



CZ NÁVOD K OBSLUZE

Experimentální sada

Mikrokontroléry 10104

Obj. č. 19 22 86



Vážený zákazníku,

děkujeme Vám za Vaši důvěru a za nákup experimentální sady.

Tento návod k obsluze je součástí výrobku. Obsahuje důležité pokyny k uvedení výrobku do provozu a k jeho obsluze. Jestliže výrobek předáte jiným osobám, dbejte na to, abyste jim odevzdali i tento návod k obsluze.

Ponechejte si tento návod, abyste si jej mohli znovu kdykoliv přečíst!



Úvod

Mikrokontrolér je programovatelný ovladač, který ve své podstatě funguje na velmi jednoduchém principu. Mikrokontroléry se nacházejí téměř ve všech elektronických domácích spotřebičích, herních zařízeních a dokonce i bezpilotních kosmických plavidlech. Zařízení provádí předem naprogramované procesy. Nejdůležitější je však vhodná volba mikrokontroléru proto, aby plně odpovídal potřebám požadované aplikace. Na výběr je celá řada těchto zařízení s různými programovacími jazyky. Většina z nich však podporuje programovací jazyk C, Basic nebo Assembler. Programování většinou vyžaduje programovací jednotku. Seznamte se proto prostřednictvím této sady se všemi možnostmi mikrokontrolérů.

Mikrokontrolér skýtá možnost širokého využití. Programování může být realizováno i pomocí dvou tlačítek. Programovací tlačítko PBC přitom pracuje jen s několika málo příkazy. Jeho funkce však lze dále rozvíjet a celý systém tak dále „učit“ (funkce Learn). Naprogramované aplikace je možné kdykoliv upravovat aniž by bylo nutné provádět další složitě úkony.

Mikrokontroléry slouží zejména pro měření, regulaci výkonu a řízení různých systémů. Naprogramovaný mikrokontrolér se používá jako součást různých elektronických obvodů a proto je nezbytná jeho alespoň základní znalost. V sadě se proto seznámíte s postupem při jeho programování. Efekt, který mikrokontrolér přinese, se projevuje mnohem výrazněji, než u jiných systémů. Všechny programovací jazyky jsou ve svém základu velmi podobné a převod do jiného jazyka je proto mnohem jednodušší.

TPS-Mikrokontrolér je vlastně jednočipový počítač (mikropočítač / mikroprocesor), který má 4 digitální vstupy E1 – E4 a čtyři digitální výstupy A1 – A4. Navíc disponuje dvěma analogovými vstupy AD1 a AD2 a kvazi-analogovým výstupem PWM. Resetovací vstup s připojeným tlačítkem Reset znovu spustí program. Kontrolér je napájen 3 bateriemi AA články s provozním napětím 4.5 V. Jeho provoz je však možný při napětí v rozsahu od 2,2 V do 5,5 V.

Technické údaje

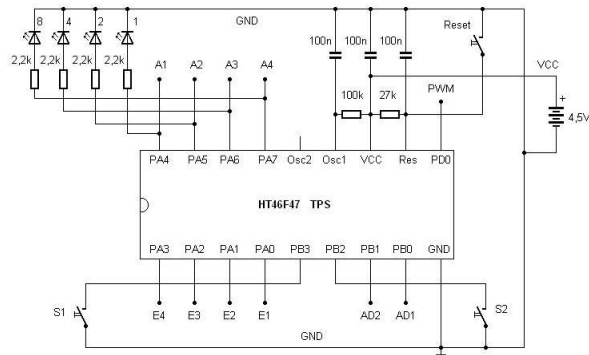
Typ mikroprocesoru	HT46F47
Hodinová frekvence	2 MHz
Interní EEPROM	128 bajtů
Zdroj napájení	2,2 V až 5,5 V DC
Spotřeba proudu	1 mA při 4,5 V
4 výstupní porty	až 10 mA
1 PWM výstup	až 10 mA
4 vstupní porty	klidový stav 1
2 analogové vstupy	0 V DC
Tlačítko (2 polohy)	klidový stav 1

Obsah experimentální sady

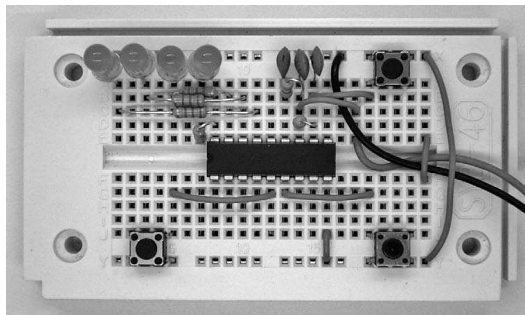
Základní deska
Příhrádka na baterie
Vodiče
Mikroprocesor HT46F47 s TPS firmware
3 tlačítka
4 LED, 5 mm, červené
1 LED, 5 mm, zelená
1 LDR (fotorezistor)
3 diskové kondenzátory 100 nF
1 elektrolytický kondenzátor 47 µF
5 rezistorů 2,2 kΩ
1 rezistor 10 kΩ
1 rezistor 27 kΩ
2 rezistory 100 kΩ

K programování jsou zapotřebí tlačítka S1 a S2 a jednoduchý LED displej pro zobrazení výstupů A1 – A4. K dispozici je celkem 14 jednoduchých příkazů (command) s dalšími daty nebo sub-commands. Příkazy a data jsou kódovány ve 4-bitová binární čísla od 0000 do 1111 (v desítkové soustavě 0 - 15) a jsou přímo viditelné na LED displeji. Příslušné číslo je naprogramováno do tlačítka S1 po jeho stisknutí při programování. S2 přepíná mezi příkazem a daty a navyšuje adresu v příkazovém řádku. Programová struktura je tak velmi jednoduchá a pochopitelná.

Schéma základního zapojení systému



Základní zapojení se dvěma tlačítky

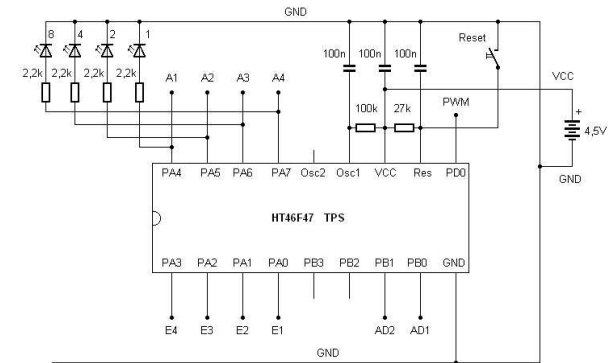


Součástí továrního nastavení mikroprocesoru je několik základních programů, které mohou být spuštěny přímo nebo v jednotlivých krocích. Seznamte se proto nejprve s hardwarovou výbavou procesoru a teprve poté se pusťte do vlastního programování.

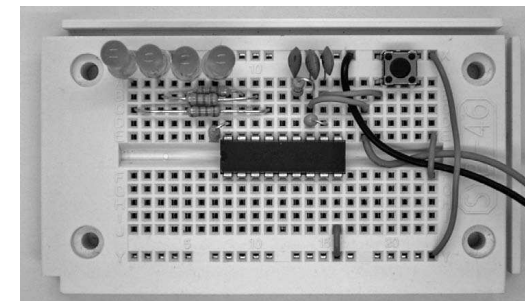
V prvním experimentu spustíte základní programy a vyzkoušíte jejich hlavní funkce. K pokusu budou také zapotřebí pouze základní komponenty. Podrobný popis jednotlivých příkazů bude v další části návodu. Zapotřebí budou následující součástky:

- připojení ke zdroji napětí (záporný **GND** „-“ a kladný pól **VCC** „+“)
- blokovací kondenzátor o hodnotě 100 nF zapojený mezi VCC a GND
- Reset-rezistor ve směru kladného proudu a reset-kondenzátor ve směru záporného proudu
- Rezistor 100 kΩ ve směru kladného proudu a kondenzátor ve směru záporného proudu

Mikroprocesor má vnitřní RC oscilační obvod. Rezistor na vstupu OSC1 udává frekvenci cyklů. Při impedanci 100 kΩ, bude frekvence přibližně 2 MHz. Připojený kondenzátor slouží pouze k blokování a nemá žádný vliv na hodnotu frekvence cyklů. Vstup OSC2 zůstává nepoužitý. Připojit jej můžete do dráhy kladného proudu a odstranit tak ¼ z frekvence cyklu.



Použijte piny v horní i dolní části desky pro připojení negativního pólu GND. Příhrádka pro baterie má dva vývody – **černý** (negativní pól) a **červený** (kladný pól). Při zapojování obvodu s mikroprocesorem vždy zachovejte správnou polaritu. V opačném případě může dojít k jeho nevratnému zničení. Použijte krátký vodič pro upevnění obou vodičů k desce. Zabráníte tím vytržení přívodních vodičů z desky. Připojené vodiče mohou být k desce připojeny trvale. Pro vypnutí / deaktivaci obvodu pak stačí pouze vyjmout jednu z baterií.



Použijte resetovací tlačítko a připojte 4 LED k rezistorům 2,2 kΩ. Rezistory jsou nezbytné pro první experimenty. Zachovejte jejich správné pořadí. A1 je připojen k levé LED. A4 k LED v pravé části. Binární zobrazení s nejvyšší bitovou hodnotou je vlevo.

Blikač

Nyní vložte 3 kusy běžných baterií o napětí 1,5 V. Použít můžete také nabíjecí Ni-MH baterie. Tím dojde ke spuštění zkušebního programu, při kterém se budou LED střídavě rozsvěcovat (levá a pravá LED) při frekvenci přibližně 1 Hz. Přehled programů zobrazuje pouze programy s 5 řádky. Přepínání mezi oběma LED je řízeno příkazem s prodlevou 0,5 s. Skok impulzu na začátek zajišťuje následné opakování procesu. Jednotlivé příkazy budou vysvětleny v následující části. Na názorném příkladu zjistíte jak je programování jednoduché. Mikroprocesor má vestavěnou funkci interpretace, která rozpoznává a vykonává jednoduché příkazy různých programovacích jazyků. Programy tak zajišťují vysokou kompaktnost, než je řízení výkonu prostřednictvím jiných systémů. V příkladu je uveden rozsah od 20h v postupné řadě (v desítkové soustavě 32). Programy v horním adresním rozsahu je možné spustit později z příslušných aplikací. Adresy mohou být přepsány vlastním programovým kódem. Mikrokontrolér navíc může být kdykoliv resetován do původního, továrního stavu s obsahem základních programů.

Adresa	Příkaz	Data	Popis
20	1	1	LED 1
21	2	8	Prodleva 500 ms
22	1	8	LED 8
23	2	8	Prodleva 500 ms
24	3	4	Jump -4 (návrat)

V případě, že nedojde k očekávanému procesu, zkontrolujte nejprve polaritu připojeného napětí k LED. Dále je možné v obvodu naměřit určité hodnoty. Použijte proto digitální multimetr s měřícím rozsahem maximálně 20 V. Napětí obvodu měřte vždy proti zemi (negativní pól).

Hodnoty napětí v obvodu:

napětí 4,5 V DC proti zemi

Reset – 4,5 V

Osc1 – 1,5 V

E1 až E4 – 4,5 V

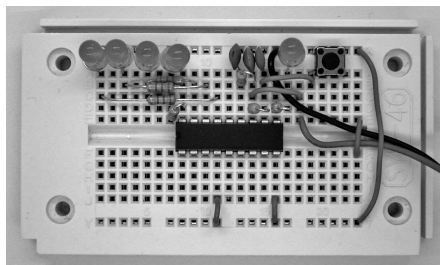
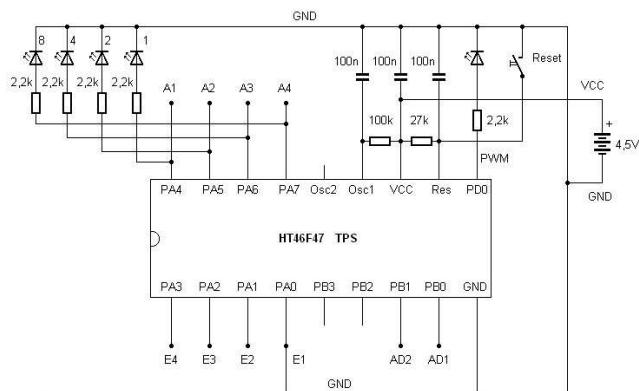
A1 – střídavě přerušované

A2, A3 – 0 V

A4 – střídavě přerušované

Binární čítač a výstup PWM (Pulse Width Modulation)

Všechny digitální vstupy jsou vedeny s vnitřním odporem proti kladnému signálu (pull-up rezistor), a mají klidové napětí ve výši provozního napětí. Připojit můžete každý ze vstupů k GND pomocí vodiče nebo svorky. Při spuštění programu dochází ke zjišťování stavu portu a jeho vyhodnocování. Konkrétní zapojení může být přivedeno na GND tak, že dojde k zaznamenání stavu 0. Různý výsledek pak představuje konkrétní program.



Použití PWM-LED

Připojte E1 k negativnímu pólu GND. Po resetu mikroprocesoru dojde ke spuštění druhého programu. Výstupní podmínky jsou pak sečteny binárně. Podmínky 0000 (v decimální soustavě 0) až 1111 (v decimální soustavě 15) pak prochází nepřetržitě. Program využívá proměnnou A pro jednoduchý součet a pro výstup do digitálních výstupu (PWM output). Příkazy 7 a 5 mají vedlejší funkce zapsané jako data.

Binární čítač s LEWD a PWM výstupem

Adresa	Příkaz	Data	Popis
25	7	1	A = A + 1
26	5	4	Port = A
27	5	9	PWM = A
28	2	6	Prodleva 100 ms
29	3	4	Jump -4 (návrat)

Tento program může být použit jako zkušební program pro čtení binárních čítačů, které musíte ovládat pro vlastní programování. Každá ze čtyř LED diod představuje jeden bit. Proto může být následně zobrazeno 4-bitové číslo. LED diody se nazývají 8, 4, 2 a 1 stejně jako jejich hodnoty ve schématu. Přidání příslušných hodnot má za následek desetinné číslo. Zapsáno hexadecimálně, čísla 10 - 15 jsou zastoupeny velkými písmeny A až F.

8	4	2	1	Decimální	Hexa-decimální
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

Tento program je možné využít jako flash dekodér při různé frekvenci. Nejbližší vyšší výstup má poloviční frekvenci nebo dvojnásobek časové periody:

A1: 200 ms

A2: 400 ms

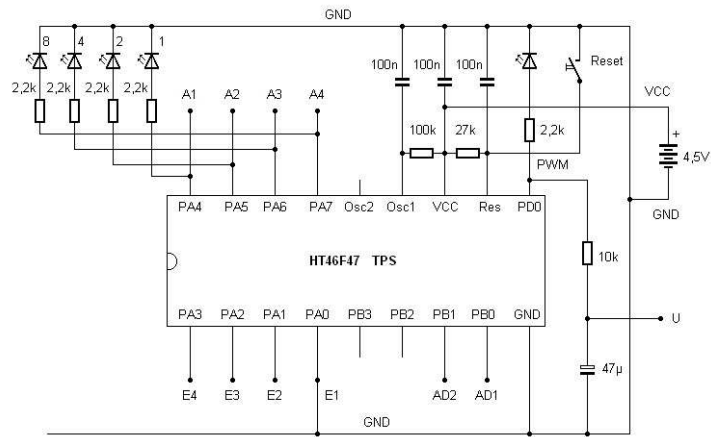
A3: 800 ms

A4: 1600 ms

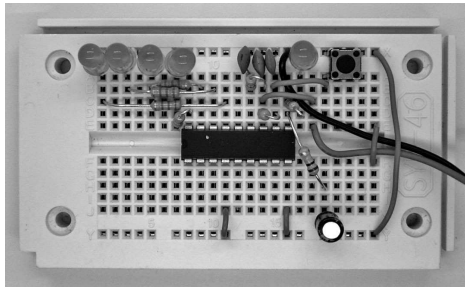
Navíc jsou navýšené číselné hodnoty výstupem **PWM**. PWM signál je obdélkový signál s frekvencí přibližně 16 kHz. Délka pulsu je řízena, proto poměr „puls-pauza“ určuje průměrnou dobu pro trvání aktivace a tím i jas LED. Jas LED je řízen v celkem 15. úrovních (0 – 15).

PWM signál může být usměrněn na stejnosměrné napětí s RC low-pass filtrem. Výstup PWM se tak stává analogovým výstupem. Tento program generuje stejnosměrné napětí, které se postupně zvyšuje z 0 V až do 4,5 V. Průběh napětí je vhodné sledovat použitím osciloskopu.

Low-Pass Filter na PWM výstupu



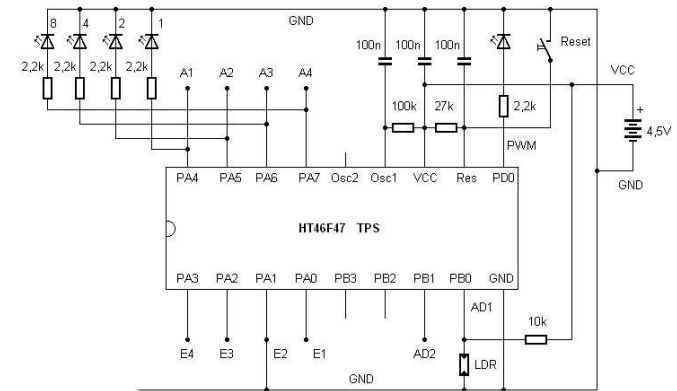
PWM výstupní napětí



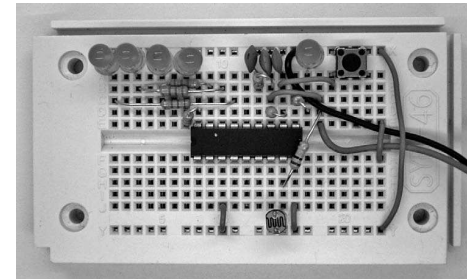
Analogově-digitální převodník (ADC)

Připojení E2 na GND a stisknutí tlačítka reset se spustí program analogově-digitálního převodníku (AD-converter). Analogové napětí se na analogovém vstupu AD1 měří a převádí na digitální číselné hodnoty. Vzhledem k tomu, že mikrokontrolér pracuje se 4-bitovými hodnotami, výsledkem analogové digitální konverze je číslo od 0 do 15. Výsledek 0 představuje vstupní napětí 0, výsledek 15 odpovídá provoznímu napětí (v tomto případě 4,5 V). Analogově-digitální hodnota je výstupem binárního čísla u 4 LED přenášena prostřednictvím PWM výstupu. Připojte napěťový dělič tvořený pevným rezistorem a fotorezistorem (LDR) do analogového výstupu AD1.

Připojení fotorezistoru



Fotorezistor připojený na vstupu AD1



Tento program je velmi podobný předchozímu programu (digitální výstupy a PWM výstup). V prvním řádku je příkaz, kterým se provádí konverze analogových hodnot.

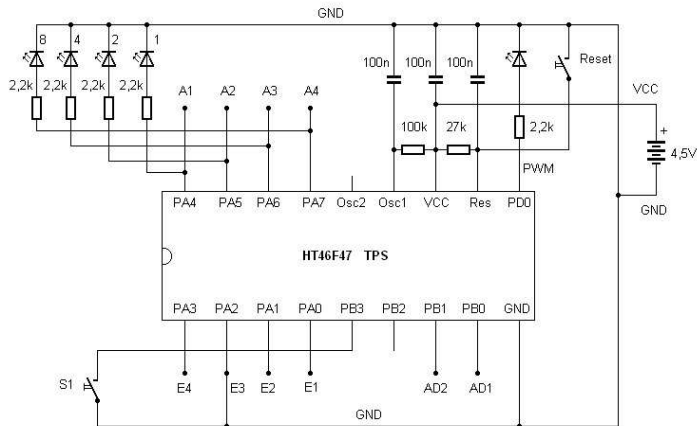
Adresa	Příkaz	Data	Popis
2A	6	9	A = AD1
2B	5	4	Port = A
2C	5	9	PWM = A
2D	2	6	Prodleva 100 ms
2E	3	4	Jump -4 (návrat)

Otestuje funkci programu při aplikaci světelné expozice – vystavení světlu. Čím intenzivnější je světelná expozice, tím menší je napětí na AD1 a naopak. Čím slabší je světelný tok, tím větší jsou AD hodnoty a tím i maximální jas LED na PWM výstupu. Zaznamenejte hodnoty binárních řádků z LED displeje a pokuste se přizpůsobit jas, například na poloviční hodnoty. Digitální hodnota je pak 0111 nebo 1000. Dochází-li k efektu „mihotavého“ světla, což znamená, že dochází k rychlému přepínání mezi oběma hodnotami.

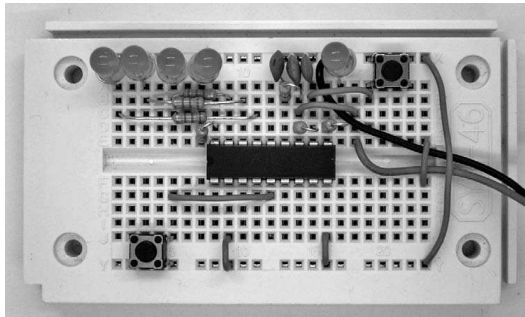
Generátor náhodných hodnot

Propojením E3 s GND dojde ke spuštění základního programu – generátoru náhodných čísel. Tlačítko S1 zajišťuje vyhodnocování podmínek. Připojený vstup má interní pull-up rezistor, kterým prochází kladné Vcc napětí. Tlačítko je připojeno k GND. Stisknutím tlačítka dojde k přepnutí vstupu S1 do hodnoty 0.

Program generátoru náhodných čísel



Propojení E3 s GND (Jumper)



Program využívá podmíněný příkaz **Jump** (skok / přeskočení). Pokud je vstupní podmínka S1 v hodnotě 0, dojde k přeskočení (Jump) následujícího příkazu. Stisknutím tlačítka je uveden do stavu hodnoty 0, a tím se zvyšuje proměnná A. To vede k rychlému součtu podmínek při spuštění. Po uvolnění tlačítka je stav dále zachován. Vzhledem k vysoké rychlosti součtu, nelze nijak ovlivnit konečný stav na výstupu, jedná se tak vždy o náhodný výsledek.

Adresa	Příkaz	Data	Popis
30	5	4	Port = A
31	C	E	S1 = ?
32	7	1	A = A + 1
33	3	3	Jump -3

Krátkým stisknutím dochází k vygenerování nového výsledku. Otestujte funkci programu opakovaným vygenerováním různých výsledků. Po dostatečném počtu spuštění, bude zřejmé, že se dochází k různým výstupním hodnotám (výsledkům). Program je tak vhodný například pro aplikaci systému „Herní kostky“.

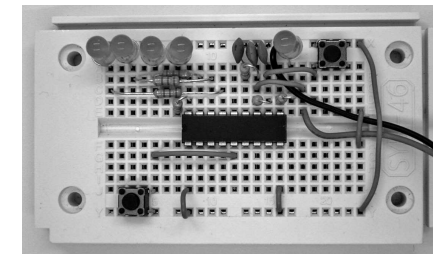
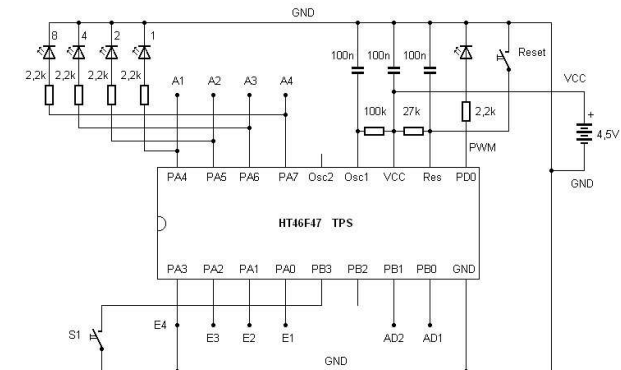
Program plní funkci čítače s maximální rychlostí matematických operací. Přitom je nejlépe patrná funkce a pracovní rychlost, při které mikroprocesor zpracovává data. Po stisknutí tlačítka dochází u výstupu A1 k generování obdélníkového signálu s frekvencí přibližně 133 Hz a průběhem periody 7,5 ms. Ke změnám stavu u tohoto portu tak dochází každých 3,75 ms. U programu proces probíhá v nepřetržité smyčce (funkce Loop). Pro vykonání příkazu je tak zapotřebí 1 milion ms. Výstup A4 poskytuje frekvenci 16,6 Hz, která je patrná ve formě blikajících LED.

Pokud však proces vyžaduje vyšší pracovní rychlost, můžete zvýšit rychlost cyklů mikroprocesoru snížením odporu na výstupu OSC1. Hodnota 100 kΩ zajišťuje rychlost cyklu při frekvenci cca 2 MHz. Vyměňte proto jeden rezistor za rezistor 27 kΩ. Tím se zvýší celková rychlost cyklu téměř 4x a doba pro vykonání příkazu o cca 0,25 ms. Mikroprocesor by však měl optimálně pracovat při impedanci 100 kΩ na OSC1. Provoz systému je pak velmi spolehlivý, bezpečný a současně při provozní hodnotě napětí 2,2 V pracuje při velmi nízkém proudovém zatížení.

Měření délky pulzu

Připojení E4 k GND spustí program pro měření délky pulzu. Přitom platí, že na vstupu S1 dochází k vyhodnocování vstupních podmínek.

Propojení E4 a GND



Spuštění měření je při podmínce S1 = 0 (stisknutí tlačítka). Před samotným vykonáním celkově 5. příkazu je prodleva 5 ms. Dalších 5 ms trvá samotný proces. Celková doba pro měření pulzu je tak 10 ms.

Adresa	Příkaz	Data	Popis
34	2	2	Prodleva 5 ms
35	C	C	S1 = 0?
36	3	2	Jump -2
37	4	0	A = 0
38	2	2	Prodleva 5 ms
39	7	1	A = A 1
3A	5	4	Port = A
3B	C	E	S1 = 1?
3C	3	4	Jump -4
3D	3	9	Jump -9

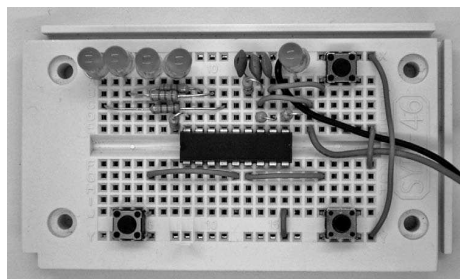
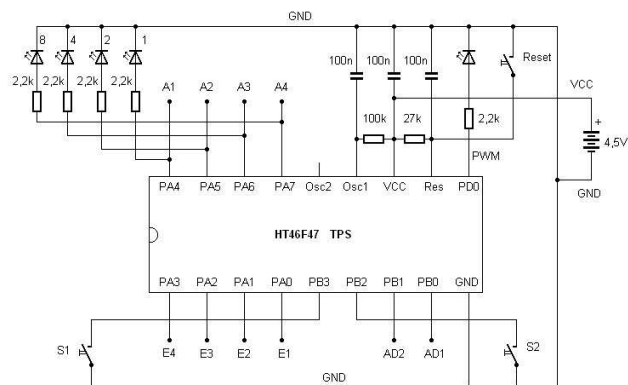
Krátce stiskněte tlačítko S1. Výsledek může být například 1010, resp. 10 (decimální soustava). Pakliže je časová jednotka programu 10 ms, doba pro zobrazení je tak 100 ms. Při dalším experimentování můžete dosáhnout hodnoty nižší než 50 ms.

Výběr programu

K programování slouží tlačítka S1 = vkládání dat a S2 = programování. Zapotřebí je i tlačítko pro reset. Tyto tlačítka se používají pro výběr programů a jejich konfiguraci. Postupně získáte vlastní zkušenosti s programováním a zjistíte, že samotné programování a úprava jednotlivých programů je velmi snadné.

Pro vstup do režimu programování, stiskněte tlačítko S2 a uvolněte jej asi po uplynutí 0,5 s. Tlačítko S2 potom použijte pro procházení v aktuálním programu a přehled příkazů, které obsahuje. Při přechodu na adresní řádek stiskněte tlačítko S2 dvakrát. Tím dojde k zobrazení příkazu a dat v adrese. Přitom se znovu krátce zobrazuje korespondující adresa.

Programovací tlačítka S1 a S2



3 tlačítka a LED

- Stiskněte tlačítko S2
- Zobrazení adresy (dolní 4 bity), 300 ms
- Vypnutí displeje, 300 ms
- Zobrazení příkazu
- Další stisknutí tlačítka S2
- Zobrazení dat
- Třetí stisk tlačítka S2
- Zobrazení další adresy. 300 ms

V případě, že chcete zobrazit pouze aktuální program a nebudete v něm provádět žádné změny, stisknutím tlačítka S2 celkem 10x přejdete až na konec programu. Přehlednost v celém procesu je velmi dobrá, vzhledem k tomu, že název programu se na displeji vždy krátce zobrazuje. Navíc je vždy dobře patrné zobrazení buď dat, nebo příkazu. V továrním nastavení se na prvních 5. adres nacházejí následující příkazy. Tyto výchozí programy pak slouží k dalším úpravám a vlastnímu programování.

Adresa	Příkaz	Data	Popis
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 14
03	8	0	AdrHi = 0
04	C	3	A = B?

4-bitový příkaz a korespondující 4-bitová data tvoří 1 byte, tj. 8-bitové číslo. ½ z byte se také někdy nazývá „Nibble“ tedy 4 bity. Čím vyšší hodnota „Nibble“ tvoří příkaz, tím nižší hodnota „Nibble“ pro data. EEPROM (paměť) mikrokontroléru zahrnuje celkem 128 bajtů. Program proto může pojmout celkem až 128 příkazů. To je dostatečný objem pro většinu aplikací, protože programový kód je vysoce kompaktní. Většina užitečných aplikací pracuje pouze s několika příkazy (většinou je to méně než 10 příkazů). Zobrazte přehled jednotlivých příkazů a dat a pak porovnejte obsah paměti. Poté znovu stiskněte tlačítko reset. Původní program se přitom nijak nezmění.

Programování

Tlačítko S1 je aktivní pouze při změně příkazu nebo dat nebo jejich novém zdávání. Zadávat je možné čísla v rozsahu 0 – 15. Prvním stiskem tlačítka S1 dojde k zadání číslíce 0. Aktuální zadání je zobrazeno prostřednictvím 4 LED. Chcete-li zadat číslo 4, stiskněte tlačítko S1 celkem 5x: 0, 1, 2, 3, 4. Binární zobrazení je pak 0100.

Pakliže jste zadali buď příkaz nebo data, popřípadě data i příkaz, dalším stisknutím tlačítka S2 dojde k přenesení do paměti EEPROM. Jako potvrzení se na 600 ms displej vypne a poté se zobrazí další příkaz a data. Tato krátká prodleva je vlastní programovací krok. V tom okamžiku navíc dochází k úspoře energie, která se použije pro přenesení dat do EEPROM. Měnit je možné současný program na stejné paměťové pozici. Pro výběr pozice použijte tlačítko S2 a úpravu příkazu nebo dat provedte tlačítkem S1. Následně úpravy uložte pomocí tlačítka S2. V první části se programuje pouze za využití dvou příkazů. To představuje zapnutí LED do nekonečné smyčky (Loop).

Rozsvícení LED

Adresa	Příkaz	Data	Popis
00	1	7	A1 - 4 = 0111
01	3	0	Jump 0

Zvolit je možné buď podrobný seznam, nebo pouze jeho zkrácenou podobu. Tyto 2 byte jsou shrnuty v hexadecimálním (hex) zápisu: 17h, 30 h. Programy jsou však zapisovány ve zkrácené podobě bez označení „h“: **17 30**. K tomuto zadání dojdete následujícím postupem:

S2 + reset, 2 x S1, S2, 8 x S1, S2, 4 x S1, S2, 1 x S1, S2

Pokud došlo k nechtěnému stisknutí tlačítka S1, je i přesto dál možné zadat správné číslo. Přejděte až za číslo 15, za kterým následuje hodnota 0. Po kompletním zadání je nový program spouštěn po stisknutí tlačítka reset. Přitom je patrné, že LED jsou rozsvícené. Přesto nedochází k žádnému dalšímu procesu. Mikroprocesor nebude reagovat na stav u vstupů E1 a E4, vzhledem k tomu, že došlo pouze k částečné úpravě továrního programu. Tovární programy tak není možné spustit. Vzhledem k tomu, že došlo ke změně pouze první dvě adresy v paměti, můžete snadno spustit původní program. Zadejte znovu první 2 příkazy (64 51) v příslušném programovacím kroku. Následně ověřte funkci původního programu.

Obnovení továrních programů

V případě, že chcete uvést mikroprocesor do továrních hodnot, zadejte 2 hodnoty FF. Po resetu dojde k vymazání paměti EEPROM. Software mikroprocesoru disponuje funkcí spuštění prvních dvou adres, čímž rozpozná prázdnou paměť. Pakliže dojde k načtení 2 FF bytů mikroprocesor předpokládá, že v paměti nejsou žádná data. Díky tomu je do EEPROM mikroprocesoru automaticky načten seznam továrních programů. Reset do továrních hodnot lze provést kdykoliv.

Adresa	Příkaz	Data	Popis
00	F	F	-
01	F	F	-

Spustěte režim programování stisknutím tlačítka S2. Zadejte hodnotu F (desítkový zápis 15), celkem 4x dokud se nerozsvítí LED A1 až A4. Zadání potvrďte stisknutím tlačítka S2. Následně proveďte reset. Tím dojde k resetu mikroprocesoru (uvedení do továrního nastavení) a obnovení všech původních programů.

Základní příkazy mikrokontroléru

Programovatelný mikroprocesor rozpoznává celkem 14 příkazů (1 – 14). Většina z těchto příkazů má parametry ve formě 4-bitového čísla 0000 – 1111 (0 – F) tj. s číselným rozsahem až 15 (desítkový zápis). Další příkazy mají sub-funkce a zadávají se s jinými parametry. Příkazový kód tak může obsahovat až 16 sub-příkazů (sub-command). Například příkaz 7 znamená „Compute A = ...“ (vypočítej A = ...). Další parametr potom označuje, který výpočet se bude provádět.

Příkazy a data jsou zapisovány společně v hexadecimálním kódu jako 1 byte. Příkaz 1 spolu s parametrem 4, se tak stává příkaz 14h (hex). Hexadecimální znak je vynechán, protože všechny příkazy a adresy jsou obecně zapisovány v hexadecimálním kódu.

První 3 příkazy jsou:

10-1F: přímý výstup na portu na A1-A4, 0-15, bin. 0000 až 1111

20-2F: doba 0 - 15
(1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 30000, 60000 ms)

30-3F: Jump (skok zpět) 0 - 15

Příkaz 1 se používá pro výstup konstantního čísla. To umožňuje výsledný výstup různého rastru, například spínání několika LED současně. Další příkaz 2, který čeká na spuštění, používá parametr, který obsahuje čas (ms) a postupně spínání 1-2-5. I přes relativně nízký rozsah 0 - 15, je možné provádět nastavení parametrů doby pro spuštění (prodleva) v rozmezí 1 ms – 1 min. Při naprogramování většího množství příkazů s nastavenou prodlevou pro spuštění, může být celkový čas i mnohem delší (Loop). „Return“ příkaz 3 je velmi jednoduchý a je postačující pro všechny aplikace, u kterých je zapotřebí proces opakovat do nekonečna. Krok návratu (Jump) je omezen až na oblast 15. Návrat (Jump) souvisí s aktuální adresou, přitom části programu mohou být přesunuty do libovolné adresy.

Program pro střídavé spuštění vyžaduje pouze tyto 3 příkazy a je zapsán do oblasti od 00 v nepatrně upravené formě. Současně se přitom mění výstupní bit samples a prodleva před spuštěním (delay).

Adresa	Příkaz	Data	Popis
00	1	1	A1 – 4 = 0001
01	2	7	200 ms
02	1	4	A1 – 4 = 0001
03	2	7	200 ms
04	3	4	Jump -4

V hexadecimální zápisu může program vypadat následovně: 11 27 14 27 34

Na základě těchto prvních třech příkazů je založen princip mnoha jednoduchých programů. Otestujte funkci následujících 3. programů. V závěru experimentování by mělo dojít k intuitivnímu používání všech příkazů. Vytvoření obdobných programů pak bude naprostou samozřejmostí. Jednoduchý příklad představuje aplikace s jednou LED a několika výstupními rastry:

Adresa	Příkaz	Data	Popis
00	1	1	LED 0001
01	2	8	500 ms
02	1	2	LED 0010
03	2	8	500 ms
04	1	4	LED 0100
05	2	8	500 ms
06	1	8	LED 1000
07	2	8	500 ms
08	3	8	Jump -8

Rozšířte program se dvěma a více výstupními rastry tak, aby se LED rozsvěcovaly z jedné strany na a zpět. Vytvořte svůj vlastní program s různými výstupními vzory a časovým zpožděním.

Adresa	Příkaz	Data	Popis
00	1	1	LED 0001
01	2	8	500 ms
02	1	2	LED 0010
03	2	8	500 ms
04	1	4	LED 0100
05	2	8	500 ms
06	1	8	LED 1000
07	2	8	500 ms
08	1	4	LED 0100
09	2	8	500 ms
0A	1	2	LED 0010
0B	2	8	500 ms
0C	3	C	Jump -12

LED „Running light“ LED rozsvěcované z jedné strany na druhou a zpět
11 28 12 28 14 28 18 28 14 28 12 28 3C

Časový spínač může obsahovat zpoždění až 1 minutu na příkaz ke zpracování. Na konci procesu, je návrat zpět k hodnotě (Jump) 0, tj. v nekonečné smyčce bez obsahu, který slouží jako ukončení programu. Restart je spuštěn (trigger) tlačítkem reset. Vytvořte vlastní program a naprogramujte si 3. minutový kuchyňský časovač tzv. „Minutka“. Můžete zobrazit zbývající čas pomocí LED coby provozní, světelný ukazatel (light bar).

Adresa	Příkaz	Data	Popis
00	1	F	LED 1111
01	2	F	1 min
02	1	0	LED 0000
03	3	0	End (konec)

Operace s proměnnými

Doposud bylo u parametrů pro jednotlivé příkazy použito pouze číselných hodnot. Programovat je v takovém případě vhodné, pakliže je program spouštěn pokaždé stejným způsobem. Složitější programy však pracují s proměnnými daty, provést lze například operace $A = A + B$ v závislosti na obsahu proměnných A a B, může být pokaždé jiný výsledek. Výsledný program pro ovládání LED může mít následující výstup.

Mikroprocesor má celkem 4 proměnné **A, B, C a D**. Nejdůležitější proměnná je A, která se také nazývá akumulátor, zkr. **Accu**. A je součástí všech početních operací a zároveň přijímá výsledek výpočtu. A se také používá pro přenos dat. Proměnná **B** je zapotřebí pro výpočet operace. Proměnné **C a D** se využívají jako dočasná paměť a jsou následně používány pro operace smyček (Loop).

K dispozici jsou také dva analogové vstupy (AD1 a AD2) a PWM výstup. Zpracovaná data jsou omezena na 4 bity a jsou přístupná pouze prostřednictvím proměnné A ($A = AD1$, $PWM = A$). Akumulátor A může být načten přímo s čísly (příkazy 40-4F). Pro zavedení B, C nebo D je zapotřebí načíst jako první A a poté přiřadit obsah k další proměnné (příkazy 51-53). A a B umožňuje některé početní procesy (příkazy 71-7A).

Příkazy 40-4F přiřazují proměnné A novou hodnotu. Skupina příkazů 51-5A převádí obsah A k čílu jako další proměnnou nebo na PWM výstup. V této skupině jsou také příkazy, které stanoví jeden bit výstupního portu.

Opačný směr dat je zastoupen skupinou příkazů 61-6A, kde jsou zdrojová data načtena do A. Skupina příkazů 71-7A nakonec provádí několik výpočetních operací s výsledkem, který se projevuje v proměnné A. Výstupní port **Dout** (Digital-out, výstup digitálních dat do mikroprocesoru) zahrnuje 4 výstupy A1-A4, který může být aktivován buď společně, nebo jako jednotlivé bity Dout.0 do Dout.3. Vstupy E1-E4 se aktivují stejně jako vstupní port Din.

```
40-4F: A = 0-15
51-5A: Target 1-9 =
51: B = A
52: C =
53: D =
54: Dout = A
55: Dout.0 = A.0
56: Dout.1 = A.0
57: Dout.2 = A.0
58: Dout.3 = A.0
59: PWM =
61-6A: = Source 1-10
61: A = B
62: A = C
63: A = D
64: A = Din
65: A = Din.0
66: A = Din.1
67: A = Din.2
68: A = Din.3
69: A = AD1
6A: A = AD2
71 -7a: = Expression 1-10
71: A = A + 1
72: A = A - 1
73: A = A + B
74: A = A - B
75: A = A * B
76: A = A / B
77: A = A B
78: A = A or B
79: A = B Xor
7A: = Not
```

Příklad programu s proměnnou A je uveden části „Binární čítač a PWM výstup“. Program byl nastaven na adresu 0 dále nepatrně rozšířen. Začátek programu je nastaven s hodnotou 0 v proměnné A. Adresa 01 obsahuje početní operaci, navýšenou o 1. Obsah proměnných je pak převáděn do PWM výstupu a na výstupní port.

Adresa	Příkaz	Data	Popis
00	4	0	A = 0
01	7	1	A = A + 1
02	5	4	Port = A
03	5	9	PWN = A
04	2	6	100 ms
05	3	4	Jump -4

Program „Navýšení o 1“
40 71 54 59 26 34

Další program je obdobou příkladu uvedeného v části „Analogově-digitální převodník“. Údaje přicházejí z analogového vstupu AD1 a jsou převáděny na výstupní port a PWM výstup. Upravený program zahrnuje další početní proces - inverzi obsahu proměnné A. Dochází tak ke změně hodnoty 0000 na novou hodnotu 1111, v tomto případě ke změně z hodnoty 0 na 15 a obráceně. Zvyšující se vstupní napětí vede k omezení PWM výstupu.

Adresa	Příkaz	Data	Popis
00	6	9	A = AD1
01	5	4	Port = A
02	7	A	A = Not A
03	5	9	PWN = A
04	2	6	100 ms
05	3	5	Jump -5

Program „Inverze“
69 54 59 26 35 7A

Jump / Branch

Doposud jsme si představili jednoduchý příkaz 3 (skokový návrat) u 15 adres. V této části dojde k rozšíření tohoto příkazu na absolutní hodnoty. Příkaz **Jump** může být indikován pouze prostřednictvím 4 bitů. Rozšířený příkaz udává „**High nibble**“ jednotlivé adresy. Dojde tak k rozšíření adresy v rozmezí 0-255, což je víc, než dokáže EEPROM zpracovávat (128 byte), tedy adresa 00-7F (0-127). Paměť je rozdělena na 8 stran (stránky 0 – 7). Strana pro příkaz Jump musí být specifikována před příkazem **Absolute Jump**. 2 sčítané smyčky s proměnnými C a D tak provádějí Absolute Jump se adresní stranou paměti, která byla předtím specifikována.

Podmíněně Jump příkazy fungují jako **Skip Commands** (vynechávání řádků). Pokud je příslušná podmínka splněna, dojde k vynechání adresy. Na vynechanou pozici je možné zapsat příkaz nebo početní operaci. U těchto podmínek dochází k porovnávání mezi A a B nebo přímým bit-query u vstupního portu.

K dispozici je možnost spuštění subprogramu (call) a souvisejícího příkazu return. Současně může probíhat několik subprogramů. Přesto subprogram nemá možnost spuštění jiného subprogramu, vzhledem k tomu, že interpret má v paměti pouze předchozí adresu.

```
80-8F: Adr-high = 0-15
90-0F: Direct jump to Adr-high, Adr-low (0-15)
A0-AF: Counting loop C-times Adr-high, Adr-low (0-15)
B0-BF: Counting loop D-times Adr-high, Adr-low (0-15)
C1-CF: Conditional jump: if (condition 1-15) then skip
C1: if A > B then Adr = Adr + 1
C2: if A > B then Adr = Adr + 1
C3: if A = B then Adr = Adr + 1
C4: if Din.0 = 1 then Adr = Adr + 1
```


C5: if Din.1 = 1 then Adr = Adr + 1
 C6: if Din.2 = 1 then Adr = Adr + 1
 C7: if Din.3 = 1 then Adr = Adr + 1
 C8: if Din.0 = 0 then Adr = Adr + 1
 C9: if Din.1 = 0 then Adr = Adr + 1
 CA: if Din.2 = 0 then Adr = Adr + 1
 CB: if Din.3 = 0 then Adr = Adr + 1
 CC: if S1 = 0 then Adr = Adr + 1
 CD: if S2 = 0 then Adr = Adr + 1
 CE: if S1 = 1 then Adr = Adr + 1
 CF: if S2 = 1 then Adr = Adr + 1
 D0–DF: Subprogramme call Adr-high, Adr-low (0-15)
 E0–EF: Return - Subprogramme

D0-DF: Subprogramme call Adr-high, ADR-nížká (0-15)
 E0-EF: Return – Subprogramme

Příklad pro použití podmíněných **Jump** příkazů se nachází v programu pro Měření délky pulzu. Program byl jen nepatrně upraven (adresa 0). Vzhledem k tomu, horní část adresy (Adr-hi) je v klidovém stavu 0, procesor začíná na straně 0. Není v tom případě zapotřebí příkazu 80. Při stisknutí tlačítka se měří a zobrazuje délka stisku. Všechny příkazy čekající na zpracování byly z programu odstraněny. Program tak v tom okamžiku pracuje s vyšším rozlišením.

Adresa	Příkaz	Data	Popis
00	C	C	S1 = 0?
01	3	1	Jump -1
02	4	0	A = 0
03	7	1	A = A + 1
04	5	4	Port = A
05	C	E	S1 = 1?
06	3	3	Jump -3
07	3	7	Jump -7

Reakční doba tlačítka S1: CC 31 40 71 54 33 37 CE

Příkaz Jump CC v adrese 00 vyhodnocuje podmínku u tlačítka S1. V klidovém stavu: S1 = 1. Podmínka tedy není pravdivá (true) a příkaz v adrese 01, není vynechán. Dochází tak k relativnímu přechodu (jump) na začátek programu. V programu se opakují příkazy na adresách 00 a 01, dokud je tlačítko stisknuté. Podmínka se stává pravdivou a adresa 01 je vynechána. Tím dojde ke spuštění aktuálního procesu měření. Akumulátor je odstraněn a kontinuálně navyšován o 1 a přiveden na výstup LED. Další podmíněný Jump příkaz CE je umístěn do adresy 05, kde je podmínkou pro vynechání příkazu je S1 = 1. Pokud je tlačítko i nadále stisknuté, podmínka není pravda. Provádí se tím příkaz 06 a způsobuje návrat k 03. Po uvolnění tlačítka program přejde na adresu 07 a následně pak na začátek programu. Zvolte tento program a otestujte jeho funkci. Časová jednotka je cca 5 ms.

Původní tovární programy jsou neustále zavedeny v paměti od adresy 34h a výše. Přepsat je možné pouze nižší adresy. Vytvořte svůj vlastní, malý program, který obsahuje pouze Jump do této adresy. Nejprve je nutné zapsat odkaz na stranu 3. Následný Absolute Jump specifikovaný adresou 4 odkazuje adresu 34.

Adresa	Příkaz	Data	Popis
00	8	3	Strana 3
01	9	4	Adresa = 34

Program pro měření – Absolute Jump: 83 94

Původní program je tak znovu vyvolán z paměti. Úplný přehled všech možných programů najdete v příloze.

Přehled příkazů

Přehled všech příkazů velmi usnadňuje programování mikroprocesoru. V následující tabulce je jejich kompletní přehled v kompaktní podobě.

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port=	Wait	Jump	A=	... = A	A = ...	A = ...	Page	Jump	C*	D*	Skip if ...	Call	Ret
0	0	1 ms	0	0				0	0	0	0		0	
1	1	2 ms	1	1	B = A	A = B	A = A + 1	1	1	1	1	A > B	1	
2	2	5 ms	2	2	C = A	A = C	A = A - 1	2	2	2	2	A < B	2	
3	3	10 ms	3	3	D = A	A = D	A = A + B	3	3	3	3	A = B	3	
4	4	20 ms	4	4	Dout = A	A = Din	A = A - B	4	4	4	4	Din.0 = 1	4	
5	5	50 ms	5	5	Dout.0 = A.0	A = Din.0	A = A * B	5	5	5	5	Din.1 = 1	5	
6	6	100 ms	6	6	Dout.1 = A.0	A = Din.1	A = A / B	6	6	6	6	Din.2 = 1	6	
7	7	200 ms	7	7	Dout.2 = A.0	A = Din.2	A = A And B	7	7	7	7	Din.3 = 1	7	
8	8	500 ms	8	8	Dout.3 = A.0	A = Din.3	A = A Or B	8	8	8	8	Din.0 = 0	8	
9	9	1 s	9	9	PWM = A	A = AD1	A = A Xor B	9	9	9	9	Din.1 = 0	9	

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
A	10	2 s	10	10		A = AD2	A = Not A		A	A	A	Din.2 = 0	A	
B	11	5 s	11	11					B	B	B	Din.3 = 0	B	
C	12	10 s	12	12					C	C	C	S1 = 0	C	
D	13	20 s	13	13					D	D	D	S2 = 0	D	
E	14	30 s	14	14					E	E	E	S1 = 1	E	
F	15	60 s	15	15					F	F	F	S2 = 1	F	

Operace Loop

Proces Loop (smýčka) umožňuje spuštění určitého procesu v předem definovaném počtu opakování. Například při provedení určitého příkazu 5x. Využívá se proto proměnné C. Jako první musí být do proměnné A načtena hodnota 5, teprve poté do C. Příkaz A2 provádí absolutní výpočet do 02 a ve stejné chvíli snižuje obsah proměnné C na 1. Pokud C dosáhne hodnoty 0, nebude příkaz Jump už dále vykonáván. Adresa Absolute Jump se odkazuje na uvedenou stránku. Pro program na straně 0, příkaz 80 může být stejně tak vynechán.

Adresa	Příkaz	Data	Popis
00	4	5	A = 5
01	5	2	C = A
02	1	5	Port = 0101
03	2	8	500 ms
04	1	A	Port = 1010
05	2	8	500 ms
06	8	0	Strana 0
07	A	2	C-časy 02
08	3	0	End (konec)

Operace Loop: 45 52 15 28 1A 80 A2 30

Vyzkoušejte funkčnost programu. Výstupní rastr LED je 0101 a 1010. Ve skutečnosti se celý proces neprovede 5x ale celkem 6x. Příkaz Jump je sice proveden 5x, ale při prvním přechodu na adresní pozici dojde ke stejnému efektu. Proto program generuje celkem 6x výstup pro LED. Změňte proměnnou na hodnotu 4 a znovu otestujte funkci programu. Teprve za těchto podmínek bude LED blikat skutečně 5x. Operaci Loop je možné použít oběma směry. V tomto případě se proces spustí skutečně 5x, protože do proměnné C se hned v úvodu načte hodnota 5. Vynechání adresy 04 znamená pouze relativní skok a dále se vykonává v nekonečné smyčce (Loop).

Adresa	Příkaz	Data	Popis
00	4	5	A = 5
01	5	2	C = A
02	8	0	AdrHi = 0
03	A	5	C-times 05 (x-krát)
04	3	0	End (konec)
05	1	5	Port = 0101
06	2	8	500 ms
07	1	A	Port = 1010
08	2	8	500 ms
09	3	6	Jump -6

5x LED Flash: 45 52 80 A5 30 15 28 1A 28 36

Porovnávání

Při této operaci se porovnávají 2 číselné hodnoty. V závislosti na výsledku porovnání dojde k vykonání funkce Jump. Číselné hodnoty musí být zadány v A a B. V následujícím příkladu se B načítá s hodnotou 5. Výsledný stav je přijat z analogového vstupu AD1. V tomto případě bude zapojen například světelný senzor obdobně jako v části „AD převodník“.

Pokud AD1 > 5
pak: všechny LED se rozsvítí
jinak: všechny LED zhasnou

Výsledný efekt je soumrakový spínač. Pakliže je LDR (fotorezistor) připojen k GND, dojde při větším osvětlení ke snížení napětí na AD1. LED zcela zhasnou, pakliže světelný tok překročí určitou úroveň a napětí zcela klesne. Limit je 6 a proto musí být výsledek měření v hodnotě pod 5.

Adresa	Příkaz	Data	Popis
00	4	5	A = 5
01	5	1	B = A
02	8	0	AdrHi = 0
03	6	9	A = AD1
04	C	1	Skip if A>B
05	9	8	Adr 08
06	1	F	LED 1111
07	3	4	Adr 03
08	1	0	LED 0000
09	3	6	Adr 03

Jednoduchý soumrakový spínač: 45 51 80 69 C1 98 1F 34 10 36

Otestujte funkci programu zakrytím světelného senzoru rukou. V té chvíli se nejvíce projeví funkce senzoru. Přesto má takový obvod i vedlejší nežádoucí efekt. Na prahových hodnotách (threshold) mezi sepnutím a rozepnutím se bude LED nekontrolovaně rozsvěcovat. Zejména při umělém osvětlení bude k tomuto efektu velmi často docházet. K problikávání LED tedy dochází po vyhodnocení programu mikroprocesorem. Výsledek tedy nemusí být vždy přesně podle očekávání. V části „Soumrakový spínač“ však je obvod výrazně vylepšen.

AND, OR a XOR

(logické funkce A, NEBO a Xnebo)

2 binární podmínky mohou být spojeny do nové. Jedním z příkladů je funkce AND: Pakliže bit 1 má stav 1 a bit 2 má stav 1, výstup stav je také 1. Binární čísla s několika bity je možné spojit stejným způsobem. Link "10 A 3 = 2" se stane logický, pokud se zapíše v binárních číslech:

1010 AND
0011 =
0010

Následující program odkazuje vstupní podmínky na konstantní číslo 3. Funkce AND potom způsobuje, že 2 spodní bity, budou maskovány (odfiltrovány). V klidovém stavu má vstupní port stav 1111. Funkce AND odkazuje 0011 a přidává podmínku 0011 do LED. Pokud připojíte jeden ze vstupů E1 nebo E2 k zápornému pólu GND, dojde k promítnutí podmínky 0 i u dalších výstupů. Změny se u vstupů E3 a E4 nijak neprojeví.

Adresa	Příkaz	Data	Popis
00	6	4	A = Din
01	5	1	B = A
02	4	3	A = 3
03	7	7	A = A And B
04	5	4	Port = A
05	3	5	Jump -5

Aplikace funkce AND: 64 51 43 77 54 35

Změňte parametry programu a použijte i ostatní logické funkce. Funkce OR (78) se používá obecně pro stanovení specifických podmínek pro hodnotu 1: 64 51 43 78 54 35

1010 OR
0011 =
1011

Pomocí funkce XOR (EXCLUSIVE-OR, 79) k inverzi jednotlivých bitů: 64 51 43 79 54 35

1010 XOR
0011 =
1001

Subprogramy

Pokud je zapotřebí opakovaně použít některých částí programu, lze je zapsat do subprogramu. Dochází tím navíc k úspoře paměťového prostoru a ubývá tím i celkový zápis programu. V následujícím příkladu je uvedeno použití subprogramu, který je z paměti vyvolán na 2 místech hlavního programu. Subprogram zahrnuje pouze 1 příkaz ($A = A - 1$) a následný Jump příkaz. Nejedná se však přitom o úsporu paměťového místa. Na příkladu pouze dojde k demonstraci příkazů CALL a RET:

Hlavní program

Adresa	Příkaz	Data	Popis
00	8	0	AdrHi = 0
01	D	8	Call 08
02	5	4	Output
03	2	9	1 s
04	D	8	Call 08
05	5	4	Output
06	2	8	0,5 s
07	3	7	Jump -7

Subprogram

Adresa	Příkaz	Data	Popis
08	7	2	$A = A - 1$
09	E	0	Ret

Použití subprogramu: 80 D8 54 29 D8 54 28 37 72 E0

Výsledkem programu je binární čítač s odpočtem a nepravidelnými časovými prodlevami. Ověřte správnost funkce programu. Existuje několik užitečných podprogramů pro různé využití. Kompletní seznam je uveden v příloze. K jejich použití je zapotřebí znát pouze vstupní adresu:

- 50: Subprogram: Long sound (zvuková signalizace – dlouhá)
- 52: Subprogram: Short sound (zvuková signalizace – krátká)
- 53: Subprogram: Any sound, length in A (libovolný zvuk, délka – A)
- 60: Subprogram: Wait for pushed button S1 (Stisknutí tlačítka S1)
- 68: Subprogram: Wait for pushed button S2 (Stisknutí tlačítka S2)
- 70: Subprogram: Number input with S1 and S2 (číselné zadání s S1/S2)

Subprogram od adresy 60 a dál se používá pro nastavení čítače ovládaného tlačítkem S1. Výchozí stav pro čítač je hodnota 0. Hlavní program je obsahově relativně malý, vzhledem k tomu, že všechny složité funkce tlačítka jsou uvedeny v subprogramu.

Adresa	Příkaz	Data	Popis
00	4	0	$A = 0$
01	5	4	Output
02	7	1	$A = A + 1$
03	8	6	Strana 6
04	D	0	Call 60, tlačítko S1
05	3	4	Jump -4

Čítač ovládaný S1: 40 54 71 86 D0 34

Otestujte funkci programu. Pokud tlačítko S1 stisknete 10x, výsledný výstup bude 1010. Pozměňte program tak, aby použití subprogramu bylo od adresy 68. V té chvíli bude čítač reagovat na S2.

Soumrakový spínač

(Twilight Switch)

Soumrakový spínač slouží pro rozsvícení světelného zdroje v případě poklesu světelného toku pod určitou mez. Pokud dojde k nárůstu světa, zdroj se vypne (světlo zhasne). Důležité je nastavení obvodu takovým způsobem, aby při prahových hodnotách, kdy jsou světelné hodnoty vyrovnané, nedocházelo k problikávání LED. To lze zajistit pomocí funkce hysterze – nastavení určitého vztahu mezi aktivací a deaktivací. Následující program je nastaven s těmito pravidly:

- Pokud napětí na AD1 není vyšší než 5 V, světlo se vypne.
- Pokud napětí na AD1 není nižší než 9 V, světlo se rozsvítí.

Dochází tak vytvoření střední oblasti, ve které se výstupní podmínka nemůže měnit. Tím se předejde nežádoucímu problikávání LED.

0-5: LED jsou zhasnuté

6-8: LED beze změny

9-15: LED se rozsvítí

Adresa	Příkaz	Data	Popis
00	1	0	LED 0000
01	4	5	$A = 5$
02	5	1	$B = A$
03	6	9	$A = AD1$
04	C	1	Skip if $A > B$
05	1	0	LED 0000
06	4	9	$A = 9$
07	5	1	$B = A$
08	6	9	$A = AD1$
09	C	2	Skip if $A < B$
0A	1	F	LED 1111
0B	3	A	Jump -10

LED stmívač

(Dimmer)

Pomocí tohoto programu je možné ovládat jas LED. Jas se ovládá na PWM výstup prostřednictvím tlačítek. Krátkým stiskem tlačítka dosáhnete následující úrovně jasu. Opakovaným stisknutím tlačítka pak úroveň jasu LED měníte postupně. Základem programu je použití příkazů **Skip**. Pokud tak nedojde k dalšímu stisku tlačítka, je příkaz pro navýšení nebo snížení obsahu akumulátoru vynechán. Problematický bod nastává při přechodu z úrovně 15 na 0 nebo z úrovně 0 na 15. Aby nedocházelo k nežádoucímu překročení hodnot, je zapotřebí provést určitá opatření. Je nezbytné nastavit obvod tak, aby rozlišil jakého konce (úrovně) dosáhl (0 nebo 15). Vzhledem k tomu, že funkce akumulátoru provádí neustálé porovnávání, obsah je nutné zavést do operační paměti pomocí proměnné C.

Adresa	Příkaz	Data	Popis
00	8	0	AdrHi = 0
01	5	9	PWM = A
02	2	7	200 ms
03	5	2	$C = A$
04	4	F	$A = 15$
05	5	1	$B = A$
06	6	2	$A = C$
07	C	2	Skip if $A < B$
08	9	B	Jump 0B
09	C	F	Skip if $S2 = 1$
0A	7	1	$A = A + 1$
0B	5	2	$C = A$
0C	4	0	$A = 0$
0D	5	1	$B = A$

0E	6	2	A = C
0F	C	1	Skip if A>B
10	9	0	Jump 00
11	C	E	Skip if S1 = 1
12	7	2	A = A - 1
15	9	0	Jump 00

Stmívač LED: 80 59 27 52 4F 51 62 C2 9B CF 71 52 40 51 62 C1 90 CE 72 90

Číselný zámeček

Princip programu je založen na PWM výstupu, který sepne v případě vložení správného číselného kódu. Číselné zadání se provádí pomocí tlačítek S1 a S2. V následujícím příkladu je uveden program s jednoduchým zadáním prostřednictvím tlačítka S1. Na začátku programování dojde po prvním stisknutí tlačítka k výsledku 0000. Dalším stisknutím tlačítka S1 se provádí navýšení o hodnotu 1. Stisknutím tlačítka S2 se zadávání ukončí a program se přepne do funkce Loop.

Adresa	Příkaz	Data	Popis
00	C	C	S1 = 0?
01	3	1	Jump -1
02	4	0	A = 0
03	5	4	Dout = A
04	2	3	10 ms
05	C	E	S1 = 1?
06	3	2	Adr 04
07	C	F	S2 = 1?
08	3	0	End
09	C	C	S1 = 0?
0A	3	3	Adr 07
0B	7	1	A = A + 1
0C	2	3	10 ms
0D	C	C	S1 = 1?
0E	3	1	Adr 0D
0F	3	C	Adr 03

Číselné zadání: CC 31 40 54 23 CE 32 CF 30 CC 33 71 23 CC 31 3C

Číselné zadání je rovněž k dispozici coby hotový subprogram od adresy 70.

Namísto funkce Loop v řádce 08, je příkaz RET. Subprogram je vynechán v případě, že je výsledek zadán do proměnné A. Následující program vyvolá číselnou hodnotu z paměti 3x a provádí její srovnání s předdefinovaným výsledkem. V tomto případě je správné zadání 3, 5, 2. Poté se výstup PWM aktivuje na hodnotu 15. Jakékoliv jiné, nesprávné zadání, způsobí přepnutí programu do nekonečné smyčky (Loop). Výstup PWM pracuje jako běžný digitální port, což je nezbytné pro všechny 4 výstupy A1 až A4 pro číselné zadání. Po každém úplném a správném zadání, dojde k indikaci prostřednictvím LED s tím, že došlo k vložení správné číselné řady (kombinace).

Adresa	Příkaz	Data	Popis
00	8	7	Page 7
01	4	3	A = 3
02	5	1	B = A
03	D	0	Call 70
04	C	3	Skip if A=B
05	3	0	End
06	1	0	LED off
07	4	5	A = 5
08	5	1	B = A
09	D	0	Call 70
0A	C	3	Skip if A=B
0B	3	0	End
0C	1	0	LED off
0D	4	2	A = 2
0E	5	1	B = A
0F	D	0	Call 70

10	C	3	Skip if A=B
11	3	0	End
12	1	0	LED off
13	4	F	A = 15
14	5	9	PWM=A
15	3	0	End

Číselný zámeček: 87 43 51 D0 C3 30 10 45 51 D0 C3 30 10 42 51 D0 C3 30 10 4F 59 30

Příloha – Přehled programů

Address	Command	Data	Comment
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 1110
03	8	0	Page 0
04	C	3	A = B?
05	9	8	Jump 08
06	8	2	Page 2
07	9	5	Jump 25, "Count up"
08	4	D	A = 1101
09	8	0	Page 0
0A	C	3	A = B ?
0B	9	E	Jump 0E
0C	8	2	Page 2
0D	9	A	Jump 2A, "AD/PWM"
0E	4	B	A = 1011
0F	8	1	Page 1

Strana 0: 64 51 4E 80 C3 98 82 95 4D 80 C3 9E 82 9A 4B 81 – výběr a spuštění programů

Address	Command	Data	Comment
10	C	3	A = B?
11	9	4	Jump 14
12	8	3	Page 3
13	9	0	Jump 30, "Random"
14	4	7	A = 0111
15	8	1	Page 1
16	C	3	A = B?
17	9	A	Jump 1A
18	8	3	Page 3

Address	Command	Data	Comment
19	9	4	Jump 34, "Stop watch S1"
1A	4	3	A = 0011
1B	8	2	Page 2
1C	C	3	A = B?
1D	9	0	Jump 20 "Alternating flash"
1E	8	4	Page 4
1F	9	0	Jump 40, "Stop watch S1/S2"

Strana 1: C3 94 83 90 47 81 C3 9A 83 94 43 82 C3 90 84 90 - výběr a spuštění programů

Address	Command	Data	Comment
20	1	1	0001 "Alternating flash"
21	2	8	Wait for 500 ms
22	1	8	1000
23	2	8	Wait for 500 ms
24	3	4	Jump -4
25	7	1	A = A + 1 "Count up"
26	5	4	Port = A
27	5	9	PWM = A
28	2	6	Wait for 100 ms
29	3	4	Jump -4
2A	6	9	A = AD1 "AD/PWM"
2B	5	4	Port = A
2C	5	9	PWM = A
2D	2	6	Wait for 100 ms
2E	3	4	Jump -4
2F	F	F	-

Strana 2: 11 28 18 28 34 71 54 59 26 34 69 54 59 26 34 FF
- střídavě blikající LED, odpočet, AD/PWM

Address	Command	Data	Comment
30	5	4	Port = A "Random"
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Jump -3
34	2	2	Wait 5 ms "Stop watch S1"
35	C	C	S1 = 0?
36	3	2	Jump -2
37	4	0	A = 0
38	2	2	Wait for 5 ms
39	7	1	A = A + 1
3A	5	4	Port = A
2B	C	E	S1 = 1?
3C	3	4	Jump -4
3D	3	9	Jump -9
3E	F	F	-
3F	F	F	-

Strana 3: 54 CE 71 33 22 CC 32 40 22 71 54 CE 34 39 FF FF – náhodné spuštění, stop S1

Address	Command	Data	Comment
40	8	6	Jump 6, "Stop watch Start/Stop"
41	D	0	Call "Waiting S1"
42	4	0	A = 0
43	7	1	A = A + 1
44	5	4	Port = A
45	2	3	Wait for 10 ms
46	C	D	S2 = 0?
47	3	4	Jump -4
48	D	8	Call "Waiting S2"
49	4	0	A = 0
4A	5	4	Port = A
4B	3	B	Jump -11

Address	Command	Data	Comment
4C	F	F	-
4D	F	F	-
4E	F	F	-
4F	F	F	-

Strana 4: 86 D0 40 71 54 23 CD 34 D8 40 54 3B FF FF FF FF – Program stopky: start/stop

Address	Command	Data	Comment
50	4	F	A = 15 "Sound long"
51	9	3	Adr 03
52	4	5	A = 5 "Sound short"
53	5	3	D = A "Sound variable"
54	1	9	A4 = 1
55	1	1	A4 = 0
56	2	1	2 ms
57	1	9	A4 = 1
58	1	1	A4 = 0
59	2	1	2 ms
5A	1	9	A4 = 1
5B	1	1	A4 = 0
5C	2	0	1 ms
5D	B	4	D-times 04
5E	1	0	Dout 0
5F	E	0	Return

Strana 5: 4F 93 45 53 19 11 21 19 11 21 19 11 20 B4 10 E0 – Subprogram: zvukový výstup

Address	Command	Data	Comment
60	2	3	Wait 10 ms "Wait S1"
61	C	E	S1 = 1?
62	3	2	Jump -2
63	2	3	Wait for 10 ms
64	C	C	S1 = 0?
65	3	1	Jump -1
66	E	0	Return
67	F	F	-
68	2	3	Wait 10 ms "Wait S2"
69	C	F	S2 = 1?
6A	3	2	Jump -2
6B	2	3	Wait for 10 ms
6C	C	D	S2 = 0?
6D	3	1	Jump -1
6E	E	0	Return
6F	F	F	-

Strana 6: 23 CE 32 23 CC 31 E0 FF 23 CF 32 23 CD 31 E0 FF Subprogram S1/S2

Address	Command	Data	Comment
70	C	C	S1 = 0? »Button input"
71	3	1	Jump -1
72	4	0	A = 0
73	5	4	Port = A
74	2	3	Wait for 10 ms
75	C	E	S1 = 1?
76	3	2	Jump -2
77	C	F	S2 = 1?
78	E	0	Return
79	C	C	S1 = 0?
7A	3	3	Jump -3
Address	Command	Data	Comment
7C	2	3	Wait for 10 ms
7D	C	C	S1 = 1?
7E	3	1	Jump -1
7F	3	C	Jump -12

Strana 7: CC 31 40 54 23 CE 32 CF 30 CC 33 71 23 CC 31 3C – Subprogram zadávání tlačítkem

Přehled příkazů

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port =	Wait	Jump -	A =	... = A	A = ...	A = ...	Page	Jump	C*	D*	Skip if ...	Call	Ret
0	0	1 ms	0	0				0	0	0	0		0	
1	1	2 ms	1	1	B = A	A = B	A = A+1	1	1	1	1	A>B	1	
2	2	5 ms	2	2	C = A	A = C	A = A-1	2	2	2	2	A<B	2	
3	3	10 ms	3	3	D = A	A = D	A = A+B	3	3	3	3	A = B	3	
4	4	20 ms	4	4	Dout.0 = A	A = Din	A = A-B	4	4	4	4	Din.0 = 1	4	
5	5	50 ms	5	5	Dout.0 = A.0	A = Din.0	A = A*B	5	5	5	5	Din.1 = 1	5	
6	6	100 ms	6	6	Dout.1 = A.0	A = Din.1	A = A/B	6	6	6	6	Din.2 = 1	6	
7	7	200 ms	7	7	Dout.2 = A.0	A = Din.2	A = A And B	7	7	7	7	Din.3 = 1	7	
8	8	500 ms	8	8	Dout.3 = A.0	A = Din.3	A = A Or B		8	8	8	Din.0 = 0	8	
9	9	1 s	9	9	PWM = A	A = AD1	A = A Xor B		9	9	9	Din.1 = 0	9	
A	10	2 s	10	10		A = AD2	A = Not A		A	A	A	Din.2 = 0	A	
B	11	5 s	11	11					B	B	B	Din.3 = 0	B	
C	12	10 s	12	12					C	C	C	S1 = 0	C	
D	13	20 s	13	13					D	D	D	S2 = 0	D	
E	14	30 s	14	14					E	E	E	S1 = 1	E	
F	15	60 s	15	15					F	F	F	S2 = 1	F	

Recyklace



Elektronické a elektrické produkty nesmějí být vyhazovány do domovních odpadů. Likvidujte odpad na konci doby životnosti výrobku přiměřeně podle platných zákonných předpisů.

Šetřete životní prostředí! Přispějte tak k jeho ochraně!

Příklad tohoto návodu zajistila společnost Conrad Electronic Česká republika, s. r. o.

Všechna práva vyhrazena. Jakékoliv druhy kopií tohoto návodu, jako např. fotokopie, jsou předmětem souhlasu společnosti Conrad Electronic Česká republika, s. r. o. Návod k použití odpovídá technickému stavu při tisku! **Změny vyhrazeny!**

© Copyright Conrad Electronic Česká republika, s. r. o.

REI/8/2015