



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

# Stavebnice a úlohy s jednočipovými počítači Atmel AVR pro základní školy

Vypracoval: Bc. Vojtěch Bušek  
Vedoucí práce: Ing. Michal Šerý

České Budějovice 2013

Diplomová práce se zabývá návrhem a realizací stavebnice použitelné pro výuku práce s jednočipovými počítači Atmel AVR a jejich periferními obvody. Součástí diplomové práce jsou návrhy elektrických schémat jednotlivých modulů stavebnice, výkresy desek plošných spojů a návrhy ukázkových úloh použitých při výuce tohoto tématu včetně pracovních a metodických listů. Pro svou jednoduchost a názornost byl jako programovací jazyk aplikací zvolen jazyk Bascom AVR.

This thesis describes the design and implementation of a building kit suitable for teaching how to work with Atmel AVR single-chip computer and its peripheral circuits. Included in the thesis are drafts of electrical schematics of each individual module of the building kit, drawings of the PCBs as well as propositions for sample tasks used in teaching this subject. These further encompass methodological and work papers. Due to its simplicity and clarity the Bascom AVR was chosen as the programming language for all applications.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové, a to v nezkrácené podobě 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.

29.6.2013

podpis

Touto formou bych rád poděkoval vedoucímu této diplomové práce p. Ing. Michalovi Šerému a Ing. Zdeňkovi Kouřilovi za cenné rady a připomínky při zpracování mé práce.

## Obsah

1	Úvod .....	8
2	Popis procesorů rodiny Atmel AVR.....	8
2.1	Vnitřní struktura a vybrané parametry .....	9
2.2	Periférie procesorů Atmel AVR.....	12
2.2.1	Vstupně výstupní porty .....	13
2.2.2	Integrované čítače a časovače .....	15
2.2.3	Sériové kanály .....	16
2.2.4	Analogový komparátor.....	18
2.2.5	Analogově digitální převodník.....	19
2.3	Bootloader .....	19
3	Stručný popis programovacího jazyka Bascom AVR .....	21
3.1	Základy jazyka .....	21
3.2	Datové typy a proměnné .....	22
3.3	Výrazy a operátory.....	23
3.4	Pedagogické aspekty jazyka Bascom a programování jednočipových počítačů Atmel.....	24
3.4.1	Čas a taktovací frekvence.....	24
3.4.2	Práce v cyklu .....	25
3.4.3	Desetinná čísla.....	25
3.4.4	Možné problémy při zápisu hodnot do portu .....	26
4	Návrh modulů stavebnice .....	28
4.1	Technické provedení .....	28
4.1.1	Propojení modulů .....	29
4.1.2	Napájení.....	29
4.2	Modul 8x LED + 1x Bargraf.....	30
4.3	Modul 2x 8 spínačů s volbou výstupu.....	33
4.4	Modul 2x sedmissegmentový displej.....	36

4.5	Modul 2x BCD dekodér .....	39
4.6	Modul 4x sedmissegmentový displej se společnou anodou .....	42
4.7	Modul izolovaný H můstek .....	45
4.8	Modul 16bit I/O expandér se sběrnici I <sup>2</sup> C .....	48
4.9	Modul konektorů .....	52
4.10	Modul s MCU .....	55
4.11	Modul napájení .....	59
4.12	Modul běžných vstupů a výstupů .....	63
5	Návrh ukázkových úloh .....	67
5.1	Ukaž, jaké spínače (ne)jsou sepnuté .....	68
5.1.1	Pracovní list úlohy .....	69
5.2	Knight Rider .....	70
5.2.1	Pracovní list úlohy .....	71
5.3	Knight Rider II. ....	72
5.3.1	Pracovní list úlohy .....	73
5.4	Dekodér .....	74
5.4.1	Pracovní list úlohy .....	75
5.5	Semafor s přechodem pro chodce .....	76
5.5.1	Pracovní list úlohy .....	77
5.6	Návrhy dalších úloh .....	78
5.6.1	4bitový dekodér BCD na HEX .....	78
5.6.2	Multiplexování čtyřmístného segmentového displeje .....	78
5.6.3	Maticová klávesnice .....	78
5.6.4	Hrací kostka .....	78
5.6.5	Teploměr s využitím termistoru .....	79
5.6.6	Stroboskop .....	79
5.6.7	Řízení směru otáčení SS motoru .....	79

5.6.8	Měření doby kmitu, nebo kyvu .....	79
6	Závěr .....	80
7	Odborná literatura.....	81

## 1 Úvod

Jednočipové počítače jsou v současné době elektronickým prvkem, který je stále častěji využíván nejen zkušenými odborníky při řízení různých elektronických systémů, ale postupně se rozšiřuje jako ovládací prvek amatérských konstrukcí a projektů. Současná masivní výroba vývojových stavebnic je toho přímým důkazem.

Cílem této práce je vytvořit jednoduše použitelné moduly pro připojení k jednočipovému počítači a navrhnout vhodné úlohy pro výuku základního programování jednočipových počítačů. Vzhledem k primárnímu určení těchto úloh pro žáky 9. ročníků ZŠ jsou voleny relativně jednoduché programy a zapojení. Důraz je kladen také na možnosti jejich vzájemného propojení – složením dvou jednoduchých programů, které mohou existovat „samy o sobě“ bez většího užitku, může vzniknout užitečná aplikace. Jako programovací jazyk je zvolen Bascom® AVR. Ten vychází ze syntaxe výukového jazyka Basic. Domnívám se, že tento programovací jazyk je vhodnější pro výuku začátečníků díky jeho snadno pochopitelným příkazům a struktuře, které z velké míry přebírá právě po zmíněném Basicu.

## 2 Popis procesorů rodiny Atmel AVR

Jednočipové počítače byly na trh uvedeny firmou Intel v roce 1977. Konkrétně se jednalo o typ Intel 8048, který se díky své univerzálnosti velmi rozšířil do nejrůznějších oblastí spotřební elektroniky a ovládacích systémů. Tyto jednočipové počítače se vyznačovaly integrací množství prvků a periférií, které bylo možné využívat pro ovládání různých elektronických zařízení a automatizaci systémů. Díky integraci těchto periférií byla velmi příznivá i cena a zároveň spolehlivost těchto specializovaných obvodů. To vedlo k jejich velké oblibě, dalšímu vývoji a rozšíření jejich použití. V současné době zažívají jednočipové počítače opravdový rozmach také díky jejich nízké ceně a snadnému využití k nejrůznějším účelům. Velmi časté je jejich nasazení v jednoduchých řídicích obvodech včetně různých hraček a tzv. „Do It Yourself“ projektů. Všeobecně známé je jejich nasazení v open source projektech založených na vývojových deskách a perifériích Arduino.

S postupem času a zvyšujícími se požadavky na funkce těchto obvodů dochází k stále větší integraci, zvyšování rychlosti zpracování instrukcí a v neposlední řadě ke snižování počtu připojených prvků potřebných pro jejich činnost.



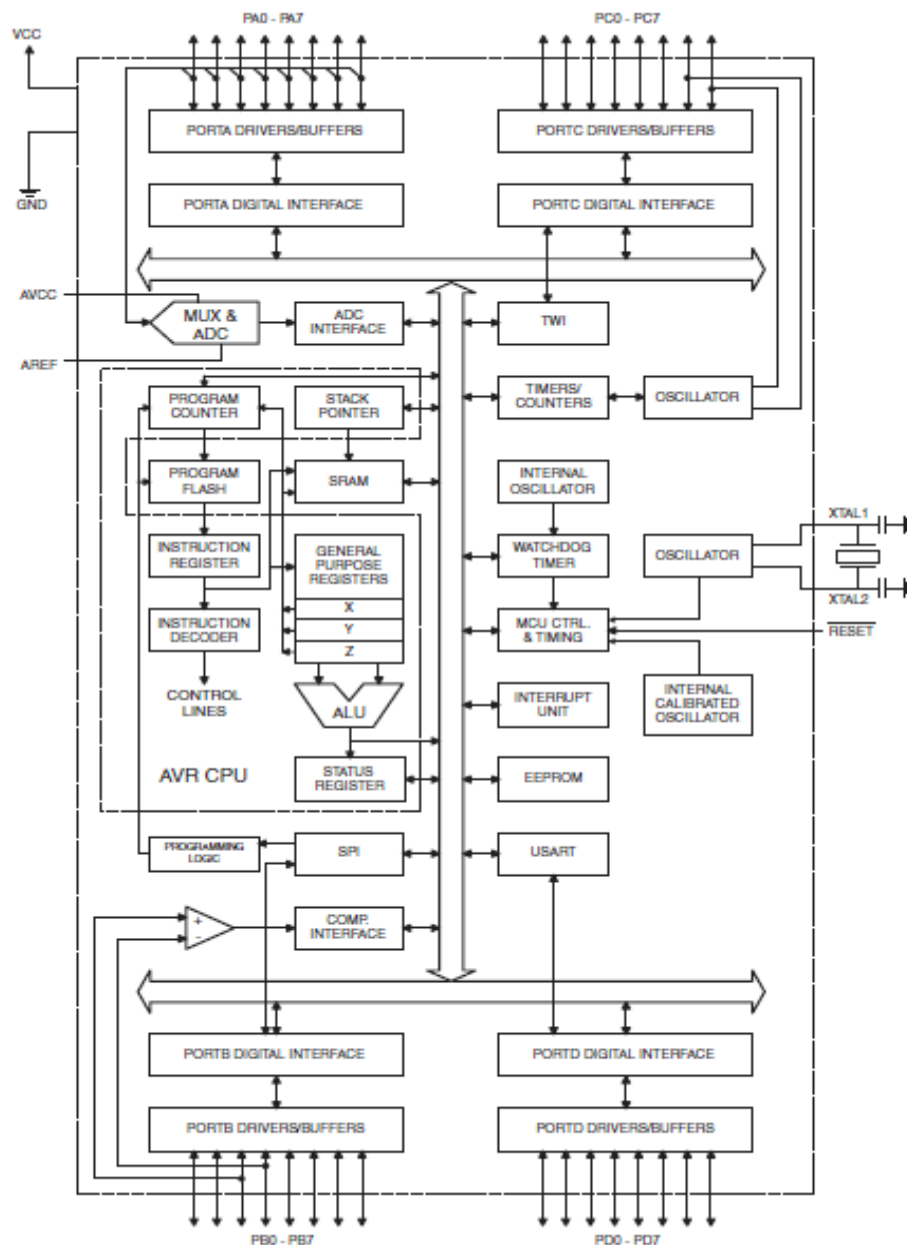
Takovými obvody jsou i jednočipové počítače rodiny Atmel AVR. Je možné je popsat jako jednočipové počítače harwardského typu s redukovanou instrukční sadou (*Reduced Instruction Set Coprocessor*), které v sobě integrují aritmeticko-logickou jednotku, paměť SRAM, non-volatilní paměti EEPROM a Flash, vstupně-výstupní porty, čítače, časovače, programovací rozhraní, komunikační rozhraní a další periferie podle konkrétního typu obvodu. Velkou výhodou zmiňovaných obvodů Atmel AVR je integrace dalších prvků potřebných pro jejich činnost. Konkrétně se jedná o RC obvody, určující taktovací frekvenci, reset a watchdog obvody, které snižují počet potřebných připojených diskrétních prvků na naprosté minimum. Tím je značně zjednodušeno zapojení těchto obvodů a otevírá se cesta pro samotnou práci s jednočipovými počítači a jejich periferiemi.

Další nespornou výhodou, alespoň některých jednočipových počítačů, je i možnost použití tzv. zavaděče. Ten je jednou standartním způsobem naprogramován do vnitřní paměti a po každém zapnutí předává k vykonání uživatelem vytvořený a nahraný program. Sám se při dalším běhu již neuplatňuje. Konkrétní výhoda spočívá v následné jednoduchosti programování jednočipového počítače. Ten může být programován pouhým nahráním vytvořeného programu (uloženého zpravidla v souboru typu *.hex*) přes sběrnici USB nebo jiné sériové rozhraní. K tomuto účelu je možné použít velké množství volně šiřitelných programů.

Standardním způsobem programování je myšleno programování paměti jednočipového počítače pomocí speciálních programovacích obvodů tzv. „programátorů“ s paralelním nebo sériovým rozhraním SPI (Serial Programming Interface).

## **2.1 Vnitřní struktura a vybrané parametry**

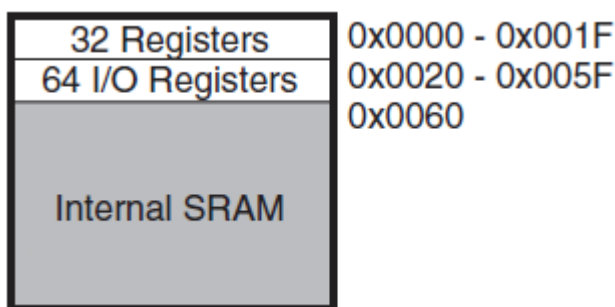
Procesory Atmel AVR jsou 8 bitové počítače harwardského typu a jako takové tedy obsahují dvě oddělené paměti. Jednu volatilní pro aktuálně zpracovávaná data, druhou non-volatilní, která uchovává činný program. Na obr. č. 1 je schématem znázorněna vnitřní struktura procesoru, konkrétně ATmega32. Vnitřní struktura jiných procesorů Atmel AVR je téměř stejná. Liší se v počtu vstupně-výstupních portů a připojených periferií.



Obr. č. 1, vnitřní struktura procesoru ATmega32, převzato z [1]

Instrukce těchto procesorů jsou reprezentovány 16 bitovým instrukčním slovem. To je dostatečné pro obsažení všech instrukcí procesoru a jejich dekódování v dekodéru instrukcí během jednoho hodinového cyklu. Jejich výkon je tedy svázan s taktovací frekvencí a dosahuje hodnoty 1 MIPS/1 MHz [1].

Procesory obsahují volatilní a non-volatilní paměti SRAM, FLASH a EEPROM. SRAM pro ukládání dat potřebných pro jejich činnost, FLASH pro uložení programu a EEPROM pro uložení proměnných, které jsou určeny k uchování při vypnutí procesoru. Jejich velikost je - typ od typu - různá a závislá na typu procesoru. Pohybuje se od jednotek bajtů do desítek kilobajtů.



Obr. č. 2, vnitřní uspořádání SRAM, převzato z [1]

## Vnitřní uspořádání paměti

SRAM je uvedeno na obrázku č. 2.

Mikroprocesory obsahují 32 univerzálních pracovních registrů, které jsou používány výkonným jádrem k načítání operandů a ukládání výsledků jeho výpočtů.

Dále obsahují 64 vstupně-výstupních registrů, které ovlivňují chování mikroprocesoru. Je zde možné nastavovat vlastnosti a chování periférií, vstupně-výstupních portů, ale i samotného mikroprocesoru. Na obrázku č. 2 je uvedeno adresování celého datového prostoru. Z pohledu jiného modelu adresování, které používá i programovací jazyk Bascom, se univerzální pracovní registr neadresuje, a tedy adresy I/O registrů jsou 0x00 – 0x3F [2].

Vstupně-výstupní registry jsou ve většině případů u jednočipových počítačů Atmel AVR totožné. Jejich kompletní přehled a popis je vždy uveden v dokumentaci výrobce.

Paměť SRAM je dále použitelná pro uložení uživatelských proměnných a její kapacita je závislá na typu jednočipového počítače. Pohybuje se v rozmezí od desítek bajtů po jednotky megabajtů.

Posledním použitým typem je paměť typu EEPROM. Její životnost je asi 100 000 čtecích/zapisovacích cyklů, kapacita se pohybuje od stovek bajtů do desítek kilobajtů. Vzhledem k omezenému počtu možných zápisů není vhodné ji používat pro ukládání zpracovávaných dat. Naopak je vhodné ji používat pro zápis programových konstant, nebo pro uložení proměnných před vypnutím jednočipového počítače a pro jejich další použití při opětovném zapnutí. Paměť EEPROM je čtena a zapisována po bytech. Při čtení z této paměti je procesorová jednotka zastavena po dobu 4 hodinových cyklů, než vykoná další instrukci. V případě zápisu do EEPROM je doba zastavení 2 hodinové cykly [3].

Jednočipové počítače jako sekvenční obvody potřebují pro svou činnost hodinový (taktovací) signál. Jeho zdroj u mikroprocesorů Atmel může být dvojitý. Externí v podobě krystalu, nebo interní v podobě kalibrovaného RC obvodu. Interní zdroj je

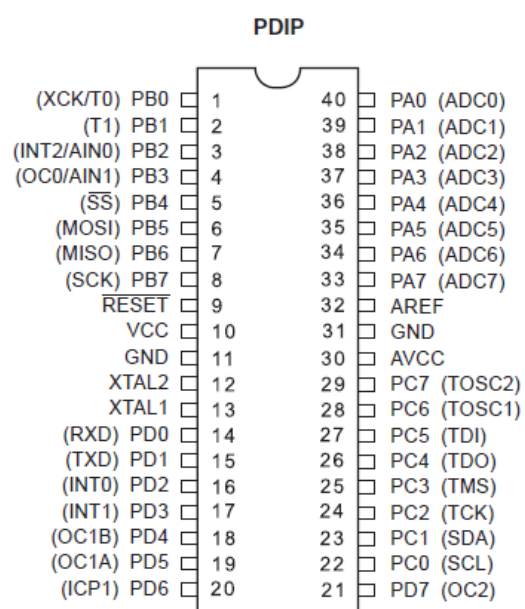
přítomný vždy. Možnost připojit externí zdroj hodin zpravidla chybí u mikroprocesorů rodiny TINY kvůli omezenému počtu jejich vstupních (nebo výstupních) pinů.

Při výběru frekvence taktovacího signálu je nutné brát ohled na některé periferie, které v aplikaci jednočipového počítače používáme. Jedná se především o ty, jejichž bezproblémová funkčnost závisí na přesném časování – tedy hlavně o sběrnice.

Interní zdroj hodin je velmi zajímavou funkcí mikroprocesoru, protože umožňuje jeho provoz bez připojení dalších potřebných prvků, jako jsou krystaly, rezistory a kondenzátory. Ta se uplatní především pro různé zkušební a jednoduché obvody, které se obejdou bez vysokých a velmi přesných frekvencí taktovacího signálu. Jednočipové počítače Atmel AVR zpravidla disponují kalibrovaným RC obvodem o volitelné frekvenci 8 MHz, 4 MHz, 2 MHz nebo 1 MHz. Frekvenci taktovacího signálu a jeho zdroj je možné zvolit při programování mikroprocesoru, a to zapsáním konfiguračních pojistek. Za provozu nemůže být taktovací frekvence měněna.

## 2.2 Periferie procesorů Atmel AVR

Mikroprocesory nejsou jen holé obvody určené k výpočtům, ale integrují v sobě i řadu dalších obvodů, které rozšiřují jejich funkčnost. Tyto obvody se nazývají periferie. Všechny periferie, jak se lze přesvědčit na obrázku č. 3, s mikroprocesorem



Obr. č. 3, rozložení pinů ATMEGA32, převzato z [1]

sdílí jeho piny, kterými je připojen do elektrického obvodu. Představují významné prvky, které umožňují efektivně, často bez použití dalších obvodů, použít jednočipový počítač pro řízení komplexních systémů [4].

Rozložení pinů a přidružených periférií se může typ od typu mikroprocesoru lišit. Stejně tak i jejich počet.

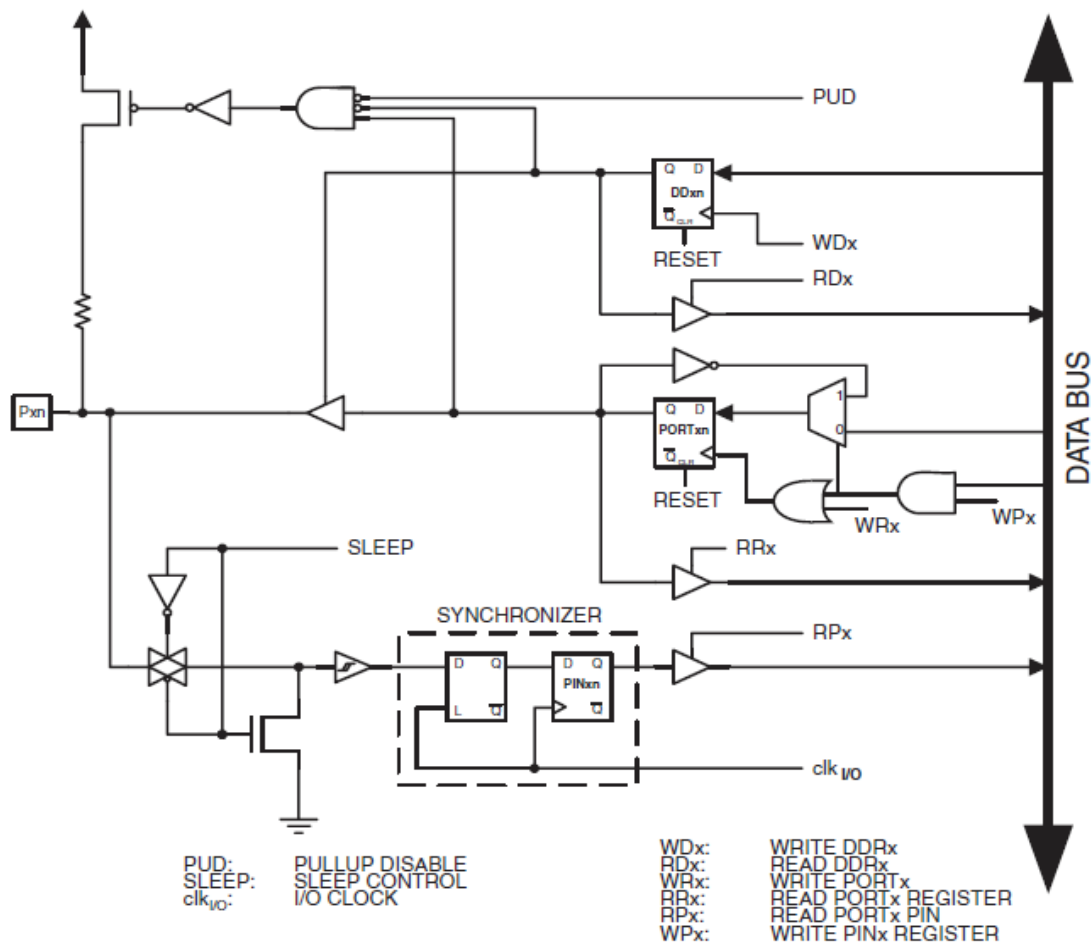
Společným rysem všech mikroprocesorů jsou právě piny, které slouží pro připojení k obvodu. Ty jsou sdruženy do maximálně 8bitových portů, které se označují písmeny A, B, ..., F. Všechny typy

jednočipových počítačů Atmel disponují portem B. Vstupně-výstupní porty jsou jednou z periférií, proto je popíšeme blíže.

### 2.2.1 Vstupně výstupní porty

Každý jednočipový počítač používá pro komunikaci s okolím několik k tomu určených vývodů (pinů). Ty mohou být nastaveny buď jako vstupní, nebo jako výstupní. Přeneseně tedy, v souvislosti s komunikací mikropočítače, je možné z těchto vývodů data číst, nebo je na ně zapisovat.

Nastavení pinů lze libovolně měnit i při chodu pouhou změnou v příslušném registru. Totéž platí i pro nastavení dalších funkcionalit pinů, jako je například zapnutí interního zdvihacího („pull-up“) rezistoru, tedy pokud je pin nastaven jako výstup. Obrázek č. 4 popisuje vnitřní zapojení digitálního vstupně výstupního pinu včetně souvislostí s příslušnými registry. Jednotlivé piny se sdružují do portů označených písmeny. Příslušné registry a jejich funkce uvádí tabulka č. 1.



Obr. č. 4, schéma digitálního I/O portu, převzato z [5]

Tabulka č. 1, registry portů

<b>Registr</b>	<b>Název</b>	<b>Funkce</b>
DDRx	Data Direction Register	Nastavení vstup / výstup pro port x
PORTx	Port x Data Register	Zápis hodnot na port, pokud je konfigurován jako výstup / aktivace pull-up rezistoru, pokud je port konfigurován jako vstup
PINx	Port x Input Pins	Čtení hodnoty na příslušném portu

Chování zdvihacího rezistoru dále ovlivňuje globální nastavení tzv. „Pull-up disable“ (PUD) bitu, který se nachází v MCUCR (MCU Control Registru). Všechny možnosti nastavení příslušných registrů a následné projevy chování pinů / portů shrnuje tabulka č. 2.

Tabulka č. 2, shrnutí chování digitálních pinů / portů [1]

<b>DDRx</b>	<b>PORTx</b>	<b>PUD</b>	<b>Směr</b>	<b>Pull-UP</b>	<b>Popis</b>
0	0	X	vstup	ne	Vysoká impedance
0	1	0	vstup	ano	Pinem teče proud, pokud je připojen k log. 0
0	1	1	vstup	ne	Vysoká impedance
1	0	X	výstup	ne	Nastavena log. 0
1	1	X	výstup	ne	Nastavena log. 1

Jak již bylo řečeno, jednotlivé piny se sdružují do portů, které jsou většinou 8 bitové. Vzhledem k celému portu pak piny plní úlohu jednotlivých bitů. Je tedy možné hovořit o nejméně nebo nejvíce významném bitu. V případě 8 bitového portu se používá značení PORTx.0 jako nejméně významný bit (pin) na portu x, a PORTx.7 jako nejvíce významný bit (pin) na portu x. Samotný port je tedy obrazně možné označit jako bajt (slovo).

Vnitřní elektrické zapojení vstupně výstupních pinů je provedeno jako poměrně odolné vůči chybnému zapojení. V obvodu každého pinu jsou přítomné i ochranné diody k signálům GND a  $V_{cc}$ . Maximální proud tekoucí pinem je omezen charakteristicky na 25 mA. Proudová zatížitelnost celého portu je pouze 100 mA. Jednočipový počítač je tedy ve většině případů schopný napájet např. LED displeje, a to bez použití dalších podpůrných obvodů.

### 2.2.2 Integrované čítače a časovače

Při práci s jednočipovými počítači je často nutné jeho práci rozvrhnout do určitých časových úseků a případně časové úseky měřit. Jednočipové počítače obsahují obvody, které tyto funkce realizují samostatně. Tím nedochází ke zbytečnému vytížení procesoru a ten je tak schopen vykonávat zcela nezávisle vlastní program. Po uplynutí měřeného nebo generovaného časového úseku je generováno přerušení a definovanou částí programu je toto přerušení obslouženo. Čítače a časovače jsou zpravidla realizovány jedním obvodem s možností nastavení módu. Obvod tak tedy může vystupovat jako časovač nebo jako čítač [3].

Čítače (counters) čítají impulzy vstupního signálu.

Časovač (timer) načítá impulzy odvozené od hodinového signálu mikroprocesoru, a tím je možné zajistit odměření přesného časového úseku [6].

Mikroprocesory Atmel AVR většinou integrují dva takové obvody, jeden 8 bitový a druhý 16 bitový. Čítače / časovače se tedy liší počtem možných čítaných impulzů, nebo délkou času, který jsou schopné měřit.

Vzhledem k architektuře RISC těchto mikroprocesorů jsou časovače schopné inkrementovat příslušný registr s každou periodou hodinového signálu. Z důvodu možnosti použití hodinového signálu vysoké frekvence obsahují obvody i tzv. předděličku tohoto signálu. Ta je tvořena samostatným 10 bitovým čítačem s možností dělit vstupní hodinový signál hodnotami 8, 64, 256 a 1024. Doba inkrementování čítače je určena vztahem (1), kde  $t$  je čas inkrementace,  $f_{osc}$  je frekvence hodinového signálu a  $n$  je hodnota nastaveného předdělení.

$$t = \frac{f_{osc}}{n} \quad (1)$$

Pro širší možnosti využití je možné do registrů příslušných čítačů/časovačů zapisovat vlastní hodnoty – tedy čítače/časovače nemusí vždy počítat od nulové hodnoty. Přirozeně je také možné hodnoty registrů načítat a dále s nimi pracovat.

Při nutnosti čítání konkrétní hodnoty, resp. odměření konkrétního časového úseku, je možné do registru TCNTx přímo zapsat konkrétní hodnotu. Uvažujeme-li frekvenci hodinového signálu, například 8 MHz, a časovači bude nastaveno předdělení

hodnotou 256, zvýší se hodnota v registru za 1s na 31250. V případě 16 bitového čítače/časovače však k přetečení a k vyvolání přerušení dojde až po napočítání 65536 pulzů. Pokud bude čítač přednastaven na hodnotu, kterou vypočteme jako rozdíl nejvyššího možného stavu čítače a stavu, kterého čítač dosáhne za požadovanou jednotku času, tj.  $65536 - 31250 = 34286$ , bude docházet v tomto případě každou sekundu k přetečení časovače, a v případě povolení příslušného přerušení i k jeho obsluze [3]. Při takovémto zacházení s registry čítačů je nutné si uvědomit, že samotný zápis konkrétní hodnoty trvá procesoru určitou dobu a tento proces je nutné při každé obsluze přerušení opakovat. Odměřený požadovaný čas tedy nemusí být (a zpravidla také nebude) úplně přesný a může se mírně lišit od požadované skutečnosti. Tento čas se také může lišit podle použitého jazyka pro program procesoru.

V úlohách uvedených v této práci není dodržení zcela přesného času nikterak fatální a měření času pro účely výukové stavebnice bude dostatečně přesné.

Další využitelnou funkcionalitou čítačů/časovačů je generování pulzně šířkové modulace.

### **2.2.3 Sériové kanály**

Podpora sériových rozhraní je se u mikroprocesorů Atmel AVR různá a liší se podle konkrétních typů. Integrované bývají následující sériová rozhraní: SPI, USART a TWI. Rozhraní SPI bývá přítomno vždy, včetně rozměrově malých mikroprocesorů „ATTiny“, a to z toho důvodu, že je toto rozhraní používané pro programování jednočipového počítače.

SPI (Serial Programming Interface) je rozšířené sériové rozhraní, používané především k programování mikroprocesorů a ke komunikaci s dalšími elektronickými prvky, které integrují rozhraní SPI. Velmi často se jedná o teplotní čidla, multiplexory a demultiplexory, LED řadiče, AD a DA převodníky, v neposlední řadě také paměťové moduly téměř všech typů (SRAM, FLASH, EEPROM apod.) a operační zesilovače.

SPI používá pro komunikaci 4 signály, jejichž seznam a stručný popis uvádí tabulka č. 3.



Tabulka č. 3, popis signálů sběrnice SPI

<b>MISO</b>	<b>Master In – Slave Out</b>	<b>Příchozí data od podřízených zařízení</b>
MOSI	Master Out – Slave In	Odchozí data k podřízeným zařízením
SCK	Serial Clock	Hodinový signál generovaný masterem
$\overline{SS}$ ( $\overline{CS}$ )	Slave Select (Chip Select)	Někdy též „Chip Select“. Slouží k určení zařízení pro příjem dat konkrétním zařízením. v jednoduchých aplikacích se často vypouští.

Rozhraní USART je univerzální synchronní a asynchronní sériové rozhraní, které je velmi často integrováno do různých jednočipových počítačů díky svojí flexibilitě a možnosti komunikace různými standardy a rychlostmi za použití pouze jednoho technického řešení.

Jednočipové počítače Atmel integrují toto rozhraní s plnou podporou všech eventuálně potřebných prvků. Jmenovitě například generování sudé nebo liché parity, její hardwarová kontrola, podpora různě dlouhých přenášených rámců, generování přerušeni při příjmu nebo odeslání dat atd.

Z pohledu hardwaru mikroprocesoru jsou, co se týče rozhraní USART, vyvedeny pouze piny TXD (vysílání) a RXD (příjem). Je vhodné uvést, že vstupní a výstupní úrovně přenášeného signálu korespondují s napájecím napětím (0 V a 5 V), a proto při potřebě úrovní odlišných je nezbytné použít převodník úrovní. Příkladem budiž sběrnice RS-232, jejíž úrovně pro příjem a vysílání dat jsou  $\pm 15$  V.

Převodníky úrovní jsou zcela běžným elektronickým sortimentem a jsou spjaty s konvertorem pro konkrétní typ použité sběrnice. Typické je jejich použití pro spojení USART se sběrnici RS-232, RS-485, RS-422. Rovněž konvertory rozhraní USART na sběrnici USB jsou v současné době zcela běžné. S jejich použitím rovněž odpadá nutnost používat převodníky úrovní, neboť USB používá stejné napěťové úrovně jako jednočipové počítače tedy, 0 V a 5V.

Pro kompletnost uvedme, že výše uvedené sběrnice jsou sběrnice v pravém slova smyslu, ačkoliv se tak velice často nepoužívají. Částečně také proto, že vysílání právě jednoho člena sběrnice musí zajistit komunikační protokol. Ten však není součástí specifikací těchto sběrnic [7].

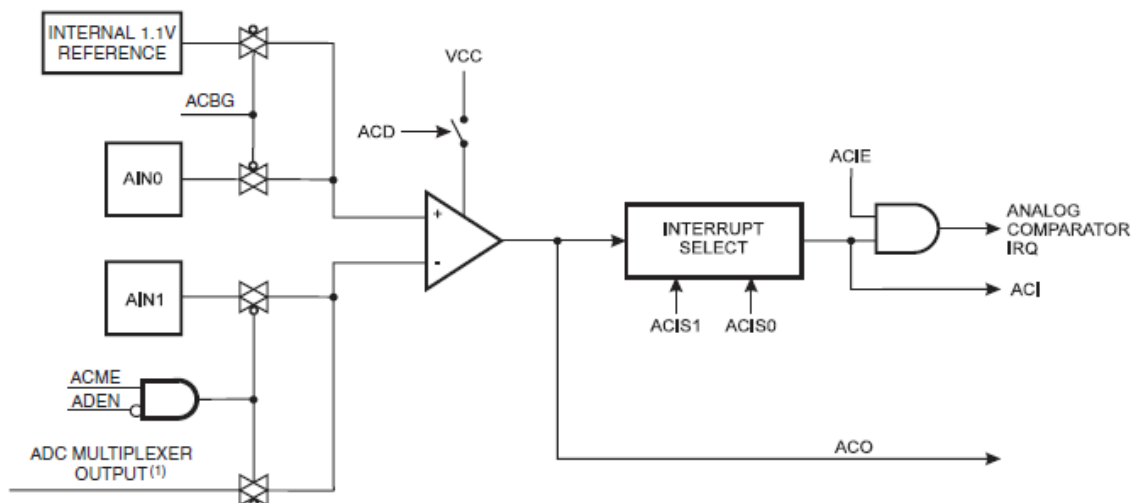
Posledním integrovaným rozhraním je TWI (Two Wire Interface), což je dvou vodičové sériové rozhraní. Pro komunikaci používá dva signálové vodiče SDA (Serial Data Line) a SCK (Serial Clock). Zařízení připojená k této sběrnici mohou vystupovat jako řídicí i řízená (master/slave). K adresování používá 7 bitový adresový prostor, což umožňuje připojení až 128 zařízení. Jeho hlavní předností je kompatibilita s velmi rozšířenou sběrnici I<sup>2</sup>C [8].

Rozhraní TWI nemusí být u jednočipových počítačů Atmel vždy integrované. Mikroprocesory jsou však dostatečně výkonné k jeho softwarové emulaci. Podpora této „softwarové“ verze sběrnice I<sup>2</sup>C je zajištěna u většiny vyšších programovacích jazyků.

Komunikační porty všech zmíněných rozhraní jsou přidružené k běžným vstupně-výstupním portům jednočipového počítače. Pokud je chceme použít právě ke komunikaci, je nutné toto nastavit v konfiguračních registrech mikroprocesoru.

## 2.2.4 Analogový komparátor

Analogovým komparátorem je vybavena většina jednočipových počítačů Atmel AVR. Jedná se o velmi jednoduché zařízení porovnávající úroveň vstupního napětí na jeho invertujícím a neinvertujícím vstupu. Celé schéma je znázorněno na obrázku č. 5.



Obr. č. 5, schéma analogového komparátoru, převzato z [5]

Komparátor porovnává napětí vstupů AIN0 a AIN1, přičemž AIN0 je neinvertující vstup (+) komparátoru a AIN1 je vstup invertující (-) [9].

Možné stavy výstupu komparátoru v závislosti na vstupních napětích uvádí tabulka č. 4.

Tabulka č. 4, výstup analogového komparátoru

Vstup	Výstup (hodnota ACO bitu v registru ACSR)
$U_{AIN0} < U_{AIN1}$	Log. 0
$U_{AIN0} \geq U_{AIN1}$	Log. 1

Důležitou schopností komparátoru je generování přerušení při různých událostech na výstupu komparátoru, možnost zachycení stavu čítače/časovače 1, nebo připojení interní napěťové reference k neinvertujícímu vstupu komparátoru.

### 2.2.5 Analogově digitální převodník

Některé jednočipové počítače Atmel AVR v sobě integrují analogově digitální převodníky. Ty jsou svou konstrukcí 10 bitové, s postupnou aproximací. Jejich vstupy jsou opět spřažené se vstupně výstupním portem – zpravidla s portem A. Analogově digitální převodníky bývají velmi často 8mi kanálové se vstupem do multiplexoru (což vychází z konstrukce aproximačních převodníků) s externími vstupy napěťové reference. Vstupní napětí může nabývat hodnot od 0 V do napájecího napětí. Rreferenční napětí je připojeno na pin AREF, nebo je volena interní napěťová reference 2,56 V.

Převodníky mají pracovní frekvenci od 50 do 200 kHz a z tohoto důvodu je přiváděný hodinový signál dělen předděličkou tak, aby bylo dosaženo požadované frekvence. Předdělička dělí frekvenci externího oscilátoru hodnotami 2, 4, 8, 16, 32, 64 a 128 [3].

Integrované A/D převodníky mají určité zkreslení a integrální nelinearitu. Udává se zkreslení  $\pm 2$  LSB a integrální nelinearita je 0,5 LSB. Převodníky po dokončení převodu mohou generovat přerušení a disponují obvodem „Sample / Hold“, který zaručuje, že konvertované napětí je po dobu konverze drženo na konstantní úrovni [1].

## 2.3 Bootloader

S rostoucí velikostí pamětí jednočipových počítačů Atmel AVR se objevují tzv. „bootloadery“ – česky zavaděče. Jsou to specializované programy, které jsou naprogramovány do příslušné části paměti. Přímá podpora zavaděčů je ze strany výrobce zavedena pouze v čípech typu ATmega nebo Xmega. Přímou podporou je myšlena právě přítomnost zmíněné vyhrazené paměti.

Funkce zavaděčů, která je v podstatě totožná se zavaděči v počítačích PC, je následující: trvale nahráný program ve speciální paměti (zavaděč) slouží ke snadnému načtení nového uživatelského programu a jeho nahrání do běžné programové paměti jednočipového počítače.

Zavaděče umí obsluhovat komunikační periférie procesoru, především sériové sběrnice. S jejich pomocí je také možné načíst uživatelský program a nahrát ho do již zmíněné programové paměti. Nejběžněji je možné se setkat se zavaděči, které podporují komunikaci po sériových sběrnících RS 232, nebo RS 485. Je však možné použít i I<sup>2</sup>C, Ethernet, SPI, atd. Po resetu následuje znovu spuštění zavaděče, kontrola komunikační periférie na přítomnost nového programu a spuštění uživatelského programu.

Programování samotného zavaděče je nutné provést standardním způsobem, tj. paralelním nebo sériovým programátorem, za současného nastavení pojistek – v dokumentaci nazývaných „fuses“. Jedná se především o správné nastavení reset vektoru. Tímto termínem je označována adresa, odkud mikroprocesor začíná načítat instrukce pro svoji práci. Běžným nastavením je samozřejmě první stránka paměti - tedy 0x0000. Paměť pro zavaděč se nachází na konci adresového prostoru.

Pokud má po naprogramování zavaděče ihned po startu dojít k jeho spuštění, je nutné nastavit při programování i adresu stránky paměti, kde se nachází. Adresa reset vektoru se tedy může lišit podle typu použitého jednočipového počítače – v závislosti na velikosti jeho paměti a velikosti vyhrazeného prostoru pro zavaděč.

Zavaděče jsou, z hlediska konstrukce mikroprocesoru, vázány na přítomnost instrukcí SPM (Store Program Memory) a LMP (Load Program Memory), nikoliv na velikost paměti ani na přítomnost speciální paměti. Lze najít projekty, které se zabývají vývojem zavaděčů i pro procesory ATTiny [10].

### **3 Stručný popis programovacího jazyka Bascom AVR**

Programovací jazyk Bascom je vyšší programovací jazyk, vyvinutý v 90. letech za účelem programování jednočipových počítačů Atmel. Konkrétně tehdy moderních Atmel 8051. Definování tohoto jazyka mělo za cíl vytvořit jednoduše použitelný programovací jazyk k programování jednočipových počítačů bez znalosti jazyka Assembler s možností konfigurace a využívání dostupných periférií procesoru, snadné nastavení vnitřních registrů a obsluha přerušení. Jeho výhodou jsou předem definované (a velice často používané) operace, cykly, práce s proměnnými různých velikostí, rozhodování, funkce pro práci s textem, předdefinované hardwarové operace s diskretními součástkami, definice vlastních funkcí a vlastních procedur.

Jazyk Bascom je inspirován vyšším programovacím jazykem Basic. Jeho příkazy, syntaxe, ale i parenteze a další použité struktury v jazyce velmi připomínají Visual Basic 6 [11]. Jazyk Bascom po Basicu přebírá i velmi vysokou míru abstrakce, což jej činí velmi snadno pochopitelným a vhodným i pro úplné začátečníky – jazyk Basic byl koneckonců vyvinut přesně za tímto účelem. Jeho použití pro náročné aplikace se tím nijak nevylučuje.

Jazyk Bascom dále disponuje vlastním vývojovým prostředím s kompilátorem. Ačkoliv je k dispozici zdarma, omezuje velikost výsledného kompilovaného programu na 4 kB. Při potřebě vývoje programů přesahujících tuto velikost je nutné vývojové prostředí zakoupit. Úlohy uvedené v této diplomové práci a programy vytvořené žáky během jejich zpracování velikost 4 kB nepřekročí. Z tohoto pohledu není pro základní seznámení s procesory Atmel nutná žádná investice.

#### **3.1 Základy jazyka**

Jednotlivé znaky použitelné v jazyce Bascom tvoří dohromady popisky částí kódu („labels“), klíčová slova, proměnné a operátory. Ty jsou slučovány do výrazů, které tvoří program.

Bascom používá znaky abecedy, číslice a speciální znaky a není citlivý na velikost písmen abecedy. Některé znaky však mají speciální význam. Shrnuje je tabulka č. 5.

Tabulka č. 5, znaky používané Bascomem, upraveno, převzato z [12]

Znak	Popis
ENTER	Ukončení řádky
	Prázdné nebo mezera
'	Jednoduchá uvozovka - apostrof
*	Hvězdička (symbol násobení)
+	Symbol součtu
,	Čárka
-	Symbol rozdílu
.	Desetinná tečka
/	Lomítko (symbol dělení); je zpracován jako \
:	Dvojtečka
"	Dvojitě uvozovky
;	Středník
<	Menší než
=	Symbol ekvivalence, přiřazení nebo relační operátor
>	Větší než
\	Zpětné lomítko (celočíslené dělení)
^	Exponent

Výrazy jazyka jsou dvojí: vykonatelné a nevykonatelné. Vykonatelný výraz říká programu co dělat v dalším kroku a souvisí přímo s vykonáním instrukcí procesorem. Nevykonatelné výrazy jsou potřebné především pro kompilátor. Jako příklad uveďme dimenzování proměnné příkazem **Dim** nebo výraz **\$crystal**, kterým informujeme kompilátor o použité taktovací frekvenci.

### 3.2 Datové typy a proměnné

Tabulka č. 6, datové typy jazyka Bascom [12]

Typ	Velikost	Rozsah
bit	1 bit	0,1
boolean	1 bit	0,1 (nelze použít true a false)
byte	8 bit	0 až 255
integer	2 Bajty	-32 768 až 32 767
word	2 Bajty	0 až 65 535
dword	4 Bajty	0 až 4294967295
long	4 Bajty	-2 147 483 648 až 2 147 483 647
single	4 Bajty	$1,5 \times 10^{-45}$ až $3,4 \times 10^{38}$
double	8 Bajtů	$5,0 \times 10^{-324}$ až $1,7 \times 10^{308}$
string	max 254 x 1 Bajt	každý znak jeden bajt + „nulový“ bajt

Datový typ proměnné určuje typ v ní uložených dat. Určuje zároveň i velikost proměnnou obsazené paměti a způsob interpretace této hodnoty. Datové typy užívané jazykem Bascom shrnuje

tabulka č. 6.

Proměnná je objekt, který popisuje určitý úsek paměti, která obsahuje související informaci. Je vždy určena svým jménem a deklarována příkazem **Dim**. Název proměnné vždy začíná písmenem a nelze ji pojmenovat rezervovaným slovem (příkazem) Bascomu. Před jejím použitím je vyžadována deklarace a je možné používat i tzv. „fantom proměnné“ [13], které plní funkci všeobecně známých pointerů.

### 3.3 Výrazy a operátory

Bascom disponuje aritmetickými, relačními a logickými operátory. Tabulka 7 shrnuje jejich zápis a význam.

Tabulka č. 7, operátory Bascomu

Operátor	Význam	Příklad použití
<b>Aritmetické</b>		
+	Sčítání	soucet = A + B
-	Odčítání	rozdíl = A - B
*	Násobení	součin = A * B
/	Dělení	podíl = A / B
\	Celočíselné dělení	podíl = A \ B
<b>MOD</b>	Modulo	zbytek = A MOD B
<b>Relační</b>		
=	Rovnost	If A >= B Then ... ElseIf ... EndIf
<>	Nerovnost	
<	Menší než	
>	Větší než	
<=	Menší než nebo rovno	
>=	Větší než nebo rovno	
<b>Logické</b>		
<b>OR</b>	Disjunkce	Loop until A = 2 OR B < 10
<b>AND</b>	Konjunkce	A = B AND &b00100000
<b>NOT</b>	Negace	B = NOT A
<b>XOR</b>	Exklusivní disjunkce	A = B XOR C

V programovacím jazyce Bascom není možné v jednom zápisu provést několik operací. Například zápis  $W = C + D + F$  nebo  $R = S \text{ XOR } T \text{ AND } U$  způsobí chybu v překladu a tuto složenou operaci bude nutné rozepsat na jednotlivé dílčí kroky.

### 3.4 Pedagogické aspekty jazyka Bascom a programování jednočipových počítačů Atmel

Komplexní popis programovacího jazyka Bascom není cílem této práce a jeho základy budou vysvětleny na samostatných úlohách. Práce s ním má však určité odlišnosti oproti běžným programovacím jazykům, které plynou z principu práce samotných programovaných mikroprocesorů. Můžeme je vyjmenovat:

- Práce v nekonečné smyčce – program „nemá konec“, neustále se opakuje do vypnutí mikroprocesoru nebo do nahrání nového programu.
- Častá práce s hodnotami ve dvojkové nebo šestnáctkové číselné soustavě.
- S tím související problémy dvojkové soustavy – desetinná čísla.
- Zaokrouhlování (Bascom zaokrouhluje podle doporučení organizace IEEE) [14].
- Pokud nepřipojíme další hardwarové komponenty, nebude vidět výstup.
- Je potřeba, alespoň okrajově, chápat funkci připojeného hardwaru.
- Programové prostředí je v anglickém jazyce, stejně tak příkazy.
- Některé příkazy jsou matoucí v souvislosti se zaběhlou terminologií.

Na tyto zvláštnosti je jistě nutné upozornit, ne všechny je ale potřeba probrat teoreticky s předstihem. Vhodným postupem může být navodit určitou problémovou situaci, kterou nebude možné vyřešit zaběhlými postupy a principy. Právě v tomto okamžiku, kdy se žáci budou sami zajímat o „správný postup“, bude vhodné danou problematiku vyložit a to s ohledem na věk a pokročilost vyučované skupiny.

Pochopení reprezentace čísel ve dvojkové soustavě, její návaznost na elektroniku, pojmy bit a bajt a neustálé opakování vykonávaného programu ve smyčce a základy algoritmů jsou pro samotný úvod do programování jednočipových počítačů nesmírně důležité.

#### 3.4.1 Čas a taktovací frekvence

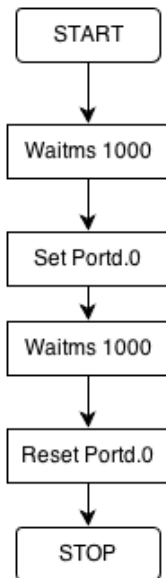
Stejně tak je důležité sdělit, jak rychle jednočipové počítače pracují. Jejich taktovací frekvence a tedy i výpočetní výkon je ve srovnání s běžnými osobními počítači opravdu malý. Začátečník si ale leckdy neuvědomí, že i pro něj komicky nízká frekvence – řekněme 1MHz, znamená vykonání 1 instrukce za dobu 1 $\mu$ s. To je přibližně 100 000x kratší čas, než který je člověk schopen postřehnout. Pokud bychom nechali



mikroprocesor např. rozsvítit LED diodu a ihned poté jí vypnutí, žáci neuvidí žádný výsledek. Platí tedy, že to, co je pro člověka pouhý okamžik, může být pro počítač opravdu dlouhá doba, za kterou je schopen vykonat mnoho příkazů.

Rychlost provádění příkazů se zdá být dobrým startem pro vysvětlení zmiňované cyklické práce.

### 3.4.2 Práce v cyklu



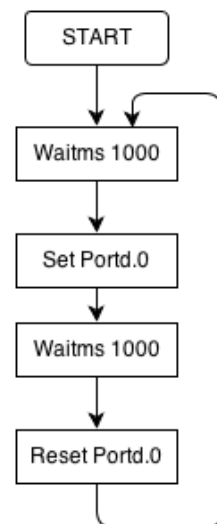
Obr. č. 6, algoritmus bez cyklu.

Zkusme si představit, že náš program v mikroprocesoru bude postupovat podle algoritmu uvedeného na obrázku č. 6. Předpokladem je připojená dioda LED na pinu 0 portu D mikroprocesoru. Po spuštění náš mikroprocesor přivede napětí na příslušný pin, tím rozsvítí diodu LED. Nechá ji zapnutou po dobu 1 sekundy a následně příslušný pin uzemní, čímž LED dioda zhasne. Další příkazy nemá. LED dioda blikne znovu až po resetu mikroprocesoru. K takové funkci je ale zbytečné používat jednočipové počítače.

Pokud ale necháme jednočipový počítač po skončení poslední instrukce pokračovat zpět od začátku, získáme již nějakou funkci.

Algoritmus se změní podle obrázku č. 7 a mikroprocesor bude plnit funkci přerušovače.

Bude ji plnit i po jeho resetu.



Obr. č. 7, práce programu v cyklu

### 3.4.3 Desetinná čísla

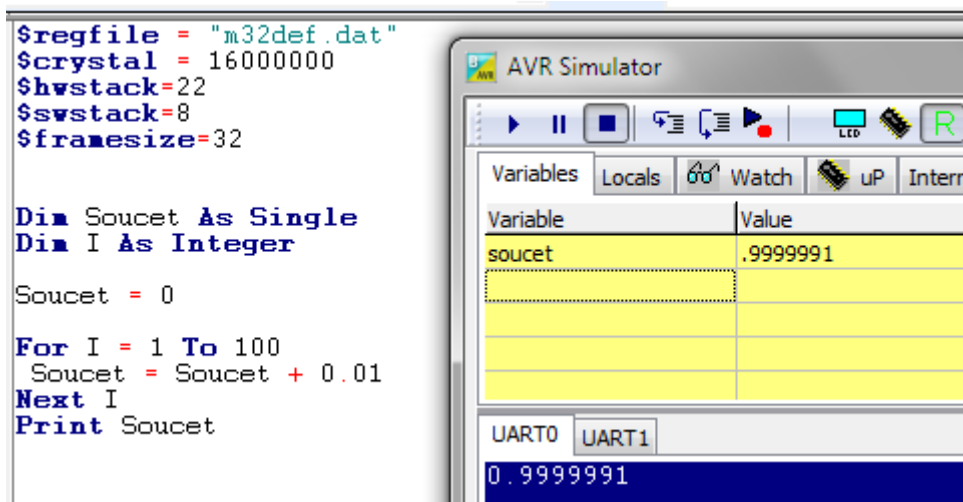
Některá čísla je nemožné přesně vyjádřit ve dvojkové soustavě – tedy pomocí mocnin čísla 2. Jedná se samozřejmě o čísla desetinná, protože jejich hodnotu nelze vždy vyjádřit součtem záporných mocnin čísla 2.

Postup převodu je všeobecně známý, ale uveďme příklad desetinného čísla, které není možné přesně vyjádřit ve dvojkové soustavě – stane se totiž periodickým.

$$0,4_{10} = 0,0\overline{1110}_2 \quad (2)$$

Tyto problémy mají ostatně všechny číselné soustavy, ale v případě programování aplikací pro běžný osobní počítač obsahují programovací jazyky algoritmy a knihovny, které tyto nepříjemné situace řeší a to většinou podle doporučení IEEE 754 [14].

V případě programování aplikací pro jednočipový počítač takové pohodlí nemáme. Proto je vhodné se těmto situacím vyhnout. Jak by mohla vypadat situace, která nás přivede k nepředpokládanému chování programu, uvádí obrázek č. 8.



Obr. č. 8, výpis hodnoty na simulátoru, vývojové prostředí Bascom

Součet by měl samozřejmě být roven hodnotě 1, ale ona malá chyba je způsobena nemožností převést bez chyby číslo 0,01 do soustavy s bází 2. Stejně tak i konstrukce uvedené na obrázku č. 9 by neměly být používány, tak jak obrázek demonstruje. Je možné použít spíše určité rozpětí hodnot, než porovnávat zda jsou si

```

Dim Cislo_a As Single
Dim Cislo_b As Single

Cislo_a = 69.82
Cislo_b = 69.20 + 0.62

If Cislo_a = Cislo_b Then
Print "Čísla jsou si rovna."
End If

```

Obr. č. 9, chybné porovnání rovnosti, vývojové prostředí Bascom

rovny. V tomto případě program nikdy nevypíše informaci o rovnosti.

### 3.4.4 Možné problémy při zápisu hodnot do portu

Zápis hodnot na port jednočipového počítače s pomocí jazyka Bascom je přirozeně velmi jednoduchý a intuitivní.

Bohužel, díky syntaxi Bascomu, je zde několik nepříjemných okamžiků, kterých je nutné se vyvarovat. Jedná se o příkazy, které souvisí se zápisem hodnoty na port, případně konkrétní pin příslušného portu,

```

Config PORTB = Output
Config PORTA = Output

PORTB = 255
PORTA = 0

Reset PORTB.1
Set PORTA.5

Config PORTB = Input

PORTB.0 = 1
Print PINB.1
Print PINB
Print PORTB

```

nebo o načtení hodnoty z portu, pokud je port/pin mikroprocesoru nastaven jako vstupní. Obrázek č. 10 demonstruje problémovou situaci.

Program nastaví porty B a A mikroprocesoru jako výstupní, přičemž v portu B nastaví všechny jeho piny na hodnotu 1, v portu A na hodnotu 0 (myšlena je přirozeně log. 0 a log.1).

Obr. č. 10, problémové příkazy Bascomu

Následně příkazem `Reset PORTB.1` nastaví druhý pin portu B na hodnotu log. 0 (resetuje). Obdobně příkazem `Set PORTA.5` nastaví 6. pin portu A na hodnotu log. 1.

Zde nastává jistý problém při výuce, hlavně poté, co se žáci s Bascomem seznámí. Podobnost příkazů `Set PORTA.5` a `PORTA.5 = 1` vede k myšlence, že oba příkazy dělají totéž a je tedy možné je používat jako ekvivalentní. Pravdou je, že druhý zmíněný příkaz se používá pro zapnutí integrovaného, tzv. zdvihacího odporu. Tento příkaz má ale smysl pouze v případě, že je port nebo pin nastaven jako vstupní.

Obdobné problémy nastávají při pokusu o čtení hodnoty z portu. Žáci mylně považují port nebo pin za jedno totožné zařízení, kterému je jednou řečeno, zda je vstupní nebo výstupní. Mělo by se dbát na vysvětlení, že to, s čím pracují, jsou tzv. registry, jejichž hodnoty se promítají na fyzicky hmatatelné piny jednočipového počítače a obráceně – že přítomnost proudu na jednotlivých pinech se může promítnout do registru a že tyto registry jsou pokaždé jiné v závislosti na tom, zda hodnoty čteme nebo zapisujeme. Tabulka č. 8 shrnuje význam jednotlivých příkazů.

Tabulka č. 8, význam příkazů pro práci s portem

Příkaz	Popis	Směr portu
<code>PORTB = 255</code>	Připojí všechny piny portu B na +5 V	výstup
<code>Set PORTB.7</code>	Připojí 8. pin portu B na +5 V	výstup
<code>PORTB.0 = 1</code>	Zapne zdvihací rezistor pro pin 1 portu B	vstup
<code>Print PINB.1</code>	Vypíše hodnotu (0,1) načtenou z pinu 2 portu B	vstup
<code>Print PINB</code>	Vypíše hodnoty načtené z celého portu B	vstup
<code>Print PORTB</code>	Vypíše naposledy zapsané hodnoty do portu B	vstup/výstup

## 4 Návrh modulů stavebnice

Moduly stavebnice používané pro výuku programování jednočipových počítačů jsou vlastní koncepce s ohledem na jednoduchost použití a levnou výrobu s možností snadné výměny elektronických prvků i v neprofesionálních podmínkách.

Rovněž byl brán ohled na přehledné uspořádání modulů, které ulehčuje orientaci při propojování jednotlivých modulů a případně jednotlivých řídicích signálů. Moduly jsou z výroby potisknuty. Potisk je navržen tak, aby jasně a přehledně čísloval každý konektor nebo řídicí signál. Zároveň je v podobě potisku vždy uvedeno stručné vysvětlení významu jednotlivých kontaktů, čímž odpadá časté hledání zapojení v dokumentaci k elektronickým součástkám. Pokud se u použité kombinace součástek objevuje dvojí značení (např.: 0; 1; 2; 3... i A; B; C; D... ), je dbáno na to, aby toto rozdílné značení mělo stejný řád. Tedy například na kontaktu č. 0 nalezneme signál A. V opodstatněných případech existují ve stavebnici samozřejmě výjimky. Vysvětlení zapojení ve formě potisku je ale dostatečně názorné.

Stavebnici lze snadno doplňovat běžně dostupnými a levnými moduly, které se v poslední době vyrábí pro domácí a amatérské použití. Často jsou označovány jako DIY („Do It Yourself“) moduly.

### 4.1 Technické provedení

Moduly stavebnice jsou navrženy v programu pro návrh elektrotechnických projektů KiCAD. Ten umožňuje kompletní vytvoření návrhu v podobě elektrického schématu, výkresu desek plošných spojů a exportu podkladů pro výrobu. V případě modulů stavebnice se jedná o export souboru pro vrtání děr ve formátu Excellon, dále export obrazce plošného spoje, nepájivé masky, uživatelského popisu a obrysu desky ve formátu Gerber RS 274-X.

Moduly stavebnice jsou tvořeny deskou plošného spoje z běžného materiálu *FR4*, což je ohnivzdorná tkanina ze skelných vláken sycená epoxidovou pryskyřicí [15], konkrétně o síle 1,5 mm. Na straně spojů je nanесena nepájivá maska, na straně součástek je umístěn potisk bílé barvy. Síla měděné fólie je 18  $\mu\text{m}$  a plošný spoj se nachází pouze na jedné straně desky. Desky jsou ostříženy na výsledný požadovaný rozměr.

Součástky použité pro moduly jsou voleny v pouzdrech běžných rozměrů THT a DIL. Prvky v pouzdech DIL jsou usazeny v odpovídajících patičkách.

#### **4.1.1 Propojení modulů**

Propojení modulů stavebnice je voleno vlastní koncepcí a to především z důvodů snazší orientace v zapojení. Běžnou praxí je vést signálové vodiče na společném konektoru spolu s napájecími vodiči - tento systém používá například vývojová platforma Arduino.

Propojení modulů je možné s pomocí 8 žilových plochých kabelů s konektory IDC, které sdružují vodiče příslušné k jednomu portu jednočipového počítače. Vodiče pro napájení modulů stavebnice jsou samostatné, průběžné – tedy z jednoho modulu je možné napájet druhý. Samostatné moduly jsou chráněné diodou proti přepólování, jejich správné napájení signalizuje kontrolní dioda LED.

Osamocené řídicí signály obvodů na modulech jsou vyvedeny zvlášť na samostatné piny. K jejich propojení je možné použít běžné propojovací vodiče.

#### **4.1.2 Napájení**

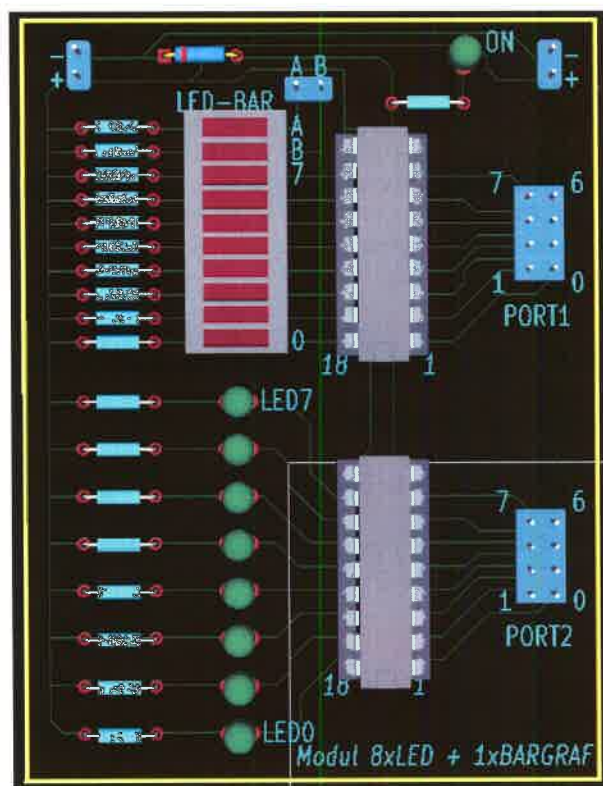
Pro napájení modulů stavebnice je potřebné stabilizované stejnosměrné napětí 5 V a případně i 12 V. Jeho zdroj není součástí návrhu a lze použít libovolný běžně dostupný bezpečný zdroj o příslušném napětí. Vzhledem k použití stavebnice při výuce je součástí návrhu obvod, který zajišťuje odolnost proti zkratu na obou napájecích napětích. Jeho provedení umožňuje montáž do přístrojové krabičky s vývodem napájení na běžně používané zdířky o průměru 4 mm. Jeho činnost signalizuje čtveřice diod LED, které uživatele informují o normálním provozu a zkratu. S přihlédnutím k proudovým nárokům jednotlivých modulů je maximální dodávaný proud nastaven na 500 mA.

Na vybraných modulech jsou k dispozici samostatné vývody napájení, které je možné použít pro dodatečné zapojení řídicích signálů bez použití jednočipového počítače. Je tak umožněno i snadné vyzkoušení funkce modulů.

V následujících kapitolách jsou stručně popsány jednotlivé moduly stavebnice.

Výkresy plošných spojů jsou zobrazeny v pohledu na vrstvu spoje.

## 4.2 Modul 8x LED + 1x Bargraf



Obr. č. 11, Modul 8xLED + Bargraf, 3D náhled

Modul s 8 LED diodami a 10 segmentovým bargrafem (jeho 3D náhled zobrazuje obrázek č. 11) je použit především pro první kroky při výuce programování a dále jako jeden z možných výstupních modulů.

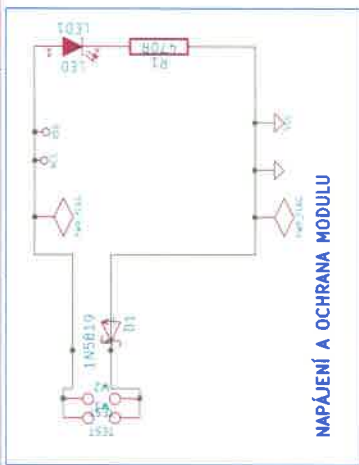
Porty 1 a 2 lze připojit k modulu s jednočipovým počítačem a nechat tak zobrazovat stavy na jeho výstupních portech. Jak již bylo uvedeno, pin 0 na portu ovládá přes budič diodu LED0 a po připojení k mikroprocesoru je ovládán nejméně významným bitem připojeného portu. Další LED diody jsou zapojeny obdobně.

Součástí modulu jsou obvody ULN2803, což jsou 8 kanálové Darlingtonové budiče, které budí jednotlivé LED. Bargraf obsahuje celkem 10 LED diod. Poslední dvě označené A a B jsou ovladatelné mimo standardní port a budič s pomocí odpovídajících vývodů A a B.

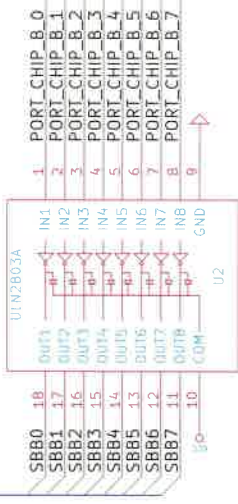
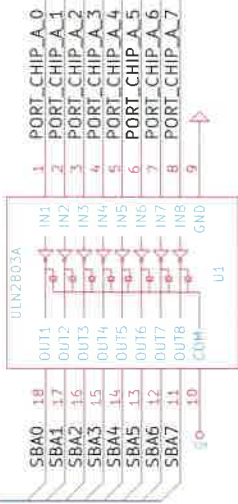
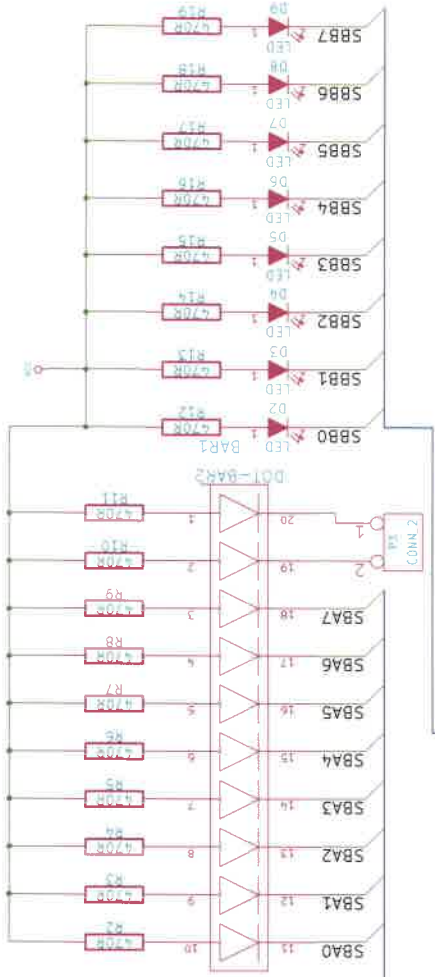
Ochrana proti přepólování je provedena Schottkyho diodou s nízkým úbytkem napětí. Rozpis součástek je uveden v tabulce č. 9. Elektrické schéma a výkres plošného spoje jsou uvedeny na následujících stranách. Osazovací plánec je uveden v příloze č. 1.

Tabulka č. 9, seznam součástek

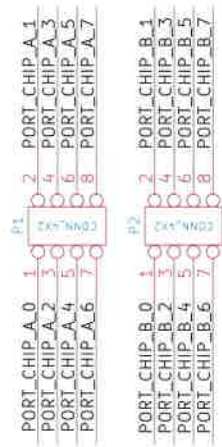
D1	1N5819
R1-R19	Rezistor 470R; 0,25 W
U1, U2	ULN2803A v patici DIL18
LED1, D2-D9	LED 3mm
DOT-BAR2	DC-10SRWA/ DC-10GWA
W1, W2, P3	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P2, P3	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm

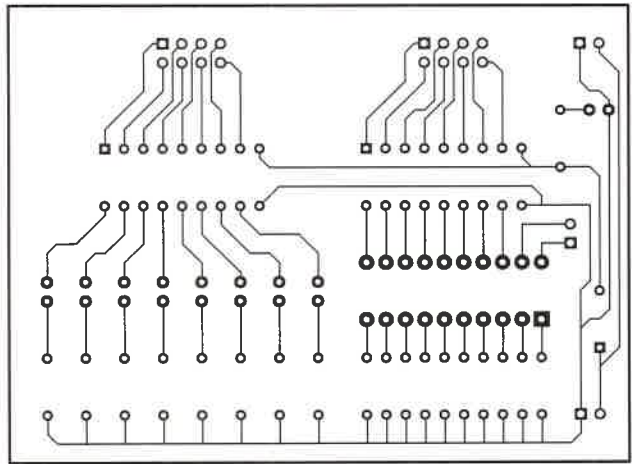


**NAPÁJENÍ A OCHRANA MODULU**



PORT\_CHIP\_A\_0, SBA0 --> LSB 1.Bajt  
 PORT\_CHIP\_B\_0, SBB0 --> MSB 1.Bajt  
 PORT\_CHIP\_B\_1, SBB1 --> LSB 2.Bajt  
 PORT\_CHIP\_B\_7, SBB7 --> MSB 2.Bajt





File: modul16\_LED\_v2.brd

Sheet: 1/1

Title: modul 8xLED + 4xBARGRAF

Size: A4 Date: 5 Jul 2013

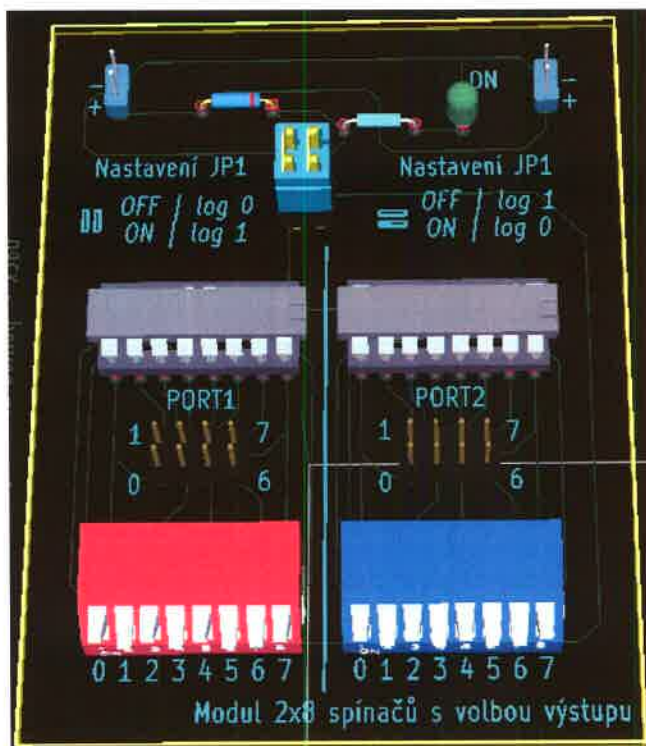
KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

Rev:

Id: 1/1



### 4.3 Modul 2x 8 spínačů s volbou výstupu



Obr. č. 12, Modul 2x8 spínačů, 3D náhled

Modul se spínači slouží jako výstupní modul, s jehož pomocí je možné volit na portech 1 a 2 mezi dvěma logickými hodnotami. Je používán především jako volič hodnoty pro vstupní porty jednočipového počítače. Je možné ho přímo připojit i k dalším modulům, a tak snadno demonstrovat jejich funkci i bez mikroprocesoru.

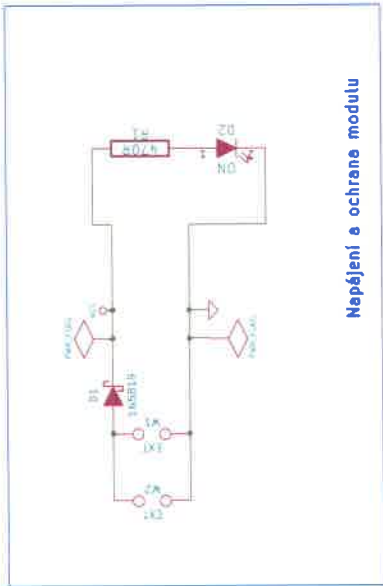
Jako spínací prvky jsou použity spínače DIP, čímž odpadá nutnost používat spínače s aretací anebo polovodičové řešení.

Součástí modulu je dvojice jumperů, která umožňuje volit smysl výstupního logického signálu: normální, kde poloha zapnuto odpovídá logické hodnotě 1 nebo invertovaný, kde tatáž poloha spínače odpovídá logické hodnotě 0. Seznam součástek uvádí tabulka č. 10.

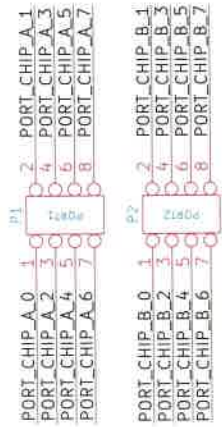
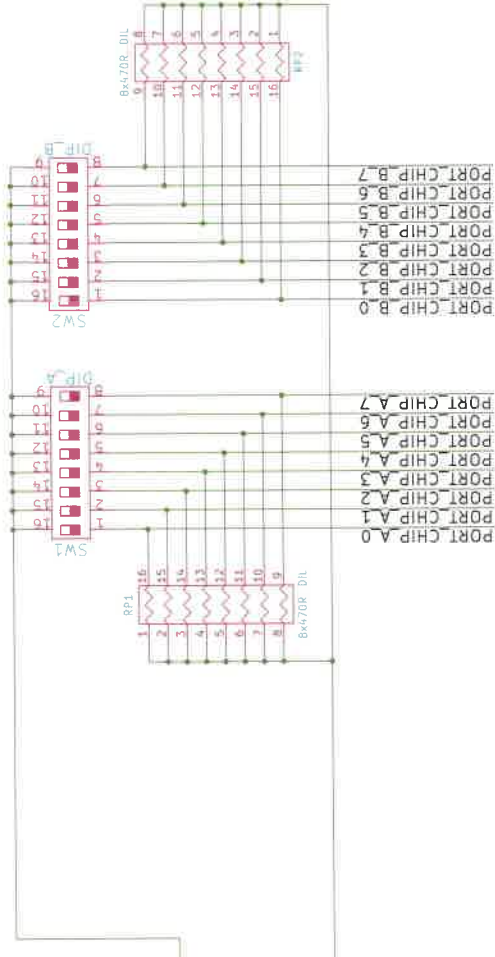
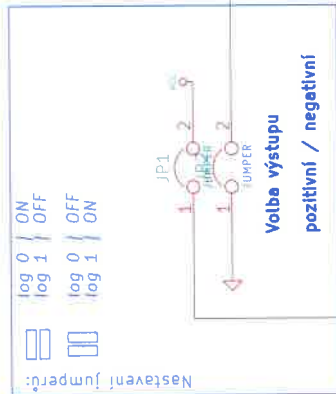
Tabulka č. 10, seznam součástek

D1	1N5819
D2	LED 3mm, zelená
RP1, RP2	8xrezistor 470R v pouzdru DIL16
R1	Rezistor 470R; 0,25 W
SW1, SW2	DIP-SWITCH: DP-08
JP1, JP2	Kolíková lišta, PIN:3, vidlice, THT, rozteč 2,54 mm
W1, W2	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P1, P2	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm

Elektrické schéma a výkres plošného spoje jsou na následujících stranách, osazovací plánec v příloze č. 2.



Napájení a ochrana modulu



Skvěbnice s jednocipý ATMEL

Vojtěch Bušek

File: modul\_16\_DIP.sch

Sheet: /

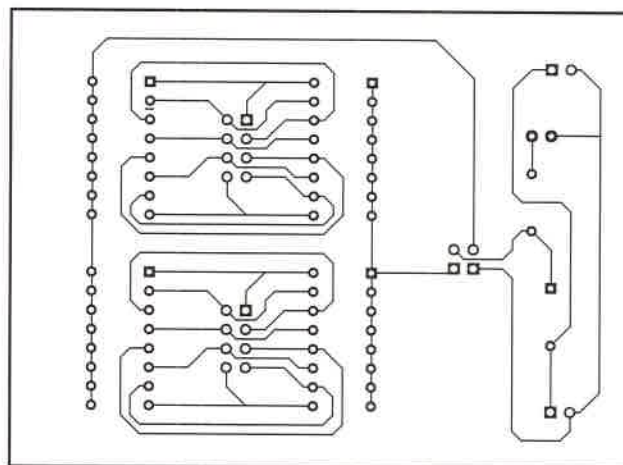
Title: Modul 2x8 spínačů s volbou výstupu

Size: A4 Date: 5. jul. 2013

Rev: 1

KiCad E.D.A. eeschema (2011-07-08 BZR 3044) - stable

Id: 1/1



Stavebnice s jednočipy ATMEL

**Vítěch Bušek**

File: modul\_16\_DIP.brd

Sheet: 1/1

**Title: Modul 2x8 spínačů s volbou výstupu**

Size: A4

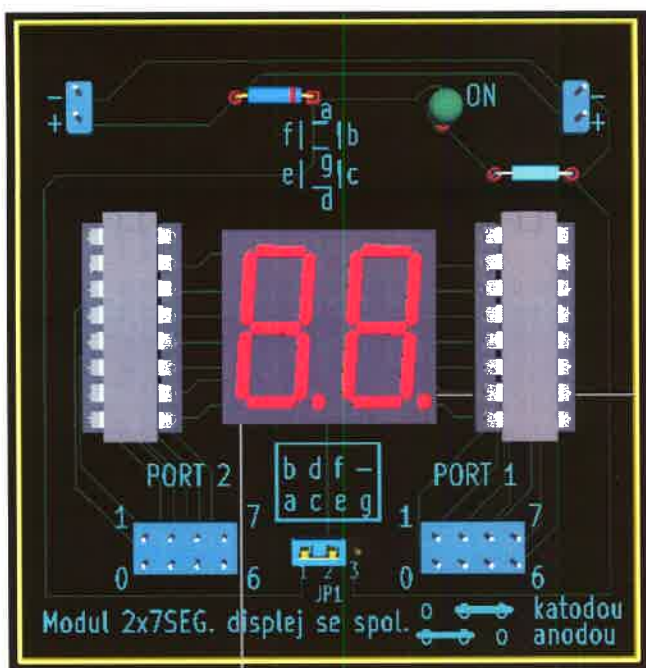
Date: 5 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

**Rev:**

Id: 1/1

## 4.4 Modul 2x sedmissegmentový displej



Obr. č. 13, Modul 2x7-segmentový displej, 3D náhled

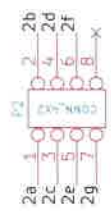
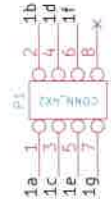
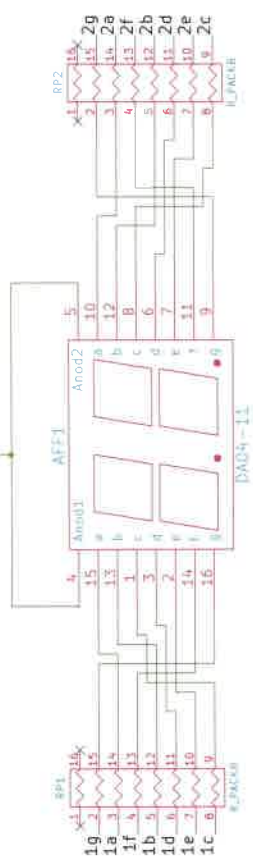
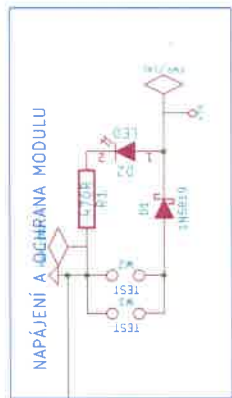
Na obrázku č. 13 je zobrazený návrh modulu s dvoučíslicovým segmentovým displejem, který je možné použít pro zobrazení číselného výstupu z mikroprocesoru.

Na modulu se nachází dva vstupní porty, každý z nich náleží k jedné číslici displeje. Rezistory jsou zvoleny v pouzdrech DIL16 zasazených do patič a jejich hodnotu lze snadno přizpůsobit napájecímu napětí. Součástí modulu není žádný budič LED diod.

Modul dále disponuje jumperem JP1, který umožňuje volbu segmentového displeje se společnou anodou nebo katodou. To předpokládá usazení displeje v kolíkové liště. Tento jumper také nemusí být osazen, ale pouze nahrazen příslušným propojem při osazování desky. Tím je umožněna výroba dvou různých modulů s využitím jednoho plošného spoje. Seznam součástek uvádí tabulka č. 11. Osazovací plánek je uveden v příloze č.3, schéma a výkres plošného spoje na následujících stranách.

Tabulka č. 11, seznam součástek

D1	1N5819
D2	LED 3mm, zelená
RP1, RP2	8x rezistor 470R v pouzdru DIL16
R1	Rezistor 470R; 0,25 W
AFF1	DA04-11 nebo DC04-11, barva dle výběru; 2x zástrčka kolíková, PIN8, THT, rozteč: 2,54 mm
JP1	Kolíková lišta, PIN:3, vidlice, THT, rozteč 2,54 mm
W1, W2	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P1, P2	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm



Stavebnice s jednočipy Atmel

Vojtěch Bušek

File: modul\_2x7SEGM\_spolA\_K.sch

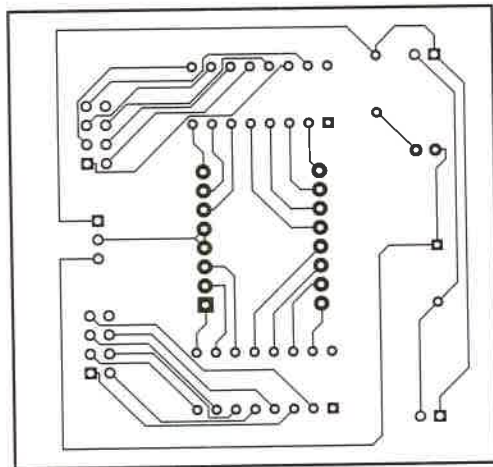
Sheet: /

Title: Modul 2x 7 SEGM. displej, volba společné elektrody

Size: A4 Date: 5 Jul 2013

KiCad E.D.A. eschema (2011-07-08 BZR 3044)-stable

Rev: 1/1



Stavebnice s jednočipý ATME1

**Vojtěch Bušek**

File: modul\_2x7SEGM\_spolA\_K.brd

Sheet: 1 / 1

**Title: Modul 2x 7 SEGM. displej, volba společné elektrody**

Size: A4

Date: 5 Jul 2013

**Rev:**

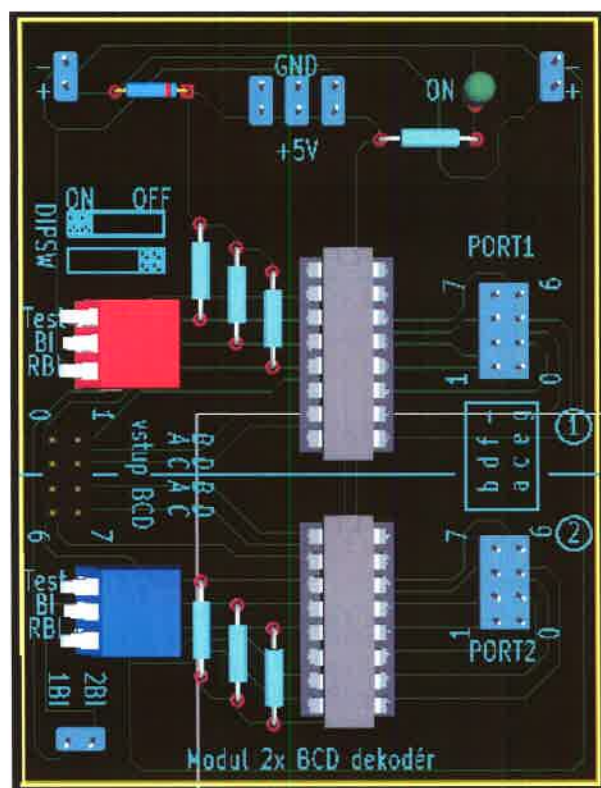
KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

5

4



## 4.5 Modul 2x BCD dekodér



Obr. č. 14, Modul 2x BCD dekodér, 3D náhled

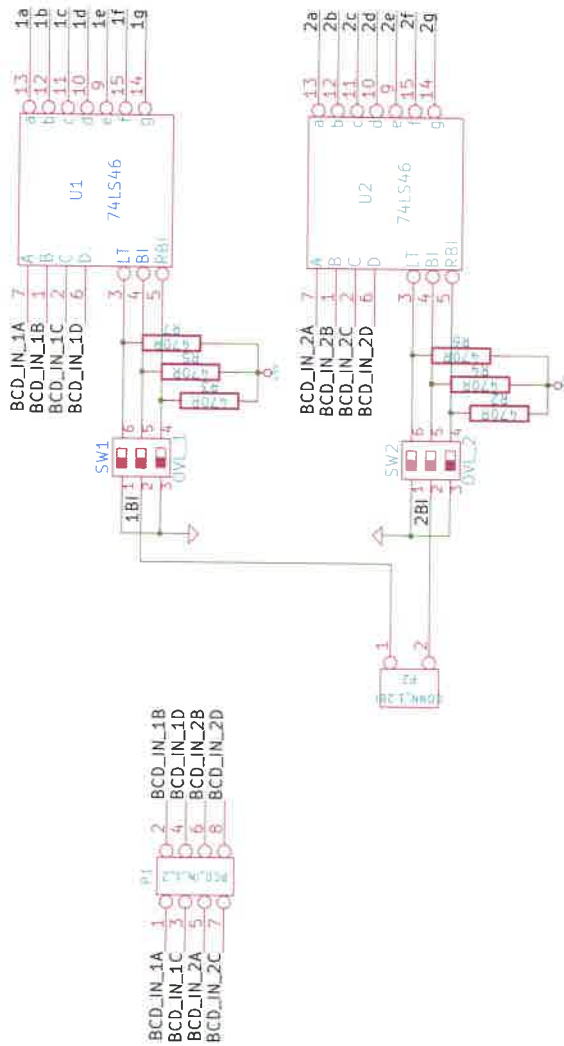
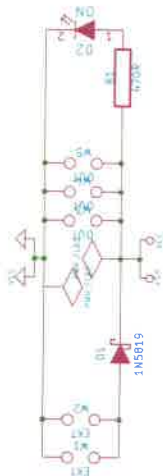
Modul stavebnice s BCD dekodéry (jeho 3D náhled je uveden na obrázku č. 14) je určen především ke spojení se segmentovými displeji. Uplatňuje se v úlohách, které mají ukázat význam kódování (úspora dat).

Samotný modul obsahuje dva dekodéry BCD a k nim příslušné porty 1 a 2. Dále se na modulu nachází vstupní port se signály A až D pro jednotlivé dekodéry. Modul lze osadit integrovaným obvodem 7447, nebo příbuznými typy se stejným rozložením vývodů: 7446, 74247. Lze použít i obvod 7449 v případě potřeby řízení displeje se společnou katodou.

Všechny zmíněné obvody disponují třemi doplňkovými funkcemi: test všech segmentů, vypnutí displeje a vypnutí zobrazení číslice 0. Tyto funkce lze manuálně aktivovat zapnutím příslušného spínače. Funkční vstupy pro vypnutí displeje („Blanking Input“) jsou vyvedeny na samostatný konektor pro další rozšíření funkce například PWM řízení jasu. Tabulka č. 12 udává seznam součástek. Osazovací plán je uveden v příloze č. 4.

Tabulka č. 12, seznam součástek

D1	1N5819
D2	LED 3mm, zelená
U1, U2	74LS47 nebo kompatibilní
R1	Rezistor 470R; 0,25 W
R6-R7	Rezistor 1k0; 0,25 W
SW1, SW2	DIP-SWITCH: DP-03
W1-W5, P2	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P1, P3, P4	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm



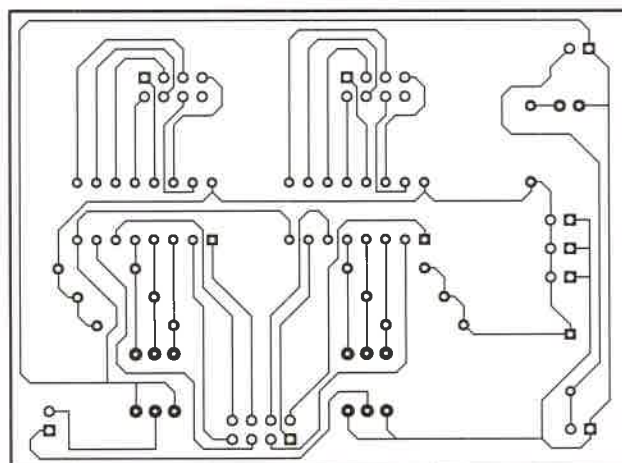
(BI) - Blanking Input (v L vypne všechny segmenty)  
 (LT) - Lamp Test (v L zapne všechny segmenty)  
 (RBI) - Ripple-Blanking Input (v L vypne zobrazení nuly)  
 všechny vstupy aktivní v L



7446 možno nahradit za:  
 7447, 74246, 74247 – aktivní úroveň L  
 NEBO  
 nahradit za 7449 – aktivní úroveň H  
 Při osazení 7449 změna displeje na DC04–11

Stavebnice s jednotky Atmel
<b>Vojtěch Bušek</b>
File: modul_2x_BCD_dekoder.sch
Sheet: /
<b>Title: Modul 2x BCD dekoder</b>
Size: A4
Date: 6 Jul 2013
KiCad E.D.A. eschema (2011-07-08 BZR 3044)-stable
<b>Rev:</b>
Id: 1/1





back – pohled ze strany součástek

Stavebnice s jednočipy Atmel

**Vojtěch Bušek**

File: modul\_2x\_BCD\_dekoder.brd

Sheet: 1/1

**Title: Modul 2x BCD dekodér**

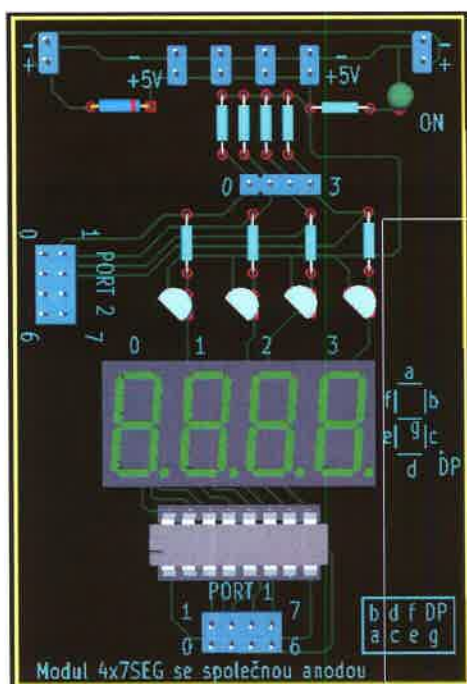
Size: A4 Date: 6 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) – stable

**Rev:**

Id: 1/1

## 4.6 Modul 4x sedmissegmentový displej se společnou anodou



Obr. č. 15, Modul 4x sedmissegmentový displej, 3D náhled

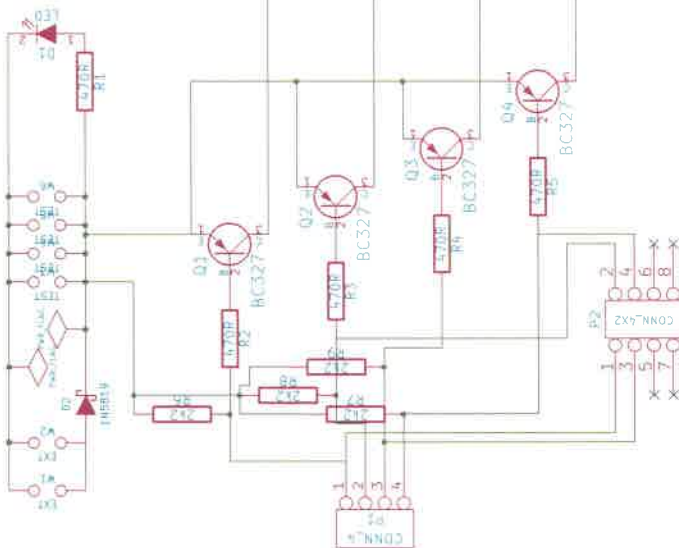
Modul stavebnice, uvedený na obrázku č. 15, obsahuje čtyřmístný sedmissegmentový displej, který má spojeny odpovídající katody jednotlivých segmentů číslic do portu 1. Anody pro každou číslici jsou spínány s pomocí čtyř tranzistorů PNP. Jejich báze jsou vyvedeny na port 2 a na samostatné piny 0 až 3. Displej je tedy multiplexovaný, což znamená, že je několikanásobně využito jedné přenosové cesty.

Rezistory pro jednotlivé segmenty jsou opět použity integrované v pouzdru DIL16, modul disponuje samostatnými vývody napájení, které lze použít pro manuální sepnutí tranzistorů a použít tak displej i bez multiplexování (patrně jako jednomístný). Modul neobsahuje žádný budič, proto je vhodné ho připojit k mikroprocesoru pomocí modulu s BCD dekodérem nebo s darlingtonovým polem.

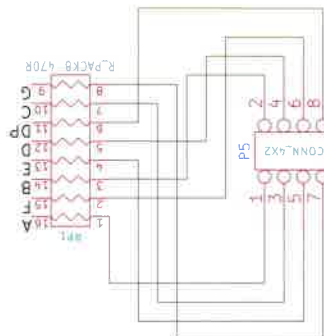
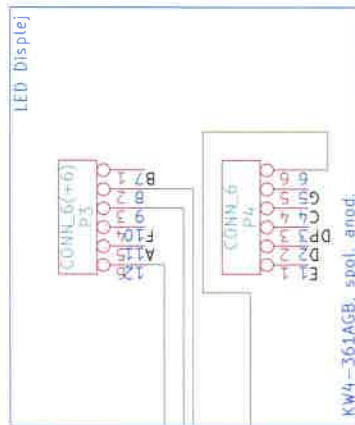
Modul je opět možné osadit i displejem se společnou katodou, což znamená i změnu tranzistorů PNP za komplementární NPN. Osazovací plánec je uveden v příloze č. 5, seznam součástí udává tabulka č. 13.

Tabulka č. 13, seznam součástí

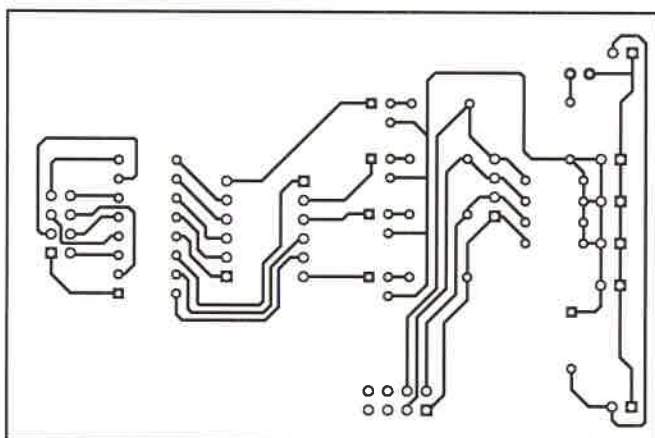
D1	LED 3mm, zelená
D2	1N5819
RP1	8xrezistor 470R v pouzdru DIL16
R1-R5	Rezistor 470R; 0,25 W
R6-R9	Rezistor 2k2; 0,25 W
P3, P4	Zástrčka kolíková, PIN6, THT, rozteč: 2,54 mm Displej KW4-361AGB
Q1-Q4	BC327
W1-W6	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P1	Kolíková lišta, PIN:4, vidlice, THT, rozteč 2,54 mm
P2, P5	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm



- DIG.1 (COM) = 12
- DIG.2 (COM) = 9
- DIG.3 (COM) = 8
- DIG.4 (COM) = 6
- A = 11
- B = 7
- C = 4
- D = 2
- E = 1
- F = 10
- G = 5
- DP = 3



Stavbenice s jednocipry Atmel  
**Voltěch Bušek**  
 File: moduL\_displej\_4x75EGM\_multiplex.sch  
 Sheet: /  
 Title: **Modul 4x sedmissegmentový displej se spol. anodou**  
 Size: A4 Date: 7 Jul 2013  
 Křičad E.D.A. - peschema (2011-07-08 BZR 3044) - stable  
 Rev: 1/1  
 Id: 1/1



Stavebnice s jednočipý Atmel

**Vojtěch Bušek**

File: modul\_display\_4x7SEG\_multplex.brd

Sheet: 1/1

**Title: Modul 4x sedmissegmentový displej se spol. anodou**

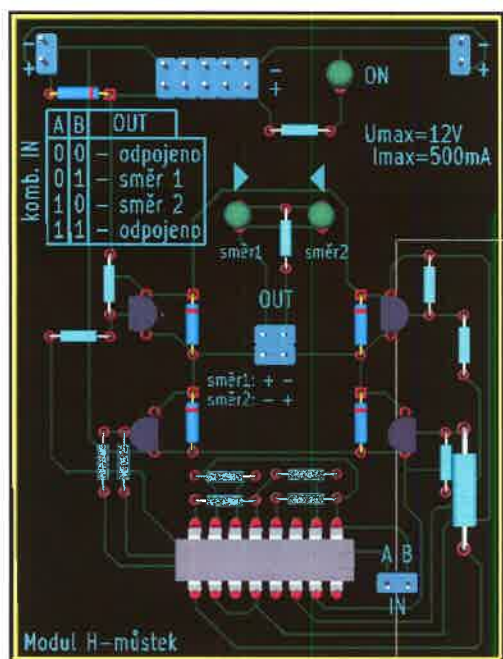
Size: A4

Date: 7 Jul 2013

**Rev:**

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

## 4.7 Modul izolovaný H můstek



Obr. č. 16, izolovaný H-můstek, 3D náhled

Na obrázku č. 16 je zobrazený 3D náhled na modul s izolovaným tranzistorovým můstkem H. Je určen pro aplikace, kde je vyžadována změna polaroty napětí na zátěži. Předpokládané použití je pro řízení stejnosměrného komutátorového motoru s odlišným napájecím napětím, které může dosahovat až 12 V, což je dáno především typem použitých tranzistorů a rezistorů v jejich obvodech. Spínání jednotlivých tranzistorů zajišťuje optočlen skrze signály A a B vstupního portu IN. Díky použití optočlenu je můstek odolný proti přivedení stejných

Tabulka č. 14, nastavení ovládacích signálů na vstup IN. Obvody optočlenu jsou přizpůsobeny k přímému řízení jednočipovým počítačem.

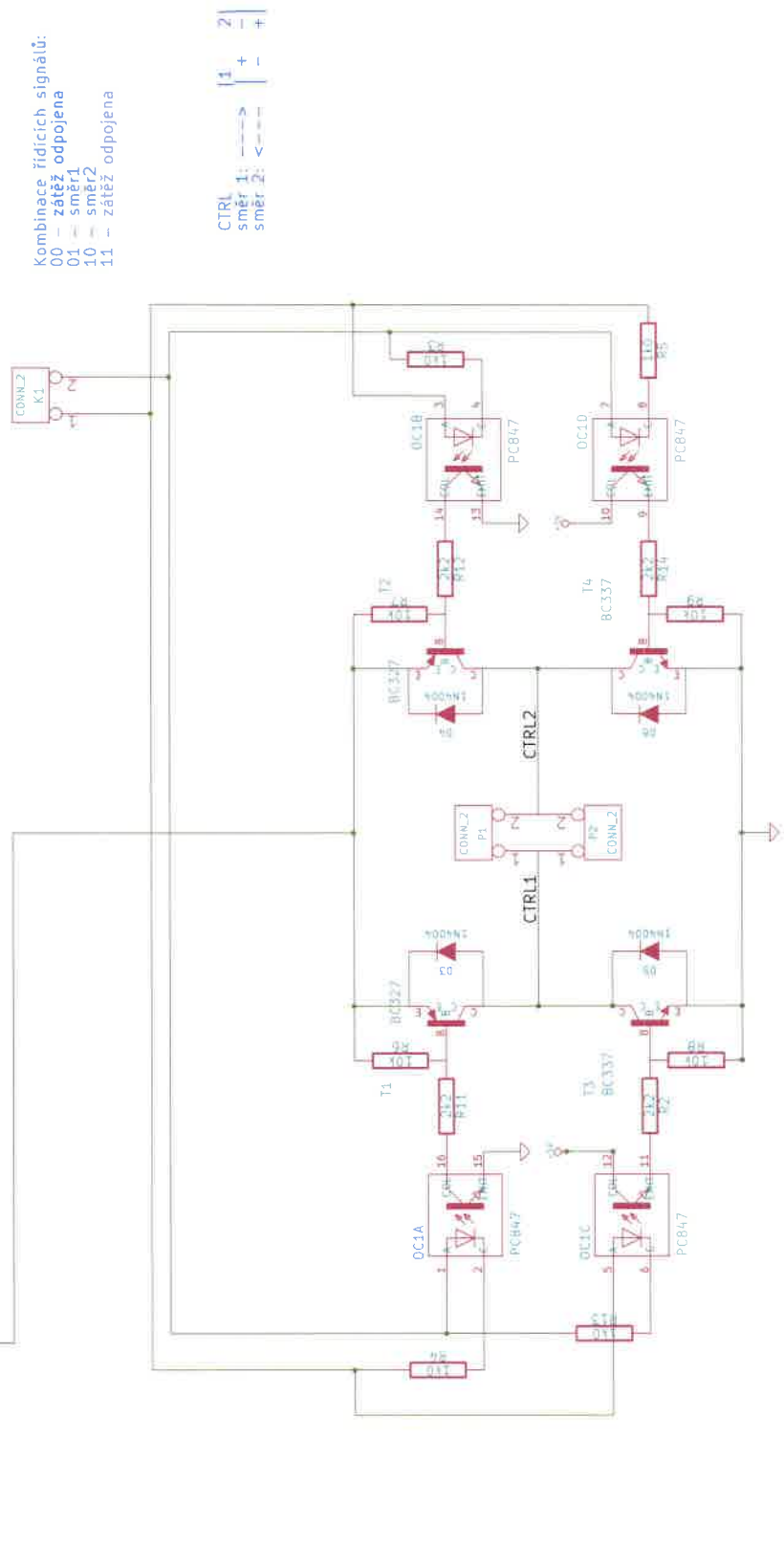
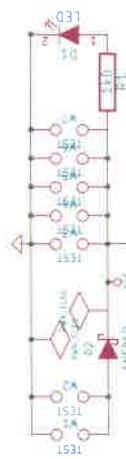
A	B	OUT
0	0	Odpojen
0	1	+ -
1	0	- +
1	1	Odpojen

Zátěž se připojuje k portu OUT. Polarita je signalizována dvojicí diod LED a závisí na vstupních signálech portu IN podle tabulky č. 14. Seznam součástek uvádí tabulka č. 15, osazovací plánek je v příloze č. 6.

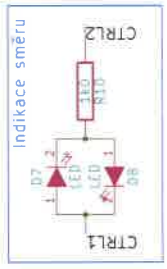
Schéma a výkres plošného spoje na následujících stranách.

Tabulka č. 15, seznam součástek

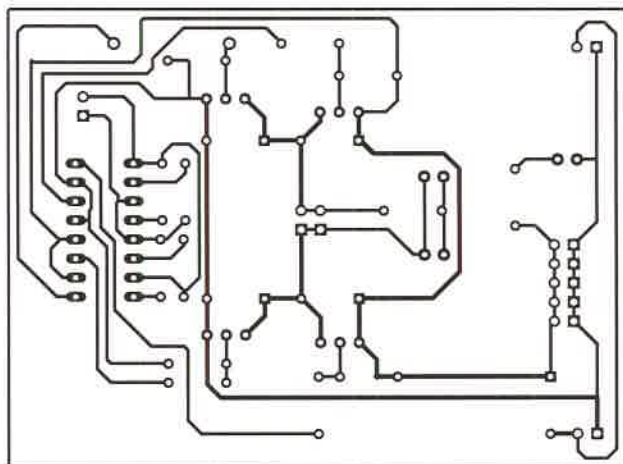
D1, D7, D8	LED 3mm, zelená
D2	1N5819
D3-D6	1N4007
R1, R3-R5, R10, R13	Rezistor 1k0; 0,25 W
R6-R9	Rezistor 10k; 0,25 W
R2, R11-R14	Rezistor 5k6; 0,25 W
T1, T2	BC327
T3, T4	BC337
W1-W7, K1, P1, P2	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
OC1	LTV847



Kombinace řídicích signálů:  
 00 – zátěž odpojena  
 01 – směr1  
 10 – směr2  
 11 – zátěž odpojena



Stavebnice s jednočipy Atmel  
**Vojtěch Bušek**  
 File: H\_mustek\_v2\_opto.sch  
 Sheet: /  
**Title: Modul izolovaný H-můstek**  
 Size: A4 Date: 7 Jul 2013  
 Křídlo E.D.A. eschema (2011-07-08 BZR 3044) – stable  
**Rev: 3**  
 Id: 1/1



back -- pohled ze strany součástek

Stavebnice s jednočipy Atmel

**Vojtěch Bušek**

File: H\_mustek\_v2\_opto.brd

Sheet: 1/1

**Title: Modul Izolovaný H můstek**

Size: A4 Date: 8 Jul 2013

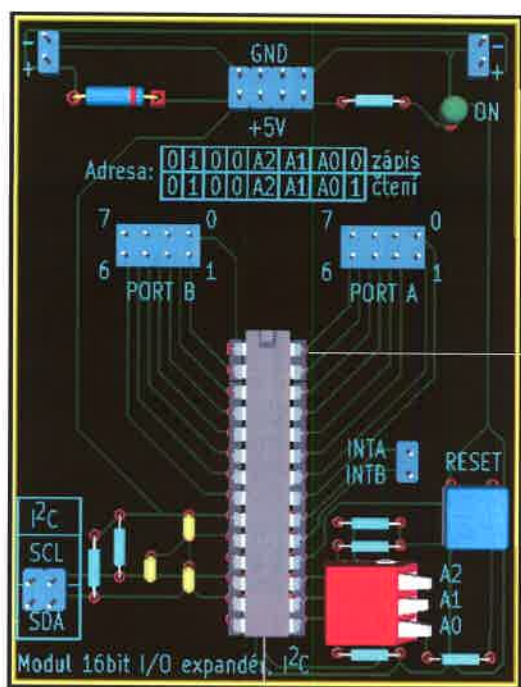
KICad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

**Rev: 3**

Id: 1/1



## 4.8 Modul 16bit I/O expandér se sběrnici I<sup>2</sup>C



Obr. č. 17, Modul I/O expandér, 3D náhled

Nastavitelný vstupně-výstupní 16 bitový expandér (jeho náhled je uveden na obr. č. 17) je hlavním prvkem dalšího modulu stavebnice. Expandér je vybaven sériovou sběrnici I<sup>2</sup>C, jejíž signály SCL a SDA jsou vyvedeny na samostatné piny. Ty jsou zdvojené pro případ připojení dalších zařízení na sběrnici.

Signály SCL a SDA sériové sběrnice jsou opatřeny vyhlazovacími kondenzátory a zdvihacími rezistory, které není vždy nutné osazovat. Zvláště v případě použití několika zařízení na sběrnici. Pak je vhodné použít externí terminátor.

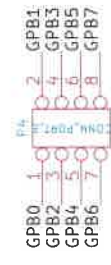
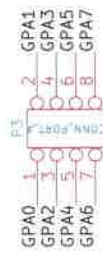
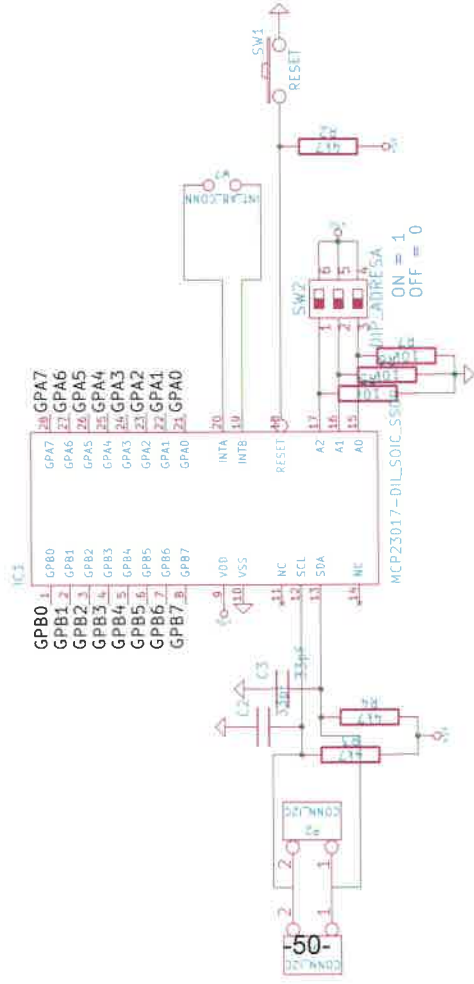
Vstupní nebo výstupní porty (záleží na nastavení obvodu) jsou vyvedeny na porty A a B. Odlišné značení respektuje katalogový list součástky. Modul dále obsahuje přepínač DIP pro volbu části adresy součástky. Zapnutí spínače odpovídá volbě logické úrovně 1. Adresu expandéru tvoří celkem 8 bitů. Její první část je určena výrobcem a je nastavena na hodnotu 0100<sub>2</sub> [16]. Další část je volena na 3 bitech s využitím spínačů A2 až A0. Poslední bit nastavuje, zda se bude do expandéru zapisovat nebo z něj číst.

Obvod poskytuje výstupy INTA a INTB, které lze použít pro generování přerušení a externí reset obvodu. Seznam součástek uvádí tabulka č. 16. Elektrické schéma a výkres desky plošného spoje jsou uvedeny na následujících stranách. Osazovací plánec v příloze č. 7.

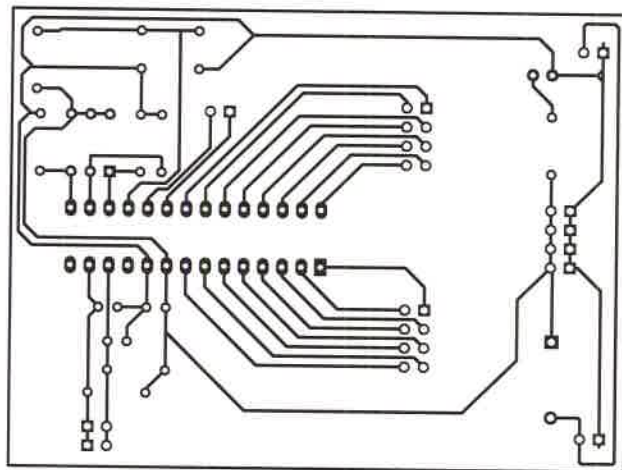


Tabulka č. 16, seznam součástek

D1	LED 3mm, zelená
D2	1N5819
R1	Rezistor 470R; 0,25 W
R2-R4	Rezistor 4k7; 0,25 W
R5-R7	Rezistor 10k; 0,25 W
IC1	MCP23017
C1	Kondenzátor 100 nF, kreamický
C2, C3	Kondenzátor 33 pF, keramický
P3, P4	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm
W1-W7, P1, P2	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
SW1	B3F-1005, bez aretace, konfigurace kontaktů: SPST-NO, THT, červený hmatník
SW2	DIP-SWITCH: DP-03



Stavebnice s jednočipý ATMEL  
**Vojtěch Bušek**  
 File: modul MCP23017-I2C\_expander.sch  
 Sheet: /  
**Title: Modul 16bit I/O expander, I2C**  
 Size: A4 Date: 7 Jul 2013  
 KiCad E.D.A. eeschema (2011-07-08 BZR 3044)-stable



Stavebnice s jednočipý ATMEL

**Vojtěch Bušek**

File: modul MCP23017-I2C expander.brd

Sheet: 1/1

Title: **Modul 16bit I/O expander, I2C**

Size: A4

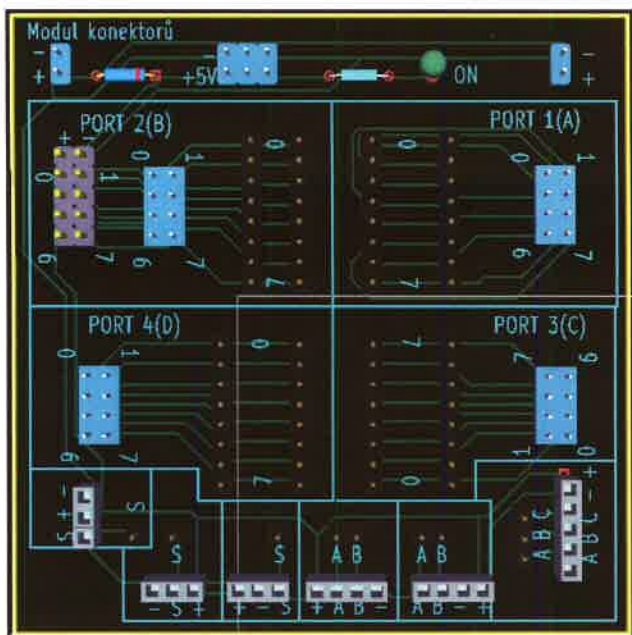
Date: 7. jul. 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) -stable

Rev: 2

Id: 1/1

## 4.9 Modul konektorů



Obr. č. 18, modul s konektory, 3D náhled

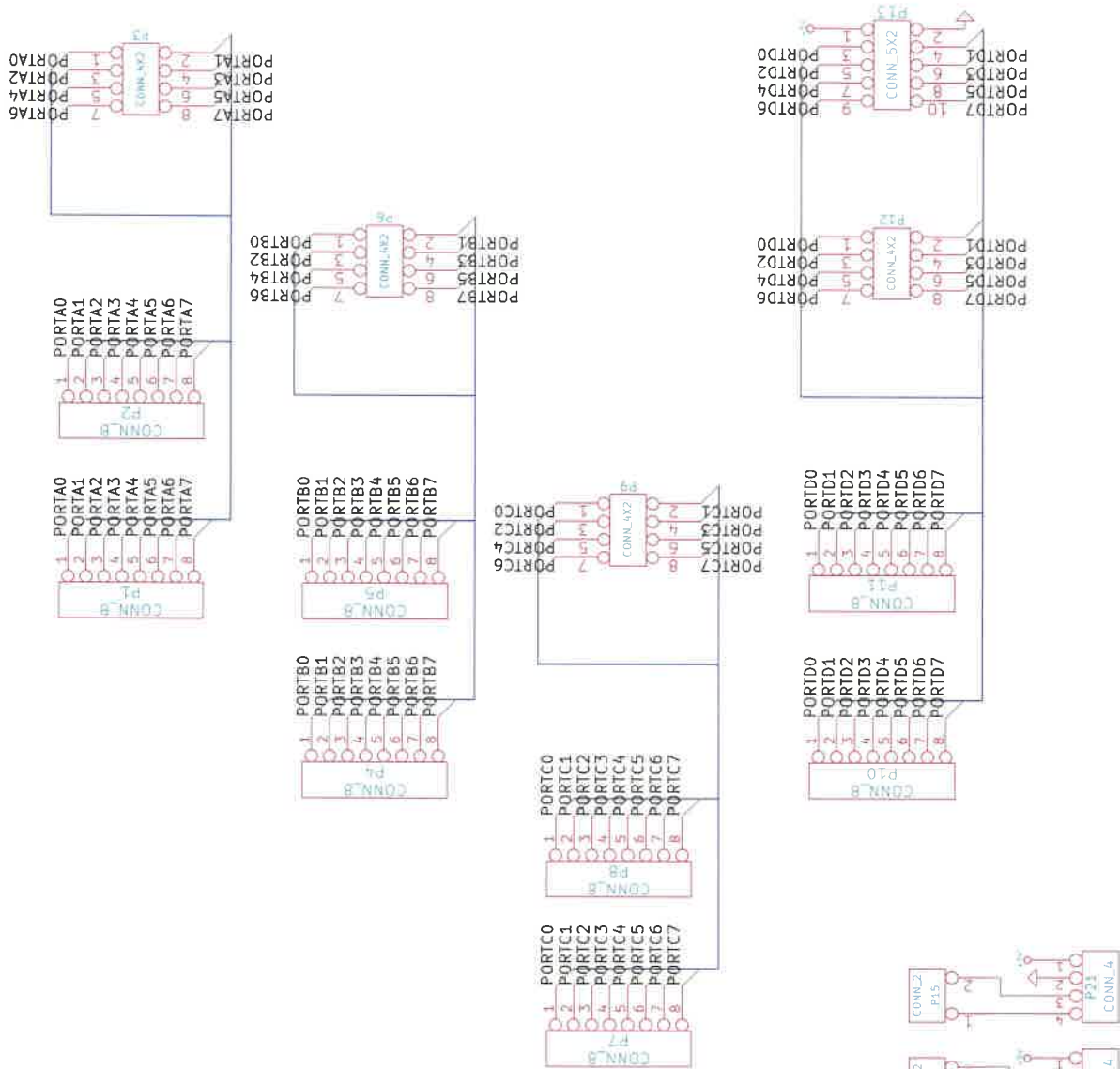
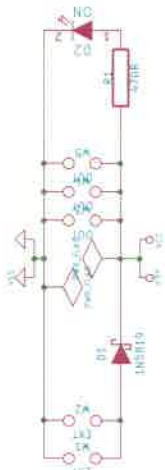
Na obrázku č. 18 zobrazený modul s konektory je velmi univerzální modul, který slouží pro připojení ostatních periferních modulů pomocí k tomu určených portů a propojovacích vodičů.

Do modulu s konektory lze přímo zasadit modul s mikroprocesorem a zajistit tak spojení s periferními moduly. Rozložení kontaktů portů 1 až 4 odpovídá rozložení kontaktů portů A až D jednočipového počítače ATmega32.

Po obvodu modulu jsou k dispozici konektory, které lze využít pro připojení periférií vývojové platformy Arduino. Z důvodu jejich velmi různorodého zapojení je zvoleno vyvedení výstupních signálů na samostatné kolíkové lišty a pro spojení s jednočipovým počítačem je potřeba tyto signály propojit manuálně. Seznam použitých součástek uvádí tabulka č. 17, schéma zapojení a deska plošných spojů jsou uvedeny na následujících stranách, osazovací plánec je uveden v příloze č. 8.

Tabulka č. 17, seznam součástek

D1	1N5819
D2	LED 3mm, zelená
R1	Rezistor 470R; 0,25 W
P19	Zástrčka kolíková, PIN5, THT, rozteč: 2,54 mm
P20, P21	Zástrčka kolíková, PIN4, THT, rozteč: 2,54 mm
K2 K4	Zástrčka kolíková, PIN3, THT, rozteč: 2,54 mm
P13	Konektor IDC, vidlice, PIN:10, THT, rozteč 2,54 mm
P1, P2, P4, P5, P7, P8	Kolíková lišta, PIN:2, vidlice/zástrčka, THT, rozteč 2,54 mm
P3, P6, P9, P12	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm
W1-W5, P14, P15	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
K1	Kolíková lišta, PIN:3, vidlice, THT, rozteč 2,54 mm



Stavebnice s jednociply Atmel

**Vojtěch Bušek**

File: prechodove\_konektory.isch

Sheet: /

Title: **Spojovací modul**

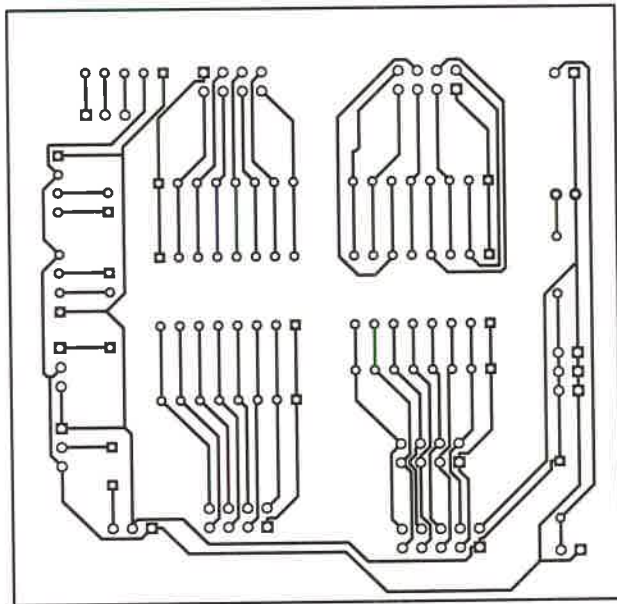
Size: A4 Date: 8 Jul 2013

KiCad E.D.A. eeschema (2011-07-08 BZR 3044)-stable

Rev:

Id: 1/1

back – pohled ze strany součástek



Stavebnice s jednočipy Atmel

**Vojtěch Bušek**

File: prechodove konektory.brd

Sheet: 1/1

**Title: Modul konektorů**

Size: A4

Date: 9 Jul 2013

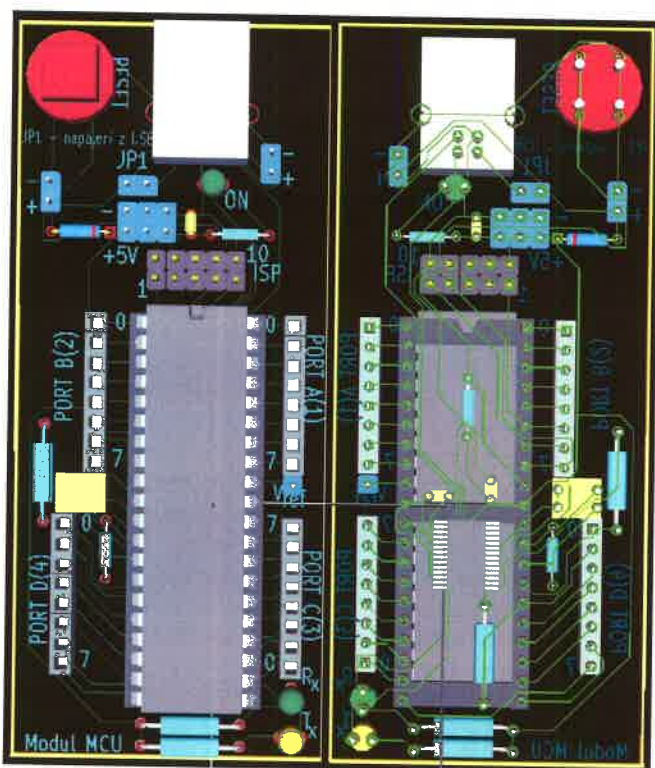
KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) -stable

**Rev:**

Id: 1/1



## 4.10 Modul s MCU



Obr. č. 19, modul s MCU, 3D náhled

Modul s řídicím jednočipovým počítačem je centrálním prvkem stavebnice. Jeho konstrukce umožňuje přímé připojení do modulu konektorů, ale není to nutnou podmínkou – lze využít i propojovacích vodičů. Aktivním prvkem tohoto modulu je jednočipový počítač Atmel ATmega32, který disponuje čtyřmi vstupně-výstupními porty a řadou periférií. Stavebnici tak lze snadno využít i mimo oblast výuky. Mikroprocesor je vybaven zavaděčem. 3D náhled na obě strany modulu je uveden na obrázku č. 19.

Stiskem tlačítka RESET dojde k resetu mikroprocesoru. Taktovací signál zajišťuje interní RC oscilátor o frekvenci 8 MHz. Modul může být osazen i běžným krystalovým rezonátorem. V takovém případě musí být mikroprocesor přeprogramován a musí být nastaveny příslušné interní pojistky.

Pro účely komunikace s osobním počítačem a snadné nahrání uživatelského programu je součástí modulu i USB převodník FT232RL, který tuto komunikaci zprostředkuje. Převodníky FT232RL a podobná rozhraní jsou vyráběny v podstatě výhradně v pouzdech SMD a kvůli zachování jednostranného plošného spoje je umístěn na straně spojů. Přenos dat mezi PC a modulem signalizují LED diody Rx a Tx.

Rozhraní FT232RL je plně kompatibilní s protokolem USB 2.0 [17]. Pro spojení s osobním počítačem je nutná instalace odpovídajících ovladačů, které jsou dostupné pro operační systémy Windows 98, 98SE, ME, 2000, Server 2003, XP, Server 2008, Vista, 7 a to včetně 64bitových verzí, dále Windows CE 4.2, 5.0 a 6.0. Podporovány jsou i operační systémy MacOS 8/9, OS-X a Linux od verze jádra 2.4 [17]. Výrobce na svých webových stránkách [18] poskytuje ovladače ke stažení, včetně návodu k instalaci [19]. Ovladač vytvoří v PC virtuální sériový port.

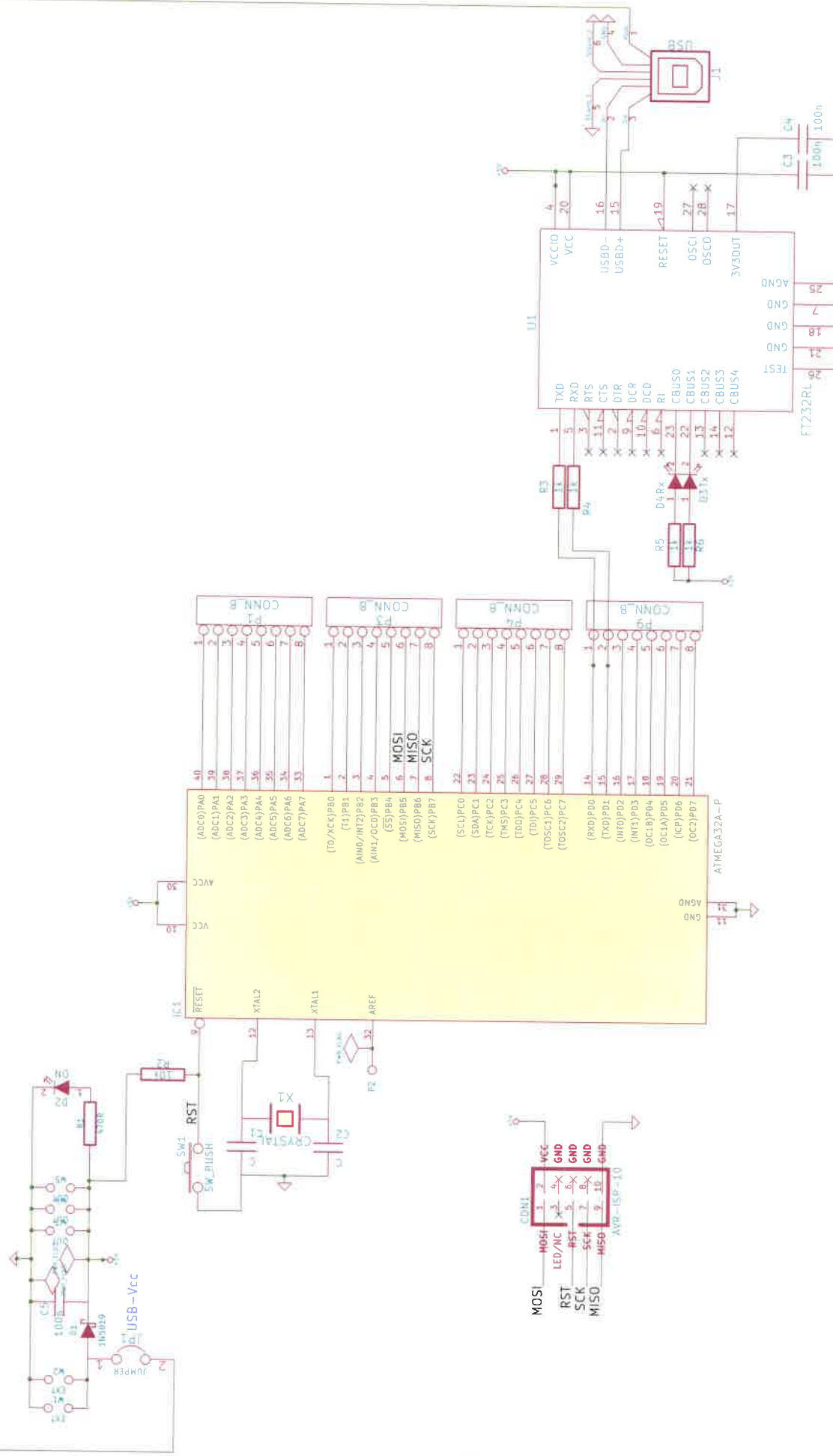
Programování mikroprocesoru se provádí zápisem příslušného souboru *.hex* s využitím zavaděče a odpovídajícího softwaru. Modul dále disponuje i programovacím rozhraním ISP s odpovídajícím 10 pinovým standardizovaným konektorem, čímž je umožněno i standartní sériové programování mikroprocesoru. Napájení během programování součástky zajišťuje programátor.

Na modulu se dále nachází jumper JP1. Jeho zkratováním je umožněno napájení modulu skrze sběrnici USB. Propojením příslušných konektorů je pak možné napájet i ostatní moduly stavebnice, nicméně je třeba mít na paměti proudové možnosti sběrnice USB a šířku použité trasy plošného spoje, která je pouhých 0,2 mm. Odběr proudu by v takovémto případě neměl překročit 100 mA [20], jak to vyžaduje standart USB. Z hlediska konstrukce plošného spoje modulu je maximální trvalý proud 800 mA. Teplota vodiče plošného spoje přitom vzroste přibližně o 15 °C [21]. Rozpis součástek uvádí tabulka č. 18. Elektrické schéma a výkres desky plošného spoje jsou na následujících stranách, osazovací plánec je uveden v příloze č. 9.

Tabulka č. 18, seznam součástek

D1	1N5819
D2 – D4	LED 3mm
R1	Rezistor 470R; 0,25 W
R2	Rezistor 10k; 0,25 W
R3 – R6	Rezistor 1k0; 0,25 W
C1, C2	Kondenzátor 22 pF, keramický
C3 – C5	Kondenzátor 100 nF, keramický
P1, P3, P4, P9	Kolíková lišta, PIN:8, vidlice/zástrčka, THT, rozteč 2,54 mm
IC1	ATMEGA32-16PU, THT, PDIP40 v patici DIL40
U1	FT232RL, SMD, pouzdro SSOP28
X1	Rezonátor krystalový, THT, 16 MHz
SW1	B3F-1005, bez aretace, konfigurace kontaktů: SPST-NO, THT, červený. hmatník
J1	USB B zásuvka, úhlová, THT
CON1	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm
W1-W5, JP1	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P2	Kolíková lišta, PIN:1, vidlice, THT, rozteč 2,54 mm

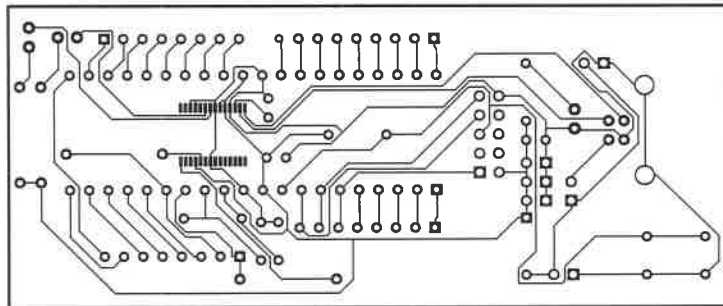




Stavebnice s jednočipový Atmel  
**Vojtěch Bušek**  
 File: Modul s MCU.isch  
 Sheet: /

**Titě: Modul s MCU**  
 Size: A4 Date: 9. jul. 2013.  
 KiCad E.D.A. eschema (2011-07-08 BZR 3044) - stable  
 Rev: 1/1

back – pohled ze strany součástek



Stavebnice s jednočipý ATME1

**Vojtěch Bušek**

File: Modul s MCU.brd

Sheet: 1/1

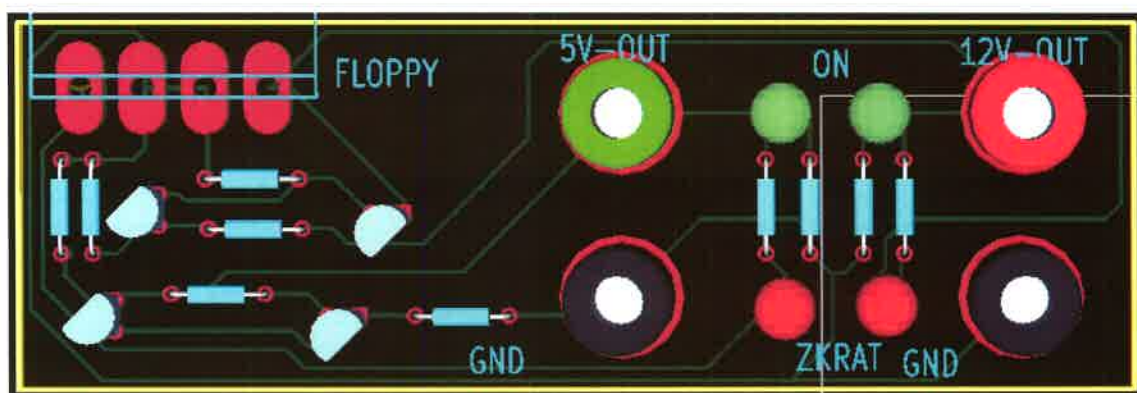
**Title: Modul s MCU**

Size: A4 Date: 9 Jul 2013

KICad: E.D.A. pcbnew (2011-07-08 BZR 3044) -stable

**Rev:**  
Id: 1/1

## 4.11 Modul napájení



Obr. č. 20, modul napájení, 3D náhled

Modul napájení, zobrazený na obrázku č. 20, slouží pro zajištění napájení celé stavebnice a chrání ji a zdroj proti proudovému přetížení.

Jako zdroj napětí lze použít libovolný stabilizovaný zdroj stejnosměrného napětí 5 V. Další úroveň napětí je volitelná a není pro moduly stavebnice potřebná, nicméně při výuce a různých experimentálních zapojeních je vhodné využívat i jiná napětí pro práci s elektromotory, relátky, žárovkami, atd. Modul umožňuje i ochranu tohoto volitelného zdroje. Pro ukázkou je zde použito napětí 12 V.

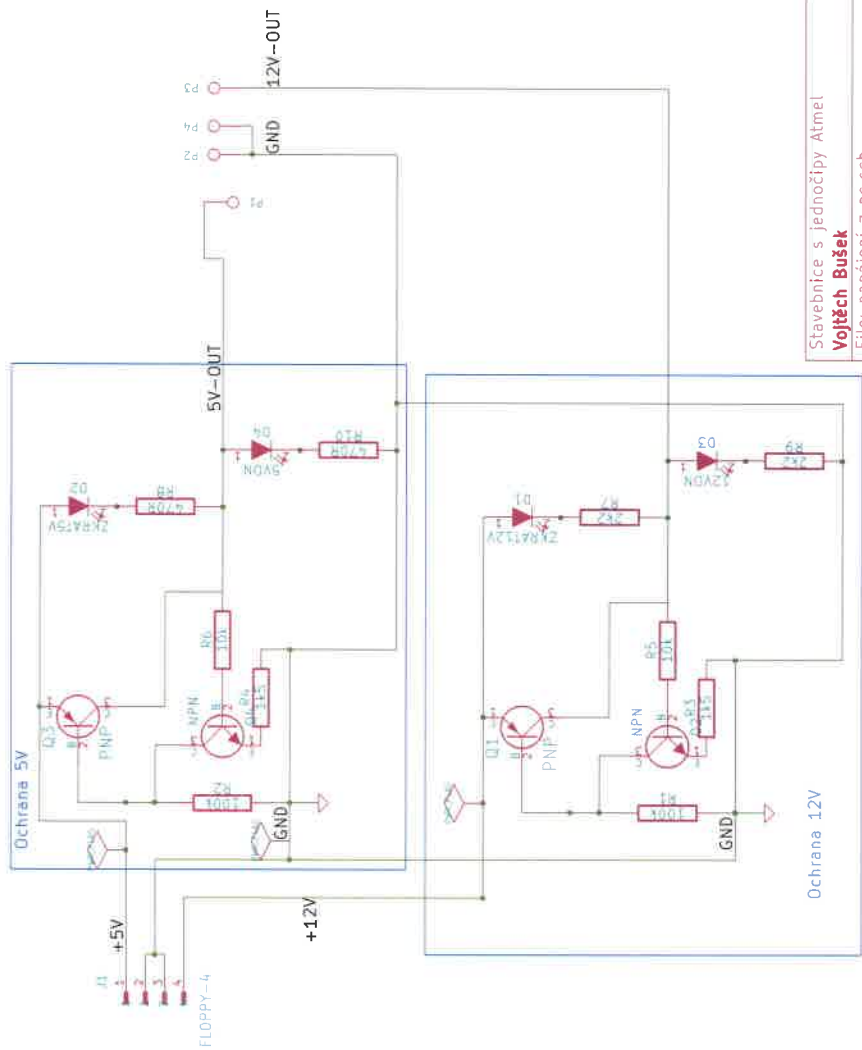
Konstrukce umožňuje instalaci modulu do samostatné přístrojové krabičky například spolu se zdrojem. Napájení je vyvedeno na běžné 4 mm banánkové zásuvky.

Ochranu proti zkratu zajišťuje dvojice tranzistorů, které se chovají jako zdroj konstantního proudu. Pokud proud dosáhne nastaveného proudového prahu, napětí rychle klesne a celý obvod se chová jako zdroj proudu s vysokým vnitřním odporem. Činnost nadproudové ochrany je signalizována diodou LED.

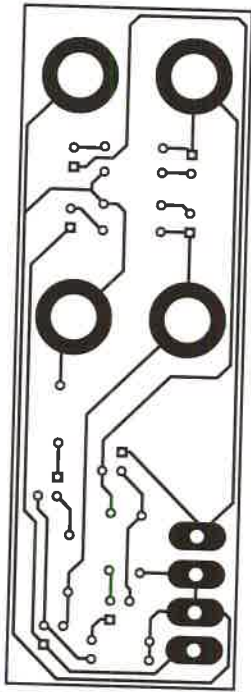
Nadproudová ochrana modulu je účinná při proudu 500 mA při napětí 5 V, a 800 mA při napětí 12 V. Práh účinnosti lze změnit volbou rezistorů R1 až R6, případně i změnou použitých tranzistorů. Všechny moduly stavebnice, konkrétně jejich plošné spoje, jsou na tyto hodnoty dimenzovány s dostatečnou rezervou. Seznam použitých součástí uvádí tabulka č. 19, elektrické schéma a výkres desky plošného spoje se nachází na následujících stranách, osazovací plánec je uveden v příloze č. 10.

Tabulka č. 19, seznam součástek

Q1, Q3	BC 327
O2, O4	BC 337
R1, R2	Rezistor 100k; 0,25 W
R3, R4	Rezistor 1k5; 0,25 W
R5, R6	Rezistor 10k; 0,25 W
R7, R9	Rezistor 2k2; 0,25 W
R8, R10	Rezistor 470R; 0,25 W
J1	Libovolný úhlový, PIN:4, vidlice/zástrčka, THT, rozteč 5,08 mm
P1 – P4	Zásuvka; banánek 4 mm; 60VDC; L:46 mm; Montážní otvor: Ø9 mm
D1, D2	LED 3 mm, červená
D3, D4	LED 3 mm, zelená



Stavebnice s jednočipy Atmel  
**Vojtěch Bušek**  
 File: napajeni z pc sch  
 Sheet: /  
**Title: Modul napájení**  
 Size: A4 Date: 9 Jul 2013  
 KiCad E.D.A. eeschema (2011-07-08 BZR 3044) - stable  
**Rev:**  
 Id: 1/1



Back – pohled shora

Stavebnice s jednočipy Atmel  
**Vojtěch Bušek**

File: napájení z pc.brd  
Sheet: 1/1

**Title: Modul napájení**

Size: A4

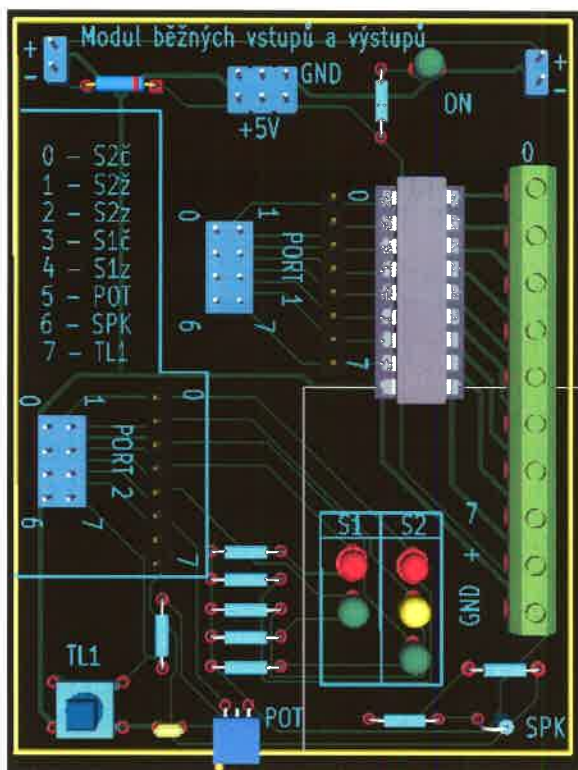
Date: 9 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) – stable

Rev:

Id: 1/1

## 4.12 Modul běžných vstupů a výstupů



Obr. č. 21, modul vstupů a výstupu, 3D náhled

Posledním navrženým členem stavebnice je modul s běžně využívanými vstupy a výstupy. Jeho náhled uvádí obrázek č. 21.

Modul obsahuje mikrospínač, potenciometr, piezoelektrický akustický měnič bez generátoru, pětice diod LED a darlingtonové pole ULN2308A.

Vstupní brány obvodu ULN2308A jsou vyvedeny na PORT 1. Odpovídající výstupy na svorkovnici u obvodu desky. Společná anoda obvodu je vyvedena rovněž na svorkovnici a není spojena s napájením modulu, kvůli možnému napájení odlišným napětím. V takovém

případě je potřeba zajistit, aby napájení spínaných prvků a napájení stavebnice zajišťoval jeden zdroj.

Na konektoru PORT 2 jsou vyvedeny vstupy a výstupy ostatních periférií, které jsou při programování velmi jednoduše použitelné. Barvy diod LED připomínají světelný semafor, což má návaznost na ukázkové úlohy. Je však možné je využít libovolně. Jezdec potenciometru má v sérii zapojenou dvojici zdířek. Tu je možné zkratovat propojkou, nebo do zdířek umístit kondenzátor o kapacitě 100 nF. Tím je možné využívat funkci `GetRC` jazyka Bascom a snadno tak odečítat polohu potenciometru. K tlačítku TL1 je připojen zdvihací rezistor. Při jeho stisku je výstupní signál připojen k GND.

Zapojení PORTu 2 uvádí tabulka č. 20. Rozpis součástek tabulka č. 21. Elektrické schéma a výkres desky plošného spoje jsou uvedeny na následujících stranách, osazovací pláněk v příloze č. 11.



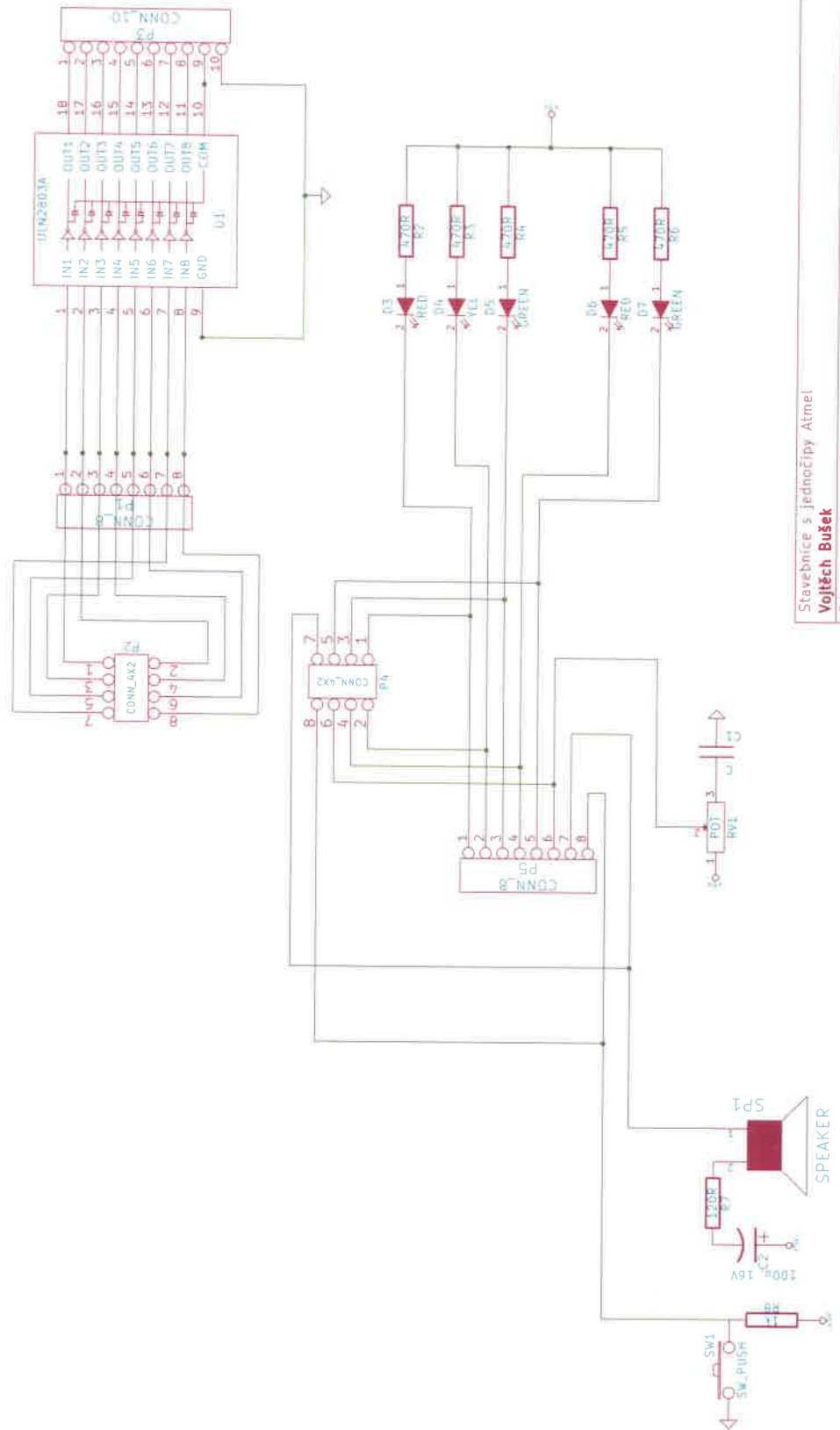
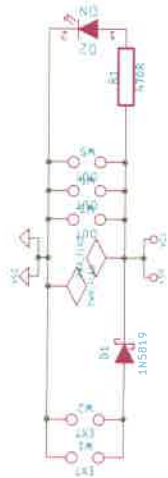
Tabulka č. 20, zapojení PORT 2 modulu

PIN	SIGNÁL	POL.
0	S2 – červená LED	-
1	S2 – žlutá LED	-
2	S2 – zelená LED	-
3	S1 – červená LED	-
4	S1 – zelená LED	-
5	Jezdec potenciometru	+/-
6	Piezoakustický měnič	-
7	Tlačítko TL1	+/-

Tabulka č. 21, seznam součástek

D1	1N5819
D2 – D7	LED 3mm, různé barvy
R1 – R6	Rezistor 470R; 0,25 W
R7	Rezistor 120R; 0,25 W
R8	Rezistor 1k0; 0,25 W
C1	Zástrčka kolíková, PIN2, THT, rozteč: 2,54 mm / Kondenzátor 100 nF, keramický
C2	Kondenzátor 100 µF, 16V, elektrolytický
P1, P5	Kolíková lišta, PIN:8, vidlice, THT, rozteč 2,54 mm
P3	Svorkovnice do PCB, rozteč 5,08 mm, THT, PIN:10
U1	ULN2803A, DIL18 v příslušné patici
SW1	B3F-1005, bez aretace, konfigurace kontaktů: SPST-NO, THT, hmatník
BPT-14	Akustický měnič: piezoelektrický; bez vnitřního generátoru
P2, P4	Konektor IDC, vidlice, PIN:8, THT, rozteč 2,54 mm
W1-W5	Kolíková lišta, PIN:2, vidlice, THT, rozteč 2,54 mm
P2	Kolíková lišta, PIN:1, vidlice, THT, rozteč 2,54 mm





Stavebnice s jednočipý Atmel

Vojtěch Bušek

File: modul ULN.sch

Sheet: /

Title: Modul s běžnými vstupy a výstupy

Size: A4

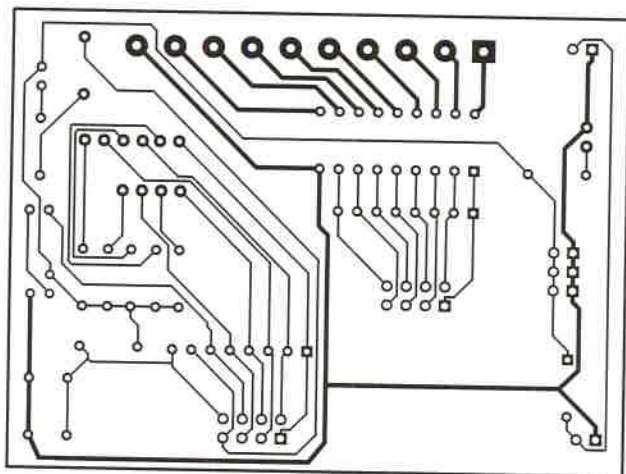
Date: 10 Jul 2013

KiCad E.D.A. eeschema (2011-07-08 BZR 3044) - stable

Rev:

Id: 1/1

back - pohled ze strany součástek



Stavebnice s jednočipy Atmel

**Vojtěch Bušek**

File: modul ULN.brd

Sheet: 1/1

Title: Modul s běžnými vstupy a výstupy

Size: A4 Date: 10 Jul 2013

KICad: E.D.A. pcbnew (2011-07-08 BZR 3044) -stable

Rev:

Id: 1/1

## 5 Návrh ukázkových úloh

S pomocí stavebnice lze realizovat různé a rozdílně obtížné úlohy. Její provedení se snaží co možná nejvíce ulehčit práci začátečníkům tím, že řeší propojení odpovídajících signálů a jasně popisuje prvky na modulech. Zapojení je ve většině případů dobře patrné, přehledné a nevyžaduje zdlouhavé vyhledávání v katalogových listech, které je při běžné práci s elektronickými součástkami nutností.

Stavebnice byla vytvořena za účelem výuky programování jednočipových počítačů. Je třeba mít na paměti, že toto téma je velmi obsáhlé a komplexní – vyžaduje znalosti programování, logiky, algoritmizace, elektroniky, funkce ovládaného hardwaru, představivost. Stavebnice se snaží používat taková elektronická zapojení, která umožňují oproštění od prvotního nezáživného probrání potřebné látky. Navíc témata související s jednočipovými počítači nejsou pro žáky snadno pochopitelná, mají-li být probrána teoreticky a spíše zdržují od samostatné práce.

Pro práci na úlohách jsou vyžadovány pouze minimální znalosti z oblasti elektroniky a informatiky. Ty mohou být postupem času prohloubeny a ukázány v souvislostech, problémy lze řešit teprve v okamžiku, když na ně narazíme. Úlohy je rovněž vhodné kombinovat a spojovat je do úloh komplexních, jak již bylo naznačeno v úvodu. Spojením dvou dílčích úloh je možné vytvořit užitečnou aplikaci s použitím jednočipového počítače.

Úlohy vedou žáky k trénování představivosti a schopnosti naplánovat svoji činnost, řešit problémy, trénují logické myšlení a algoritmizaci úloh. V tomto případě plánují činnost mikroprocesoru, který udělá vše, co mu sdělí pomocí příkazů. Zpětná vazba je žákům poskytnuta velmi rychle v případě volby vhodných úloh. Při prvních zkušenostech s programováním je důležité nechat žáky vytvářet jednoduché a funkční, byť velmi krátké programy. S tím souvisí i volba programovacího jazyka. Úlohy předpokládají využití jazyka Bascom AVR, jehož příkazy a syntaxe jsou snadno pochopitelné a velmi brzy přicházejí první viditelné výsledky, které žáky motivují.

Součástí diplomové práce je návrh několika ukázkových úloh vhodných pro začátečníky včetně metodických listů.

## 5.1 Ukaž, jaké spínače (ne)jsou sepnuté

Pro první seznámení s jednočipovým počítačem je vhodné ukázat a vyzkoušet, co vlastně dělá. Podle našich příkazů umí číst logické stavy na svých pinech nebo svým pinům logické stavy nastavovat.

Tato definice je poněkud degradující. Samozřejmě, že pro mikroprocesor tyto stavy představují nějakou informaci, kterou může zpracovat, uložit ji, odeslat na port atd. Máme na paměti, že program pracuje v nekonečné smyčce a naše příkazy se provádějí velice rychle.

Tuto skutečnost budeme demonstrovat na dvou krátkých úlohách, pro které propojíme následující moduly:

- Modul s MCU spojený s modulem konektorů
- Modul 8 x LED + 1x Bargraf
- Modul 2x8 spínačů s volbou výstupu (pozice OFF dává log. 0)

Volba konkrétních portů je libovolná, žáci si pouze musí uvědomit, který z modulů – a tedy i odpovídající port – bude použit jako vstupní a který jako výstupní. Tato informace je pro mikroprocesor nezbytná.

Načtenou informaci musíme uložit do proměnné – její velikost koresponduje s počtem použitých spínačů. Těch je 8, proměnná by tedy měla mít velikost bajtu. Tato proměnná bude zároveň říkat, jaké LED diody má mikroprocesor rozsvítit. Rozsvítí se pouze ty diody LED, na jejichž odpovídající pozici je sepnutý spínač – příslušný bit je ve stavu log. 1.

Další „podúloha“ ukazuje zmiňované zpracování dat po načtení z portu. Data jsou příkazem NOT hodnota invertována. Rozsvítí se tedy LED diody na pozicích, kde NEjsou sepnuty spínače.

Pravdou je, že k realizaci této úlohy vůbec není jednočipový počítač potřeba a v praxi bychom se s takovouto aplikací setkali jen stěží. Po určitém čase může být vhodné se k této úloze vrátit a zapojit ji bez jednočipového počítače, namalovat elektrické schéma a vysvětlit si činnost obvodu ULN2803.

## 5.1.1 Pracovní list úlohy

# Ukaž, jaké spínače (ne)jsou sepnuté!

1 K jednočipovému počítači jsou připojeny 2 moduly. Jeden se spínači, druhý s LED diodami. Který je vstupní? Který výstupní? Zapiš si odpověď! K jakému portu je připojen?

Vstupní: \_\_\_\_\_

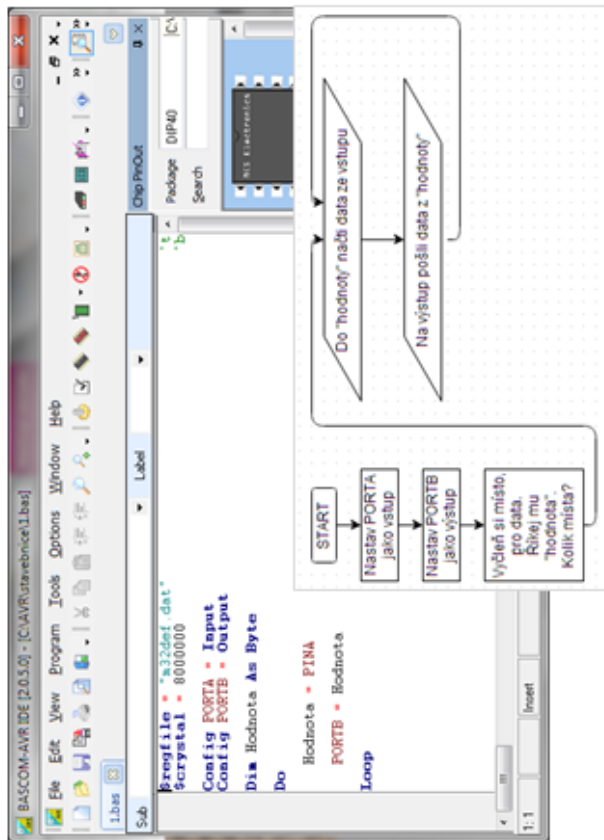
Výstupní: \_\_\_\_\_

2 Na obrázcích vpravo je vývojový diagram a samotný program. Označ číslicemi (nebo jinými symboly) řádek v programu a odpovídající místo v diagramu.

3 Naprogramuj svůj vlastní program, kterým mikropočítač rozsvítí LED diody na stejných pozicích, kde jsou zapnuté spínače. Zapiš programu bude velmi podobný jako na obrázku vpravo. V čem se může lišit?

\_\_\_\_\_

\_\_\_\_\_



4

Počítač může po načtení data i různě zpracovávat, než nám je zobrazí. Mohl by nám ukázat, jaké spínače jsme **NE**zapnuli? Jak program upravit? Na jakém místě?

Příkaz, který by nám mohl pomoci se jmenuje **NOT** – má význam **negace**.

Vyzkoušej ve svém programu a pokus se odpovědět na otázku, co dělá negace.

Program si nezapomeň uložit!

## 5.2 Knight Rider

Inspirací pro tuto úlohu je dobře známý „inteligentní“ automobil, který sehrál hlavní roli ve známém populárním stejnojmenném seriálu. Pro výuku se bude jednat o jednu jeho konkrétní část – přední světelnou lištu, ve které se světlo plynule přesouvalo zleva doprava a zpět, jak naznačuje obrázek č. 22.

Pro tuto úlohu budeme potřebovat:



- Modul s MCU spojený s modulem konektorů
- Modul 8x LED + 1x Bargraf

Obr. č. 22, záběr ze seriálu Knight Rider, převzato z [22]

Žáci by měli být již seznámeni s dvojkovou soustavou – ideálně formou nějaké hry. Měli by mít jasnou představu, že pokud jednočipovému počítači přikážeme, aby na svůj port poslal nějaké číslo, poslechne nás. Číslo ale napíše v „jemu přirozené“ dvojkové soustavě tak, že na odpovídajících pinech nastaví logické úrovně 1. Z předchozí úlohy víme, že tento stav rozsvítí LED diody na modulu.

Chceme tedy, aby na port zapsal číslo 1 (LSB), poté 2, pak 4, 8, 16, atd. V okamžiku, kdy dojdeme k číslu 128 (MSB), budeme zapisovat čísla v obráceném pořadí zpět k hodnotě 1.

Úloha trénuje převody čísel mezi soustavami a vede k neočekávanému výstupu. Ačkoliv by nás měl jednočipový počítač poslechnout, žáci uvidí, že svítí všechny připojené LED diody. Jak je to možné? Jednočipový počítač pracuje velice rychle. Diody se tedy rozsvěcují postupně, ale tak rychle, že to nemůžeme postřehnout! Náš program tedy budeme muset poněkud upravit přidáním příkazů, které jednočipový počítač v jeho práci pozdrží. Budou to příkazy `waitms`.

## 5.2.1 Pracovní list úlohy



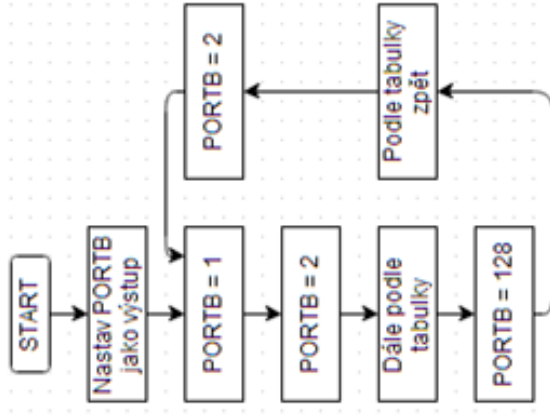
### Knight Rider

**1** K jednočipovému počítači je připojen modul s diodami LED. Jaká čísla musíme zadat, aby se postupně rozsvěcovali jednotlivé LED diody? Víme, že se rozsvítí dioda na stejné pozici, kde jednočipový počítač nastaví pin portu na log. 1 (+5 V). Doplň následující tabulku:

Bin	Dec
1	
10	
100	
1000	
10000	
100000	
1000000	
10000000	

**2**

Naprogramuj svůj vlastní program, kterým mikropočítač rozsvítí LED diody stejně jako to dělá automobil jménem KITT. Postupovat můžeš podle vývojového diagramu na ohrázkü.



**3**

Opravdu program dělá, to co chceš? Ne? Možná je to proto, že pracuje příliš rychle! Příkazem `waitms 1000` můžeš program zastavit na dobu 1 sekundy. Uprav svůj program a vyzkoušej.

**4**

Nakresli nový vývojový diagram, který odpovídá tvému programu. Nezapomeň svou práci uložit!

### 5.3 Knight Rider II.

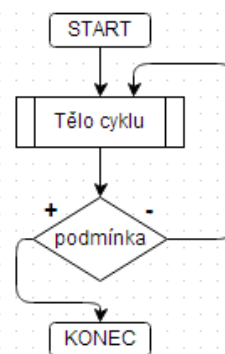
Úloha, která následuje je pouze obměnou předchozí. Zápís příslušného programu je sice jednoduchý a pochopitelný, ale jeho zápís je poměrně neúsporný. Jednočipový počítač navíc neustále opakuje téměř to samé. Pro tyto účely je ale vhodné používat cykly. Ty jsou také předmětem této úlohy. Konkrétně jednodušší z nich: Do...Loop Until. Jak je zpracován zobrazuje algoritmus na obrázku č. 23. Cyklus je opuštěn, je-li podmínka splněna. V češtině by bylo možné ho nazvat „opakuji...dokud není splněna podmínka“.

Úloha ukazuje, že výpočty může procesor provádět sám bez naší pomoci. Není nutné mu hodnoty přesně sdělovat, existuje-li mezi nimi matematický vztah.

Úlohu je možné mírně obměnit a místo výpočtu následující hodnoty pouze posouvat desetinnou čárku příkazem Shift. Všichni žáci zcela jistě vědí, co se stane při posunu desetinné čárky v desítkové soustavě. Jaký bude rozdíl, pokud budeme pracovat v soustavě dvojkové? Významná je přitom skutečnost, že počítač tuto operaci provede výrazně rychleji, než násobení nebo dělení. To platí obecně pro všechny počítače, ať jednočipové, nebo jiné.

Málokdo z žáků také prohlédne, že i tento program povede k neočekávanému chování aplikace. Při prvním zapnutí se totiž nerozsvítí první LED dioda, ale až druhá. Dále bude náš program dělat přesně to, co opravdu chceme. Proč? Odpověď je skryta v algoritmu tohoto cyklu.

Žáci mohou rovněž trénovat svou představivost. Pouze tento cyklus lze využít, pokud dopředu neznáme počet opakování – procesor přeci „opakuje dokud ...“. Dokud nestiskneme tlačítko nebo dvě nebo správnou kombinaci tlačítek, dokud nepřijde příkaz od jiného počítače, dokud neuplyne nějaký stanovený čas. Závěrečná diskuze na toto téma jistě probudí fantazii žáků.



Obr. č. 23, algoritmus cyklu Do - Loop Until



## 5.3.1 Pracovní list úlohy



### 1 Knight Rider II.

Použij zapojení a tabulku z předchozí úlohy. V tomto cvičení se naučíš pracovat s cyklem. Nejprve ale malý úkol. Vyplň tabulku podle příkazů na obrázku a pracuj **pouze s aktuální hodnotou!**

```
START
A = 1
zapiš do tabulky podle vzorce A^2
A = 128
Tabulka je vyplněna
```

Proměnná	A
A=	A=1
A=	A=
A=	A=
A=	A=
A=	A=
A=	A=
A=	A=
A=	A=

Při programování se často používá následující zápis:

$$A = A * 2$$

Výpočet může být samozřejmě jiný. Jak bychom to řekli slovy? Odpověď promysli. Promluvíme si o ní společně.

### Syntaxe a význam příkazu

Do Dělej následující příkazy

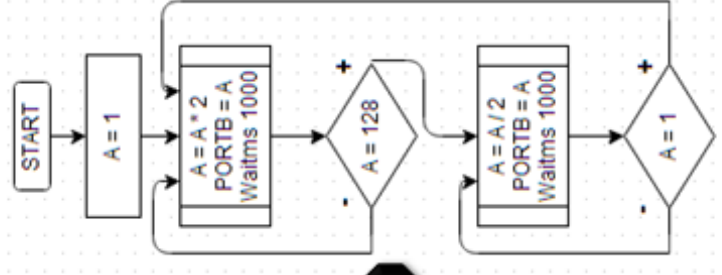
**2**  
 $A = A * 2$   
 PORTB = A  
 Waitms 1

**Loop Until A = 128**  
 Dokud A není rovno 128

Naprogramuj svůj vlastní program pro KITTa s pomocí cyklu.

Můžeš využít vývojového diagramu vpravo.

Nezapomeň svou práci uložit!



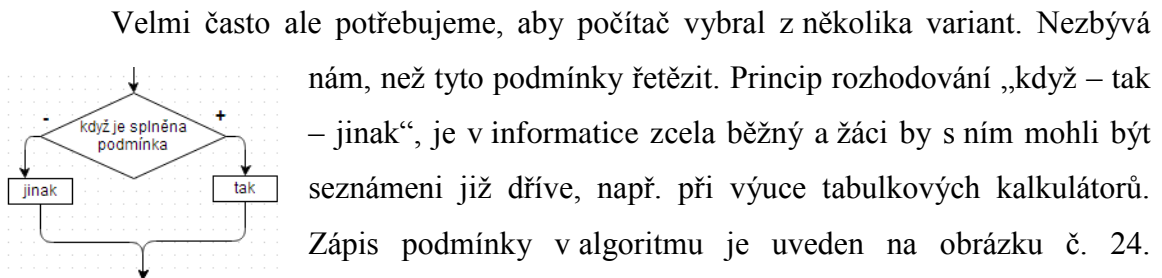
**3**

**4**

K jakému účelu by se podle tebe dal tento cyklus použít?

## 5.4 Dekodér

Důležitou programovací strukturou jsou podmínky. Je to jediný způsob, jak se může počítač rozhodovat mezi dvěma alternativami. Ty jsou, jak dobře známo, 0 (nepravda) a 1 (pravda). Jiné rozhodování je pro něj nemožné, což vychází ze zákona Booleovy algebry o vyloučení třetího. Žádná třetí možnost prostě není.

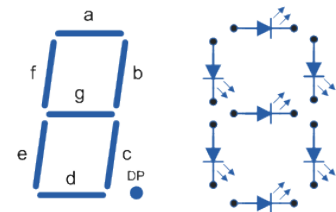


Obr. č. 24, podmínka Parametr „jinak“ nemusí být vždy zapsán. V případě, že chybí a podmínka není splněna, pokračuje počítač v programu dále.

V tomto cvičení budou žáci potřebovat následující moduly:

- Modul s MCU spojený s modulem konektorů
- Modul 2x sedmissegmentový displej
- Modul 16 spínačů s volbou výstupu (nastavit poloha ON=log. 1)

Z jednočipového počítače uděláme dekodér, který podle námi navoleného čísla v binární soustavě (s pomocí spínačů) rozsvítí správné segmenty na displeji. A to tak, abychom hodnotu viděli tak, jak jsme zvyklí – v desítkové soustavě. Jak se segmenty značí, udává obrázek č. 25.



Obr. č. 25, značení segmentů displeje

Na úvod je také vhodné nakreslit zapojení displeje. Zdůraznit, že jeden pól je společný a podle toho, také musíme volit náš výstup. Záleží na tom, zda má displej společnou anodu, nebo katodu. V úloze se předpokládá displej se společnou anodou. Výsledné číslo zapisované do portu jednočipového počítače tomu musíme přizpůsobit.

```
If PINA = 5 Then  
    PORTB = &B0010010|  
End If
```

Obr. č. 26, zobrazení čísla 5 na displeji

Použitou programovou konstrukci uvádí obrázek č. 26. Později se opět můžeme k úloze vrátit a osvětlit činnost obvodů, zvaných dekodéry BCD na segmentový displej. Pracují totiž stejně. Žáci si tak mohou udělat konkrétní představu o tom, co vlastně hardware je a co dělá.

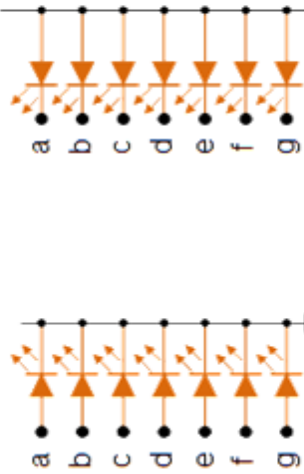
## 5.4.1 Pracovní list úlohy



# Dekodér

1

Prohlédni si následující schémata a obrázky a odpověz na otázky.



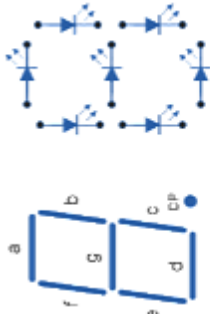
Jaký je společný pól LED diod?  
Jak je rozsvícíme? Přivedením + neb o - ?

3

Pokud si za jednotlivé křížky v tabulce představíš jedničky, za prázdné místo nuly a celý řádek budeš považovat za dvojkové číslo, jakou hodnotu bude představovat v desítkové? Vyplň v tabulce sloupec DEC.

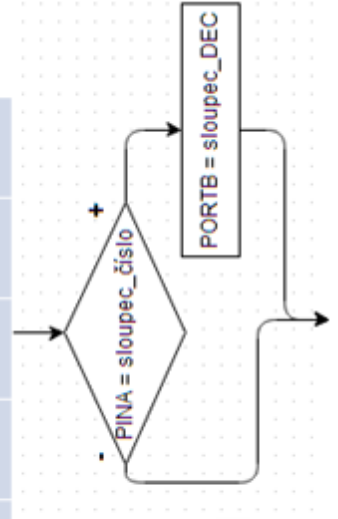
2										
Segmenty	g	f	e	d	c	b	a	číslo		
DEC	x	x	x	x	x	x	x	0		
								1		
								2		
								3		
								4		
								5		
								6		
								7		
								8		
								9		

Segmenty displeje se značí písmeny. Jaké segmenty musíme rozsvítit, abychom viděli zobrazenou číslici? Vyplň tabulku kromě sloupečku DEC.



Naprogramuj svůj vlastní program s pomocí podmínek.  
Můžeš využít vývojového diagramu vpravo.  
Nezapomeň svou práci uložit!

4



## 5.5 Semafor s přechodem pro chodce

Programování nás často přesvědčí, že naše představy o daném problému jsou velmi mělké. Říká se, že programátor věnuje pouze malou část času programování samotného programu. Mnohem více času potom věnuje tomu, aby program nedělal to, co nemá. Ke stejné situaci nás zavede tato úloha. Budeme k ní potřebovat:

- Modul s MCU spojený s modulem konektorů
- Modul běžných vstupů a výstupů

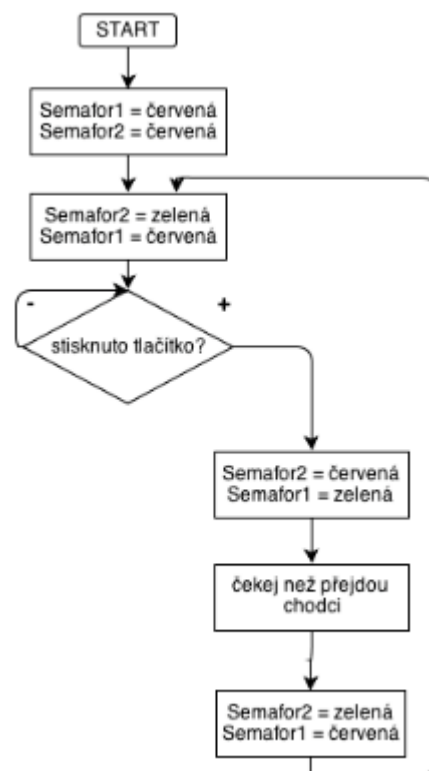
Úkolem je vytvořit program, který bude řídit světelné semaforey na přechodu pro chodce. Na modulu stovebnice se nachází vše potřebné včetně tlačítka. Program není nikterak složitý, problém je spíše ve zřetězení potřebných kroků. Malý návod se nachází na obrázku č. 27. Při výuce je vhodné algoritmus vytvořit společně, případně si pozorováním v terénu připomenout, jaké jsou jednotlivé stavy semaforu.

V úloze není potřeba využívat žádných časovačů, pouze vhodně doplnit prodlevy.

Tímto směrem se také bude ubírat ona skrytá problémová úloha. Žáci zcela jistě neprohlédnou, že jimi naprogramovaný přechod pro chodce bude snadné obelstít a stálým držetím tlačítka v podstatě zamezit jízdě automobilů.

Žáci by měli již umět nastavit nebo resetovat určitý pin portu. Umějí nastavit port jako vstupní nebo výstupní. Stavebnice ale sdružuje semaforey i tlačítka do jednoho portu. Bude potřeba nastavit jen určité piny jako vstupní. Využijeme k tomu příkaz  $DDR_x = \&b01111111$ , kterým nastavíme jako vstupní pouze MSB portu. Za x je třeba dosadit písmeno portu, kam je modul připojen.

Úloha může být časově náročnější než předchozí uvedené úlohy.



Obr. č. 27, zjednodušený algoritmus řízení semaforu

## 5.5.1 Pracovní list úlohy

### Semafor s přechodem pro chodce

1

K úkolu budeme potřebovat modul běžných vstupů a výstupů propojený portem 2 s jednočipovým počítačem.

Pozorně si запиš, na které piny počítače jsou připojeny jednotlivá světla semaforu a tlačítko a zda se jedná o vstup, nebo výstup.

Pozor na polaritu, kterou se LED diody na tomto modulu spínají!



2

Pozoruj, jak se chová semafor u přechodu pro chodce. Můžeš si vyhledat i vhodné video. Zapiš si jednotlivé kroky. Porovnej je se svými spolužáky a následně se pokuste sestavit algoritmus. Nakresli.

Propojení	PORT.PIN	VSTUP/VÝSTUP
S1 červená		
S1 zelená		
S2 červená		
S2 žlutá		
S2 zelená		
Tlačítko		

Podle tvého algoritmu vytvoř řídicí program pro semafor.

4

Nezapomeň svou práci uložit!

Pozorně proveď chování programu. Věnuj pozornost tlačítku. Stiskni ho opakovaně, třeba i na dlouhou dobu a pozoruj co se stane.

5

Navrhni, jakou část programu upravit.

3

Port, ke kterému je připojen modul běžných vstupů a výstupů nemůže být jednoduše nastaven jako vstupní nebo výstupní. Vstupní bude jen jeho část. Jak se projeví příkaz `DDRx` = programovacího jazyka Bascom? Zapiš do sešitu!

DDR znamená u jednočipových počítačů „Data Direction Register“. Zkus si tento název přeložit do češtiny.

## 5.6 Návrhy dalších úloh

S elektronickou stavebnicí lze realizovat poměrně velkou škálu úloh. Její moduly jako je můstek H, darlingtonové pole nebo I2C expandér dovolují i náročnější aplikace, které svým rozsahem mohou vyhovět i na středních odborných školách.

Stavebnice také může sloužit jako základ pro připojení jiných součástek a obvodů, které nejsou její součástí. Moduly toto propojení umožňují.

Návrhy úloh, které je možné dále realizovat. Bez rozlišení, zda se jedná o úlohy pro základní nebo střední školy.

### 5.6.1 4bitový dekodér BCD na HEX

S využitím sedmissegmentových displejů je možné zobrazovat i jiné znaky než 0 až 9. Dovedou zobrazit i písmena A až F. Jedná se o rozšíření úlohy jménem Dekodér. V té využíváme 4 bity pro zapsání dvojkového čísla, přičemž je 6 kombinací nevyužitých. Hodili by se právě pro zobrazení zmíněných znaků. Žáci se tak mohou naučit hexadecimální soustavu jaksi mimochodem.

### 5.6.2 Multiplexování čtyřmístného segmentového displeje

Práce s polem a přesným načasováním příkazů je hlavním posláním této úlohy. Mikroprocesor musí ve správném pořadí posílat hodnoty na port displeje a zároveň vždy pomocí tranzistoru zapnout odpovídající pozici. Úloha procvičí i práci s textovými řetězci. Zobrazená čísla je opět možné volit s pomocí připojených spínačů nebo je načítat ze sériové linky. Načtení a zobrazení čísla z maticové klávesnice může být vhodným pokračováním následující úlohy.

### 5.6.3 Maticová klávesnice

Bascom sice disponuje přímo funkcí `GetKBD`, která umí provést vše potřebné, ale obsluhu maticové klávesnice je přirozeně možné vytvořit. Po nakreslení a pochopení schématu vede příklad k podmínkám.

### 5.6.4 Hrací kostka

Úloha může jednoduše využít již naprogramovaného dekodéru se segmentovým displejem. Zde se pouze změní vstup, kterým je náhodné generované číslo a ne řada spínačů. Rovněž je vhodné upozornit, že generované číslo je pouze tzv. pseudonáhodné. Krátký program o tom spolehlivě přesvědčí.

### 5.6.5 Teploměr s využitím termistoru

Další úloha, kde lze využít displejů. Připojením termistoru k libovolnému pinu mikroprocesoru a uzemněním přes kondenzátor bude možné načítat příkazem `GetRC` hodnoty, které se budou měnit v závislosti na odporu termistoru. Přesný popis, co příkaz dělá je uveden v manuálu. Nemůžeme čekat, že bude teploměr závratně přesný. Výsledek je ale překvapivý. Teploměr je potřeba kalibrovat.

Obdobná úloha může zjišťovat například polohu potenciometru, nebo s určitými výpočty vypsat hodnotu kapacity kondenzátoru nebo rezistoru.

### 5.6.6 Stroboskop

Jednočipové počítače disponují několika kanály pulzně šířkové modulace. Někteří žáci hrají na strunné hudební nástroje. Společným tématem je frekvence a kmitání. Pokud necháme jednočipový počítač rozsvěcovat výkonné LED diody v určitých časových intervalech a svítit s nimi na kmitající struny, uvidíme, že některé nekmitají. Nebo se to alespoň zdá. Frekvenci je opět možné odečítat z displeje.

### 5.6.7 Řízení směru otáčení SS motoru

Modul s H můstkem je přímo určen k tomuto účelu. Změna směru otáčení stejnosměrných motorů se provádí změnou polarit napájecího napětí na jeho vývodech. K tomuto účelu se běžně využívala relátka. Polovodičové řešení je ovšem také možné a vcelku jednoduché. Procvičujte a prohlubuje pochopení práce polovodičů. Na odborné škole může vést téma k diskuzi, jaké problémy mohou nastat při nesprávném zapojení podle běžně známých a dostupných schémat.

### 5.6.8 Měření doby kmitu, nebo kyvu

Připojením optické závory a prací s přerušením a s časovači můžeme snadno měřit čas mezi průchody kyvadla optickou závorou. Kolik průchodů musíme změřit, abychom vypočetli délku trvání periody a kolik, abychom vypočetli délku trvání kyvu. Kam je nejvhodnější umístit závoru. To jsou otázky, které mají návaznost na fyziku a velmi dobře dokreslují a doplňují probíranou látku.

## 6 Závěr

Návrh modulární stavebnice pro výuku programování jednočipových počítačů má za cíl přiblížit a umožnit práci s elektronikou a zmíněnými počítači i úplným začátečníkům. Jako přínos hodnotím její nenáročnost, jasnou koncepci spojování jednotlivých dílů a snadné programování samotného mikroprocesoru. Žákům to dovoluje soustředit se na to, co je opravdu podstatné. Mohou si lámat hlavu s algoritmem, což je pro ně v jejich věku dostatečně zatěžující, nikoliv se zapojením. Použití vývojových kitů ve výuce začátečníků je problematické, protože se ve většině případů jedná o profesionální výrobky, které, obsahují velké množství periferních obvodů, není vidět jejich zapojení a často působí nepřehledně. Přílišná náročnost snadno vede u žáků k domněnì, že probíraná látka je příliš těžká, komplexní a neproniknutelná. Proto není vhodné ji komplikovat složitým zapojením nebo programovacím jazykem s nízkou abstrakcí. Navržená stavebnice těmito problémy netrpí.

Některé úlohy spolu se stavebnicí byly zároveň použity při výuce předmětu PXE (praxe) na Střední průmyslové škole, Na Třebešíně 2299, Praha 10. Konkrétně v prvním ročníku oboru Informační technologie ve strojírenství. Pochopení principů práce s elektronickými obvody přichází postupně a žáci si práci s nimi osvojují až poté, co zvládnou alespoň základní kroky a funkce samotného jednočipového počítače.

Při práci s jednočipovými počítači dochází k pochopení významu dvojkové soustavy v informatice a k rozvoji logického myšlení. Jako vhodné úlohy se jeví především ty, které vedou k nějakému problému nebo neočekávanému chování aplikace. Žáci sami kladně hodnotili využití dřívějších dílčích úloh při řešení komplexního zadání (například při tvorbě teploměru). Vytvářejí si vlastní představu o rychlosti práce počítače, učí se používat přístroje pro měření a případně i zobrazení průběhu elektrických veličin.

Tyto aktivity mezi žáky vzbudily zájem o daný předmět a v mnoha případech navrhovali vlastní cíle. Snáze také chápali matematické úlohy související se zjednodušováním logických funkcí (Karnaughovy mapy), které se ve zmíněném předmětu vyučují. Jako pozitivní dále hodnotím i práci s polovodičovými prvky, jelikož se výuce tohoto tématu věnuje na školách malá pozornost.



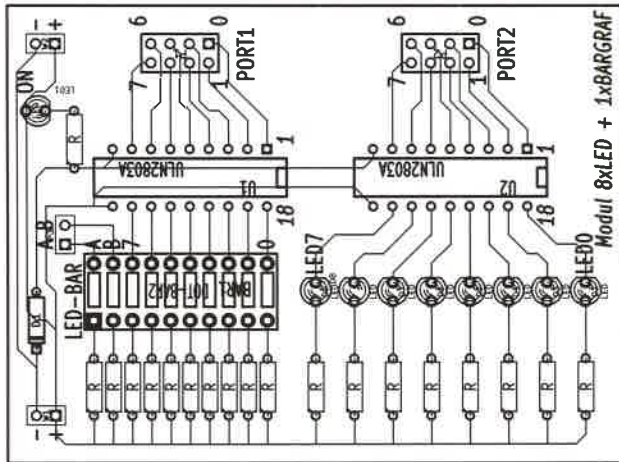
## 7 Odborná literatura

- [1] ATMEL CORPORATION. *DATA SHEET, 8-bit microcontroller ATmega32*. Dostupné z: <http://www.atmel.com>
- [2] BASCOM-AVR: AVR Internal Registers. ALBERTS, Mark. MCS ELECTRONICS. *BASCOM-AVR* [online]. [cit. 2013-01-04]. Dostupné z: [http://avrhelp.mcselec.com/avr\\_internal\\_registers.htm](http://avrhelp.mcselec.com/avr_internal_registers.htm)
- [3] BUŠEK, Vojtěch. *Řízení mikrokontroléry ATMEL*. Bakalářská práce. Jihočeská univerzita, Pedagogická fakulta, Katedra fyziky. Vedoucí práce Ing. Michal Šerý.
- [4] ŠUBRT, Vladimír. *Mikrokontroléry ATMEL AVR: vývoj aplikací*. 1. vyd. Praha: BEN - technická literatura, 2002, s. 71. ISBN 8073000555.
- [5] ATMEL CORPORATION. *DATA SHEET, 8-bit microcontroller ATtiny25*. Dostupné z: <http://www.atmel.com>
- [6] MATOUŠEK, David. *Práce s mikrokontroléry Atmel AT89C2051*. 1. vyd. Kubicová Iveta. Praha: BEN - technická literatura, 2002. ISBN 80-7300-048-2. Popis čítačů a časovačů, s. 136.
- [7] RS-485 - Wikipedie: Dvou vodičová verze RS-485. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-19]. Dostupné z: <http://cs.wikipedia.org/wiki/RS-485>
- [8] ATMEL CORPORATION. *8-bit Microcontrollers: Application Note* [online]. 2010, 14 s. [cit. 2013-03-20]. <http://www.atmel.com/Images/doc2564.pdf>. Dostupné z: <http://www.atmel.com/Images/doc2564.pdf>
- [9] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR AT90S*. 2. vyd. Praha: BEN, 2006, s. 288. ISBN 80-7300-209-4.
- [10] *TinySafeBoot: A tiny and safe Bootloader for AVR-ATtinys and ATmegas* [online]. 2013, 10.4.2014 [cit. 2013-04-14]. Dostupné z: [http://jtxp.org/tech/tinysafeboot\\_en.htm](http://jtxp.org/tech/tinysafeboot_en.htm)
- [11] VÁŇA, Vladimír. *Mikrokontroléry Atmel AVR: programování v jazyce Bascom*. 1. vydání. Praha: BEN, technická literatura, 2004, s. 15. ISBN 80-7300-115-2.
- [12] BASCOM-AVR: Language Fundamentals. *MCS Electronics* [online]. 2012 [cit. 2013-05-01]. Dostupné z: [http://avrhelp.mcselec.com/language\\_fundamentals.htm](http://avrhelp.mcselec.com/language_fundamentals.htm)

- [13] BASCOM-AVR: BASCOM Language Reference, DIM. *MCS Electronics* [online]. 2012 [cit. 2013-05-01]. Dostupné z: <http://avrhelp.mcselec.com/dim.htm>
- [14] IEEE 754-2008. *IEEE Standard for Floating-Point Arithmetic*. Revision of ANSI/IEEE Std 754-1985. 2008. Dostupné z: <http://ieeexplore.ieee.org/servlet/opac?punumber=4610933>
- [15] NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. LI 1-1998. *NEMA STANDARDS PUBLICATION NO. LI 1-1998: Industrial Laminating Thermosetting Products*. Rosslyn, Virginia 22209, 1998. Dostupné z: <http://www.nema.org/Standards/ComplimentaryDocuments/LI1.pdf>
- [16] MICROCHIP TECHNOLOGY INC. *MCP23017/MCP23S17: 16-Bit I/O Expander with Serial Interface*. 2007, 44 s. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf>
- [17] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *FT232R USB UART IC*. 2010, 43 s. Dostupné z: <http://www.tme.eu/cz/Document/9d2a0f24d19320b803e2081a185f270d/ft232r.pdf>
- [18] <http://www.ftdichip.com/FTDrivers.htm>, 16.6.2013
- [19] <http://www.ftdichip.com/Support/Documents/InstallGuides.htm>, 16.6.2013
- [20] USB.org - Documents: Developers. *USB.org - Documents* [online]. [cit. 2013-06-09]. Dostupné z: <http://www.usb.org/developers/docs/>
- [21] ZÁHLAVA, Vít. ČVUT, Fakulta elektrotechniky. *Proudové zatížení DPS – trvalé* [online]. [cit. 2013-06-20]. Dostupné z: [http://www.micro.feld.cvut.cz/home/zahlava/ppn/prednasky/vlastnosti\\_DPS.pdf](http://www.micro.feld.cvut.cz/home/zahlava/ppn/prednasky/vlastnosti_DPS.pdf)
- [22] *ZIMBIO: David Hasselhoff At The 2nd Annual Knight Rider Convention* [online]. 2013 [cit. 2013-05-11]. Dostupné z: <http://www.zimbio.com/pictures/g5fo9HBIJVE/David+Hasselhoff+2nd+Annual+Knight+Rider+Convention/BY5imophlst>

# Příloha č. 1

back - pohled ze strany součástek



File: moduL16\_LED\_v2.brd  
Sheet: 1/1

Title: modul 8xLED + 1xBARGRAF

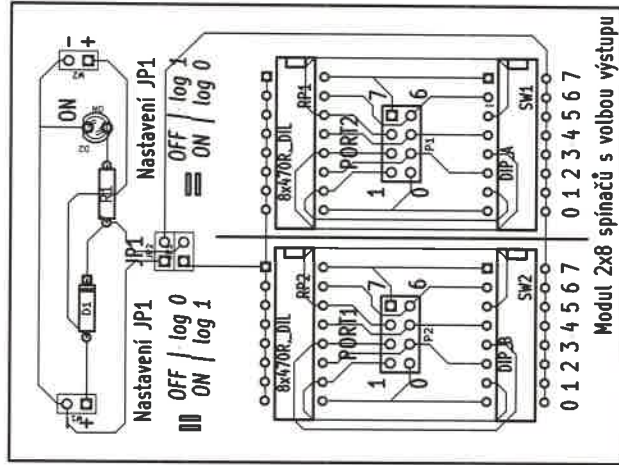
Size: A4 Date: 5 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

Rev: 1/1

Id: 1/1

# Příloha č. 2



back - pohled ze strany součástek

Stavebnice s jednočipy ATMEL

Vojtěch Bužek

File: modul\_16\_DIP.brd

Sheet: 1/1

Title: Modul 2x8 spínačů s volbou výstupu

Size: A4

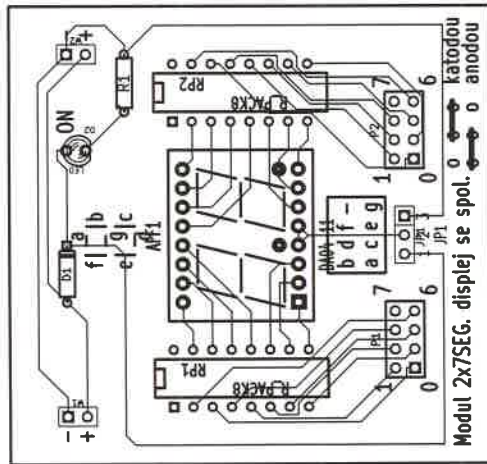
Date: 5 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR\_3044) - stable

Rev:

Id: 1/1

# Příloha č. 3



Stavebnice s jednočipy ATMEL

**Vojtěch Bušek**

File: modul\_2x7SEG\_m\_spo\_L\_A\_K.brd

Sheet: 1/1

**Title: Modul 2x 7 SEGM. displej, voľba spoločné elektrody**

Size: A4 Date: 5. jul. 2013

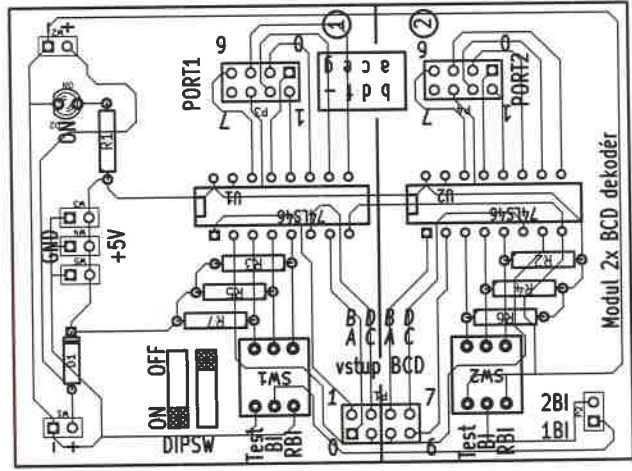
KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

**Rev:**

Id: 1/1

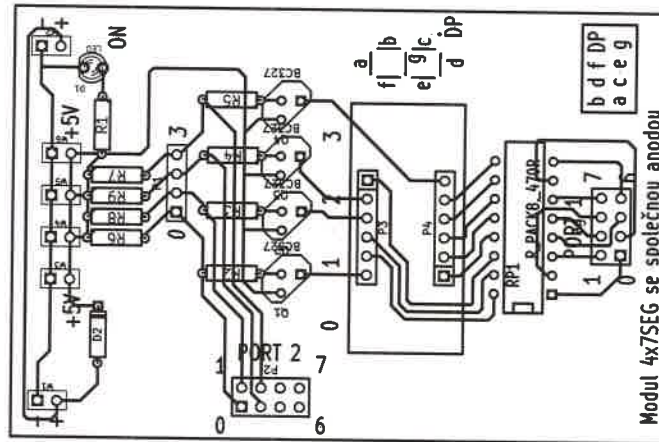
# Příloha č. 4

skrz zásuvku vstříže se bsljloq – řack  
 potisk – pohled shora



Stavebnice s jednočipy Atmel  
**Vojtěch Bušek**  
 File: modul2x\_BCD\_dekoder.brd  
 Sheet: 1/1  
**Title: Modul 2x BCD dekodér**  
 Size: A4 Date: 6 Jul 2013  
 KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable  
**Rev:**  
**Id:** 1/1

# Příloha č. 5



Stavebnice s jednočipy Atmel

**Vojtěch Bušek**

File: modul\_displej\_4x7SEG\_multiplex.brd

Sheet: 1/1

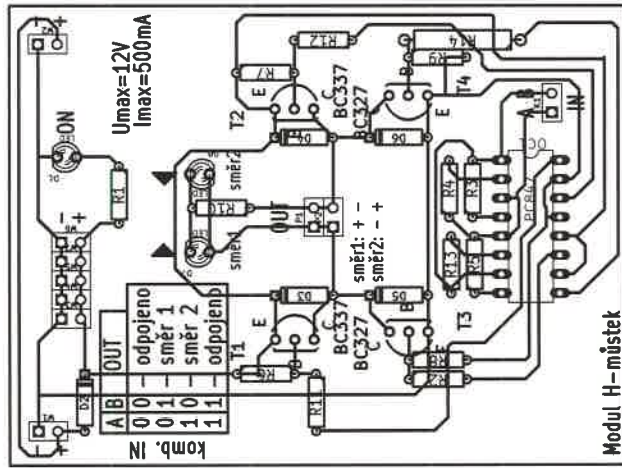
Title: Modul 4x sedmísegmenový displej se spol. anodou

Size: A4 Date: 7 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) -stable

Rev:  
Id: 1/1

# Příloha č. 6



Stavebnice s jednočipy Atmel

**Vojtěch Bušek**

File: H\_mústek\_v2\_opto.brd

Sheet: 1/1

Title: **Modul izolovaný H mústek**

Size: A4

Date: 7 Jul 2013

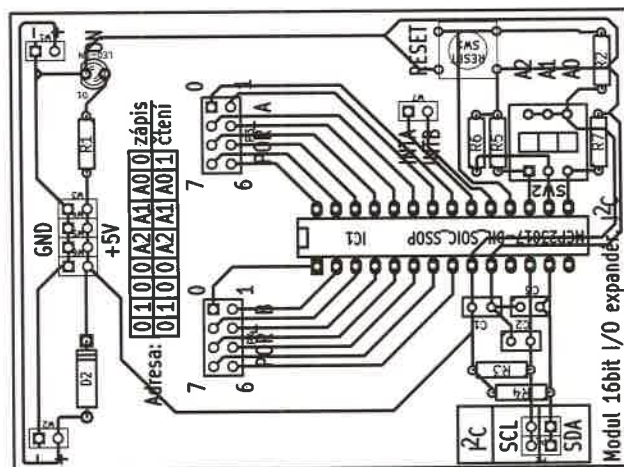
KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

Rev: **3**

Id: 1/1



# Příloha č. 7



Stavebnice s jednočipový ATMEL

**Vojtěch Bušek**

File: modul MCP23017-12C expandér.brd

Sheet: 1/1

Title: **Modul 16bit I/O expandér, I2C**

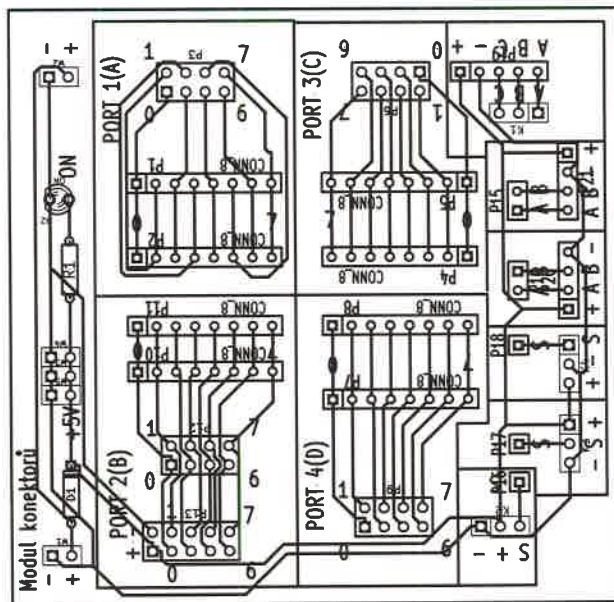
Size: A4 Date: 7 jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

Rev: 2

Id: 1/1

# Příloha č. 8



pack - bujeloq - bpad

Stavebnice s jednočipý Atmel

Vojtěch Bušek

File: prechodove konektory.brd

Sheet: 1/1

Title: Modul konektorů

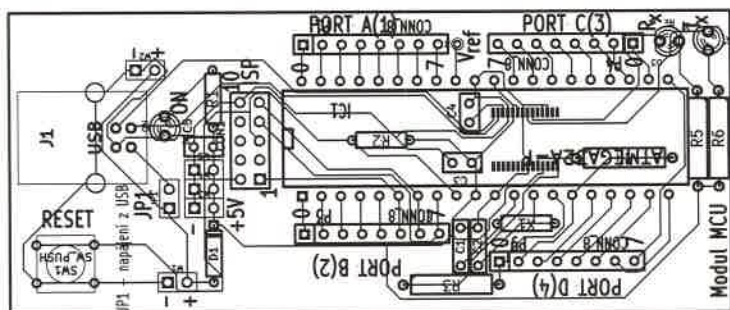
Size: A4 Date: 9. jul. 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044)-stable

Rev:

Id: 1/1

# Příloha č. 9



pacik – bojujete se s tímto součástí

Stavebnice s jednočipový ATME

Vojtěch Bužek

File: Modul s MCU.brd

Sheet: 1/1

Title: Modul s MCU

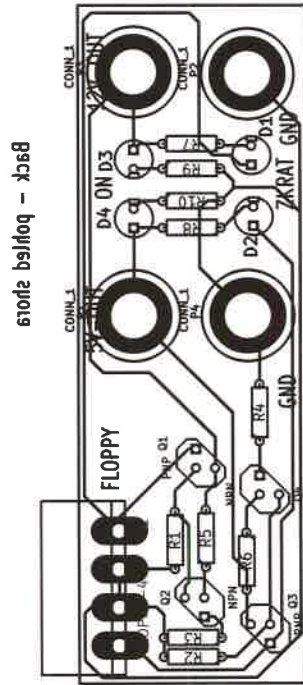
Size: A4 Date: 9 Jul 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

Rev:

Id: 1/1

# Příloha č. 10



Stavebnice s jednočipy Atmel

**Voltěch Bušek**

File: napájení z pc.brd

Sheet: 1/1

Title: Modul napájení

Size: A4

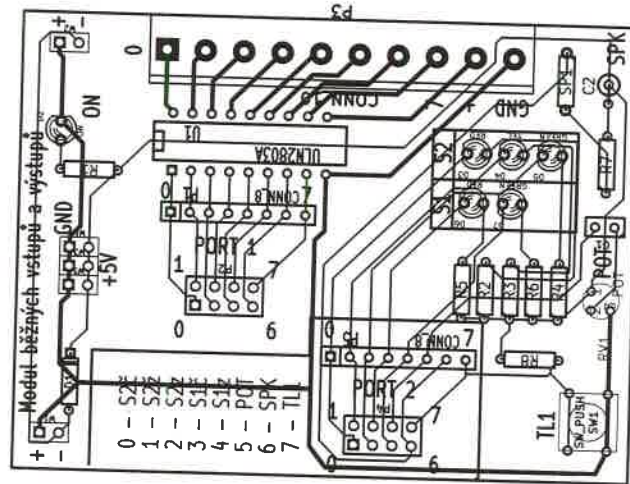
Date: 9. jul. 2013

KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable

Rev:

Id: 1/1

# Příloha č. 11



stavba modulu vstupů a výstupů

Stavebnice s jednočipy Atmel  
**Vojtěch Bušek**  
 File: modul\_ULN.brd  
 Sheet: 1/1  
**Title: Modul s běžnými vstupy a výstupy**  
 Size: A4 Date: 10 jul 2013  
 KiCad E.D.A. pcbnew (2011-07-08 BZR 3044) - stable  
 Rev: 1/1  
 Id: 1/1