

Przełącznik sensorowy z procesorem AVR AT90S1200

Bohaterem artykułu jest premierowa konstrukcja opracowana w AVT, wykorzystująca i testująca możliwości sterownika z rodziny AVR. Przyświecały nam dwa cele: stworzenie układu posiadającego walory użytkowe i pokazanie na jego przykładzie jak można wykorzystywać i oprogramować nowe sterowniki sięgając po ogólnie dostępne narzędzia.

Procesory AVR produkowane przez firmę ATMEL są prostymi 8-bitowymi sterownikami jednocukrowymi. Cechy wyróżniające je to duża szybkość działania i niewielki pobór prądu, co pozwala im pracować nawet w układach zasilanych bateryjnie. Dodatkowymi atutami jest duża wydajność prądowa portów wyjściowych zdolnych sterować bezpośrednio diodami LED lub małymi przekaźnikami oraz możliwość ustawiania wyprowadzeń portów w stan wysokiej impedancji. Takie właściwości umożliwiają wykorzystanie procesora jako przełącznika

sensorowego, czyli układu przełączającego, działającego pod wpływem dotknięcia. Oczywiście, przełączniki takie znane są od dawna. Można je zbudować z ogólnie dostępnych elementów dyskretnych i bramek logicznych. Konstrukcja przełącznika z procesorem jest jednak niezwykle prosta. W dodatku łatwo można zwiększyć jego możliwości modyfikując jedynie program, a również o to w całości chodzi.



Schemat ideowy przełącznika pokazano na rys. 1. Jak widać, w swej zasadniczej części urządzenie składa się z procesora AVR i niewielu więcej elementów. Dwa styki sensorów dołączone są do sterownika poprzez pojemności C3 i C4. Świecenie diod D1 i D2 sygnalizuje dotknięcie któregoś z sensorów. Procesor do pracy (oczywiście oprócz zasilania +5V) potrzebuje jedynie ukła-

du wytwarzającego impulsy taktujące o częstotliwości stabilizowanej rezonatorem kwarcowym. W prototypie użyto kwarcu o częstotliwości ok. 4MHz, jednak modyfikując program można zastosować kwarc o częstotliwości z zakresu od 1 do 10MHz.

Działanie układu jest bardzo proste. Dotknięcie któregoś z sensorów powoduje zapalenie się odpowiadającej

mu diody LED. Jednocześnie następuje zmiana stanu portu PB. W cyklu ok. 0,6s na kolejnych wyjściach portu pojawia się aktywny stan wysoki. Kierunek przesuwania się stanu aktywnego zależy od tego, czy dotknięty zostanie sensor SW1 czy SW2. Poziom wysoki na wyprowadzeniach portu może przesuwać się pomiędzy dwoma położeniami skrajnymi: wprowadzeniem

PB.7 oraz przyjęciem przez wszystkie bity portu poziomu nieaktywnego, czyli niskiego. Wyprowadzenia portu poprzez złącze JP2 mogą bezpośrednio sterować LED-ami lub innymi układami pobierającymi do 20mA prądu.

Układ w prosty sposób można rozbudować. Dołączając dwa zewnętrzne rejestry tranzaskowe można uzyskać sterowany dotykowo przełącznik,

List. 1.

```

;*****
;* Przełącznik dotykowy na
;* procesorze AT90S1200*
;* zegar ok.4.0MHz
;*****
; dane wyjściowe podawane są na
; port PB
; diody sygnalizujące przesuw
; dołączone są do PD6 i|PD5
; wejście sensora przesuwu
; "w dół" dołączone jest
; do PD3 (SW1)
; wejście sensora przesuwu
; "w górę" dołączone jest
; do PD4 (SW2)
; wejście wyboru opcji PD2
; wyjście sygnału dla
; zatrasku sterującego
; wyświetlaczem PD0
; wyjście sygnału dla
; zatrasku danych
; równoległych PD1
.include "1200def.inc"
.DEF zeg_06s=r16
; rejestr stopera
.DEF port_out=r17 ;rejestr
; stanu portu równoległego
.DEF test=r18
;rejestr pomocniczy
.DEF trans=r19
;rejestr pomocniczy
.DEF dig_number=r20
;wskaźnik aktywnego wyjścia
.DEF opcja=r21
;rejestr aktywnej opcji
;obsługa wyświetlacza
.DEF sw1_test=r22
;znacznik aktywnego wejścia 1
.DEF sw2_test=r23
;znacznik aktywnego wejścia 2
.EQU clk_paral=1
;przypisanie nazwy do wyjścia
;sygnału zatrasku danych
;równoległych
.EQU clk_disp=0
;przypisanie nazwy do wyjścia
;sygnału zatrasku wyświetlacza
.EQU pin_opcja=2
;przypisanie nazwy do wejścia
;wyboru opcji
.EQU dig_table=5 ;adres
;tabeli danych sterujących
;segmentami wyświetlacza

ldi trans,0xff ;zapis
;wartości FFh do rejestru DDRB
out DDRB,trans ;programuje
;port B jako port wyjściowy
ldi trans,0x63 ;zapis do
;rejestru DDRD programuje port D
out DDRD,trans ;w części
; jako port wyjściowy w części
; jako wyjściowy
sbi PORTD,pin_opcja
ldi opcja,0xff
sbis PIND,pin_opcja
;testowanie wyprowadzenia,
;którego stan steruje

ldi opcja,0x0 ;włączeniem lub wyłączeniem
;obsługi wyświetlacza

ldi r30,dig_table ;zapis
;danych do tabeli sterującej
;segmentami wyświetlacza
ldi trans,0x3f ;wartość
;reprezentująca zapalone
;segmenty cyfry "0"
st Z,trans
inc r30
ldi trans,0x06 ;wartość
;reprezentująca zapalone
;segmenty cyfry "1"
st Z,trans
inc r30

;w tym miejscu należy dopisać
;instrukcje zapisujące do
;pamięci wygląd pozostałych
;cyfr 2 - 7

inc r30
ldi trans,0x7f ;wartość
;reprezentująca zapalone
;segmenty cyfry "8"
st Z,trans

ldi test,0x0
;ustawienie wartości
;początkowych rejestrów
ldi trans,0x0
out PORTD,trans
ldi port_out,0x01
ldi dig_number,0x01
rcall set_paralel
;po resecie aktywne będzie
;wyjście 1
rcall wyswietl_dig
;wyświetlenie cyfry
;określającej aktywne wyjście
ldi zeg_06s,110 ;wartość
;początkowa rejestru stopera
ldi trans,0x3 ;preskaler /64
out TCCR0,trans
ldi trans,0x0 ;wartość
;początkowa rejestru licznika
;sprzętowego
out TCNT0,trans

pl:
;początek pętli głównej
;programu
in trans,TIFR
sbrc trans,TOV0 ;badanie
;stanu flagi TOV0 licznika,
;jeśli ustawiona
rjmp zegar
;skok do podprogramu "zegar"
in trans,0x0
sbic PIND,3
;badanie stanu SW1
ori sw1_test,0x40 ;stan
;wysoki, wejście jest
;pobudzone poprzez dotyk
sbic PIND,4

;badanie stanu SW2
ori sw2_test,0x20 ;stan
;wysoki, wejście jest
;pobudzone poprzez dotyk
rcall signal
;zapalenie lub zgaszenie diod
;sygnalizacyjnych D1,D2
rjmp pl

;podprogram wyświetlania cyfry
;na wyświetlaczu
;siedmiosegmentowym
wyswietl_dig:
cpi opcja,0x0
breq w_d1 ;obsługa
;wyświetlacza jest wyłączona
ldi r30,dig_table
add r30,dig_number
ld trans,Z
out PORTB,trans
nop
nop
cbi PORTD,clk_disp
;generacja impulsu dla
;rejestru
sbi PORTD,clk_disp
;zatraskującego
cbi PORTD,clk_disp
nop
nop
w_d1:
ret
;podprogram ustawiania
;aktywnego wyjścia
;równoległego
set_paralel:
out PORTB,port_out
nop
nop
;przesuwanie bitu portu
;wyjściowego gdy aktywne
;jest SW1
cpi port_out,0x80 ;bit portu
;nie może już być przesuwany
breq z_1 ;stan wyjścia
;nie ulegnie zmianie
cpi port_out,0x0
nop
brne z_31
ldi port_out,0x1
;aktywne wyjście numer 1
ldi dig_number,0x1
rjmp z_1
z_31:
lsl port_out
;przesuwanie bitu wyjścia
inc dig_number
rjmp z_1
z_2:
;przesuwanie bitu portu
;wyjściowego gdy aktywne
;jest SW2
cpi port_out,0x0 ;bit portu
;nie może już być przesuwany
breq z_1 ;stan
;wyjścia nie ulegnie zmianie
lsr port_out
dec dig_number
z_1:
rcall set_paralel ;ustawienie
;aktywnego wyjścia

;*****
;badanie stanu SW2
ori sw2_test,0x20 ;stan
;wysoki, wejście jest
;pobudzone poprzez dotyk
rcall signal
;zapalenie lub zgaszenie diod
;sygnalizacyjnych D1,D2
rjmp pl

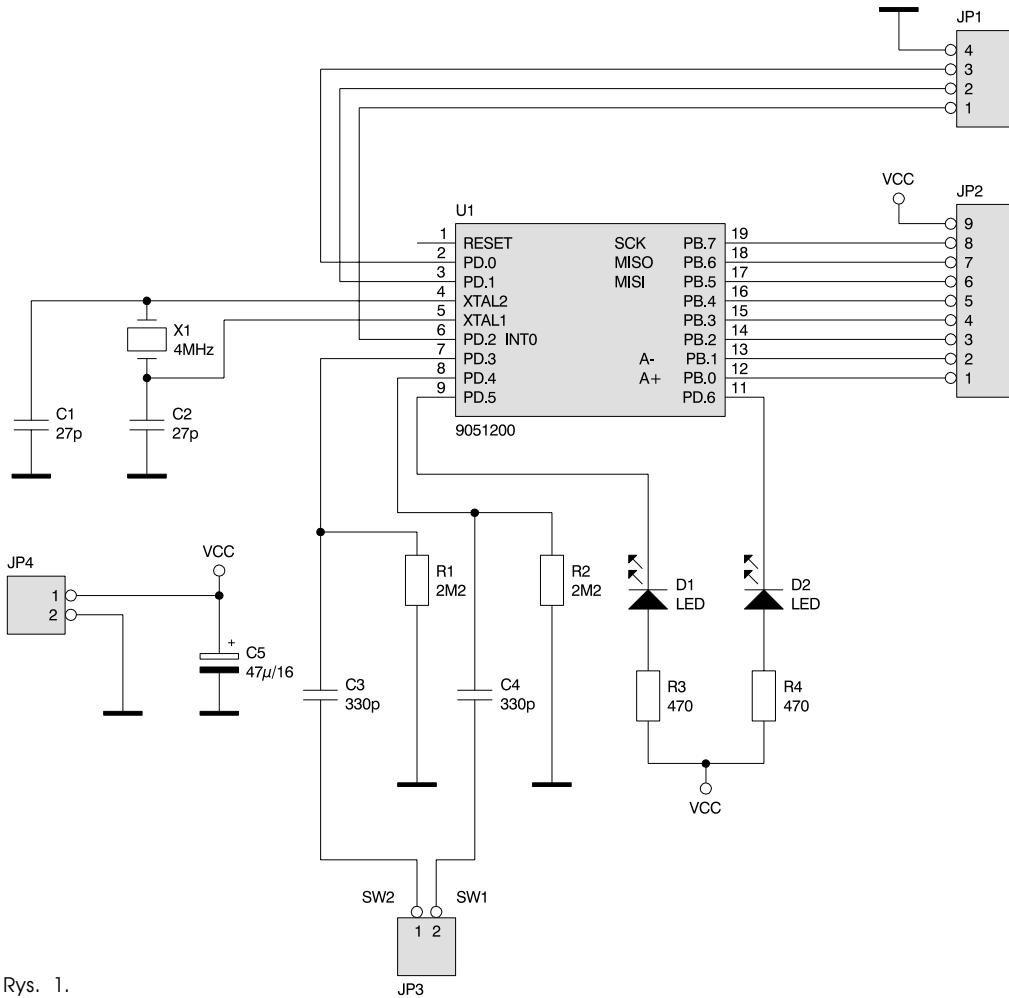
;koniec pętli głównej programu

zegar:
;podprogram zegara-stopera
ldi trans,0x2
out TIFR,trans
;wyzerowanie flagi TOV0
dec zeg_06s
tst zeg_06s
brne pl ;stoper jeszcze nie
;odmierzył 0,6s

ldi zeg_06s,110 ;odliczone
;zostało ok. 0,6s
or test,sw1_test
or test,sw2_test
cpi test,0x60
;jeżeli aktywne oba
;wejścia SW1 i SW2
breq z_1 ;stan portu PB
;nie ulegnie zmianie
cpi test,0x20
breq z_2 ;aktywne SW2
cpi test,0x40
breq z_3 ;aktywne SW1
rjmp z_1
;żadne z wejść nie jest aktywne
z_3:
;przesuwanie bitu portu
;wyjściowego gdy aktywne
;jest SW1
cpi port_out,0x80 ;bit portu
;nie może już być przesuwany
breq z_1 ;stan wyjścia
;nie ulegnie zmianie
cpi port_out,0x0
nop
brne z_31
ldi port_out,0x1
;aktywne wyjście numer 1
ldi dig_number,0x1
rjmp z_1
z_31:
lsl port_out
;przesuwanie bitu wyjścia
inc dig_number
rjmp z_1
z_2:
;przesuwanie bitu portu
;wyjściowego gdy aktywne
;jest SW2
cpi port_out,0x0 ;bit portu
;nie może już być przesuwany
breq z_1 ;stan
;wyjścia nie ulegnie zmianie
lsr port_out
dec dig_number
z_1:
rcall set_paralel ;ustawienie
;aktywnego wyjścia
;*****
rcall wyswietl_dig
;wyświetlenie cyfry na
;wyświetlaczu
ldi sw1_test,0x0
ldi sw2_test,0x0
rjmp pl

;podprogram wyświetlania cyfry
;na wyświetlaczu
;siedmiosegmentowym
wyswietl_dig:
cpi opcja,0x0
breq w_d1 ;obsługa
;wyświetlacza jest wyłączona
ldi r30,dig_table
add r30,dig_number
ld trans,Z
out PORTB,trans
nop
nop
cbi PORTD,clk_disp
;generacja impulsu dla
;rejestru
sbi PORTD,clk_disp
;zatraskującego
cbi PORTD,clk_disp
nop
nop
w_d1:
ret
;podprogram ustawiania
;aktywnego wyjścia
;równoległego
set_paralel:
out PORTB,port_out
nop
nop
;przesuwanie bitu portu
;wyjściowego gdy aktywne
;jest SW1
cpi port_out,0x80 ;bit portu
;nie może już być przesuwany
breq z_1 ;stan wyjścia
;nie ulegnie zmianie
cpi port_out,0x0
nop
brne z_31
ldi port_out,0x1
;aktywne wyjście numer 1
ldi dig_number,0x1
rjmp z_1
z_31:
lsl port_out
;przesuwanie bitu wyjścia
inc dig_number
rjmp z_1
z_2:
;przesuwanie bitu portu
;wyjściowego gdy aktywne
;jest SW2
cpi port_out,0x0 ;bit portu
;nie może już być przesuwany
breq z_1 ;stan
;wyjścia nie ulegnie zmianie
lsr port_out
dec dig_number
z_1:
rcall set_paralel ;ustawienie
;aktywnego wyjścia
;*****

```



WYKAZ ELEMENTÓW

Rezystory

R1, R2: 2,2MΩ
R3, R4: 470Ω

Kondensatory

C1, C2: 27p
C3, C4: 330p
C5: 47µF/16

Półprzewodniki

U1: AT90S1200 (procesor zaprogramowany)
D1, D2: diody LED (np. żółta i zielona)

Różne

X1: kwarc ok. 4MHz

Zamiast sensorów można także użyć zwykłych przycisków zamontowanych w miejsce kondensatorów C3 i C4. W takim przypadku wprowadzenia złącza JP3 należy wtedy zewrzeć i połączyć z napięciem zasilania procesora, natomiast oporniki R1 i R2 można zamienić na inne o oporności ok. 10kΩ.

Po dotarciu do tego miejsca opisu, Czytelnicy zainteresowani jedynie wykonaniem działającego przełącznika mogą sobie darować lekturę dalszej części tekstu. Posiadają już wiedzę niezbędną do jego uruchomienia i użycia. Natomiast wszystkich zainteresowanych sterownikami AVR, możliwościami ich wykorzystania oraz samodzielnym pisaniem programów dla tych procesorów zapraszam do zapoznania się z list. 1. Szczegółowy opis programu wraz z kodem źródłowym dostępny jest na stronie <http://www.ep.com.pl/ftp/other.html>.

Ryszard Szymaniak, AVT
ryszard.szymaniak@ep.com.pl

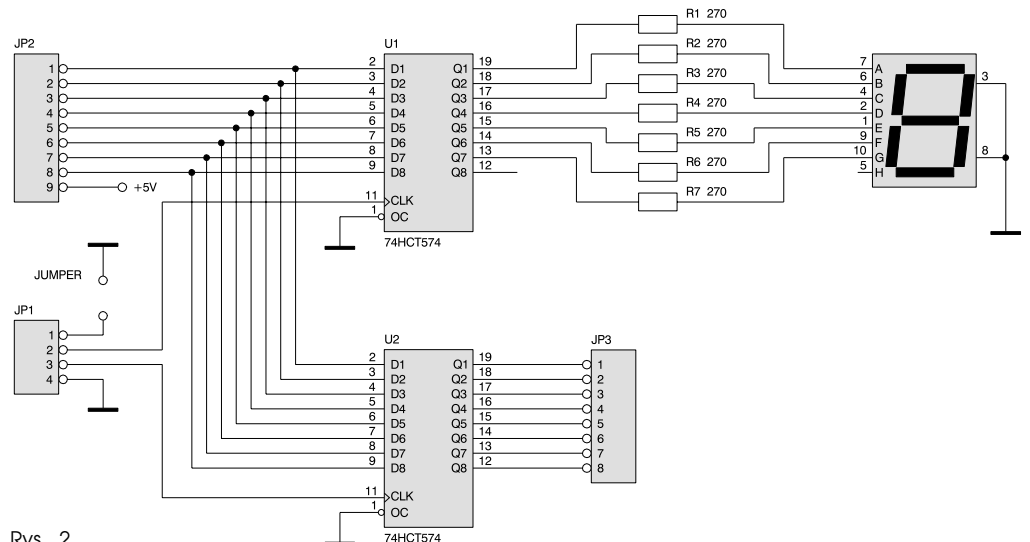
Rys. 1.

którego aktywny stan włączy 7-segmentowy wyświetlacz LED. Schemat układu poszerzającego możliwości przełącznika pokazuje rys. 2. Jako zatrząsków można użyć zarówno kostek 74HCT574 (wygodniejsze podczas projektowania druku) jak i 74LS374 (użyte w modelu) czy jakichkolwiek innych 8-bitowych rejestrów zatrząskowych typu D. Jako wyświetlacza można użyć dowolnego wyświetlacza LED o wspólnej katodzie.

Sygnaly zatrząskujące dane w dodatkowych rejestrach podawane są złączem JP1. Wyjaśnić należy obecność na rysunku 2 jumpera. Otóż zworka ta steruje wyborem trybu pracy przełącznika. Jeżeli w momencie włączenia zasilania pozostanie zwarta do masy, sterownik nie będzie obsługiwał wyświetlacza i sterowane zewnętrzne układy można dołączyć wprost do portu PB. Pozostawienie zwory rozwartej spowoduje multipleksowanie danych na tym porcie i konieczność zastosowania zewnętrznych rejestrów zatrząskowych.

Montaż przełącznika jest banalnie prosty ze względu na niewielką liczbę elementów wchodzących w jego skład. Do złącza JP3 dołącza się sensory o dowolnym kształcie, wykonane z kawałka drutu lub blaszki. Nie powinny być jednak zbyt długie i oddalone od płytki. Układ jest podatny na zakłócenia wywoływane działa-

niem dużych odbiorników energii elektrycznej, takich jak lodówki, silniki itp. Podatność układu na zakłócenia można stłumić zmniejszając wartość rezystancji oporników R1 i R2 (rys. 1) jednak jej zbyt obniżenie może spowodować, że układ przestanie reagować na dotykaniu wyprowadzeń senso-



Rys. 2.