vývojový diagram programu stanice.

Pro počítání času je použit vnitřní časovač, který pracuje na frekvenci krystalu 16 MHz. Při konfiguraci časovače je nastavena dělička kmitočtu krystalu na 256, takže kmitočet pro počítání časovače je 62,5 kHz. Pro přetečení čítače za 100 ms potřebujeme tedy 6250 pulzů. Jelikož je tento čítač 16bitový, potřebuje k přetečení 65 536 pulzů. Musíme ho tedy přednastavit na hodnotu 59 286 ( $65\,536 - 6\,250$ ). K tomu slouží interní registry TCNT1H (Counter Register High Byte) a TCNT1L (Counter Register Low Byte). [5] Protože  $59\,286 = 231 \cdot 256 + 150$ , musíme nastavit TCNT1H=231 a TNCT1L=150. Díky tomu vyvolá čítač přerušení a spustí proceduru, která do proměnné *\_desetina* přičte jedničku, vždy

přesně po uplynutí jedné desetiny sekundy. Na níže uvedené části kódu jsou vidět proměnné, ze kterých se skládá řetězec času zobrazovaný na LCD.

```
Tim1_isr:
    TCNT1H = 231
    TCNT1L = 150
    _desetina = _desetina + 1

    If _desetina >= 10 Then
        _desetina = 0
        _sec = _sec + 1
        If _sec >= 60 Then
            _sec = 0
            _min = _min + 1
        End If
        If _min >= 60 Then
            _min = 0
            _hour = _hour + 1
        End If
    End If
Return
```

Obr. č. 33, Procedura volaná při přetečení timeru.

Výsledný řetězec ve tvaru **HH:MM:SS,D** se na displeji zobrazuje průběžně, tím závodník při letmém pohledu snadno zjistí, že stopky pořád počítají a kolik času už uběhlo. Při stisku tlačítka STOP zůstane na displeji hodnota výsledného času až do té doby, než rozhodčí na základně zmáčkne tlačítko vysílající povel k vynulování všech stopek.

Na displeji stanice se kromě času zobrazují i jiné informace o činnosti stanice. Při příjmu povelu IDEN se zobrazí informace o probíhající identifikaci stanice a příslušné číslo stanice. Po povelu NULL displej zobrazí informaci, že je stanice vynulovaná a čeká na start. Kromě těchto stavových informací se na LCD zobrazují i chybová hlášení, např. neúspěšný přenos změřeného času na základnu nebo chybný příjem – neznámý povel.

## 5.2 Program pro základnu

Tato část programového vybavení systému zabezpečuje několik víceméně oddělených funkcí. Vnitřně je každá funkce oddělenou procedurou programu.



### 5.2.1 Identifikace (IDENT)

Příkaz IDENT vyšle základna na kanálu č. 001. Každá z právě zapnutých stanic, která tento příkaz zachytí, se přeladí na svůj soukromý kanál a očekává další povel. Základna začne procházet všechny možné obsazené kanály a na každém z nich vyšle opět povel k identifikaci. Stanice, která na příslušném kanále poslouchá, odpoví a přeladí se zpět na kanál č. 001. Základna si čísla stanic, které odpověděly, zapíše do paměti a dále komunikuje jen na kanálech, kde našla při identifikaci aktivní stanice. Při identifikaci základna kontroluje i číslo stanice, jestli odpovídá příslušnému kanálu. Takto provedená počáteční identifikace umožňuje provoz celého systému s jakýmkoliv počtem zapnutých stanic a není ani potřeba, aby jejich identifikační čísla šla v pořadí za sebou. Pokud v průběhu soutěže nastane potřeba většího počtu stanic, stačí další stanici zapnout a znovu spustit identifikaci.

### 5.2.2 Nulování (NULL)

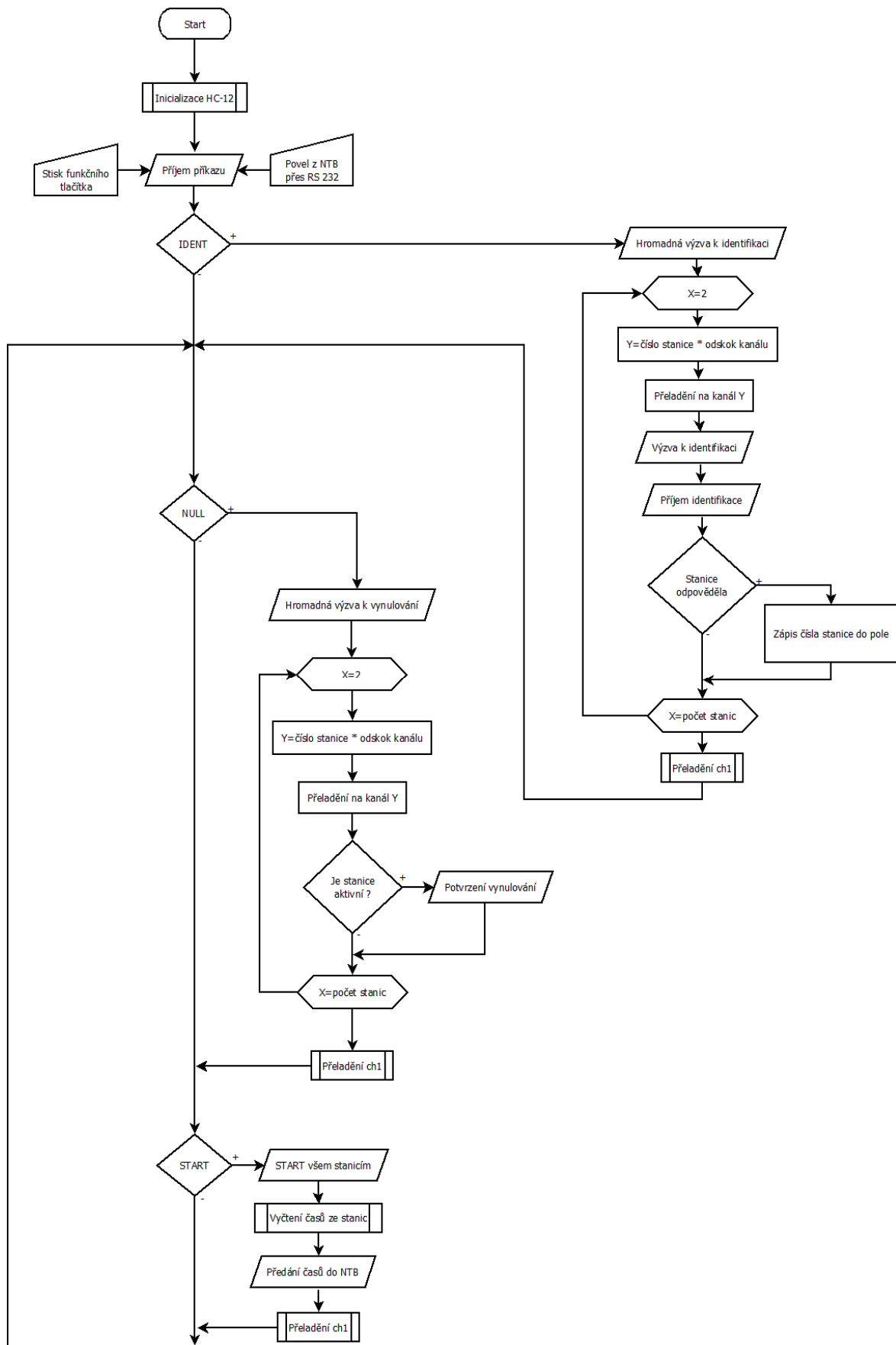
Před zahájením každého kola soutěže je třeba všechny stopky vynulovat a připravit stanice na příjem příkazu ke spuštění časomíry. Příkaz NULL stejně jako v předešlém případě odešle základna na kanále č. 001. Další komunikace probíhá opět na soukromém kanále každé stanice a základna při tom kontroluje, zda jsou aktivní všechny stanice, které se přihlásily během identifikace. Pokud některá stanice neodpoví, přestože předtím byla aktivní, základna ukáže na displeji upozornění s číslem nekomunikující stanice. Například když se během soutěže vybije stanici akumulátor. Po provedení příkazu NULL se stanice přeladí zpět na kanál č. 001 a očekává na něm příkaz START, kterým se spustí čítač, a stopky se rozběhnou.

### 5.2.3 Vyčítání dat ze stanic (COUNT)

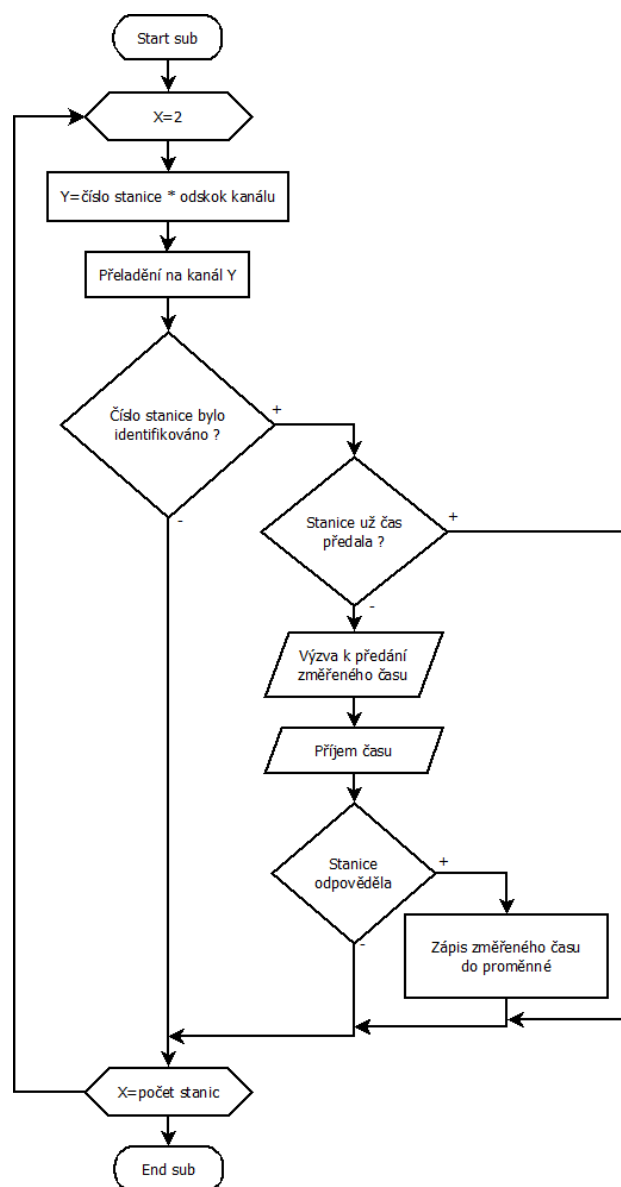
Během počítání času jednotlivé stanice neodpovídají. Po zmáčknutí tlačítka STOP zastaví časovač a nastaví se na příjem na soukromém kanálu. Základna mezitím přeladuje v určitých intervalech mezi všemi soukromými kanály zapojených stanic a testuje, zda příslušná stanice už zastavila stopky. Pokud ano, požádá stanici příkazem COUNT o odeslání výsledného času, který si zapíše do paměti, a přeladí se na kanál další stanice. Toto přeladování se provádí, dokud se nevyčtou všechny stanice. Vyčtené časy základna postupně posílá do programu v připojeném notebooku, kde jsou vidět průběžné výsledky.

Poté, co rozhodčí odsouhlasí odečtené časy, může zmáčknout tlačítko NULL, čímž se hodnoty časů ve stanicích i v základně smažou a celý systém se připraví na nový povel ke startu.





Obr. č. 34, Vývojový diagram programu základny.



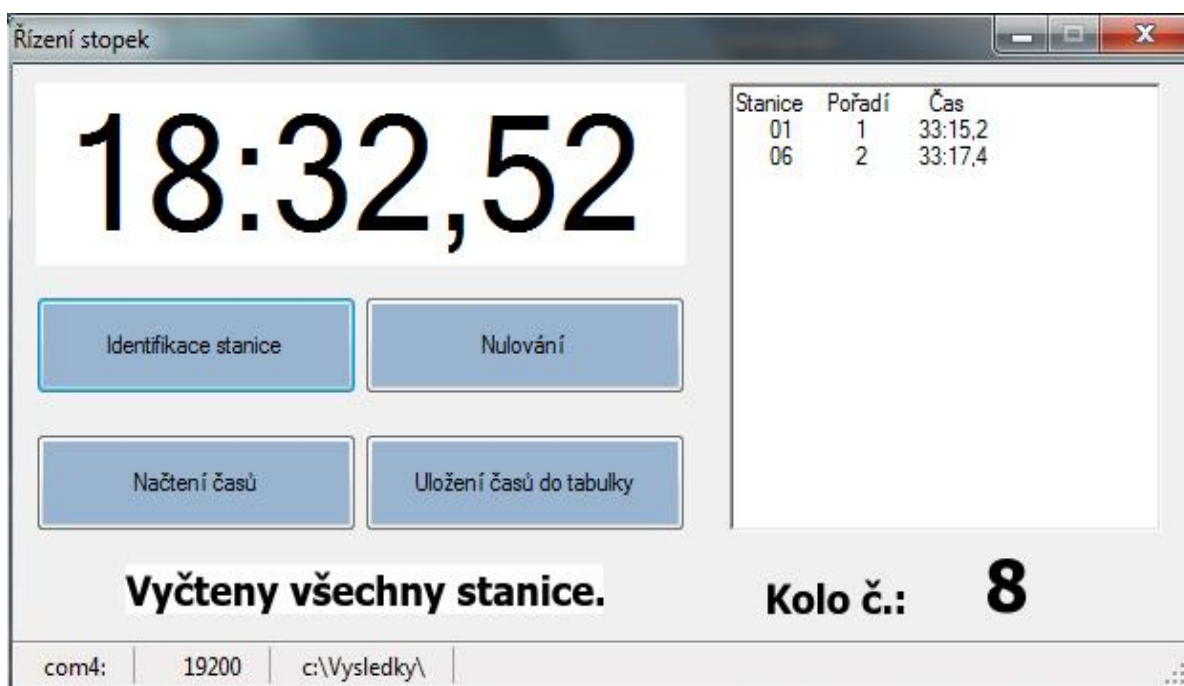
Obr. č. 35, Vývojový diagram podprogramu *Vyčítání času ze stanic*.

### 5.3 Program pro **připojený** notebook.

Tato část programového vybavení je napsaná v prostředí jazyka Visual Basic. Důležitou funkcí tohoto programu je komunikace se základnou. Ta je realizovaná sériovým portem RS 323 na straně základny a portem USB s převodníkem u notebooku. Díky tomu lze základnu a tím celý systém ovládat a následně vyčítat ze základny naměřená data.

Pro ovládání systému jsou použity stejné povely, které posílají tlačítka na základně, jen je reprezentují softwarové příkazy. Proto můžeme celý systém ovládat také myší z notebooku. Na následujícím obrázku (Obr. č. 36) je zobrazeno hlavní okno programu.

Vlevo nahoře se ukazuje ubíhající času v případě, že stopky běží. Toto pole zobrazující čas se chová stejně jako LCD na stanicích. Pokud ze základny dorazí příkaz NULL, vynuluje se a příkazem START se spustí počítání. Čas se počítá podle vnitřních hodin notebooku, není tedy synchronizován se stanicemi ani základnou. Pod časovým údajem jsou tlačítka pro ovládání systému. Funkce *Identifikace stanic*, *Nulování* a *Načtení časů* už byly popsány dříve. Funkce *Uložení časů do tabulky* veškeré získané časy zapíše do excelovské tabulky pro další zpracování.



Obr. č. 36, Hlavní okno programu pro NTB.

V pravé části okna se průběžně objevují časy, které základna získá ze stanic. Hodnoty časů se porovnávají a řadí v pořadí od nejkratšího času shora dolů. Pod tabulkou časů je číslo kola soutěže, které právě probíhá (proběhlo). Vždy po odstartování se číslo kola zvětší o jedničku. Pokud se načte poslední čas, pošle základna zprávu do notebooku, že už odečetla data ze všech připojených stanic, a pod tlačítky se objeví zpráva, že jsou výsledky kompletní. Potom obsluha uloží výsledky do tabulky.

Ve spodní části okna se zobrazuje číslo použitého Com-portu, rychlost přenosu mezi základnou a notebookem a složka, kde je umístěna tabulka s výsledky. Pokud na zmíněné nápisy klikneme myší, můžeme hodnoty měnit podle potřeby.

Tabulka výsledků ve své struktuře zahrnuje číslo soutěžního kola, které právě proběhlo, číslo stanice a naměřený čas. Takto formátovaná data se ukládají chronologicky za sebou a je věcí dalšího software, aby data zpracoval, přiřadil jednotlivým časům jména závodníků, doplnil trestné body za špatný počet nalezených zlatěnek atd.

## 6 Závěr

Na začátku práce jsem kromě celé řady technických parametrů systému, jako jsou požadované funkce, typ a dosah bezdrátového přenosu, zdroj napájení a jeho výdrž, odolnost proti povětrnostním vlivům atd., musel zohlednit i určitou modularitu, možnost různých budoucích úprav a samozřejmě i finanční náročnost celého projektu. To kvůli jeho určení, jímž jsou soutěže dětských a mládežnických klubů, jak jsem zmínil v úvodu. Při navrhování systému jsem uvažoval o tom, že bych pro stavbu základny použil místo ATmegy 32 počítač Raspberry Pi, který je mnohem výkonnější, univerzálnější a lépe rozšiřitelný, ale v té době byl finančně oproti Atmelu mnohem nákladnější.

Raspberry Pi je jednodeskový počítač s procesorem z rodiny ARM Cortex, takže je výkonově srovnatelný s běžným smartphonem. Na desce spolu s procesorem je několik portů USB v. 2, HDMI, LAN, Wi-Fi, Bluetooth, audio, konektor pro Micro SD kartu, paměť RAM atd. Většina kitů Raspberry Pi obsahuje i 40pinový GPIO konektor, který se dá použít stejně jako porty na Atmelu. Vlastnostmi, výkonem a vybavením daleko překonává mikrokontrolér AT Mega 32, ale při zvážení jeho ceny a funkcí, které od základny opravdu nutně potřebujeme, jsem zvolil Atmel.

Dnes už je situace jiná. Na trh se letos dostaly klony Raspberry Pi - moduly Orange Pi a Banana Pi. Posledně jmenované jsou sice výkonnější a lépe vybavené, jsou ale dražší, než Raspberry Pi. Za to Orange Pi mají při obdobném vybavení desky i výkonu v některých případech až třetinovou cenu oproti svému předobrazu. Je tomu nejspíš proto, že za vývojem projektu Orange Pi stojí čínská společnost Shenzhen Xunlong Software CO., Limited, kdežto původní Raspberry Pi vyvíjí britská nadace Raspberry Pi Foundation.

Základna osazená Orange Pi má oproti současné konstrukci s Atmelem několik výhod. Podporuje operační systémy Linux a Android, takže se dá naprogramovat téměř cokoliv jako na běžném PC nebo smartphonu. Do základny můžeme umístit Apache HTTP server, který bude získaná data umisťovat na webové stránky a díky vestavěné Wi-Fi se budou výsledky online zobrazovat na notebooku, chytrém telefonu nebo tabletu. Zároveň by

mohl importovat aktuální startovní listinu a zobrazené stránky by tak byly kompletní a zároveň dostupné komukoliv, kdo se připojí k internetu.

Variantu „Orange“ základny, jak jsem ji teď zhruba popsal, včetně ovládání celého systému pomocí tabletu nebo telefonu s Androidem, bych chtěl zabudovat do další verze časomíry, pokud bude o tento systém u sportovních klubů ještě zájem.

## 7 Přílohy:

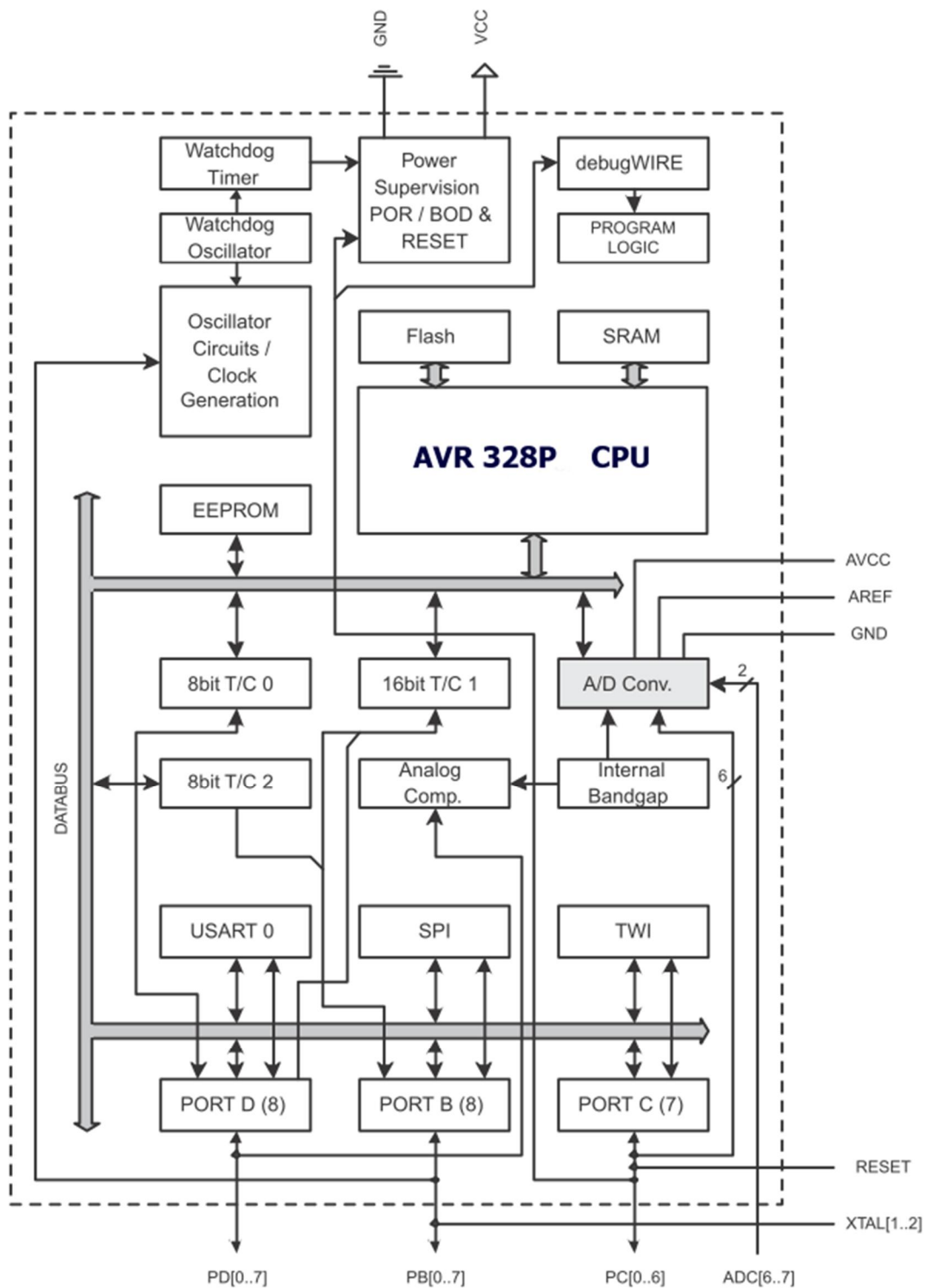
### 7.1 Stupně krytí (ČSN EN 60 529) [13]

Tab. č. 2 Popis stupňů krytí (ČSN EN 60 529).

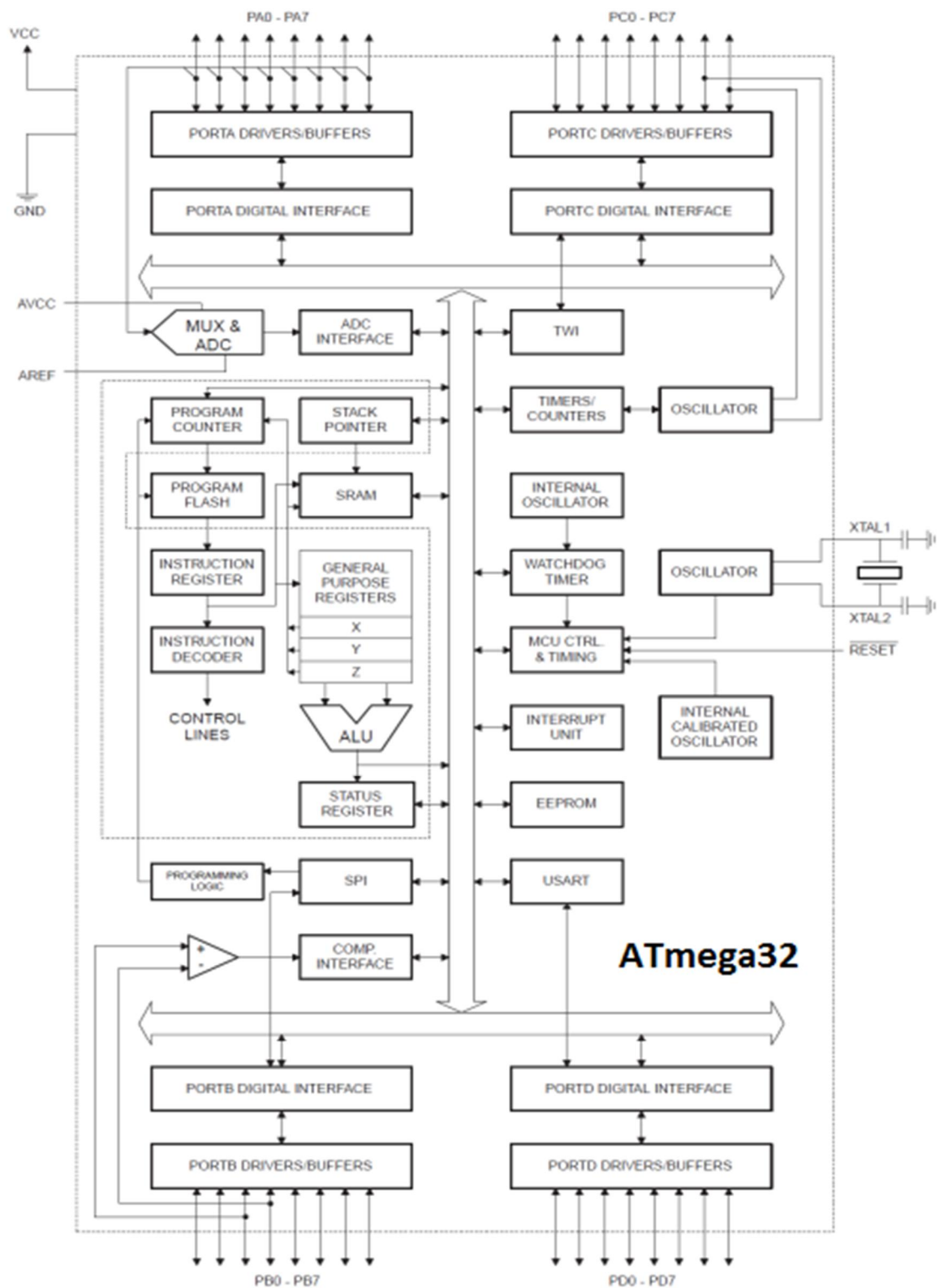
Stupně ochrany před dotykem nebezpečných částí a před vniknutím cizích pevných těles udávané první číslicí:	
IP 0x	Nechráněno
IP 1x	Zařízení je chráněno před vniknutím pevných cizích těles o průměru 50mm a větších a před dotykem dlaně
IP 2x	Zařízení je chráněno před vniknutím pevných cizích těles o průměru 12,5mm a větších a před dotykem prstem
IP 3x	Zařízení je chráněno před vniknutím pevných cizích těles o průměru 2,5mm a větších a před dotykem nástrojem
IP 4x	Zařízení je chráněno před vniknutím pevných cizích těles o průměru 1mm a větších a před dotykem drátem
IP 5x	Zařízení je chráněno před prachem a před dotykem jakoukoli pomůckou
IP 6x	Zařízení je prachotěsné a je chráněno před jakoukoli pomůckou
Stupně ochrany proti vniknutí vody udávané druhou číslicí:	
IP x0	Nechráněno
IP x1	Chráněno proti padající vodě při ekvivalentu deště 3–5 mm padající vody za minutu v průběhu 10 minut. Jednotka je umístěna ve své pracovní poloze
IP x2	Chráněno proti padající vodě, když je přístroj v poloze 15 stupňů. Stejně jako IPX-1, jen s rozdílem, že jednotka je testována ve 4 pozicích, nakloněna o 15° v každé poloze od normální provozní polohy
IP x3	Chráněno proti vodní tříšti. Voda stříká na přístroj v úhlu 60° vertikálně, v množství 10 litrů za minutu a při tlaku 80–100kN/m <sup>2</sup> po dobu 5 minut
IP x4	Chráněno proti stříkající vodě. Stejně jako u IPX-3, jen s rozdílem, že voda stříká ve všech úhlech
IP x5	Chráněno proti vodním proudům. Voda míří 6,3 mm tryskou ve všech úhlech při průtoku 12,5 litrů za minutu při tlaku 30 kN/m <sup>2</sup> po dobu 3 minuty ze vzdálenosti 3 metry
IP x6	Chráněno proti vlnobití. Voda míří 12,5 mm tryskou ve všech úhlech při průtoku 100 litrů za minutu při tlaku 100 kN/m <sup>2</sup> po 3 minuty ze vzdálenosti 3 metry
IP x7	Chráněno proti ponoření do vody. Ponoření na 30 minut do hloubky 1 metr
IP x8	Chráněno proti potopení do vody. Zařízení je schopné nepřetržitého potopení do vody za podmínek, které určí výrobce zařízení



## 7.2 Bloková schémata použitých mikrokontrolérů



Obr. č. 37, Blokové schéma ATmega328P, převzato a upraveno z [8].



Obr. č. 38, Blokové schéma ATmega32, převzato a upraveno z [14].

## 7.3 Seznam obrázků

Obr. č. 1, Závodnice chce právě stisknout tlačítko.....	8
Obr. č. 2, Zařízení kabelové časomíry. ....	9
Obr. č. 3, Nástříkové terče (vpravo naplněn vodou), převzato a upraveno z [1]. ....	10
Obr. č. 4, Mechanismus sklopných terčů, převzato a upraveno z [2]. ....	11
Obr. č. 5, Editor elektrických schémat. ....	13
Obr. č. 6, Editor plošného spoje. ....	14
Obr. č. 7, 3D pohled na desku pro LCD. ....	15
Obr. č. 8, Základní okno programového prostředí Bascom. ....	17
Obr. č. 9, Menu v Bascomu ....	17
Obr. č. 10, Okno AVR programmeru. ....	18
Obr. č. 11, Hlavní okno Visual Studia při návrhu. ....	19
Obr. č. 12, Modul 328P Pro Mini s popisem pinů, převzato a upraveno z [9]. ....	21
Obr. č. 13, Nahrazení sériové komunikace, převzato a upraveno z [10]. ....	22
Obr. č. 14, Deska modulu HC-12 s anténou, převzato a upraveno z [10]. ....	22
Obr. č. 15, Modul s ATmega 32, převzato a upraveno z [10]. ....	25
Obr. č. 16, LCD 1602, převzato a upraveno z [10]. ....	25
Obr. č. 17, Krabice s hřibovým tlačítkem, převzato a upraveno z [12]. ....	26
Obr. č. 18, Blokové schéma stanice. ....	26
Obr. č. 19, Foto sestavené stanice. ....	27
Obr. č. 20, 3D model pouzdra pro Arduino. ....	28
Obr. č. 21, Schéma desky s AT mega 328. ....	28
Obr. č. 22, Návrh DPS stanice ze strany součástek. ....	29
Obr. č. 23, Schéma desky LCD. ....	30
Obr. č. 24, Návrh DPS s LCD ze strany součástek. ....	30
Obr. č. 25, Foto otevřené stanice s popisem. ....	31
Obr. č. 26, Powerbanka, převzato a upraveno z [10]. ....	32
Obr. č. 27, Foto otevřené základny s popisem. ....	33
Obr. č. 28, Blokové schéma základny. ....	34
Obr. č. 29, Schéma propojovací desky. ....	35
Obr. č. 30, DPS propojovací desky. ....	35

Obr. č. 31, Procedura přeladění na soukromý kanál.....	37
Obr. č. 32, Vývojový diagram programu stanice. ....	38
Obr. č. 33, Procedura volaná při přetečení timeru.....	39
Obr. č. 34, Vývojový diagram programu základny. ....	41
Obr. č. 35, Vývojový diagram podprogramu <i>Vyčítání času ze stanic</i> . ....	42
Obr. č. 36, Hlavní okno programu pro NTB.....	43
Obr. č. 37, Blokové schéma ATmega328P, převzato a upraveno z [8].....	47
Obr. č. 38, Blokové schéma ATmega32, převzato a upraveno z [14].....	48

## 7.4 Seznam tabulek

Tab. č. 1, Seznam AT příkazů HC-12. [11].....	23
Tab. č. 2 Popis stupňů krytí (ČSN EN 60 529).....	46

## 8 Použité prameny a literatura

- [1] Nástříkové terče. *SDH Darkovice* [online]. Darkovice: SDH Darkovice, 2007 [cit. 2016-12-21]. Dostupné z: <http://www.sdh-darkovice.cz/index.php?p=130>
- [2] Sada 2 sklopných terčů. *Firesport - internetový obchod pro hasiče* [online]. Radkov: Marie Hynčicová, 2016 [cit. 2016-12-21]. Dostupné z: <http://shop.firesport.cz/sada-2-sklopných-tercu-se-svetly-a-spinaci-elektronikou-nezavislou-na-casomire/>
- [3] FRYDRYCH, Michal. *Bezdrátová elektronická časomíra s velkým LED zobrazovačem* [online]. Brno: Frydrych, Michal, 2010 [cit. 2016-12-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/handle/11012/17672>. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. David Jaroš.
- [4] *KiCad EDA* [online]. Fr: The KiCad Team, © 2010-2015 [cit. 2016-12-21]. Dostupné z: <http://kicad-pcb.org/>
- [5] VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR: programování v jazyce Bascom*. 1. vyd. Praha: BEN - technická literatura, 2004. ISBN 80-730-0115-2.
- [6] *Visual Studio* ® [online]. USA: Microsoft, 2016 [cit. 2016-12-21]. Dostupné z: <https://www.visualstudio.com/cs/>
- [7] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL*. 2. vyd. Praha: BEN - technická literatura, 2006.  $\mu$ C & praxe. ISBN 8073002094.
- [8] ATMEGA328P Datasheet (PDF) - ATMEL Corporation. *ALLDATASHEET.COM* [online]. US: Alldatasheet, 2016 [cit. 2016-12-21]. Dostupné z: <http://www.alldatasheet.com/datasheet-pdf/pdf/241077/ATMEL/ATMEGA328P.html>
- [9] Elektro Pro-Mini pinout diagram. *Pinterest* [online]. California, USA: Pinterest, Inc., 2016 [cit. 2016-12-22]. Dostupné z: <https://cz.pinterest.com/pin/552113235551633755/>
- [10] DealeXtreme. *DealeXtreme* [online]. N.T.Hong Kong: Epro E-commerce Limited, 2005 [cit. 2016-12-21]. Dostupné z: <http://www.dx.com/>
- [11] HC-12 Wireless Serial Port Communication Module User Manual V1.1. *WWW.SEEEDSTUDIO.COM* [online]. San Leandro, United States: Seeed Studio, © 2008-2016 [cit. 2016-12-21]. Dostupné z: [http://wiki.seeedstudio.com/images/b/b0/HC-12\\_User\\_Manual.pdf](http://wiki.seeedstudio.com/images/b/b0/HC-12_User_Manual.pdf)
- [12] *Elektro Hartman* [online]. Česká Skalice: ELEKTROHARTMAN, 2016 [cit. 2016-12-21]. Dostupné z: <http://www.elektrohartman.cz>
- [13] *ČSN EN 60 529 (33 0330) Stupně ochrany krytem (krytí-IP kód) / zpracovali Zdeněk Strmiska, Radka Horká*. 1. Praha: Český normalizační institut, 1993.



- [14] ATMEGA32 Datasheet (PDF) - ATMEL Corporation. *ALLDATASHEET.COM* [online]. USA: Alldatasheet, 2016 [cit. 2016-12-21].  
Dostupné z: <http://pdf1.alldatasheet.com/datasheet-pdf/view/77378/ATMEL/ATMEGA32.html>