

Blok č. 5 Řídicí jednotka se zobrazovačem LCD a kanálem - UART

Úkol

Navrhněte a realizujte řídicí jednotku se semigrafickým **zobrazovačem** s řadičem HD44780 a **sériovým komunikačním kanálem** navázaným na interní obvod UART v STM32F100 pomocí nějž bude spolupracovat s terminálem. Přenos dat do LCD bude s využitím **sérioparalelního posuvného registru**. Jednotka bude obsahovat **uživatelské tlačítko spuštění** automatu na desce STM32VL Discovery a **tři externí řídicí tlačítka**, stejně jako v předchozí úloze. Funkce jednotky bude také vycházet z řešení předchozí úlohy, bude však rozšířena dalšími funkcemi v souvislosti s doplněnými obvody dle dalšího popisu.

Zobrazovač LCD použitý v úloze je minimálně ve verzi 1x16 znaků.

Popis funkce

Základní funkce časovacího automatu ovládaného tlačítkem na STM32VL Discovery kitu, které simuluje ovládací uživatelská tlačítka, zůstává. Jednotka může realizovat **časovací jednotku schodišťového automatu pro řízení osvětlení**, nebo řídicí **jednotku pro ovládání střešního ventilátoru** odsávání v bytovém domě. V případě řídicí jednotky odsávání není využit cyklus několika zapnutí a vypnutí před ukončením času sepnutí, na rozdíl od řešení předchozí úlohy. **Externí tlačítka** budou mít opět funkci „**zvyš**“, „**sniž**“ a „**potvrdit**“ nastavení času, navíc však bude čas možno **nastavovat též zvolenými klávesami na terminálu** připojeném prostřednictvím UART. Může se jednat např. o klávesy „+“, „-“, „**mezera**“, (ASCII znaky space - kód 0x20), lze ale využít i jiné klávesy.

Na připojeném **LCD** se bude v režimu nastavování zobrazovat právě nastavovaná velikost doby sepnutí (podobně, jako v minulé úloze) ve formě **dvouciferného čísla**. Stejně zobrazení bude též na terminálu. Při aktivace funkce automatu se bude zobrazovat **zbývající** čas v sekundách.

Prostřednictvím terminálu je možno řešit i sofistikovanější metody nastavení parametrů, např. přímé zadávání velikosti doby v sekundách. Toto není požadováno, může to však být implementováno. Bylo by vhodné, aby se v jednotce ukládala nastavená hodnota do paměti FLASH na čipu STM32F100x.

Pro přehled bude řídicí jednotka na LCD i na terminálu informovat obsluhu textem o právě aktivním **režimu** (RUN, SET, READY). Zobrazovač LCD bude informovat o času právě **nastavovaném** (režim **SET**) , času **již nastaveném** (režim READY) , času **zbývajícím** do zhasnutí (režim **RUN**) V **horním řádku** LCD bude zobrazen aktuální režim a čas (nastavovaný, nastavený nebo zbývající) a aktuální režim (RUN, SET, READY).

Zobrazení na LCD:

SET TIME = xx (v režimu nastavování- právě nastavený čas)

RUN TIME = xx (zbývající čas)

READY TIME = xx (nastavený čas, čeká spuštění tlačítkem)

Možné režimy **RUN** běží odměr času svícení

SET nastavování času tlačítka nebo terminálem

READY připraven a čeká na stisknutí schodišťového tlačítka pro rozsvícení

V **dolním řádku** LCD bude **jméno autora**.

Na **terminálu** se budou zobrazovat stejné zprávy. Vzdálené nastavení a ovládání automatu bude pomocí kláves „+“, „-“, „mezera“, „s“.

„+“, zvýš 0x2B (ASCII),
„-“, sníž, 0x2D,
mezera - potvrzení 0x20,
s start svícení 0x53

Tabulka ACII znaků je na www.asciitable.com

Pro zjednodušení návrhu jednotky není nutné řešit souběh činností, ke kterému by došlo, pokud by jednotka byla v režimu nastavení a současně by přišel požadavek na start časovače. Pro jednoduchost se bude předpokládat, že se vždy daná činnost dříve dokončí, než se započne s činností další.

Z hlediska funkce reálného přístroje by tento předpoklad představoval omezení, které by snížilo použitelnost zařízení. Bylo by proto vhodné řešit úlohu jako nezávislé procesy. První proces je funkcí časovače, druhý je nezávislý a zajišťuje nastavení času. Dokud se nepotvrdí nastavení nové hodnoty času tlačítkem „=“, využíval by časovač naposled nastavenou hodnotu. Oba procesy by předávaly data do vybraných míst paměti RAM, kde by se tvořil „datový obraz“ v kódech ASCII znaků, který by následně využíval třetí, nezávislý, proces, který by zajišťoval zobrazení na LCD a na terminálu.

Na úrovni assembleru by bylo možné řešení opět způsobem prezentovaným na přednášce, tedy během celého programu ve smyčce, kde by se periodicky při každém průchodu prováděly činnosti odpovídající jednotlivým procesům, které by si navzájem předávaly výsledky prostřednictvím paměti RAM ve formě příznaků a dat.

Jiný způsob (za použití vyšších programových nástrojů) by byl orientován na některý operační systém reálného času, např. FREE RTOS (www.freertos.org) . To však není náplní problematiky HW vestavěných systémů.

Pokud by se při řešení úlohy řešitel shledal, že *potřebuje ještě detailněji specifikovat popis chování*, **doplň si sám** specifikaci, případně upraví a vše popíše.

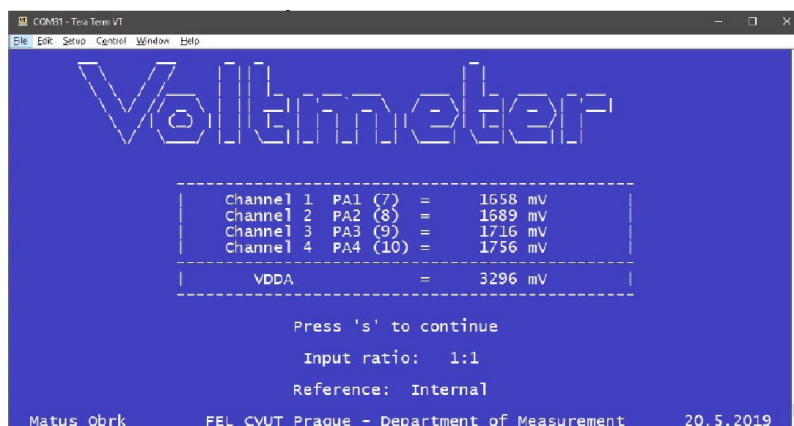
Formu zobrazení na terminálu je možno si zvolit podle uvážení. Je možno vypisovat do jednoho řádku a tento řádek přepisovat (znak CR - návrat kurzoru na začátek řádku 0x0A). V programu musí být vypnuta automatická generace znaku LF (Line Feed 0x0A) – odřádkování.

Je možno využít způsob zobrazení označovaný jako TUI - „Terminal User Interface“, případně také „Text User Interface“, kdy se s vyžitím programu PuTTY a [ANSI sekvencí](https://en.wikipedia.org/wiki/ANSI_escape_code) (např. na https://en.wikipedia.org/wiki/ANSI_escape_code) přepisuje celá obrazovka, anebo její vybrané části bez odřádkování. To je možné díky adresaci kurzoru. Příklad takového zobrazení je v BP Matuš Obrk, nebo BP Jan Holčák – viz DSPACE.CVUT.CZ. Popis využití ANSI sekvencí je také k dispozici v BP M. Obrka.

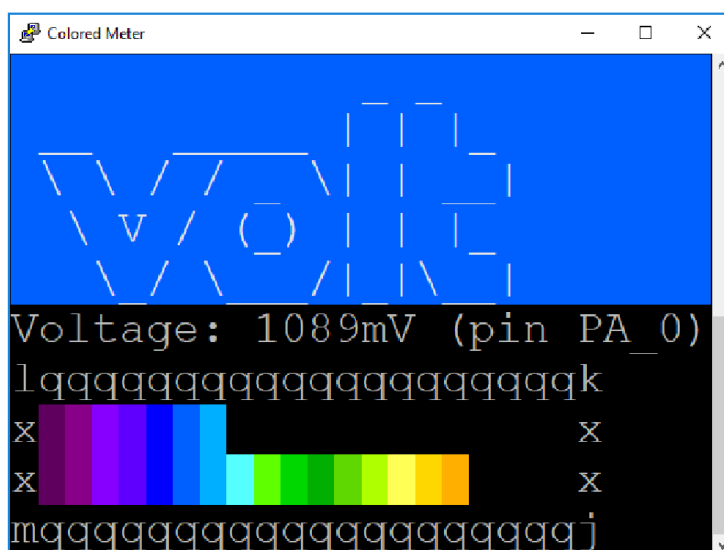
Je možno též měnit *barvu pozadí* i *barvu písma*. Tak je možno i realizovat jistou formu „bar – grafu“ ve směru řádku. Využije se znak mezera s nastavením vhodné barvy pozadí. Konkrétní vzhled a barvy se mírně odlišují podle daného terminálového programu. Velikost písma nelze programově měnit, je dána „ručním“ nastavením terminálového programu. V „obrázku“ se může využít velké písmo, jeho velikost je však všude stejná.

Použitím TUI může být komunikace s přístrojem názornější a přehlednější. Použití TUI v této úloze je však pouze doporučené. Pomocí bar- grafu by bylo možno znázorňovat velikost

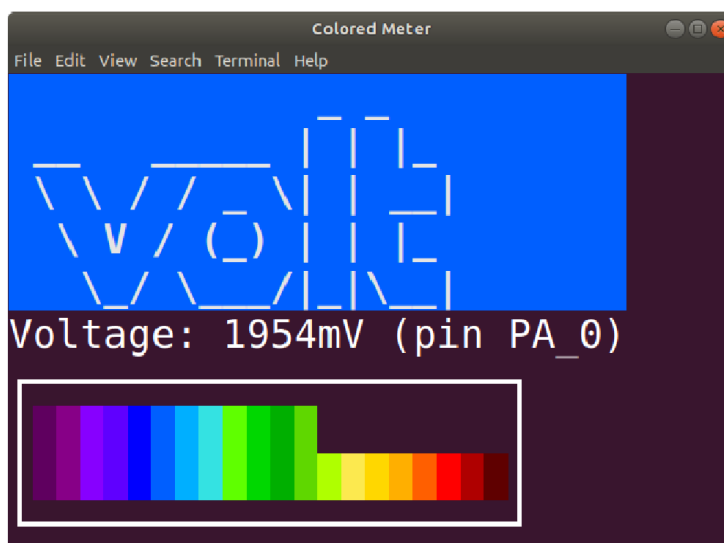
zbývajících času. Níže jsou příklady použití TUI z bakalářských prací řešených na katedře měření.



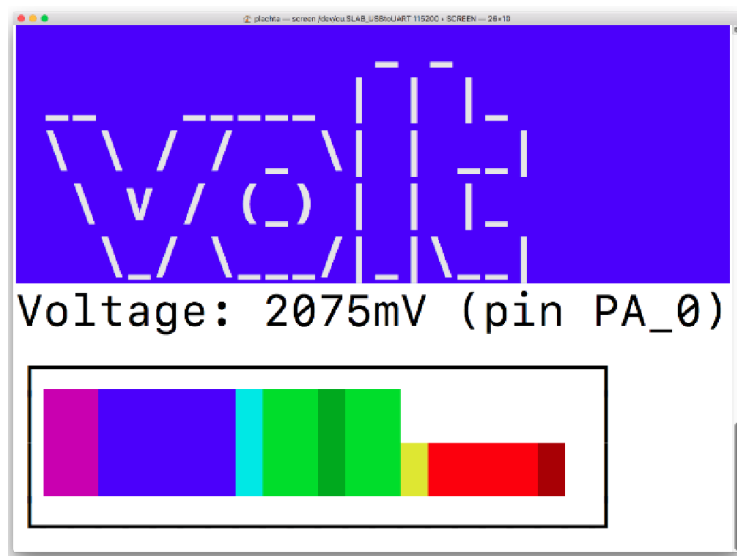
Příklad TUI z BP pana M. Obrka s využitím TeraTerm



Příklad TUI z BP pana J. Holčáka s využitím PuTTY



Příklad TUI z BP pana J. Holčáka s využitím gnome-terminal -u



Příklad TUI z BP pana J. Holčáka s využití terminal.app

Motivační problém

Podobné řídicí jednotky jsou umístěny v domovním rozvaděči, kde slouží k ***zapínání stykače motoru odsávače*** umístěného na střeše. Podle ročního období je různý požadavek na běh odsávače po stisku tlačítek v jednotlivých bytech. V létě je požadavek delšího běhu, v zimě je požadavek na kratší běh, protože by jinak docházelo k ochlazení bytů dlouhým odvětrávání. Tak by bylo možno naprogramovat možnost ručního nastavení různé velikosti doby sepnutí podle ročního období. Na základě denní doby (např. ráno v pracovní dny) a nastavení by jednotka podle nastavení mohla také sama aktivovat odsávání s vhodným intervalem.

Připojení jednotky pomocí obvodu UART navázaného na RS-232, nebo ještě lépe RS-485, by umožňovalo dálkový monitoring činnosti, změnu nastavení, kontrolu funkce, zastavení odsávání v krizových stavech, či opačně – dálkovou nucenou aktivaci odsávání, nastavení parametrů jednotky a kontrolu činnosti. Bylo by možno dálkově kontrolovat, zda nedošlo k poškození některého z paralelně zapojených tlačítek sloužících pro aktivaci časovače, tak, že by zůstalo stále sepnuté a zablokovalo tak činnost celého systému. Protože procesor STM32F100 má na čipu polovodičový teploměr, bylo by možno též dálkově zjišťovat teplotu v rozvaděči.

S využití dalšího bloku, jako např. optického snímače by bylo možno i monitorovat, zda se ventilátor po sepnutí skutečně otáčí „nezadírá“ a má obvyklé standardní otáčky. (*Pozn.: využití snímače teploty a optického snímače bude mimo jiné též náplní poslední řešené úlohy-návrhu systému sběru dat.*) Dalšími doplňkovými funkcemi pro účely monitorování by mohlo být počítání počtu sepnutí a aktivací ventilátorů a určení celkové doby jejich běhu, což by bylo možno využít pro plánování servisního zásahu při dosažení dané hodnoty.

Doplněním řídicí jednotky o vstupy pro připojení snímačů vnitřní a vnější teploty, vstupy pro připojení snímačů oxidu uhličitého a dalších plynů, by se mohla měnit v řídicí jednotku použitelnou v jednoduché vzduchotechnice domu.

Rozšířit by bylo možné i funkce tlačítka ovládajícího start. Např. počtem následných rychlých stisků uživatelského tlačítka nastavovat aktuální dobu aktivace ventilátoru jako násobek nastavené doby. Dlouhý stisk by pak mohl naopak sloužit k vypnutí právě běžícího odsávače.

Domácí příprava

V rámci přípravy se dopředu bude specifikovat funkce zařízení a jeho chování a komunikace s obsluhou. Vnější chování jednotky má být uživatelsky „vstřícné“, tedy aby se jednotka při servisním nastavování chovala tak, jak by obsluha právě očekávala a aby nebylo nutno číst uživatelský manuál. Tam, kde zadání nepostačuje pro specifikaci, si každý řešitel doplní požadavek na chování jednotky.

Obvodové řešení

Sériový komunikační kanál UART

Ve standardním zařízení ovládaném sériovým kanálem by se použilo rozhraní RS-485 (případně RS-232) pro které by se využil příslušný budič (ADM-485, příp. ADM-232L). Pro zjednodušení se pro komunikaci s blokem UART (signály RxD, TxD) budou využívat přímo signály z procesoru o napěťových úrovních LV CMOS (0, +3,3 V), které se napojí na převodník UART- USB s obvodem CH340G. Osadí se **propojka** na kontaktech mezi **+3,3 V a V_{CC}**, čímž se definují napěťové vysoké úrovně signálů v hodnotě + 3,3 V.



Převodník UART – USB s obvodem CH340

Je potřeba nainstalovat příslušný ovládač převodníku UART- USB s obvodem CH340. Otestování funkce převodníku- propojit signály Rx a Tx na převodníku UART- USB („loop back“). Spustit terminálový program, např. PuTTY (bez vlastního echování). Po stisku klávesy se díky propojení TxD na RxD objeví echování stisknuté klávesy.

Propojení převodníku UART – USB na procesor - **nejdříve propojit země** GND na GND. Propojit přijímač RxD (převodníku) na vysílač TxD a TxD z převodníku na RxD procesoru.

Připojení LCD zobrazovače

Zobrazovač **LCD** se ovládá prostřednictvím osmi datových signálů **DB7** až **DB0**, signálu **RS** (výběr „register select“), určující zda se přenášejí ASCII **znaky pro zobrazení**, nebo se předávají řídicí **povely**. Přenos se pak děje 8-bitově po jednotlivých Byte. Vstupním signálem R/W do LCD se rozlišuje zápis (R/W=0) nebo čtení (R/W =1) dat. Spádovou hranou signálu **E** se zapisují data do řadiče LCD.

Pro zjednodušení využijeme pevného nastavení **R/W = 0** a budeme data do řadiče LCD jen zapisovat.

Počet vodičů potřebných pro přenos dat lze dále **snížit** na polovinu, tedy **na čtyři**, pokud se využije **4-bitový** přenos. Zjednodušené řešení vystačí s přenosem **8-bitového datového slova** postupně ve **dvou nibble** (čtveřicích bitů) po vodičích DB7 až DB4 postupně za sebou. Nejdříve se však řadič do tohoto režimu musí nakonfigurovat zápisem příslušné sekvence **nibble** do řadiče.

Pro **přenos dat** do řadiče LCD využijeme **sério- paralelní posuvný registr**. Pokud se použije vstup dat do HD44780 v **šíři 8 bitů**, budou se signály **E** a **R/S** generovat samostatnými výstupními branami procesoru (GPIO). Pokud se použije přenos dat v **šíři 4 bitů**, může se přidat i signál **R/S** jako výstup posuvného registru. S ohledem na přepínání výstupů posuvného registru při jeho sériovém plnění je nutné **signál E** generovat výstupem **GPIO**.

Při komunikaci s řadičem LCD je nutno zohlednit jeho pomalou reakci, tedy je vhodné vkládat prodlevy tak, aby časy odpovídaly doporučením dle katalogového listu řadiče HD44780. Stejně tak je vhodné zařadit delší **prodlevu** (např. 0,5 sekundy) po zapnutí napájení a počátkem komunikace s řadičem.

Příloha – ovládání řadiče HD44780 použitého v zobrazovači LCD (2x20):

(Na internetu lze k heslu **HD44780** nalézt velké množství informací a popisů použití.)

Níže uvedené pomocné informace vycházejí z *původní laboratorní úlohy* z předmětu praktika z mikroprocesorové techniky <http://measure.feld.cvut.cz/groups/edu/pmt/index.html>

„*Připojení LCD k mikropočítači řady Atmel AT89C51*“, na stránce <http://measure.feld.cvut.cz/groups/edu/pmt/uloha10.html>

Protože některé tyto informace *mohou být užitečné i nyní*, jsou i zde přiloženy. Pokyny pro řešení úlohy uvedené v dalším textu se týkaly pouze úlohy v předmětu PMT.

Funkce zobrazovače s mikropočítačem:

Po počáteční úspěšné inicializaci displeje se na něm objeví zpráva **Displej Funguje .** Po příchodu ASCII znaku CR (Carriage Return) - kód 0Dh se displej vymaže a ukazatel vrátí na výchozí pozici vlevo.

Na horním řádku se bude zobrazovat text, který přichází do mikropočítače prostřednictvím rozhraní RS232. Nezobrazitelné znaky (znaky mimo rozmezí 20h až 7Fh) jsou ignorovány. První přichodící znak se zapisuje na první pozici zleva, další budou zaplňovat displej postupně doprava. Po příchodu ASCII znaku CR se displej opět vymaže a ukazatel vrátí na výchozí pozici vlevo.

Nepovinná část:

Doporučení pro vývoj programu:

Pro zobrazování zpráv se doporučuje vytvořit nejdříve rutinu pro zápis jednoho znaku (např. z akumulátoru) na displej, která se použije při odesílání delších zpráv. Pro ověření správné funkce využijte zobrazování znaků odesílaných z Mon51 po sériové lince (pozor – na konci displeje je nutné provést přechod na další řádek nebo návrat kurzoru zpět!!).

Zdrojový kód koncipujte jako univerzální knihovnu, kterou budete moci jednoduše využívat i v navazujících úlohách.

Zobrazovače LCD

Pro komunikaci s obsluhou se v řadě přístrojů používají LCD zobrazovače (zobrazovače s kapalnými krystaly). Tyto zobrazovače mohou být v několika základních provedeních.

- **Segmentový zobrazovač LCD**

Segmentové zobrazovače (např. sedmisegmentové), známé např. z jednoduchých digitálních multimetrů, se používají pro zobrazení číslic.

Specializované LCD zobrazovače obsahují segmenty pro znázornění alfanumerických znaků a současně mívají vně textového pole i oddělené speciální symboly (piktogramy). Používají se např. ve spotřební elektronice (CD přehrávačích, radiopřijímačích, programovatelných regulátorech teploty,...), automobilech nebo v měřicích přístrojích. Tyto specializované a na zakázku vyvinuté a vyráběné LCD zobrazovače se uplatňují pouze v zařízeních, která se vyrábějí ve velkých sériích, kde se proto rozpustí nutné vývojové náklady.

- **Mozaikový grafický zobrazovač LCD**

V případě, že je potřeba zobrazovat číslice a texty, případně grafiku, je možno použít LCD mozaikové zobrazovače. Takový zobrazovač LCD je tvořen maticí stejných malých čtvercových plošek, které se jeví jako body symetricky uspořádané ve obdélníkovém rastru. Aktivace (zvýraznění) bodu a jeho odlišení od neaktivního okolí se děje dynamicky po jednotlivých řádcích (případně sloupcích) matice. Díky optické setrvačnosti se matice jeví, jako by najednou byly aktivovány všechny vybrané body v matici.

Takovéto displeje lze vidět např. v digitálních osciloskopech, PDA nebo v mobilních telefonech, kde jsou doplněny o piktogramy pro indikaci síly signálu nebo stavu baterií.

LCD zobrazovací matice, používané v přístrojích mohou být určeny buď pouze pro zobrazení textů a dalších znaků, nebo i pro grafické zobrazení. Malé LCD zobrazovače pro použití v přístrojích mívají organizaci např. 64 x 64, 128 x 64, 128 x 128 bodů a mohou se využívat pro zobrazení znakové i grafické informace.

- **Mozaikový znakový zobrazovač LCD**

Pokud je v přístroji zapotřebí zobrazovat pouze znaková hlášení (obsahující text, číslice, případně jednoduché symboly) je použití velké matice zbytečně drahé a postačuje použít pouze zobrazovač LCD s maticí o organizaci 80 x 8 bodů. Místa zobrazení znaků jsou pevně určena.

Znaky v této mozaice se zobrazují v matici 5 x 8 bodů. Lze tedy znázornit vedle sebe až 16 znaků. Pro zjednodušení návrhu zařízení se takový zobrazovač LCD vyrábí ve formě modulu, který již obsahuje integrované obvody řadiče. Řadič, např. často používaný typ HITACHI [HD44780](#), zajišťuje dekódování znaků a dynamické obnovování zobrazované informace na LCD. Pro uživatele je činnost spojená s řízením vlastního LCD zobrazovače zcela transparentní.

Řadič v sobě obsahuje paměť RAM, do níž se zapisují ASCII kódy znaků, které mají být zobrazeny. Uživatel pouze pomocí paralelního rozhraní předává tyto znaky do řadiče.

- **Použití zobrazovače LCD s řadičem Hitachi HD44780**

Řadič Hitachi HD44780 je nejpoužívanějším řadičem znakových LCD displejů. Jeho obsluha spočívá v zápisu ASCII hodnoty znaků na určené adresy v paměti tohoto řadiče. Řadič je schopen zobrazovat běžné ASCII znaky, které obsahuje ROM generátoru znaků. Řadič má také 8 pozic pro uživatelsky definované znaky.

Rozhraní řadiče je přizpůsobeno procesorům firmy Motorola, proto jej není možné připojit přímo k adresové sběrnici a řídicím signálům procesoru řady 8051, ale je nutné řídicí signály dekódovat nebo emulovat programově.

Řídící signály HD44780

Signál	Význam / Funkce
E	Enable – určuje platnost dat na sběrnici (aktivní v 1)
RS	Register Select – určuje zda se jedná o data (RS = 1) nebo o instrukci (RS = 0)
R/W	Read/Write – určuje zda se jedná o čtení (R/W = 1) zápis (R/W = 0)

Z tabulky vidíme první odlišnost řídících signálů od architektury i51, u které jsou signály *Read* a *Write* rozděleny do dvou pinů a jejich aktivita rozhoduje i o platnosti dat na sběrnici.

Při návrhu zařízení tak musíme rozhodnout, zda budeme řídící signály dekódovat logickou sítí, nebo emulovat programově např. pomocí brány *Px*.

Rozhraní displeje obsahuje také 8 datových vodičů, po kterých se zasílají vlastní data nebo instrukce. Při inicializaci displeje lze zvolit mezi komunikací po 8 bitech, nebo po 4 bitech. V prvním případě zasíláme data v rámci jednoho přístupu, v druhém případě je nutné použít dva přístupové cykly. Druhý případ ale šetří piny procesoru.

Instrukce řadiče HD44780

Instrukce	Kód instrukce										Popis	Délka provádění (fosc = 250 kHz)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Smaž displej (Clear Display)	0	0	0	0	0	0	0	0	0	1	X	1,64 ms
Návrat kurzoru (Cursor At Home)	0	0	0	0	0	0	0	0	1	-	X	1,64 ms
Nastavení módu vstupu dat (Entry Mode Set)	0	0	0	0	0	0	0	1	ID	SH	X	39 μs
Nastavení módu displeje Display On/Off Control)	0	0	0	0	0	0	1	D	C	B	X	39 μs
Nastavení posouvání (Cursor or Display Shift)	0	0	0	0	0	1	SC	RL	-	-	X	39 μs
Nastavení funkce (Function Set)	0	0	0	0	1	DL	N	F	-	-	X	39 μs
Nastavení adresy v paměti CGRAM (Set CGRAM Address)	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	X	39 μs
Nastavení adresy v paměti DDRAM	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	X	39 μs

(Set DDRAM address)												
Čtení příznaku <i>BF</i> a adresy (Read Busy Flag and Address)	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	X	0 μ s
Zápis dat do RAM (Write Data to RAM)	1	0	D7	D6	D5	D4	D3	D2	D1	D0	X	43 μ s
Čtení dat z RAM (Read Data from RAM)	1	1	D7	D6	D5	D4	D3	D2	D1	D0	X	43 μ s

CG RAM - paměť znakového generátoru (paměť fontů)

DD RAM - paměť displeje

ID = 0 - po zápisu do DD RAM je ukazatel dekrementován

ID = 1 - po zápisu do DD RAM je ukazatel inkrementován

SH = 0 - nastaví posun kurzoru po zápisu (směr dle ID)

SH = 1 - nastaví posun celého displeje po zápisu (směr dle ID)

D = 0 - vypnutí displeje (D = 1 – zapnutí)

C = 0 - vypnutí kurzoru (C = 1 – zapnutí)

B = 0 - vypnutí blikání znaku na kurzoru (B = 1 – zapnutí)

SC = 0 - posunutí kurzoru (bez zápisu)

SC = 1 - posunutí displeje (bez zápisu)

RL = 0 - nastavení směru posunu doleva (RL = 1 – doprava)

DL = 0 - nastavení datové sběrnice na 4 bity (DB7..DB4)

DL = 1 - nastavení datové sběrnice na 8 bitů (DB7..DB0)

N = 0 - nastavení počtu řádek na 1

N = 1 - nastavení počtu řádek na 2 nebo 4

BF = 0 - připraveno pro příjem (BF = 1 – nesmíme zapisovat)

- - na hodnotě nezáleží

Adresy v paměti DD RAM

Jednotlivé zobrazované znaky jsou uchovány v paměti DD RAM. Přiřazení jednotlivých pozic paměťovým buňkám je závislé na počtu řádků displeje.

Pro použitý displej odpovídá adresa 00h znaku vlevo nahoře (tedy začátku prvního řádku). Začátku druhého řádku odpovídá adresa 40h.

00 h	01 h	02 h	03 h	04 h	05 h	06 h	07 h	08 h	09 h	0A h	0B h	0C h	0D h	0E h	0F h	10 h	11 h	12 h	13 h	14 h
40 h	41 h	42 h	43 h	44 h	45 h	46 h	47 h	48 h	49 h	4A h	4B h	4C h	4D h	4E h	4F h	50 h	51 h	52 h	53 h	54 h

Vzorová posloupnost příkazů pro ověření funkce displeje

Krok	Instrukce										Displej	Operace
	R S	R W	DB 7	DB 6	DB 5	DB 4	DB 3	DB 2	DB 1	DB 0		
1	Zapnutí napájení, RESET											Inicializace . Nic se nezobrazuj e
2	Nastavení funkce											8bitové rozhraní, 2 řádky, znaky 5 x 8
	0	0	0	0	1	1	1	0	*	*		
3	Nastavení módu displeje										—	Zapne displej a kurzor. Všechny znaky jsou po inicializaci mezery.
	0	0	0	0	0	0	1	1	1	0		
4	Nastavení módu vstupu dat										—	Po každém zápisu se kurzor posune doprava.
	0	0	0	0	0	0	0	1	1	0		
5	Zápis dat do RAM										D_	Zápis “D”. Kurzor se posune o jeden znak doprava.
	1	0	0	1	0	0	0	1	0	0		
6	...											
7	Zápis dat do RAM										DISPLEJ_	Zápis “J”. Kurzor se posune o jeden znak doprava.
	1	0	1	0	0	0	1	0	1	0		
8	Nastavení adresy v DD RAM										DISPLEJ	Nastaví novou adresu v paměti dat. Tím se změní poloha kurzoru.
	0	0	1	1	0	0	0	0	0	0	—	
9	Zápis dat do RAM										DISPLEJ	Zápis “F”.

	1	0	0	1	0	0	0	1	1	0	F_	Kurzor se posune o jeden znak doprava.
10	...											
11	Zápis dat do RAM										DISPLEJ	Zápis "E".
	1	0	0	1	0	0	0	1	0	1	FUNGUJE —	Kurzor se posune o jeden znak doprava.

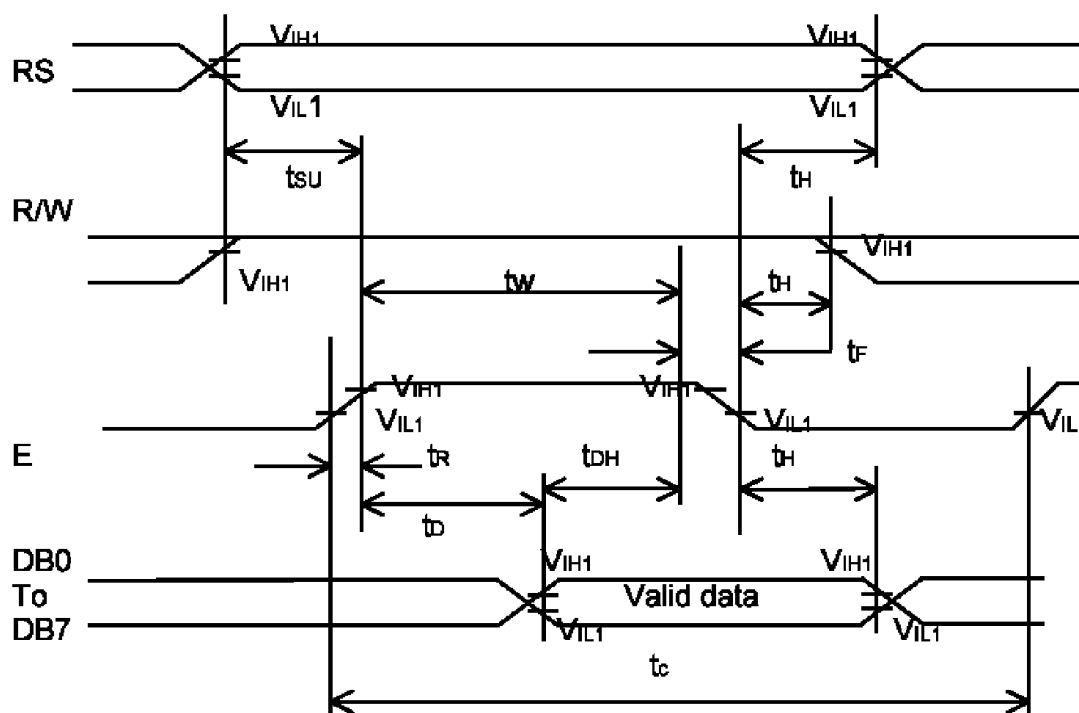
Sekvence instrukcí uvedená v tabulce zaručuje funkční (lze použít i jiné) nastavení displeje a zobrazení požadovaného textu na displeji.

Zapojení vývodů displeje - dvouřadý konektor

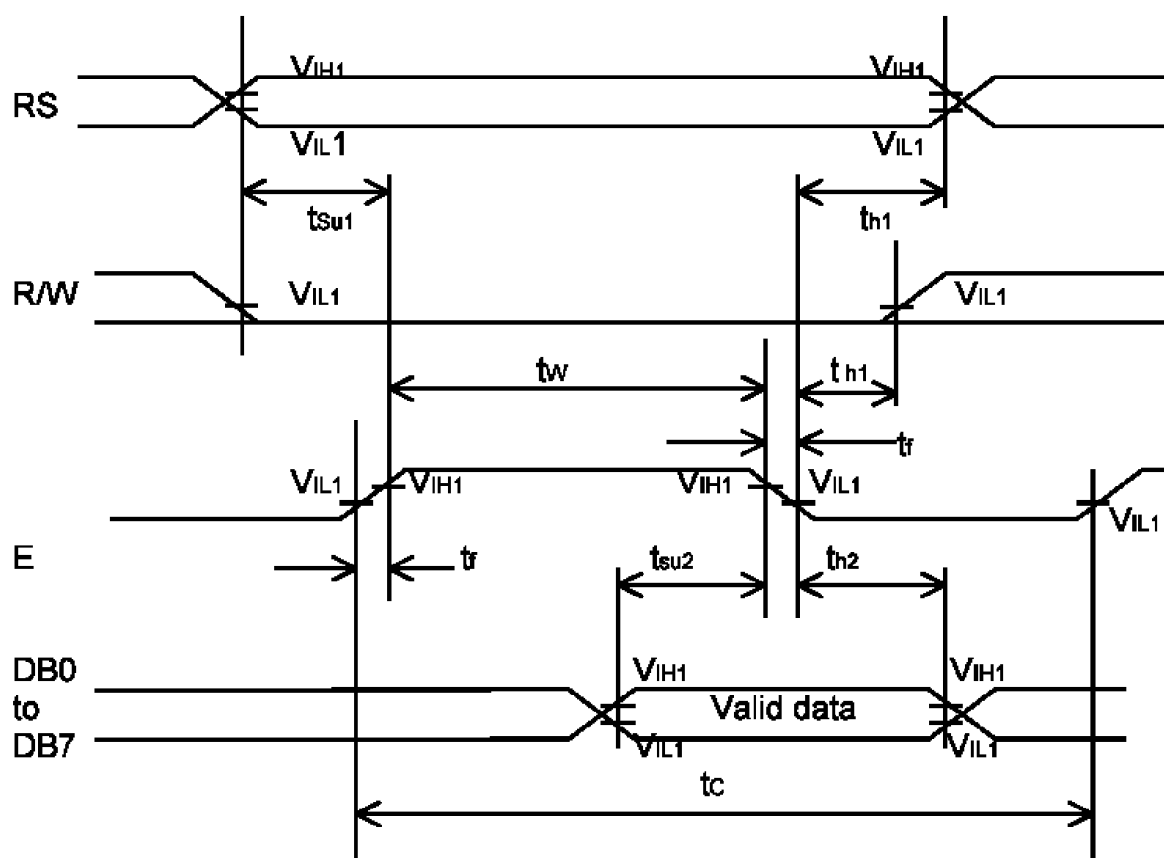
Pin	Označení	Funkce
1	VSS	GND
2	VDD	Napájení logiky 5V
3	VEE	Napětí pro budič LCD (0 až +5V)
4	RS	Data/instrukce
5	R/W	Čtení/zápis
6	E	Signál povolení přenosu (sestupná hrana)
7	DB0	Datová sběrnice bit 0
8	DB1	Datová sběrnice bit 1
9	DB2	Datová sběrnice bit 2
10	DB3	Datová sběrnice bit 3
11	DB4	Datová sběrnice bit 4
12	DB5	Datová sběrnice bit 5
13	DB6	Datová sběrnice bit 6
14	DB7	Datová sběrnice bit 7

- Řízení kontrastu *VEE* je vhodné připojit přes odpor $150\ \Omega$ na zem.

Časové diagramy zápisu a čtení LCD displeje



Čtecí cyklus



Zápisový cyklus

Dekódování řídicích signálů

Pro připojení řadiče do prostoru externí datové paměti lze vyhradit např. zvláštní adresu pro zápis do řadiče a čtení z řadiče. Rovněž lze vyhrazením adresy rozlišovat data od instrukcí. Signál *E* v takovém případě můžeme dekodovat ze signálů */RD* a */WR*.

A0	A1	RS	R/W	Funkce
0	0	0	0	Zápis instrukce do řadiče
0	1	1	0	Zápis dat do paměti řadiče
1	0	0	1	Čtení příznaku <i>BF</i> a stavu čítače adres z řadiče
1	1	1	1	Čtení dat z paměti řadiče

Jedna z možností dekodování řídicích signálů je uvedena v tabulce. Signál *E* bude v takovém případě odvozen od signálů */RD* a */WR* a adresových vodičů daných přiděleným adresovým prostorem (např. *A15* pro mapování do prostoru *08000h ÷ 0FFFFh*).

Při dekodování řídicích signálů je nutné dbát na dodržení správného časování! Bohužel u použitého řadiče displeje je minimální doba trvání jednoho cyklu delší než cyklus čtení/zápisu u použitého procesoru 89C52. Řešením by bylo použití rychlejšího provedení řadiče, nebo snížení hodinového kmitočtu procesoru.

Emulace řídicích signálů

V předchozím případě se vyskytnul problém s nedodržením minimálních požadovaných délek trvání jednotlivých řídicích signálů. Maximální délky však nijak omezeny nejsou, proto je možné řídicí signály emulovat např. piny brány *Px*. V takovém případě zapojíme datovou sběrnici displeje na jednu bránu (např. *P1*) a řídicí signály na volné piny jiné brány.

Setb/Clr RS ;rozlišuje data / instrukce
Setb/Clr RW ;rozlišuje mezi čtením / zápisem
Setb E ;aktivuje rozhraní – select
Mov P1,DATA ;zapiše data / instrukci do displeje
Clr E ;určuje platnost dat

Po provedení zápisu je nutné počkat na dokončení požadované operace. Dokončení indikuje příznak *BF*, ale při dodržení časů uvedených v tabulce instrukcí jej není nutné testovat.