INSTRUKCJA OBSŁUGI

CE

Zestaw do nauki Conrad Components Profi Mikrocontroller 10104

Nr produktu 192286

Szanowni Państwo

Dziękujemy za zakup tego produktu. Produkt jest zgodny z obowiązującymi wymogami krajowymi i europejskimi.

Aby utrzymać ten stan i zapewnić bezpieczną pracę, należy przestrzegać niniejszej instrukcji obsługi! Podręcznik ten należy do tego produktu. Zawierają one ważne informacje dotyczące prawidłowego działania i obsługi. Należy brać pod uwagę zasady prawidłowej eksploatacji oraz obsługi, zwłaszcza, gdy oddajemy produkt osobom trzecim. Pamiętaj, aby przechowywać niniejszą instrukcję do wykorzystania w przyszłości!

W razie jakichkolwiek pytań technicznych należy skontaktować się z nami pod adresem/telefonem:

Klient indywidualny:

bok@conrad.pl

801 005 133*
(12) 622 98 00

(12) 622 98 10

Klient biznesowy:

b2b@conrad.pl

(12) 622 98 22

(12) 622 98 10

UWAGA!

W razie uszkodzeń spowodowanych nieprzestrzeganiem instrukcji obsługi następuje ustanie roszczeń z tytułu gwarancji! Nie ponosimy żadnej odpowiedzialności za szkody powstałe w dalszej konsekwencji takiego postępowania!

Sterowanie programowalne

Mikrokontrolery są wszędzie: w urządzeniach domowych, w urządzeniach elektroniki rozrywkowej, w metrach, a nawet w bezzałogowych pojazdach kosmicznych. Wszędzie robią rzeczy, które program mówi im. Tworzenie samych prostych programów sterujących również jest bardzo ekscytujące. Pierwszym krokiem zawsze jest wybór mikrokontrolera lub procesora pasującego do wybranego zadania tak dokładnie, jak to możliwe. Możesz wybrać dowolną liczbę typów z różnych firmy. Możesz także wybrać swój język programowania. Asembler i C są oferowane najczęściej; w wielu przypadkach jest to język podstawowy lub inny język wysoki. Programowanie zazwyczaj wymaga skomplikowane oprogramowanie i jednostki programistycznej. Oprócz wysiłku finansowego, czasochłonnej nauki.

Zastosowany mikrokontroler jest zupełnie inny. Można go zaprogramować za pomocą tylko dwóch przycisków. Sterownik programowalny za pomocą przycisków (tasten program mierbare Steuerung; TPS) zna niewiele poleceń, których można łatwo nauczyć się i które można zaprogramować w programie za pomocą przycisków. Program może być zmieniany w dowolnym czasie i bez żadnych specjalnych zmian.

System ten nadaje się szczególnie do kompaktowych zastosowań w obszarach pomiaru, sterowania i kontroli i regulacji. Wiele zadań można już teraz całkowicie rozwiązać za pomocą tego systemu. Dodatkowo, mogą Państwo zainstalować mikrokontroler do własnych obwodów po pomyślnym jego zaprogramowaniu. Wymagana jest do tego podstawowa wiedza w dziedzinie elektroniki.

Jednocześnie system nadaje się do szkolenia i pierwszych kroków w programowaniu mikrokontrolera. Sukces nastąpi szybciej niż w innych systemach. Struktury są jednak bardzo podobnie jak w innych językach programowania, aby późniejszy transfer był łatwiejszy.

1 Wprowadzenie

Zasada działania sterownika TPS jest prosta. Mają cztery wejścia cyfrowe E1 do E4 i cztery wyjścia cyfrowe A1 do A4. Istnieją również dwa wejścia analogowe AD1 i AD2 oraz quasi analogowe wyjście PWMM. Wejście resetowania za pomocą podłączonego przycisku reset zresetuje program do początku programu. Regulator jest dostarczany z trzema ogniwami AA o napięciu ok. 4,5 V i może pracować w zakresie 2.2 V do 5,5 V.

Dane techniczne:

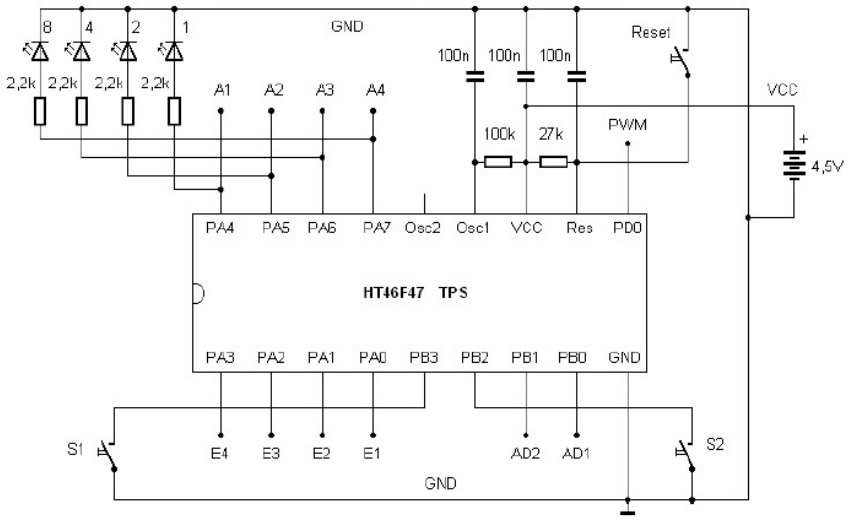
- Mikrokontroler: HT46F47F47
- Częstotliwość zegara: 2 MHz
- Wewnętrzna EEPROM: 128 bajtów
- Zasilanie VCC: 2,2 V do 5,5 V V
- Pobór prądu: 1 mA przy 4,5 V
- 4 porty wyjściowe: odporność do 10 mA
- 1 wyjście PWM: odporność do 10 mA
- 4 porty wejściowe: spokojny stan 1
- 2 wejścia analogowe: 0 V.... VCC
- 2 wejścia klawiszy: spokojny stan 1

Komponenty w pakiecie edukacyjnym:

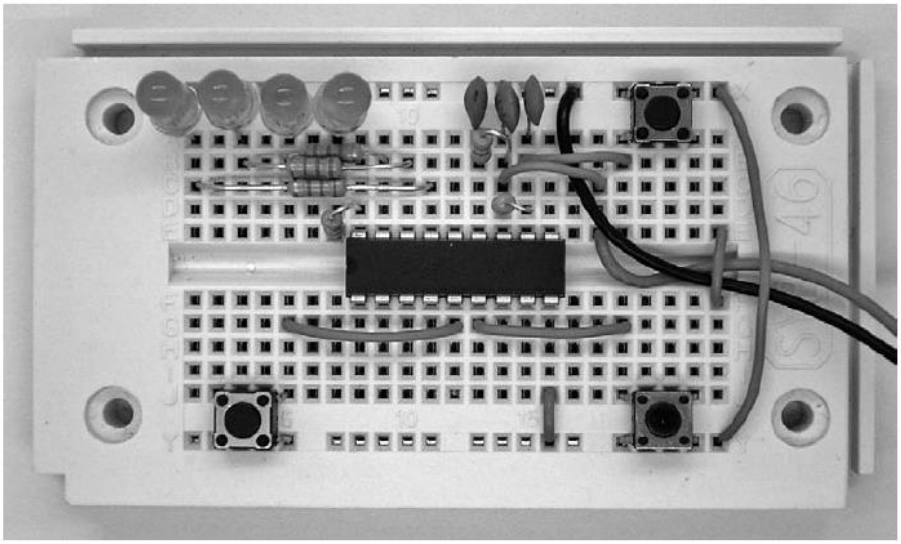
- Tablica korkowa
- Komora baterii 3 * AA
- Drut
- HT46F47 z oprogramowaniem układowym TPS
- 3 Przyciski
- 4 diody LED 5 mm, czerwony
- 1 LED 5 mm, zielony
- 1 LDR
- 3 dyskowe kondensatory 100 nF
- 1 kondensator elektrolitowy 47 µF
- 5 rezystorów 2,2 kO
- 1 rezystor 10 kO
- 1 rezystor 27 kO
- 2 rezystory 100 kO

Do programowania potrzebne są dwa przyciski S1 i S2 oraz prosty wyświetlacz LED z czterema diodami LED na wyjściach od A1 do A4. Istnieje łącznie 14 prostych poleceń z powiązanymi danymi lub podkomendy. Polecenia i dane są kodowane jako 4-bitowe liczby binarne od 0000 do 1111 (dziesiętne od 0 do 15) i są bezpośrednio widoczne na wyświetlaczach LED. Odpowiednia liczba są zaprogramowane na S1 poprzez naciśnięcie przycisków podczas programowania. S2 przełącza pomiędzy poleceniem i zwiększa dane oraz adres w wierszu poleceń. Cała struktura programu jest taka proste, że będziesz wiedział to na pamięć przy odrobinie praktyki.

Schemat podstawowego połączenia systemu



Podstawowe połączenie z dwoma przyciskami



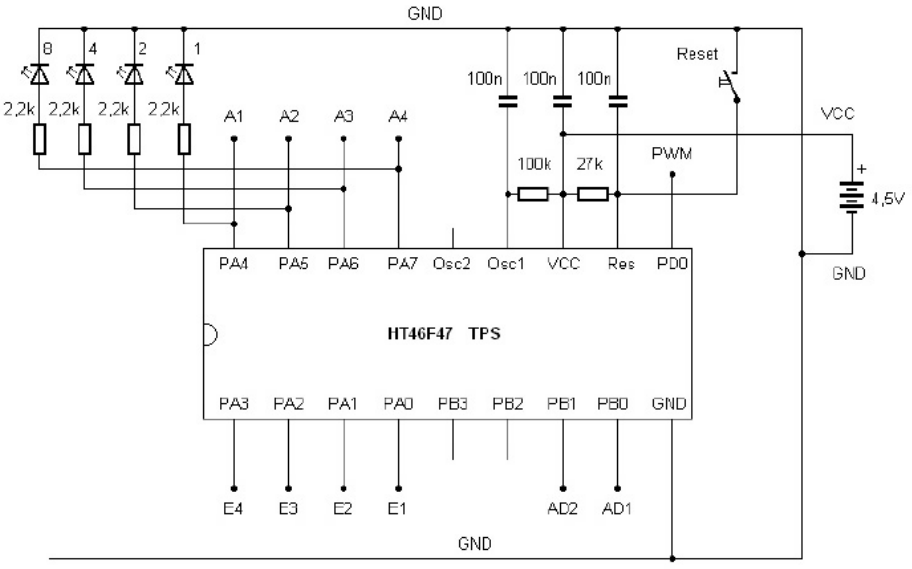
Niektóre podstawowe programy, które można uruchomić bezpośrednio lub w pojedynczych krokach, są częścią ustawień fabrycznych mikroprocesora. Zapoznaj się najpierw ze sprzętem procesora, a następnie przejdź do własnego programowania.

W pierwszym eksperymencie uruchamiasz podstawowe programy i testujesz ich główne funkcje. Do wypróbowania będą wymagane tylko podstawowe komponenty. Szczegółowy opis poszczególnych poleceń znajduje się w następnej części instrukcji. Wymagane będą następujące komponenty:

- Podłączenie zasilania pod napięciem GND (minus) i VCC (plus)
- Kondensator blokujący 100 nF między VCC i GND
- Rezystor resetujący za VCC i resetujący kondensator za GND
- Rezystor oscylatora 100 kO za VCC i kondensatorem za GND

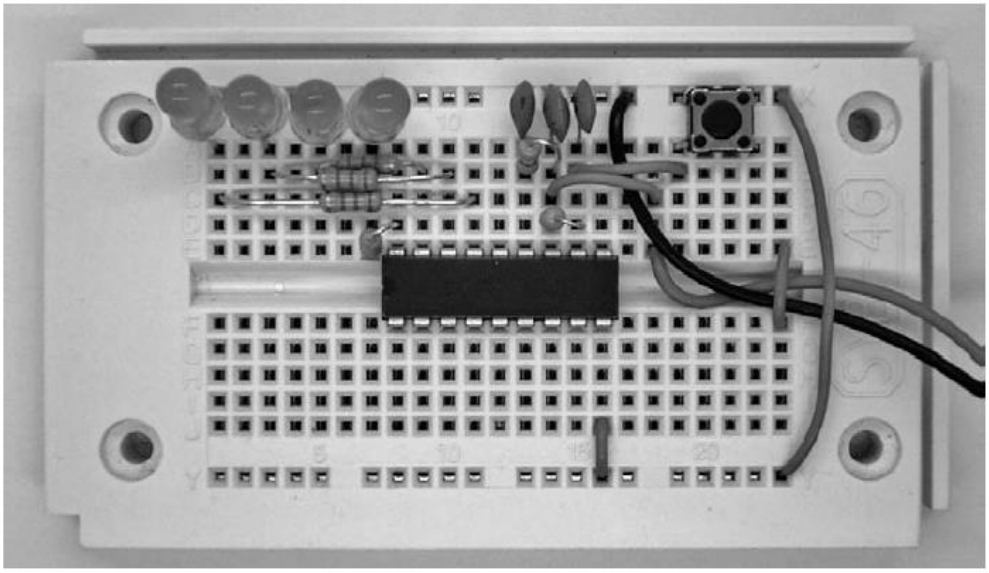
Mikroprocesor ma wewnętrzny obwód oscylacyjny RC. Rezystor wejściowy OSC1 wskazuje częstotliwość cykli. Przy impedancji 100 kΩ częstotliwość będzie wynosić około 2 MHz. Podłączony kondensator jest tylko zablokowany i nie ma wpływu na wartość częstotliwości cyklu. Wejście OSC2 pozostaje nieużywane. Można podłączyć go do dodatniej ścieżki prądu i usunąć ¼ częstotliwości cyklu.

Dla diod LED na wyjściach



Użyj górnej i dolnej szyny zasilania na tablicy połączeń jako połączenia uziemienia GND. Komora baterii ma dwa gniazda - czarny (biegun ujemny) i czerwony (biegun dodatni). Zawsze zachowuj prawidłową polaryzację podczas podłączania obwodu mikroprocesora. W

przeciwnym razie może zostać nieodwracalnie zniszczone. Użyj krótkiego przewodu, aby przymocować dwa przewody do płytki. Zapobiega to wyciągnięciu przewodów z płyty. Podłączone przewody mogą być trwale połączone z płytą. Aby wyłączyć / wyłączyć obwód, wystarczy wyjąć jedną z baterii.



Użyj przycisku reset i podłącz 4 diody LED do rezystorów 2,2 kΩ. Rezystory są niezbędne do pierwszych eksperymentów. Zachowaj je we właściwej kolejności. A1 jest podłączony do lewej diody LED oraz A4 do prawej. Wyświetlacz binarny z najwyższą wartością bitu znajduje się po lewej stronie. Jest to pomocne szczególnie podczas późniejszego programowania.

2 Alternatywna lampa błyskowa

Teraz włóż 3 sztuki standardowych baterii 1,5 V. Możesz także użyć akumulatorów Ni-MH. Spowoduje to uruchomienie programu testowego, w którym diody LED naprzemiennie włączają się (lewe i prawe diody LED) z częstotliwością około 1 Hz. Przegląd programu pokazuje tylko programy 5-liniowe. Przełączanie między dwiema diodami LED jest sterowane opóźnieniem 0,5 s. Przeskok impulsu do początku zapewnia kolejne powtórzenie procesu. Poszczególne polecenia zostaną wyjaśnione w następnej sekcji. W tym przykładzie dowiesz się, jak programowanie jest proste. Mikroprocesor ma wbudowaną funkcję interpretacji, która rozpoznaje i wykonuje proste polecenia z różnych języków programowania. Programy te zapewniają wysokie zagęszczenie niż kontrola wydajności za pośrednictwem innych systemów. W tym przykładzie zakres mieści się w przedziale od 20 godzin w porządku sekwencyjnym (w systemie dziesiętnym 32). Programy w górnym zakresie adresów można uruchamiać później z odpowiednich aplikacji. Adresy mogą być nadpisywane własnym

kodem programu. Ponadto mikrokontroler można w dowolnym momencie zresetować do pierwotnego, fabrycznie ustawionego stanu programu.

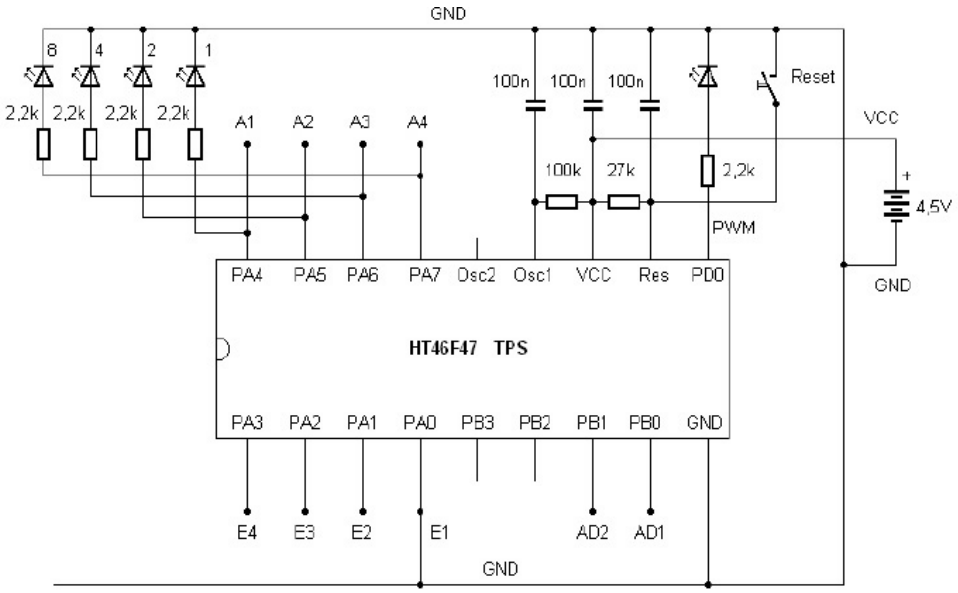
Adres	Komenda	Dane	Komentarz
20	1	1	LED 1
21	2	8	Wait for 500 ms
22	1	8	LED 8
23	2	8	Wait for 500 ms
24	3	4	Jump –4

W przypadku braku spodziewanego procesu najpierw sprawdź polaryzację podłączonego napięcia do diody LED. Możliwe jest również mierzenie pewnych wartości w obwodzie. Dlatego należy korzystać z multimetru cyfrowego o zakresie pomiarowym maksymalnie 20 V. Zmierz napięcie obwodu zawsze w stosunku do ziemi (biegun ujemny).

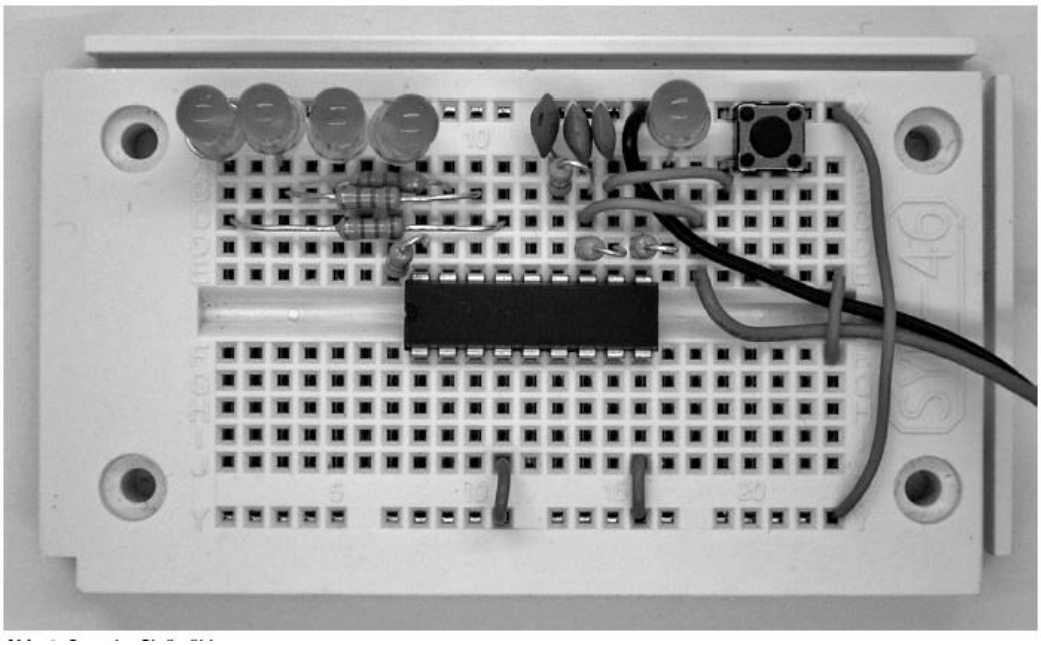
- Napięcie obwodu: 4,5 V DC
- Reset masy - 4,5 V
- Osc1 - 1,5 V E1 do E4 - 4.5 V
- A1 - Przerywany
- A2, A3 - 0 V
- A3 - Przerywany

3 Licznik binarny i wyjście PWM

Wszystkie wejścia cyfrowe są pobierane z wewnętrznego rezystora przed VCC (rezystor pull-up) i mają napięcie spoczynkowe przy napięciu roboczym. Możesz jednak umieścić każdy z nich wejścia do GND za pomocą przewodu lub styku. Podczas uruchamiania, program odczytuje domyślny port stan i ocenia je. Poszczególne połączenia można umieścić na GND, tak że stan zero. W zależności od wyniku wywoływane są różne programy.



Wykorzystanie diody PWM



Uruchamianie licznika binarnego

Połącz E1 z ujemnym GND. Po zresetowaniu mikroprocesora rozpoczyna się drugi program. Warunki wyjściowe binarne. Warunki 0000 (w systemie dziesiętnym od 0) do 1111 (w systemie dziesiętnym 15) są następnie przekazywane w sposób ciągły. Program wykorzystuje zmienną A dla prostej sumy i wyjścia do wyjść cyfrowych (wyjście PWM). Polecenia 7 i 5 mają drugorzędne funkcje zapisane jako dane.

Adres	Komenda	Dane	Komentarz
25	7	1	A = A + 1
26	5	4	Port = A
27	5	9	PWM = A
28	2	6	Wait for 100 ms
29	3	4	Jump –4

Ten program może być używany jako program testowy do odczytu liczników binarnych, które musisz kontrolować dla własnego programowania. Każda z czterech diod LED jest jednobitowa. W związku z tym można wyświetlić 4-bitową liczbę. Diody LED są nazywane 8, 4, 2 i 1, a także ich wartości na diagramie. Dodanie odpowiednich wartości skutkuje liczbą dziesiętną. Nagłówek szesnastkowy, liczby 10-15 są oznaczone wielkimi literami od A do F.

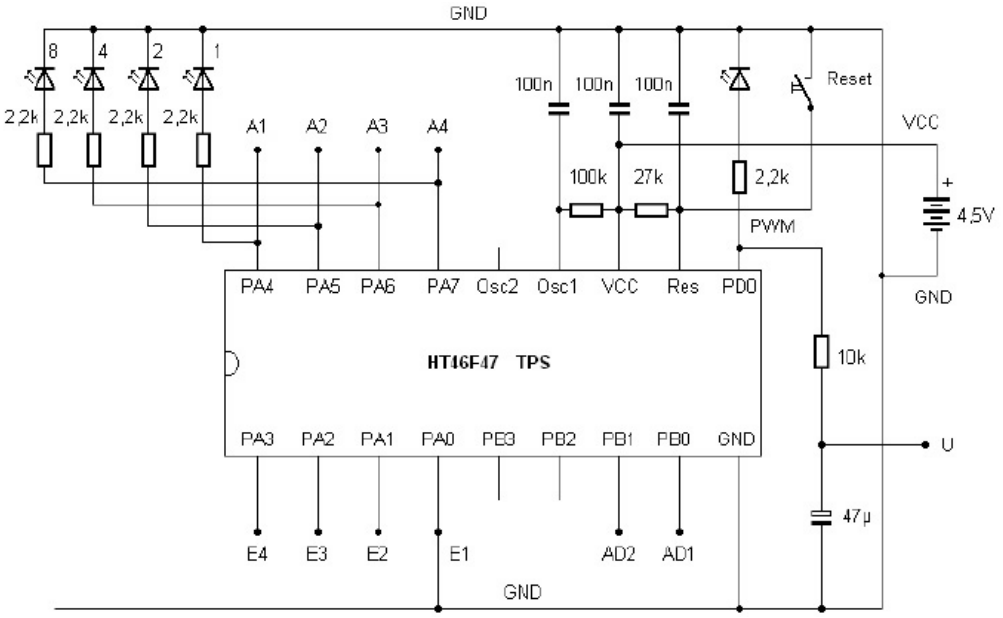
8	4	2	1	Dziesiętny	szesnastkowy
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

Ten program może być używany jako dekoder błysków na różnych częstotliwościach. Kolejna wyższa wydajność ma połowę częstotliwości lub dwa razy więcej niż czas:

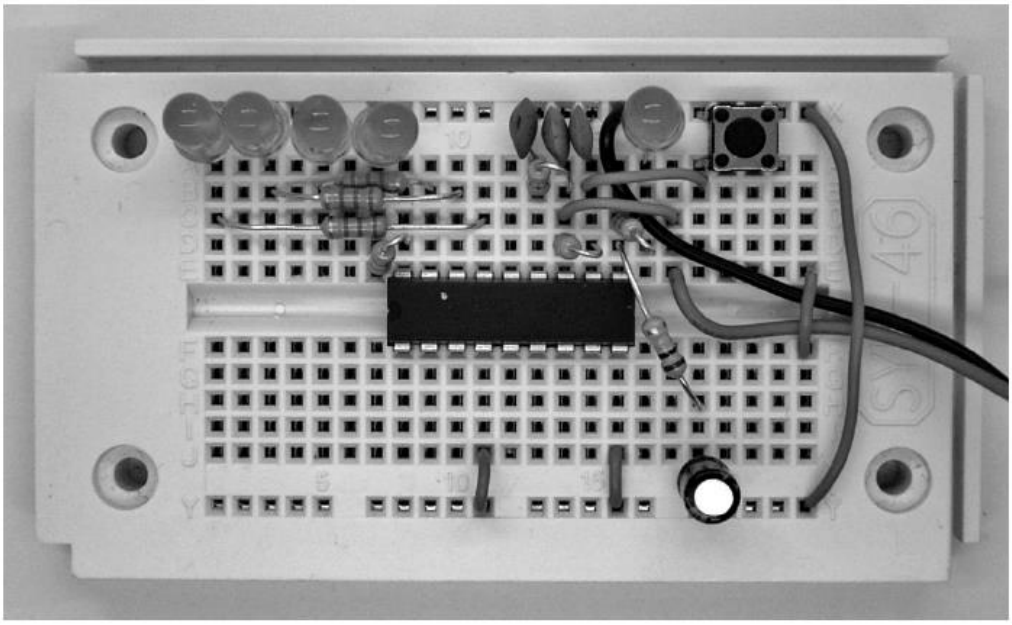
A1: 200 ms A2: 400 ms A3: 800 ms A4: 1600 ms

Dodatkowo wartości liczbowe są zwiększane przez wyjście PWM. Sygnał PWM jest prostokątnym sygnałem o częstotliwości około 16 kHz. Czas trwania impulsu jest kontrolowany, więc stosunek "przerwa w impulsie" określa średni czas trwania aktywacji, a zatem jasność LED. Jasność LED kontrolowana jest w sumie na 15 poziomach (0-15).

Sygnał PWM może być skierowany do napięcia stałego z filtrem dolnoprzepustowym RC. Wyjście PWM staje się zatem wyjściem analogowym. Ten program generuje napięcie prądu stałego, które stopniowo wzrasta od 0 V do 4,5 V. Trend napięcia należy monitorować za pomocą oscyloskopu



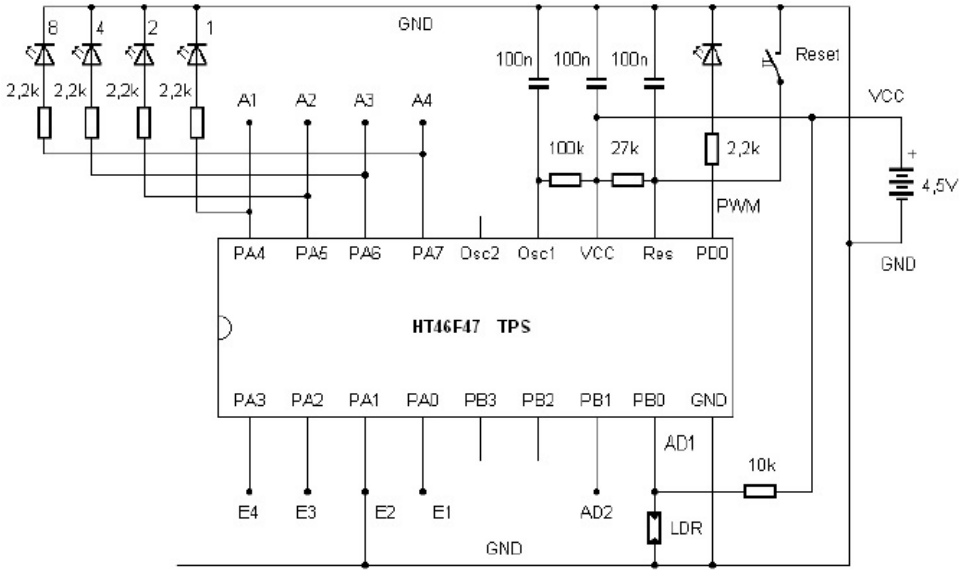
Filtr dolnoprzepustowy na wyjściu PWM



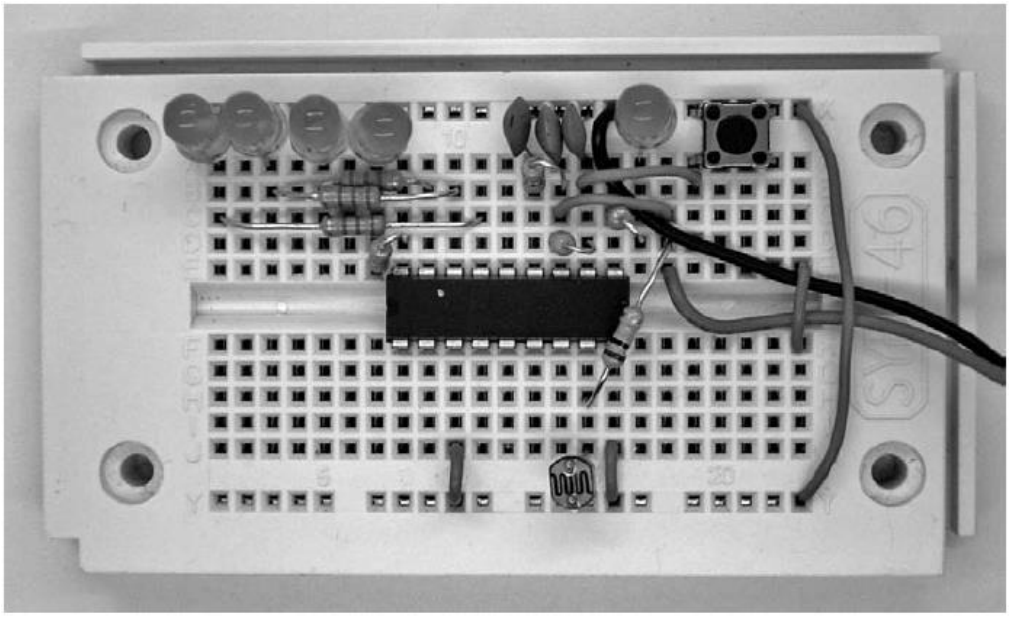
Rys. 8: Wygładzone napięcie wyjściowe PWM

4 Przetwornik analogowo-cyfrowy (ADC)

Podłącz E2 do GND i naciśnij przycisk resetowania, aby uruchomić program konwertera AD. Napięcie analogowe jest mierzone na wejściu analogowym AD1 i konwertowane na cyfrowe wartości numeryczne. Ponieważ mikrokontroler działa z 4-bitowymi wartościami, wynikiem konwersji analogowo-cyfrowej jest liczba od 0 do 15. Wynik 0 reprezentuje napięcie wejściowe 0, wynik 15 odpowiada napięciu roboczemu (w tym przypadku 4,5 V). Wartość analogowo-cyfrowa jest sygnałem wyjściowym liczby binarnej 4 diod LED transmitowanych przez wyjście PWM. Podłącz dzielnik napięcia składający się z solidnego rezystora i fotorezystora (LDR) do wyjścia analogowego AD1.



Rys. 9: Podłączenie czujnika światła



Rys. 10: LDR na wejściu AD1

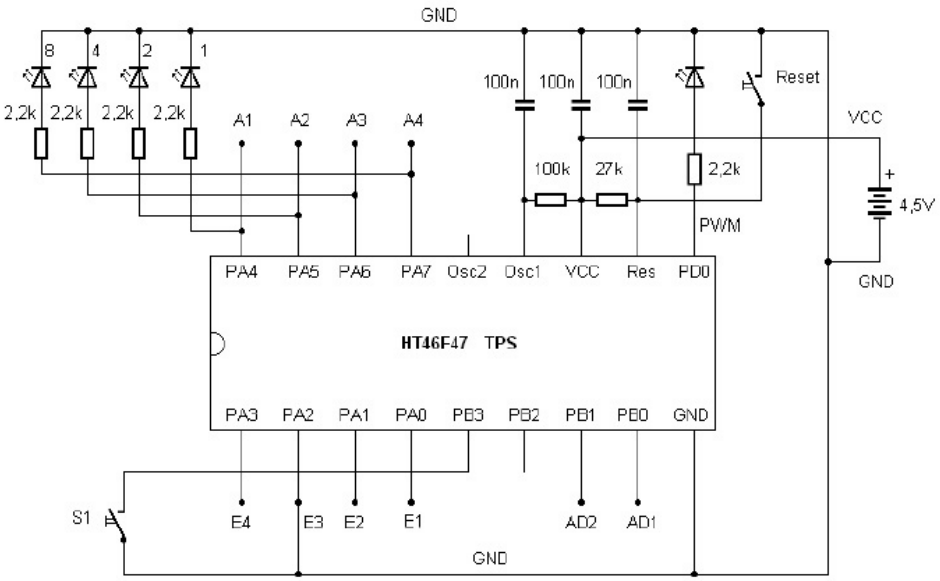
Ten program jest bardzo podobny do poprzedniego programu (wyjścia cyfrowe i wyjście PWM). Pierwsza linia to polecenie, które wykonuje konwersję wartości analogowych

Adres	Komenda	Dane	Komentarz
2A	6	9	A = AD1
2B	5	4	Port = A
2C	5	9	PWM = A
2D	2	6	Wait for 100 ms
2E	3	4	Jump –4

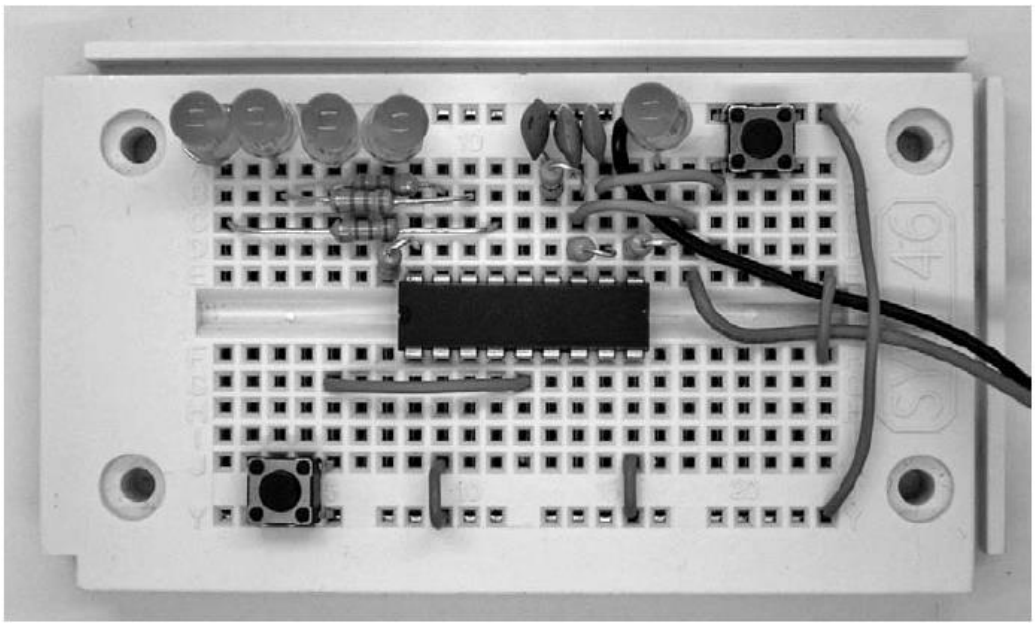
Testuje działanie programu podczas ekspozycji światłem - ekspozycja na światło. Im wyższa ekspozycja, tym niższe napięcie na AD1 i na odwrót. Im lżejszy strumień świetlny, tym wyższe wartości AD, a tym samym maksymalna jasność LED na wyjściu PWM. Przeczytaj linie binarne na wyświetlaczu LED i spróbuj ustawić jasność, np. dokładnie o połowę. Wartość cyfrowa wynosi wtedy 0111 lub 1000. Migotanie sztucznego oświetlenia może spowodować przełączanie między dwoma poziomami.

5 Generator wartości losowych

Łącząc E3 z GND, uruchamiany jest program podstawowy - generator liczb losowych. Przycisk S1 zapewnia ocenę warunków. Podłączone wejście ma wewnętrzny rezystor podciągający przechodzący dodatnie napięcie Vcc. Przycisk jest podłączony do GND. Naciśnij przycisk, aby przełączyć S1 na 0.



Rys. 11: Uruchomienie przełącznika losowego



Rys. 12: Mostek między E3 a GND

Program używa warunkowego polecenia skoku. Jeśli warunek wejściowy S1 wynosi 0, następujące polecenie Jump (Przeskok). Po naciśnięciu przycisku wartość wynosi 0, zwiększając wartość A. Prowadzi to do szybkiej sumy warunków uruchamiania. Po zwolnieniu przycisku status zostaje zachowany. Ze względu na wysoką prędkość sumowania nie można wpłynąć na ostateczny stan wyjścia, więc zawsze jest to wynik losowy.

Adres	Komenda	Dane	Komentarz
30	5	4	Port = A
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Jump –3

Naciśnięcie krótko generuje nowy wynik. Przetestuj funkcję programu, wielokrotnie generując różne wyniki. Po wystarczającej liczbie uruchomień będzie oczywiste, że występują różne wartości wyjściowe (wyjścia). Program jest zatem odpowiedni, na przykład, do zastosowania "Jako kotki do gry".

Program wykonuje funkcję licznika z maksymalną prędkością operacji matematycznych. Funkcja i prędkość robocza, przy których mikroprocesor przetwarza dane, są najlepiej widoczne. Po naciśnięciu przycisku wyjście A1 generuje prostokątny sygnał o częstotliwości około 133 Hz i okresie 7,5 ms. Zmiany w tym porcie występują co 3,75 ms. Program działa w pętli ciągłej (funkcja pętli). Wymaga to miliona ms, aby wykonać polecenie. Wyjście A4 zapewnia 16,6 Hz, co jest widoczne w postaci migających diod LED.

Jeśli jednak proces wymaga większej szybkości pracy, można zwiększyć szybkość cyklu mikroprocesora, zmniejszając opór na wyjściu OSC1. Wartość 100 kΩ zapewnia prędkość cyklu przy częstotliwości około 2 MHz. Zastąp jeden rezystor rezystorem o wartości 27 kΩ. Zwiększa to całkowitą prędkość cyklu o prawie 4x, a czas wykonania polecenia o około 0,25 ms. Jednak mikroprocesor powinien działać optymalnie przy impedancji 100 kΩ na OSC1. Działanie systemu jest wtedy bardzo niezawodne, bezpieczne i jednocześnie przy wartości roboczej, napięcie 2,2 V działa przy bardzo niskim obciążeniu prądowym.

Zmierz długość impulsu

Połączenie E4 do GND uruchamia program pomiaru długości impulsu. W tym przypadku oceniane jest wejście wejściowe S1.

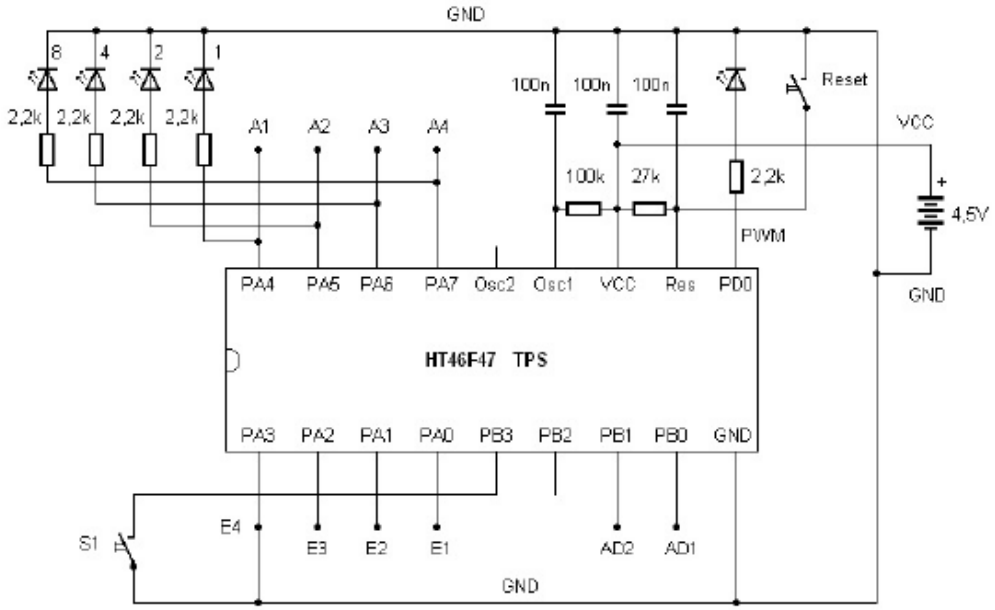
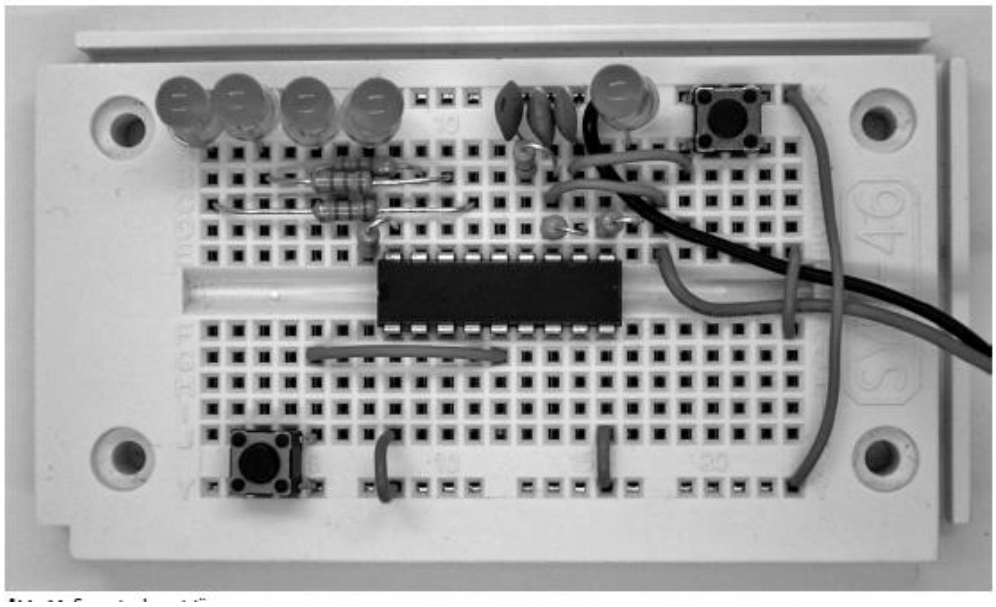


Fig 13 Połączenia E4 i GND



Pomiar rozpoczyna się od S1 = 0 (naciśnij przycisk). Przed wykonaniem całości 5. polecenia są opóźnione o 5 ms. Następne 5 ms zajmuje sam proces. Całkowity czas pomiaru tętna wynosi więc 10 ms.

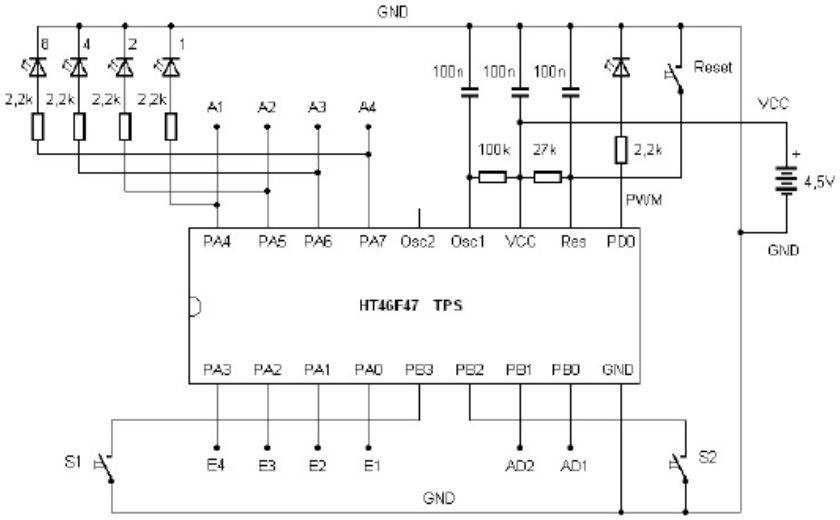
Adres	Komenda	Dane	Komentarz
34	2	2	Wait for 5 ms
35	C	C	S1 = 0?
36	3	2	Jump –2
37	4	0	A = 0
38	2	2	Wait for 5 ms
39	7	1	A = A + 1
3A	5	4	Port = A
3B	C	E	S1 = 1?
3C	3	4	Jump –4
3D	3	9	Jump –9

Naciśnij krótko S1. Wynik może wynosić, na przykład, 1010, odpowiednio. 10 (system dziesiętny). Jeśli tak jest jednostką czasu programu wynoszącą 10 ms, czas wyświetlania wynosi 100 ms. Podczas dalszych eksperymentów możesz osiągnąć wartości mniejsze niż 50 ms.

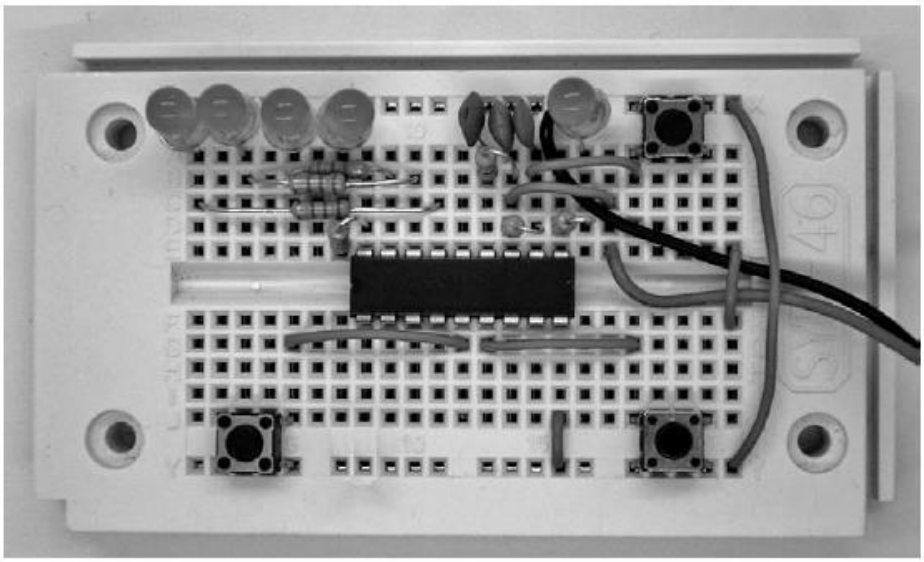
7 Wybór programu

Do programowania potrzebne są przyciski S1 (wprowadzanie danych, lewa strona) i S2 (programowanie, prawa strona). Potrzebny jest również przycisk resetowania. Tylko te przyciski mogą być używane do odczytu programów i wprowadzania programów. Przy odrobinie praktyki można bardzo szybko wprowadzać nowe programy i modyfikować je.

Aby wejść w tryb programowania, naciśnij S2 i zwolnij go po około 0,5 s. Następnie użyj przycisku S2, aby przeglądać bieżący program i przegląd zawartych w nim poleceń. Po przejściu do paska adresu naciśnij dwukrotnie S2. Spowoduje to wyświetlenie polecenia i danych w adresie. Tutaj ponownie pojawia się odpowiedni adres.



Przyciski programowania S1 i S2



3 przyciski i diody LED

- Naciśnij przycisk S2
- Wyświetlanie adresu (niższe 4 bity), 300 ms
- Wyłącz wyświetlanie, 300 ms
- Zobacz polecenie
- Naciśnij ponownie przycisk S2
- Zobacz dane
- Trzecie naciśnięcie przycisku S2
- Wyświetlaj inny adres. 300 ms

Jeśli chcesz tylko wyświetlić bieżący program i nie wprowadzać w nim żadnych zmian, naciśnij przycisk S2 przez 10x, aby przejść do końca programu. Ponieważ bieżący adres jest wyświetlany krótko, orientacja jest łatwa. Zawsze będziesz wiedział, czy na wyświetlaczu jest aktualnie wyświetlane polecenie lub dane. Domyślnie znajduje się pierwszych 5 adresów następujące polecenia. Te domyślne programy są następnie wykorzystywane do dalszej edycji i programowania niestandardowego.

Adres	Komenda	Dane	Komentarz
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?

Komenda 4-bitowa i związane z nią 4-bitowe dane razem tworzą jeden bajt, czyli 8-bitową liczbę. A pół bajt nazywany jest również "Nibble"."Nibble" dla danych. EEPROM (pamięć) mikrokontrolera zawiera łącznie 128 bajtów. Dlatego program może pomieścić łącznie do 128 poleceń. Jest to wystarczająca ilość dla większości aplikacji, ponieważ kod programu jest bardzo zwarty. Większość przydatnych aplikacji działa tylko z kilkoma poleceniami (zwykle mniej niż 10 poleceń). Zobacz przegląd poszczególnych poleceń i danych, a następnie porównaj zawartość pamięci. Następnie naciśnij ponownie przycisk resetowania. Oryginalny program w ogóle się nie zmienia.

8. Programowanie

Przycisk S1 jest aktywny tylko po zmianie polecenia lub danych lub ponownym pojawieniu się. Możliwe jest wejście liczby w zakresie 0 - 15. Pierwsze naciśnięcie przycisku S1 spowoduje wprowadzenie cyfry 0. Bieżący wpis to wyświetlane za pomocą 4 diod LED. Aby wprowadzić cyfrę 4, naciśnij przycisk S1 5 razy: 0, 1, 2, 3, 4.

Obraz binarny ma wtedy wartość 0100.Jeśli wprowadziłeś komendę lub dane, ponownie naciśnięcie przycisku S2 powoduje przesłanie danych do EEPROM. Jako potwierdzenie wyświetlacz wyłączy się na 600 ms, a następnie wyświetli się inny ekran. To krótkie opóźnienie jest niestandardowym krokiem programowania. Jest ono spowodowane oszczędzaniem energii, która jest używana do przesyłania danych do EEPROM. Możesz zmienić bieżący program na tej samej pozycji pamięci. Aby wybrać pozycję, użyj przycisku S2, aby edytować polecenie lub dane. Następnie zapisz zmiany za pomocą przycisku S2. Pierwsza

część jest zaprogramowana tylko do użytku dwa polecenia. Ma to na celu włączenie diody LED do pętli.

Oświetlenie LED

Adres	Komenda	Dane	Komentarz
00	1	7	A1–4 = 0111
01	3	0	Jump 0

Możesz wybrać szczegółową listę lub tylko skróconą formę. Te dwa bajty są podsumowane w zapisie heksadecymalnym (szesnastkowym): 17 h, 30 h. Jednak programy są napisane w skróconej formie bez etykieta "h": 17 30. Aby ukończyć to zadanie, wykonaj następujące czynności: S2 + reset, 2 x S1, S2, 8 x S1, S2, 4 x S1, S2, 1 x S1, S2

Jeśli przycisk S1 został przypadkowo naciśnięty, nadal można wprowadzić prawidłową liczbę. Przewiń w dół do numeru 15, a następnie do 0. Po wprowadzeniu nowego wpisu wprowadź nowy program jest on uruchamiany po naciśnięciu przycisku resetowania Widać czy podłączone są trzy diody LED. Nic więcej się nie dzieje. Regulator nie reaguje już na to, aby warunki przy wejściach E1 do E4 już więcej, ponieważ pierwotny program był częściowo realizowany nadpisywany. Programy przykładowe również nie mogą już być uruchamiane. Ponieważ zmieniłeś tylko dwa pierwsze adresy pamięci, możesz łatwo uruchomić oryginał. ponownie. W tym celu po prostu wpisz na nowo dwa pierwsze polecenia (64 51) zgodnie z lista z ostatniej sekcji. Sprawdzić pierwotną funkcję programów przykładowych. Najlepiej przystąpić do nowego programu ćwiczeń ponownie. Szybko zyskasz bezpieczeństwo podczas obsługi wprowadzania programu.

9 Przywrócenie wzorcowych programów

Jeśli chcesz przywrócić mikroprocesor do wartości fabrycznych, wprowadź 2 wartości FF. Po wystąpieniu resetowania aby wymazać pamięć EEPROM. Oprogramowanie mikroprocesorowe ma funkcję uruchamiania dwóch pierwszych adresów, rozpoznając w ten sposób pustą pamięć. Jeżeli załadowane są 2 bajty FF, mikroprocesor zakłada, że nie ma danych w pamięci. Dzięki temu jest automatycznie ładowany do EEPROM mikroprocesora lista programów fabrycznych. Zresetuj do wartości fabrycznych można zrobić w dowolnym momencie.

Adres	Komenda	Dane	Komentarz
00	F	F	-
01	F	F	-

Uruchom tryb programowania, naciskając S2. Wprowadź wartość F (dziesiętnie 15), ogółem 4x, aż dioda LED A1 do A4 będzie świecić. Potwierdź wpis, naciskając S2. Więc zrób to aby zresetować. Spowoduje to zresetowanie mikroprocesora (przywrócenie ustawień fabrycznych) i przywrócenie wszystkich oryginalnych programów.

10 Podstawowe polecenia mikrokontrolera

Programowalny mikroprocesor rozpoznaje łącznie 14 poleceń (1 - 14). Większość tych poleceń ma parametry w postaci 4-bitowego numeru 0000-1111 (0-F), tj. z zakresem do 15 (notacja dziesiętna). Dodatkowe polecenia mają podfunkcje i są wprowadzane wraz z innymi parametrami. Kod polecenia więc może zawierać do 16 pod-poleceń. Na przykład polecenie 7 oznacza "Oblicz A = ..." (obliczyć A = ...). Kolejny parametr wskazuje, które obliczenia zostaną wykonane.

Polecenia i dane są zapisywane razem w kodzie heksadecymalnym jako 1 bajt. Polecenie 1 razem za pomocą parametru 4 następuje polecenie 14h (hex). Szesnastkowa postać jest pomijana, ponieważ wszystkie polecenia i adresy są generalnie zapisane w kodzie heksadecymalnym.

Pierwsze trzy polecenia to:

10-1F: bezpośrednie wyjście portu na A1-A4, 0-15, bin. 0000 do 1111

20-2F: Czas 0-15

(1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 30000, 60000 ms)

30-3F: Jump 0 - 15

Polecenie 1 służy do wyprowadzania stałej liczby. Pozwala to na otrzymanie wyniku z innego rastra, na przykład przełączanie kilku diod LED w tym samym czasie.

Kolejne polecenie 2, które czeka na uruchomienie, używa parametru, który zawiera czas (ms) i przełączanie sekwencyjne 1-2-5. Pomimo stosunkowo niskiego zakresu 0-15, jest to możliwe aby ustawić parametr czasu początkowego (opóźnienie) w ciągu 1 ms - 1 min. Na programowanie wielu poleceń z ustawionym opóźnieniem do uruchomienia może być czasem całkowitym nawet znacznie dłużej (Loop).

Polecenie "Powrót" 3 jest bardzo proste i wystarczające dla wszystkich aplikacji, gdzie konieczne jest powtórzenie procesu do nieskończoności. Krok powrotu (skok) jest

ograniczony do obszaru 15. Return (Jump) jest związany z bieżącym adresem, podczas gdy części programu można przenieść na dowolne adresy

Program flash zmienny potrzebuje tylko tych trzech poleceń. Jest on wpisywany do obszar adresowy od 00 tutaj w nieznacznie zmienionej formie. Próbki bitów wyjściowych i czasy oczekiwania są również zmieniane.

Adres	Komenda	Dane	Komentarz
00	1	1	A1–4 = 0001
01	2	7	Wait for 200 ms
02	1	4	A1–4 = 0100
03	2	7	Wait for 200 ms
04	3	4	Jump -4

Lista 9: Migający program

W trybie szesnastkowym program może wyglądać następująco: 11 27 14 27 34

Na podstawie tych pierwszych trzech poleceń bazuje zasada wielu prostych programów. Sprawdź działanie następujących 3 programów. Pod koniec eksperymentu powinien być intuicyjny użycie wszystkich poleceń. Stworzenie podobnych programów będzie kwestią oczywistą.Prostym przykładem jest aplikacja z jedną diodą LED i kilkoma rastrami wyjściowymi:

Adres	Komenda	Dane	Komentarz
00	1	1	LEDs 0001
01	2	8	Wait for 500 ms
02	1	2	LEDs 0010
03	2	8	Wait for 500 ms
04	1	4	LEDs 0100
05	2	8	Wait for 500 ms
06	1	8	LEDs 1000
07	2	8	Wait for 500 ms

Adres	Komenda	Dane	Komentarz
08	3	8	Jump –8

11 28 12 28 14 28 18 28 38

Lista 10: Światło bieżące 1

Rozszerzenie programu o dwa kolejne schematy wyjściowe, tak aby punkt świetlny cofał się, oraz kroczył naprzód. Eksperymentuj z innymi wzorcami wyjściowymi i czasami opóźnienia.

Adres	Komenda	Dane	Komentarz
00	1	1	LEDs 0001
01	2	8	Wait for 500 ms
02	1	2	LEDs 0010
03	2	8	Wait for 500 ms
04	1	4	LEDs 0100
05	2	8	Wait for 500 ms
06	1	8	LEDs 1000
07	2	8	Wait for 500 ms
08	1	4	LEDs 0100
09	2	8	Wait for 500 ms
0A	1	2	LEDs 0010
0B	2	8	Wait for 500 ms
0C	3	C	Jump –12

11 28 12 28 14 28 18 28 14 28 12 28 3C

Lisa 11: Uruchomienie światła 2, tam i z powrotem.

Przełącznik czasowy może zawierać opóźnienie do jednej minuty za pomocą polecenia oczekiwania. Na końcu, istnieje skok w tył z szerokością skoku 0, to znaczy nieskończoną pętlą bez zawartości, która służy jako koniec programu. Restart uruchamiany jest za pomocą przycisku reset. Rozwiń program do trzy minutowego minutnika kuchennego. Możesz wyświetlić pozostały czas przez liczbę zapalonych diod LED jako lekki pasek.

Adres	Komenda	Dane	Komentarz
00	1	F	LEDs 1111
01	2	F	Wait for 1 min
02	1	0	LEDs 0000
03	3	0	End

1F 2F 10 30

Lista 12: Przełącznik czasowy na jedną minutę

11 Obliczanie za pomocą zmiennych

Do tej pory parametry używały tylko stałych wartości liczbowych w parametrach poszczególne polecenia. Jest to sensowne, gdy program ma działać w ten sam sposób za każdym razem. Bardziej złożone programy działają jednak ze zmiennymi danymi. Na przykład. obliczenia takie jak A = A + B może zostać wykonany. W zależności od zawartości zmiennych A i B, wynik będzie za każdym razem inny. Wynik może na przykład kontrolować diody LED na wyjściach w następujący sposób.

Kontrola ma cztery zmienne A, B, C i D. Najważniejszą zmienną jest A. Jest również nazywana akumulatorem lub skrótem Accu. A bierze udział we wszystkich operacjach komputerowych i otrzymuje wynik obliczeń. A służy również do transportu danych. B jest głównie potrzebny do operacji obliczeniowych. C i D mogą być używane jako pamięci tymczasowe i są później potrzebne jako liczniki do zliczania pętli.

Istnieją również dwa wejścia analogowe (AD1 i AD2) oraz wyjście PWM. Przetwarzane dane są ograniczone do czterech bitów i dostępne tylko za pośrednictwem zmiennej A (A = AD1, PWM = A). Akumulator A można również załadować bezpośrednio za pomocą liczby (polecenia 40-4F). Aby wypełnić B, C lub D, najpierw załaduj A, a następnie przypisz zawartość do drugiej zmiennej (polecenia 51-53). A i B pozwalają na pewne kroki obliczeniowe (polecenia 71-7A).

Polecenia 40-4F przydzielają A nową wartość. Grupa poleceń 51-5A przenosi zawartość A do celu, takiego jak inna zmienna lub wyjście PWM. W tej grupie znajdują się również polecenia ustawiające pojedynczy bit portu wyjściowego.

Drugi kierunek danych jest obecny w grupie poleceń 61-6A, gdzie dane ze źródła są odczytywane do A. Grupa poleceń 71-7A w końcu wykonuje kilka operacji obliczeniowych z wynikiem ogólnie pojawiającym się w A. Port wyjściowy D out zawiera cztery wyjścia od A1 do A4, które mogą być wyzwalane albo razem albo jako pojedyncze bity D out.0 do D out.3. Wejścia od E1 do E4 są wyzwalane równo jako port wejściowy Din.

40–4F: A = 0–15

51–5A: Target 1–9 = A

51: B = A

52: C = A

53: D = A

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 = A

61–6A: A = 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: A = 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 And B

78: A = A Or B

79: A = A Xor B

7A: A = Not A

Jeden przykład użycia zmiennej A znajduje się w przykładach programu w rozdziale 3. Program został ustawiony tutaj na adres Zero i nieco rozwinięty. Dodatkowo istnieje zdefiniowany początek z wartością 0 w zmiennej A. Adres 01 zawiera komendę obliczeniową, tutaj zwiększa się o 1. Zawartość zmiennych A jest następnie przekazywana do wyjścia PWM i portu wyjściowego.

Adres	Komenda	Dane	Komentarz
00	4	0	A = 0
01	7	1	A = A + 1
02	5	4	Port = A
03	5	9	PWM = A
04	2	6	Wait for 100 ms
05	3	4	Jump –4

40 71 54 59 26 34

Lisa 13: Zwiększenie o 1

Inny przykład został już pokazany w rozdziale 4. Dane pochodzą z wejścia analogowego AD1 i są przesyłane do portu wyjściowego i wyjścia PWM. Zmodyfikowany program zawiera dodatkowy etap obliczeniowy, tj. Inwersję zawartości zmiennej A. W ten sposób zmienia wartość 0000 na nową wartość 1111, tj. 0 staje się 15 i na odwrót. Rosnące napięcie wejściowe prowadzi zatem do zmniejszenia wyjściowego PWM.

Adres	Komenda	Dane	Komentarz
00	6	9	A = AD1
01	5	4	Port = A
02	7	A	A = Not A
03	5	9	PWM = A
04	2	6	Wait for 100 ms
05	3	5	Jump –5

69 54 7A 59 26 35

Lista 14: Invert

12 skoki i gałęzie

Do tej pory istniał tylko prosty powrót (polecenie 3), który powrócił do 15 adresów. Teraz dodajemy skok absolutny. Ponieważ cel skoku można wskazać tylko za pomocą 4-bitów, istnieje dodatkowe polecenie określające wysoki przekór adresu. Daje to przestrzeń adresową 0-255. Jest to więcej, niż potrzebujesz, ponieważ EEPROM kontrolera zawiera tylko 128 bajtów, tj. Obszar 00 do 7F (dziesiętnie 0 do 127). Pamięć jest wirtualnie podzielona na osiem stron, strony od 0 do 7. Strona celu skoku musi być określona przed skokiem bezwzględnym.

Do tej pory wprowadziliśmy proste polecenie 3 (skokowy powrót) na 15 adresów. Ta sekcja rozwinie to polecenie do wartości bezwzględnych. Polecenie przeskoku może być wskazane tylko przez 4 bity. Polecenie rozszerzone określa indywidualne adresy "High nibble". To rozszerza adres w zakresie 0-255, czyli więcej niż może obsłużyć EEPROM (128 bajtów), adres 00-7F (0-127). Pamięć jest podzielona na 8 stron (strony 0 - 7). Strona polecenia Skok musi zostać określona przed poleceniem Skok absolutny. 2 zsumowane pętle ze zmiennymi C i D są wykonywane przez Absolute Jump jest adresowana przez stronę pamięci, która została wcześniej określona.

Dwie pętle zliczające ze zmiennymi C i D wykonują również skoki bezwzględne, przy czym strona adresu musi być tu również podana.

Skoki warunkowe działają jako polecenia pominięcia. Gdy odpowiedni warunek jest prawdziwy, adres jest pomijany. Tam, na przykład może zostać napisane polecenie skoku lub polecenie obliczeniowe. Warunki mogą być porównaniami między A i B lub bezpośrednimi zapytaniami bitowymi portu wejściowego.

Istnieje możliwość uruchomienia pod procedury (wywołanie) i powiązanej instrukcji return. Jednocześnie może mieć miejsce kilka podprogramów. Jednak podprogram nie ma możliwości uruchomienia innego podprogramu, ponieważ interpreter ma tylko poprzedni adres w pamięci.

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 from subprogramme

Jeden przykład użycia poleceń skoku warunkowego znajduje się w przykładowym programie w rozdziale 6. Tutaj został nieznacznie zmodyfikowany i wprowadzony w adres 0. Ponieważ górna część adresu (Adr-hi) znajduje się w stanie spoczynku 0 i Sterownik uruchamia się na stronie 0, nie trzeba tutaj używać polecenia 80. Długość naciśnięcia przycisku jest mierzona i

wyświetlana ponownie. Wszystkie polecenia oczekujące zostały usunięte z programu, więc teraz działa z wyższą rozdzielczością.

Adres	Komenda	Dane	Komentarz
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

CC 31 40 71 54 CE 33 37

Lista 15: Reakcje na przycisk S1

Polecenie skoku CC w adresie 00 ocenia stan za pomocą przycisków S1. W stanie spoczynku S1 = 1. Warunek jest zatem niezgodny z prawdą, a polecenie w adresie 01 nie jest pomijane. Na początek jest polecenie względnego skoku. Program powtarza polecenia w adresach 00 i 01, aż przycisk zostanie naciśnięty. Warunek staje się prawdziwy i adres 01 jest pomijany. To rozpoczyna właściwy proces pomiarowy. Akumulator jest kasowany, a następnie stale zwiększany o 1 i wysyłany do diod LED. Kolejne warunkowe polecenie skoku CE znajduje się w adresie 05. Warunkiem pominięcia polecenia jest S1 = 1. Ponieważ przycisk jest nadal wciśnięty na początku, warunek nie jest spełniony. Komenda w 06 jest zatem wykonywana i powoduje powrót do 03. Dopiero po zwolnieniu przycisku program przejdzie do adresu 07 i tym samym powróci do początku.

Wejdź do programu i przetestuj go. Czas reakcji nie jest dużo szybszy. Jednostka czasu wynosi około. 5 ms.

Oryginalny przykładowy program jest nadal w pamięci od adresu 34h, ponieważ tylko niższe adresy zostały nadpisane. Napisz mały program zawierający tylko skok do tego adresu. Najpierw musisz wskazać stronę 3. Następujący skok bezwzględny o podanym adresie 4 jest następnie kierowany na adres 34.

Adres	Komenda	Dane	Komentarz
00	8	3	Page 3
01	9	4	Address = 34

83 94

Lista 16: Skok bezwzględny do programu pomiaru czasu

Oryginalny program przykładowy jest więc ponownie wywoływany. Sprawdź to także w przypadku innych przykładów. Pełny przegląd wszystkich możliwych programów znajduje się w załączniku.

13 Przegląd poleceń

Wszystkie polecenia na pierwszy rzut oka - to upraszcza pracę z kontrolerem. Poniższa tabela zawiera całe zasoby poleceń w zwartej formie.

	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	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	

	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	

14 Zliczanie pętli

Należy przeprowadzić proces, np. pięciokrotnie. W tym celu tworzy się pętlę zliczającą. Polecenie skoku wykonuje się w tym przypadku pięć razy, a następnie już nie. Zmienna licznika nazywa się C. Wartość zliczania 5 musi być załadowana najpierw w A, a stamtąd w C. Polecenie A2 wykonuje wartość bezwzględną na 02 i jednocześnie redukuje zawartość zmiennej C o 1. Gdy C osiągnął wartość 0, skok nie jest już wykonywany. Skok bezwzględny odnosi się do wskazanej strony. W przypadku programu na stronie 0, polecenie strony 80 może również zostać pominięte. Jest to jednak konieczne podczas przeskakiwania do dowolnej innej strony.

Adres	Komenda	Dane	Komentarz
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	Page 0
07	A	2	C-times 02
08	3	0	End

45 52 15 28 1A 28 80 A2 30

Listia 17: Pętla zliczania

Przetestuj program. Diody LED pokazują wzory 0101 i 1010 przy każdym przejściu. Jednak ta część programu oczywiście nie przeszła pięciokrotnie, ale dokładnie sześć razy. Polecenie skoku w adresie 07 wykonuje się dokładnie pięć razy, ale aby przejść do tego punktu po raz pierwszy, wykonywany jest migający proces. Dlatego program będzie migał sześć razy w sumie

Zmień zmienną licznika na wartość 4 i ponownie przetestuj program. Teraz diody LED będą błyskać dokładnie pięć razy.

Możesz także użyć pętli licznika, aby nie skakać z powrotem, ale do przodu. Tym razem proces jest faktycznie wykonywany pięciokrotnie, kiedy C zostało załadowane wartością 5 na początku. Pomijany adres 04 zawiera względny skok do siebie, a zatem nieskończoną pętlę, która służy jako koniec programu.

Adres	Komenda	Dane	Komentarz
00	4	5	A = 5
01	5	2	C = A
02	8	0	AdrHi = 0
03	A	5	C-times 05
04	3	0	End
05	1	5	Port = 0101
06	2	8	Wait for 500 ms
07	1	A	Port = 1010
08	2	8	Wait for 500 ms
09	3	6	Jump –6

45 52 80 A5 30 15 28 1A 28 36

Lista 18: Pięć błysków

15 Porównanie

Należy porównać dwie wartości liczbowe. W zależności od wyniku porównania wykonywany jest skok. Dwie wartości liczbowe muszą znajdować się w A i B. W poniższym przykładzie ładuje się B z liczbą 5. A odbiera swój wynik z wejścia analogowego AD1. Tutaj, na przykład można podłączyć czujnik światła, jak w rozdziale 4. Program powinien teraz wykonywać następujące czynności w sposób ciągły:

Jeśli AD1> 5

następnie: wszystkie diody LED są włączone

w przeciwnym razie: wszystkie diody LED zgasną

W sumie otrzymasz przełącznik zmierzchowy. Ponieważ LDR jest podłączony do GND, większa jasność powoduje niższe napięcie w AD1. Diody LED zgasną, gdy przekroczona zostanie pewna jasność, a zatem podcięte zostanie określone napięcie. Limit wynosi 6, ponieważ wynik pomiaru musi przekraczać 5.

Adres	Komenda	Dane	Komentarz
00	4	5	A = 5
01	5	1	B = A

Adres	Komenda	Dane	Komentarz
03	6	9	A = AD1
04	C	1	Skip if A>B
05	9	8	Adr 08
06	1	F	LEDs 1111
07	3	4	Adr 03
08	1	0	LEDs 0000
09	3	6	Adr 03

45 51 80 69 C1 98 1F 34 10 36

Lista 19: Prosty przełącznik zmierzchowy

Przetestuj program, zmniejszając zasłonę światła ręką. Przekonasz się, że podstawowa funkcja jest spełniona. Jednak zwykle występuje nieprzyjemny efekt uboczny. Dokładnie na progu między On i Off, diody LED będą migać niekontrolowane. Zwłaszcza w sztucznym świetle, jasność będzie szybko zmieniać się wokół pewnej średniej. To migotanie jest poprawnie oceniane przez program, ale wynik nie jest tym, czego można oczekiwać od przełącznika zmierzchowego. Rozdział 18 pokazuje ulepszony przełącznik zmierzchowy.

16 AND, OR i XOR

Dwa binarne warunki można połączyć w nowy warunek. Jednym z przykładów jest funkcja AND: Gdy bit 1 ma warunek 1 AND bit ma warunek 1, warunek wyjściowy jest równy 1. Liczby binarne z kilkoma bitami mogą być również połączone w ten sposób. Link "10 AND 3 = 2" staje się zrozumiały przy zapisywaniu go w liczbach binarnych:

1010 AND

0011 =

0010

Poniższy program łączy warunki wejściowe ze stałą liczbą 3. Funkcja AND praktycznie powoduje zamaskowanie dwóch niższych bitów (odfiltrowanie). W stanie spoczynkowym port wejściowy ma warunek 1111. Łącze AND z wartością 0011 dostarcza wtedy stanu 0011 na diodzie LED. Jeśli jednak podłączysz jedno z wejść E1 lub E2 do GND, warunek 0 jest również widoczny na innych wyjściach. Zmiany w E3 i E4 nie mają żadnych skutków.

Adres	Komenda	Dane	Komentarz
00	6	4	A = Din
01	5	1	B = A
02	4	3	A = 3

Adres	Komenda	Dane	Komentarz
03	7	7	A = A And B
04	5	4	Port = A
05	3	5	Jump –5

64 51 43 77 54 35

Lista 20: Zastosowanie funkcji AND

Zmień program i przetestuj również inne funkcje logiczne. Funkcja OR (78) może być używana do generowania określonych warunków wejściowych do 1: 64 51 43 78 54 35

1010 OR

0011 =

1011

Użyj funkcji XOR (wyłącznej lub 79) do odwrócenia poszczególnych bitów: 64 51 43 79 54 35

1010 XOR

0011 =

1001

17 programy podrzędne

Kiedy części programu mają zostać ponownie użyte, zapisz je w podprogramie. To często oszczędza miejsce w pamięci, a czasem także dużo pisania. Poniższy przykład pokazuje użycie

podprogramu wywoływanego w dwóch miejscach w programie głównym. Podprogram zawiera tylko jedną instrukcję (A = A-1) i polecenie skoku powrotu tutaj. Nie oszczędza to pamięci, ale przykład ma jedynie na celu zademonstrowanie poleceń CALL i RET.

Główny program:

Adres	Komenda	Dane	Komentarz
00	8	0	AdrHi = 0
01	D	8	Call 08
02	5	4	Output
03	2	9	Waiting 1 s
04	D	8	Call 08
05	5	4	Output
06	2	8	Waiting 0.5 s
07	3	7	Jump –7

Podprogram:

Adres	Komenda	Dane	Komentarz
08	7	2	A = A-1
09	E	0	Ret

80 D8 54 29 D8 54 28 37 72 E0

Lista 21: Pod program calls

Wynikiem programu jest licznik binarny zliczający w dół z nierównomiernym opóźnieniem. Przetestuj także inne polecenia w podprogramie.

Istnieje kilka przydatnych podprogramów do ogólnego zastosowania wśród programów przykładowych w warunkach realizacji. Są one wymienione w załączniku. Tylko adres wejściowy musi być znany do ich użycia:

50: Podprogram: Długie brzmienie

52: Podprogram: Krótki dźwięk

53: Podprogram: Dowolny dźwięk, długość w A

60: Podprogram: Poczekaj na naciśnięty przycisk S1

68: Podprogram: Poczekaj na wciśnięty przycisk S2

70: Podprogram: Numer wejścia z S1 i S2

Podprogram od adresu 60 jest używany tylko do ustawienia licznika sterowanego przyciskiem S1. Odczyt licznika rozpoczyna się od 0. Główny program jest stosunkowo krótki, ponieważ złożone zadanie zapytań o przyciski zostało zlecone do podprogramu.

Adres	Komenda	Dane	Komentarz
00	4	0	A = 0
01	5	4	Output
02	7	1	A = A+1
03	8	6	Page 6
04	D	0	Call 60, button S1
05	3	4	Jump –4

40 54 71 86 D0 34

Listia 22: Licznik kontrolowany przez S1

Przetestuj program. Jeśli dziesięć razy popchniesz S1, wynik powinien wynosić 1010. Zmień program tak, aby podprogram był używany od adresu 68. Teraz licznik reaguje na S2.

18 Przełącznik zmierzchowy

Przełącznik zmierzchowy włącza lampę, gdy jasność otoczenia spada poniżej pewnej granicy. Kiedy światło staje się jaśniejsze, lampa ponownie gaśnie. Należy upewnić się, że światło nie migocze na progu między światłem a ciemnością. Jest to możliwe w przypadku histerezy, tj. Pewnej odległości między jasnością aktywacji i dezaktywacji. Program przedstawiony tutaj działa z następującymi regułami:

• Jeśli napięcie w AD1 nie przekracza 5, lampa jest wyłączona.

• Jeśli napięcie w AD1 nie jest niższe niż 9, lampa jest włączona.

Zapewnia to obszar środkowy, w którym warunek wyjściowy nie może się zmienić. Ta luka zapobiega migotaniu diod LED.

0-5: Diody LED wyłączone

6-8: Diody LED niezmienione

9-15: Diody LED włączone

Adres	Komenda	Dane	Komentarz
00	1	0	LEDs 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	LEDs 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	LEDs 1111
0B	3	A	Jump –10

10 45 51 69 C1 10 49 51 69 C2 1F 3A

Lista 23: Przełącznik zmierzchowy z histerezą

19 LED-Dimmer

Celem tego przykładu programowania jest sterowana lampa LED. Jasność diody na wyjściu PWM należy regulować za pomocą przycisków. Możesz krótko nacisnąć przycisk, aby przejść do następnego etapu jasności, lub możesz naciskać, aby ciągle zmieniać jasność.

W rdzeniu programu używane są polecenia pominięcia, które są już znane. Jeśli odpowiedni przycisk nie zostanie naciśnięty, powiązane polecenie zwiększania lub zmniejszania zawartości akumulatora jest pomijane. Problem polega na tym, że zazwyczaj może to doprowadzić do przekroczenia od 15 do 0 lub od 0 do 15. Zapobieganie temu przekroczeniu wymaga nieco więcej wysiłku. W tym celu należy zapytać, czy dolny koniec (0) lub górny koniec (15) zostały już osiągnięte. Ponieważ akumulator zawsze bierze udział w porównywaniu, jego zawartość musi być umieszczona w pamięci tymczasowej. Jest to używane przez ustawienie zmiennej C.

Adres	Komenda	Dane	Komentarz
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

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

Lista 24: Kontrola jasności

20 Blokada numeru

Przedstawiona tutaj blokada numeru włącza wyjście PWM, jeśli użytkownik wprowadzi poprawną sekwencję numerów. Wejście numeryczne powinno być dokładnie zgodne ze schematem programowania za pomocą przycisków S1 i S2. Poniższy program przedstawia wejście pojedynczego numeru za pomocą przycisku S1. Podobnie jak przy programowaniu, pierwsze naciśnięcie przycisku prowadzi do wyniku 0000. Każde kolejne naciśnięcie przycisku S1 zwiększa wyjście o 1. Pchnięcie S2 kończy wejście. W takim przypadku program zakończy się nieskończoną pętlą.

Adres	Komenda	Dane	Komentarz
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

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

Lista 25: Wprowadzanie liczby

Wejście numeryczne jest również dostępne jako gotowy podprogram od adresu 70 i następne. Zamiast niekończącej się pętli w linii 08 istnieje tutaj polecenie RET. Podprogram jest pozostawiony z wynikiem wprowadzenia liczby w A.

Następująca blokada numeru wywołuje trzykrotnie wprowadzoną liczbę i porównuje wyniki z predefiniowanymi liczbami. W tym przykładzie prawidłowe wejście to 3, 5, 2. Następnie wyjście PWM jest w pełni aktywowane z wartością 15. Każde błędne wejście prowadzi jednak do nieskończonej pętli, którą można pozostawić tylko po zresetowaniu.

Wyjście PWM jest traktowane w tym przykładzie jak normalny port cyfrowy. Jest to konieczne, ponieważ wszystkie cztery wyjścia od A1 do A4 są potrzebne do wprowadzania liczb. Po każdym pełnym wprowadzeniu, cztery diody LED są kasowane, aby dać obserwatorowi jak najmniej informacji o tajnej kombinacji.

Adres	Komenda	Dane	Komentarz
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	LEDs 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	LEDs 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	LEDs off
13	4	F	A = 15
14	5	9	PWM=A
15	3	0	End

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

Lista 26: Blokada numeru

21 Załącznik

Lista przykładowych programów

Adres	Komenda	Dane	Komentarz
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

64 51 4E 80 C3 98 82 95 4D 80 C3 9E 82 9A 4B 81

Strona 0: Wybór i uruchomienie przykładowych programów

Adres	Komenda	Dane	Komentarz
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

Adres	Komenda	Dane	Komentarz
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"

C3 94 83 90 47 81 C3 9A 83 94 43 82 C3 90 84 90

Strona 1: Wybór i uruchomienie przykładowych programów

Adres	Komenda	Dane	Komentarz
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	-

11 28 18 28 34 71 54 59 26 34 69 54 59 26 34 FF

Strona 2: Przykładowe programy: Naprzemienny błysk, odliczanie, AD / PWM

Adres	Komenda	Dane	Komentarz
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	-

54 CE 71 33 22 CC 32 40 22 71 54 CE 34 39 FF FF

Strona 3: Przykładowe programy: losowe, stoper S1

Adres	Komenda	Dane	Komentarz
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

Adres	Komenda	Dane	Komentarz
4C	F	F	-
4D	F	F	-
4E	F	F	-
4F	F	F	-

86 D0 40 71 54 23 CD 34 D8 40 54 3B FF FF FF FF

Strona 4: Przykładowy program zobacz start/stop

Adres	Komenda	Dane	Komentarz
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

4F 93 45 53 19 11 21 19 11 21 19 11 20 B4 10 E0

Strona 5: Podprogramowe wyjście dźwięku

Adres	Komenda	Dane	Komentarz
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	-

23 CE 32 23 CC 31 E0 FF 23 CF 32 23 CD 31 E0 FF

Strona 6: Podprogramy Czekaj S1 i czekaj 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
7B	7	1	A = A + 1

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

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

Strona 7: Wprowadzenie przycisku podprogramu

Tabela poleceń

	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	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	

Zużyte urządzenie

Nie należy wyrzucać razem z odpadami z gospodarstwa domowego baterii ani urządzenia!!!!

Produkt ten jest oznaczony zgodnie z wymaganiami Dyrektywy WEEE (2002/96 / WE). Załączona (pokazana) etykieta wskazuje, że ten elektryczny / elektroniczny produkt nie powinien być wyrzucony razem z odpadami gospodarstwa domowego.

Kategoria produktu: Produkt ten jest sklasyfikowany, jako urządzenie kategorii 9 ("przyrządy do nadzoru i kontroli") w odniesieniu od kategoryzacji urządzenia zawartego w załączniku I do dyrektywy WEEE.

Skonsultuj się z przedstawicielem handlowym lub odpowiedzialnym biurem sprzedaży, jeśli chcesz uzyskać więcej informacji odnośnie sposobów i możliwości utylizacji produktów. Dodatkowe informacje znajdują się na stronie internetowej producenta. Stare urządzenia nie są bezwartościowymi śmieciami. Poprzez recykling można odzyskać cenne surowce. Zapytaj w swoim mieście lub w jednostce administracyjnej miasta gdzie znajdują się miejsca przyjaznej dla środowiska utylizacji odpadów elektronicznych. Dlatego właśnie jak mówi Ustawa o odpadach (DzU nr/62/2001 poz. 628) - odpady sprzętu elektrycznego i elektronicznego powinny być traktowane przez wytwórców, jako odpady niebezpieczne i zgodnie z obowiązującym prawem zbierane w sposób selektywny oraz poddawane procesom odzysku lub unieszkodliwiania. Ze względów ekologicznych - albo, jak kto woli finansowych, bo za wyrzucanie elektroniki na śmietnik grozi dotkliwa kara - nie możemy tak po prostu wrzucić sprzętu do kubła.

Dyrektywa 2002/96/WE (WEEE):

Dyrektywa Unii Europejskiej, która wskazywała na konieczność prowadzenia spójnej dla wszystkich krajach UE polityki gospodarowania ZSEE w celu ochrony środowiska. Dyrektywa określa szereg nowych obowiązków, wśród których najważniejszą rolę odgrywają artykuły 8 i 9, na podstawie, których państwa członkowskie zobowiązane są do uchwalenia aktu prawnego nakładającego na producentów obowiązek finansowania kosztów zbierania, przetwarzania, odzysku, recyklingu i przyjaznego dla środowiska usuwania ZSEE pochodzącego z gospodarstw domowych, jak również z innych źródeł np. od instytucji i firm.