

7seg. Zobrazovač

Lukáš Kořínek – SakulRaider@seznam.cz – www.sakul.cz

Tento zobrazovač vznikl jako nástupce všech předchozích zobrazovačů pro velké displeje. Je zde použit výkonnější procesor, což umožnilo přidat další funkce a hlavně do budoucna možnost aktualizace firmware pomocí jednoduché aplikace. Nyní je možno se zobrazovačem komunikovat a měnit jeho konfiguraci pomocí příkazů a zároveň z něho vyčítat provozní data.

Technické specifikace:

Napájecí napětí	12V/DC
Spotřeba proudu	cca 200mA (záleží na použitých diodách, MAX je 0,5A)
Komunikační rozhraní	UART konfigurovatelný AT příkazy (default 19,2kbps)
Zobrazované znaky	16 znaků ("NIC", 0-9, -, L, P, E, U) platí pro základní verzi
Počet adres	16 v základní verzi
Řízení jasu	PWM
Konfigurace	Pomocí AT příkazů a Jumper spojkami
Měření napětí	5-15V
Řízení teček	Samostatně tečka na zobrazovači + výstup pro dvojtečky

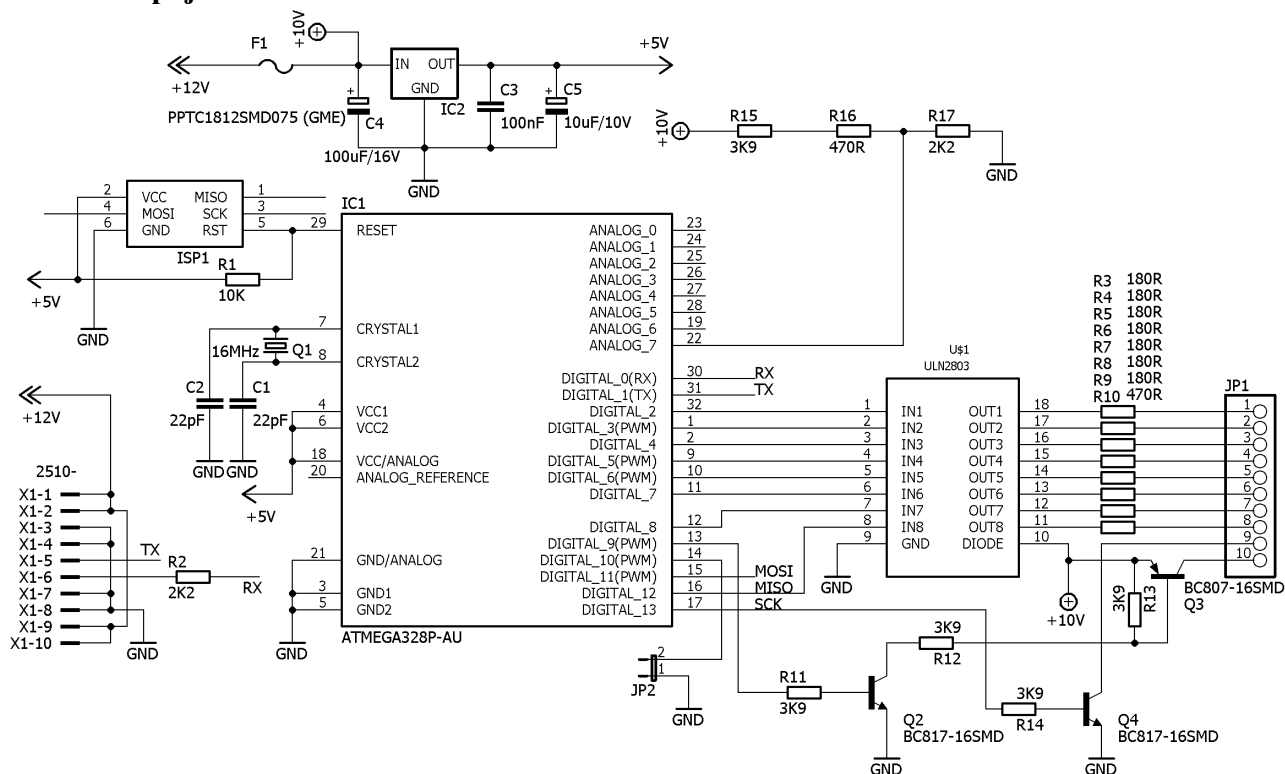
Popis konstrukce:

Jak už jsem psal v úvodu, jedná se o nástupce předchozích zobrazovačů pro velké displeje. Byla zachována zpětná kompatibilita s předchozími zobrazovači, takže je možno připojit velký displej sestavený z těchto zobrazovačů i ke starším konstrukcím. Na druhou stranu, použití procesoru **ATMEGA328P** umožnilo značně rozšířit možnosti tohoto zobrazovače. Co se týká zapojení, to se příliš nezměnilo. Srdcem zapojení je procesor **IC1**, který komunikuje s okolím přes rozhraní UART vyvedeným do konektoru **X1**. **Zde je nutno upozornit, že zapojení tohoto konektoru bylo oproti předchozím verzím upraveno a není zpětně kompatibilní!** Bylo změněno rozložení napájecích pinů, tak aby se omezila možnost poruchy vlivem špatně zapojeného konektoru a zároveň byla posílena proudová zatížitelnost celé sběrnice. Taktéž přibyl i kanál TX z procesoru, pro obousměrnou komunikaci s nadřazenou aplikací. Obvod napájení byl doplněn o vratnou pojistku **F1**, jež by měla ochránit elektroniku při zkratu na připojeném zobrazovacím prvku (LED diody, LED pásy, LED zobrazovače a podobně). Taktéž byla doplněna detekce napájecího napětí pomocí děliče tvořeného rezistory **R15-R17**. Drobné změny se dočkal i výkonový výstup pro řízení zobrazovacího prvku. Původní 7mi portový driver byl nahrazen 8mi portovým **US1**. Řízení dvojteček bylo zachováno přes tranzistor **Q4**. Bylo však doplněno řízení jasu přes tranzistory **Q2** a **Q3** jež je realizováno PWM.

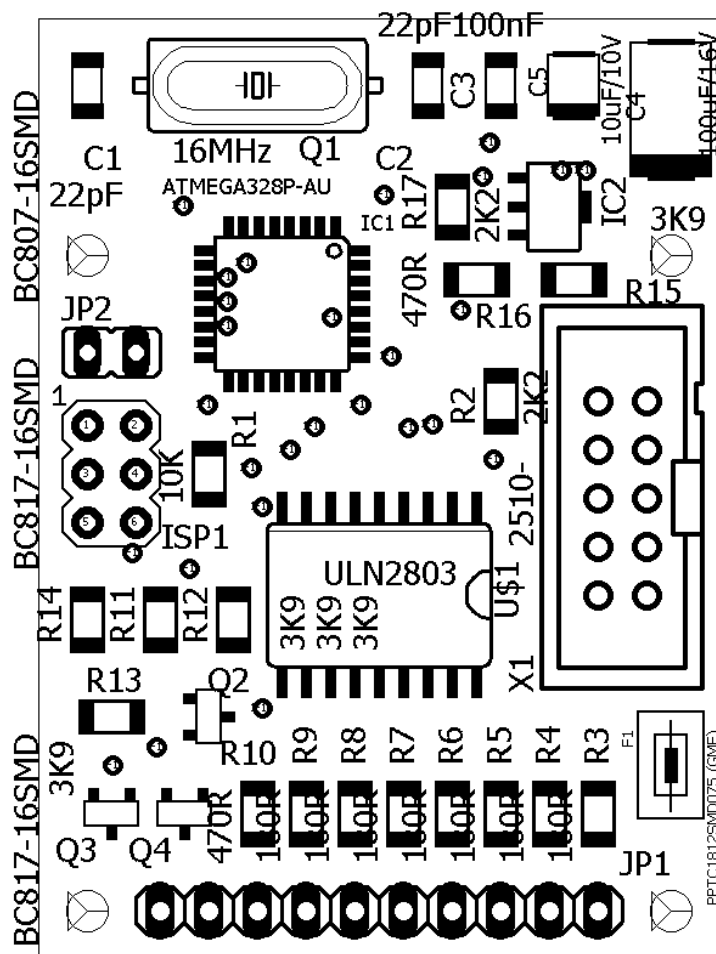
Protože většinu nastavení lze provádět AT příkazy nebyl nutný již DipSwitch a místo něho byla zařazena jumper spojka **JP2**. Osazením této propojky je možno volit komunikační protokol. Pokud je osazena používá se stará verze protokolu, jež je kompatibilní se staršími konstrukcemi (používá 1Bytový protokol). Pokud osazena není, je použit nový protokol. Vzhledem k tomu, že celá konstrukce je založena na SMD součástkách, je použit i servisní konektor **ISP1**, přes který je možno programovat procesor a propojením pinu MOSI s GND je možno přepnout procesor do defaultní komunikační rychlosti 19,2kbps. To je vhodné zejména, pokud nevíme jaká je nastavena komunikační rychlost a nedaří se nám navázat se zobrazovačem komunikací.

Výstup pro zobrazovací prvek je tvořen konektorem **JP1** a jako u předchozích verzí lze připojit libovolný zobrazovací prvek se společnou ANODOU.

Schéma zapojení:



Osazovací plán:



Při osazování je třeba pracovat pečlivě, aby nedošlo k záměně některé součástky, případně nedošlo ke zkratu vlivem pájení. Vývody zobrazovacího prvku můžeme zapájet přímo do PCB nebo můžeme osadit vhodný konektor na pozici JP1.

Důležité je upozornit na fakt, že hodnoty některých součástek jsou pouze orientační, zejména rezistorů **R3-R10** a je nutné je přizpůsobit zamýšlenému použití (v tomto případě použitému zobrazovacímu prvku).

Schéma plošného spoje:

Výkresy plošného spoje neuvádím, protože jde o spoj oboustranný s prokovenými otvory a jeho výroba z obrázkové dokumentace není možná. Nicméně v dokumentaci na konci článku najdete kompletní projektovou dokumentaci pro program Eagle v7.7, odkud si následně můžete vyexportovat potřebná data.

Význam zapojení konektorů:

JP1 - Připojení zobrazovacího prvku

JP1-1 - Segment **A**

JP1-2 - Segment **B**

JP1-3 - Segment **C**

JP1-4 - Segment **D**

JP1-5 - Segment **E**

JP1-6 - Segment **F**

JP1-7 - Segment **G**

JP1-8 - Segment **DP**

JP1-9 - Ovládání dvojtečky :

JP1-10 - **ANODA** zobrazovacího prvku

JP2 - Volba komunikačního protokolu

JP2-Osazen - Stará verze protokolu bez podpory AT příkazů (One-Byte)

JP2-Neosazen - Nová verze protokolu s podporou AT příkazů (Multi-Byte)

ISP1 - Servisní konektor

ISP1-1 - MISO

ISP1-2 - VCC (+5V)

ISP1-3 - SCK

ISP1-4 - MOSI (uzemněním volba komunikační rychlosti 19,2kbps)

ISP1-5 - RESET

ISP1-6 - GND

X1 - Interface

X1-1 - **+12V** napájení

X1-2 - **+12V** napájení

X1-3 - **GND**

X1-4 - **GND**

X1-5 - TX (UART)

X1-6 - RX (UART)

X1-7 - **GND**

X1-8 - **GND**

X1-9 - **+12V** napájení

X1-10 - **+12V** napájení

Programování (AT příkazy):

Před zařazením zobrazovače do většího celku, jako je například displej, je nutno jednotlivé zobrazovače správně nastavit. V současnosti je možno upravovat následující parametry: Jas zobrazovače, Adresu zobrazovače, Komunikační rychlost a Diagnostiku. Následně tedy popíšu nastavení těchto parametrů pomocí AT příkazů. Aby bylo ovládání pomocí AT příkazů možné, je nutné přepnout zobrazovač do režimu **Multi-Byte** (JP2 není osazen).

Každý AT příkaz (**AT-Jas255***) se skládá z návěští **AT-**, za návěštím následuje již daný příkaz například **Jas**. Některé příkazy obsahují ještě navíc parametr **255**. Konec příkazu je pak označen symbolem hvězdičky *****.

AT-Jas255* - Tímto příkazem se nastavuje jas zobrazovacího prvku. Řízení jasu se provádí pomocí PWM a proto je možno nastavit hodnotu **0-255**. Přičemž čím větší je hodnota, tím větší je výsledný jas. Nastavení se projeví okamžitě.

AT-Adresa3* - Tímto příkazem nastavujeme adresu daného zobrazovače. Momentálně je podporováno **16** adres, přičemž adresy **0** a **8** jsou pevně přiděleny pro ovládání dvojteček, z libovolného zobrazovače při použití protokolu **One-Byte**. Pokud je tato adresa nastavena zobrazovači, bude přenášet blikání dvojtečky na segment DP a G. Přičemž každý zobrazovač bez ohledu na nastavenou adresu bude na vývodu **JP1-9** přenášet stav dvojtečky (pokud je nastaven protokol **One-Byte**).

AT-Baud19200* - Tímto příkazem se nastavuje komunikační rychlost. Podporovány jsou následující rychlosti: 1200, 2400, 4800, 9600, **19200**, 38400, 57600. Pokus o nastavení jiné než podporované rychlosti bude mít za následek chybové hlášení. Nastavení komunikační rychlosti se neprojeví okamžitě, ale teprve až po potvrzení příkazem **AT-Save*** a restartu zobrazovače.

AT-Diag0* - Tímto příkazem je možno zapnout automatické diagnostické zprávy. Parametr **1** diagnostiku zapne a parametr **0** zase vypne. Toto se za běžného provozu nepoužívá a slouží to spíše pro ladění programu. Nastavení se projeví okamžitě.

AT-Info* - Tímto příkazem si můžeme nechat vypsát veškerá nastavení a hodnoty, uložené v zobrazovači. A to včetně aktuální hodnoty napájecího napětí nebo verze firmware.

AT-Save* - Tímto příkazem uložíme všechny nastavené hodnoty do EEPROM, takže v případě restartu zobrazovače budou obnoveny. Pokud bychom to neudělaly hodnoty, které jsme předtím zeditovaly, by byly platné pouze do prvního restartu a poté by se obnovily staré hodnoty uložené v EEPROM.

Ovládání:

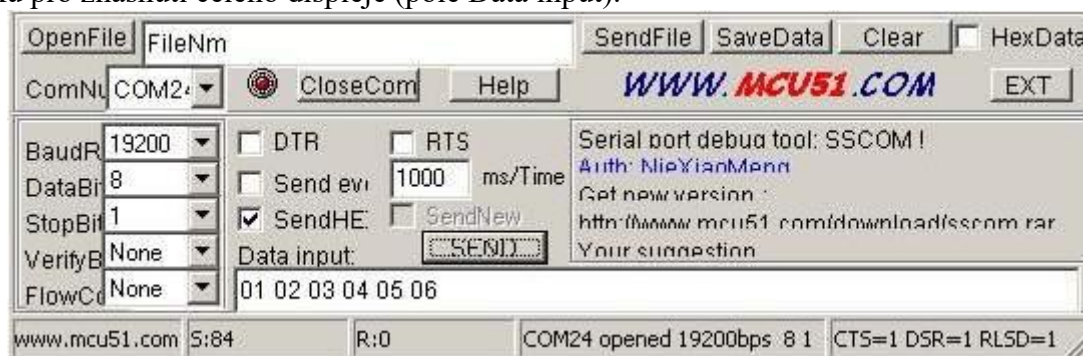
Ovládání zobrazovače je poměrně jednoduché. Momentálně je podporován pouze protokol **One-Byte**, což je stará verze kompatibilní se všemi dosud publikovanými zařízeními, která podporují výstup pro velký displej (Stopky, GPS hodiny atd.) Takže k těmto zařízením je možno zobrazovače připojit.

Samozřejmě můžete použít i vlastní zařízení, které bude posílat data do zobrazovačů. Z tohoto důvodu si prostudujte tabulku s popisem protokolu jenž je přiložena. Pro jednoduché testy lze použít nějaký sériový terminál

Komunikace se zobrazovači v displeji:

Jak už jsem zmínil, data se do displeje zasílají přes rozhraní UART. Pro komunikaci byl vytvořen jednoduchý protokol, který používá 1 byte. V tomto bytu je uložena adresa zobrazovače a znak, který se na něm má zobrazit. Zobrazovat je možno následující znaky: („NIC“, 0,1,2,3,4,5,6,7,8,9,L,P,U,E,-). Znak NIC znamená, že daný zobrazovač je vypnutý a nezobrazuje se na něm nic. Následují čísla, několik písmen a pomlčka. Adresa a zobrazovaný znak jsou v bytu rozděleny tak, že nižší 4bity jsou ADRESA a vyšší 4bity jsou zobrazovaný ZNAK. Teoreticky je možno použít až 16 adres. Pro lepší orientaci jsem přidal tabulku s významem jednotlivých bitů v odesílaném bytu.

Dobře se v tom dá orientovat, pokud budeme hodnoty zapisovat v HEX formátu. Pro pochopení uvedu několik příkladů. Pokud chceme například na zobrazovač s adresou 1 odeslat číslo 6, bude odeslaný byte vypadat takto 71h, kde 7 je právě hodnota znaku 6 (Zde si můžete všimnout, že je hodnota znaku vždy o jedna vyšší než co se skutečně zobrazí. To je způsobeno posunem, neboť 0 nereprezentuje nulu, ale hodnotu NIC, tedy zhasnutý zobrazovač. Tím pádem znak 0 na displeji, reprezentuje číslo 1 a tak dále.) a 1 je adresa jedna, tedy zobrazovač 1. Když budeme chtít na displej zapsat čísla 123456, odešleme následující: 71h 62h 53h 44h 35h a 26h. Budeme-li chtít zhasnout celý display, odešleme: 01h 02h 03h 04h 05h 06h. Myslím, že je to poměrně jasné a po prostudování tabulky není co dodat. Přesto přidám ještě obrázek z terminálu, kde je patrné nastavení komunikační rychlosti a sekvence znaků pro zhasnutí celého displeje (pole Data input).



Veškerá data, která jsou do displeje odeslána, jsou uložena a displej je zobrazuje stále, dokud nejsou přepsána novějšími daty, nebo dokud nedojde k restartu displeje. Není tedy nutné neustále posílat data do displeje. Stačí poslat pouze změnu na konkrétní zobrazovač. Pokud by jste chtěli odeslat do displeje nějaká neplatná data (nejspíše se špatnou adresou), bude je ignorovat a stav se nezmění.

Mechanické provedení:

Pro tuto konstrukci jsem vytvořil i krabičku ve formě 7mi segmentového zobrazovače. Krabičku lze vytisknout na 3D tiskárně. Jednotlivé segmenty vytiskneme z bílého plastu a krabičku s víčkem pak z plastu černého. Do jednotlivých segmentů vtlačíme LED diody, které propojíme do série (dobře patrné je to na obrázcích). Následně tyto segmenty vložíme do krabičky a zalepíme například tavnou pistolí. Pak už stačí jen jednotlivé diody segmentů propojit do elektroniky, kterou přišroubujeme šrouby M2 do krabičky. Zadní kryt můžeme také přišroubovat šrouby M3 nebo jednoduše zalepit. Tím nám vznikne kompletní zobrazovač, který můžeme namontovat do většího displeje. Všechny konektory jsou zezadu dobře přístupné, takže propojení zobrazovačů plochým kabelem nečiní žádné potíže. Tímto způsobem můžeme zkonstruovat různé velké zobrazovače. U menších můžeme použít LED diody a u větších pak LED pásy.

Seznam použitých součástek:

C1	22pF	C-EUC1206	C1206
C2	22pF	C-EUC1206	C1206
C3	100nF	C-EUC1206	C1206
C4	100uF/16V	CPOL-EUSMCD	SMC_D
C5	10uF/10V	CPOL-EUSMCB	SMC_B
F1	PPTC1812SMD075 (GME)	PTCFUSE-1812	1812
IC1	ATMEGA328P-AU	ATMEGA168-AU	TQFP32-08
IC2	78L05	78LXX-SOT89	SOT89
ISP1	AVRISP-6	AVRISP-6	AVRISP
JP1	PINHD-1X10	1X10	
JP2	JP1E	JP1	
Q1	16MHz	CRYSTALSM49	SM49
Q2	BC817-16SMD	BC817-16SMD	SOT23-BEC
Q3	BC807-16SMD	BC807-16SMD	SOT23-BEC
Q4	BC817-16SMD	BC817-16SMD	SOT23-BEC
R1	10K	R-EU_R1206	R1206
R2	2K2	R-EU_R1206	R1206
R3	180R	R-EU_R1206	R1206
R4	180R	R-EU_R1206	R1206
R5	180R	R-EU_R1206	R1206
R6	180R	R-EU_R1206	R1206
R7	180R	R-EU_R1206	R1206
R8	180R	R-EU_R1206	R1206
R9	180R	R-EU_R1206	R1206
R10	470R	R-EU_R1206	R1206
R11	3K9	R-EU_R1206	R1206
R12	3K9	R-EU_R1206	R1206
R13	3K9	R-EU_R1206	R1206
R14	3K9	R-EU_R1206	R1206
R15	3K9	R-EU_R1206	R1206
R16	470R	R-EU_R1206	R1206
R17	2K2	R-EU_R1206	R1206
U\$1	ULN2803	ULN2803	SO18L
X1	2510-	2510-	PAK100/2500-10