

Obsługa wyświetlaczy graficznych w Bascom, część 2

W tym odcinku kontynuujemy przedstawianie przykładów obsługi wyświetlaczy graficznych o rozdzielczości 128*64 pikseli, z kontrolerami typu KSx i T6963C. Obsługa wyświetlacza z kontrolerem SED jest identyczna jak wyświetlacza z kontrolerem KSx, więc i z tego typu wyświetlaczami nie powinno być problemów.



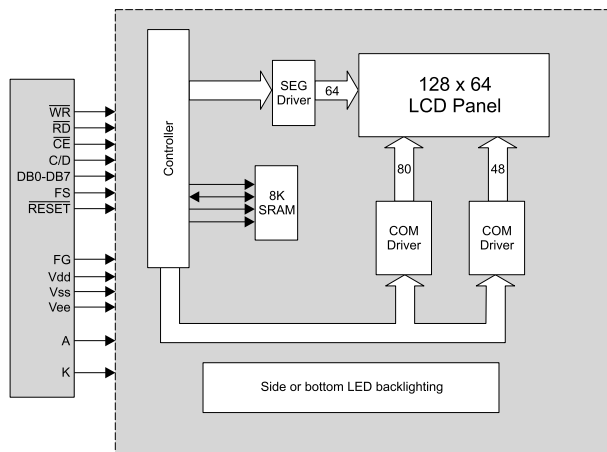
Wyświetlacze z kontrolerem T6963C

Dość popularne są wyświetlacze graficzne z kontrolerem T6963C. W ramach przykładu wykorzystano wyświetlacz graficzny JM12864b z kontrolerem T6963C obsługującym matryce LCD o rozdzielczości 128*64 pikseli i z podświetleniem LED. Wyświetlacz z tego typu kontrolerem ma magistralę 8-bitową i co jest bardzo pomocne, generator znaków 6x8 lub 8x8. Na rys. 6 przedstawiono schemat blokowy wyświetlacza JM12864b. W tym przykładzie zostanie przedstawiony przykład realizacji analogowego zegara z wykorzystaniem LCD JM12864b.

Na rys. 7 przedstawiono schemat dołączenia wyświetlacza graficznego JM12864b do mikrokontrolera. Linie D0...D7 wyświetlacza są liniami danych, natomiast linie /WR, /RD, /CE, C/D, /RESET, FS są liniami sterującymi. Wykorzystywany wyświetlacz ma podświetlenie, którego prąd jest ograniczany przez rezystor R1. Do prawidłowego działania wyświetlacza, także w tym przypadku jest potrzebne ujemne napięcie na wejściu obwodu regulacji kontrastu (V_{eg}) bliskie -10 V. Tak jak w poprzednim przykładzie jest ono wytwarzane przez przetwornicę pojemnościową zawartą w układzie MAX232. Przyciski S1 oraz S2 służą do ustawiania zegara. Rezonator kwarcowy X1 o częstotli-

wości 32768 Hz do taktuje licznik Timer2, w oparciu o który działa zrealizowany programowo zegar czasu rzeczywistego (RTC). Sam mikrokontroler jest taktowany wewnętrznym oscylatorem o częstotliwości 8 MHz. Na list. 2 przedstawiono program realizujący analogowy zegar przy użyciu wyświetlacza graficznego z kontrolerem T6963C. Instrukcje typowe dla wyświetlaczy graficznych z kontrolerem T6963C zostaną przedstawione na przykładzie programu zegara. Aby mogły być obsługiwane wyświetlacze z TC6963C, należy załączyć do programu bibliotekę *gclid.lib*. W programie stałe *Srodekx* i *Srodeky* określają środek tarczy zegara. Do konfiguracji tego typu wyświetlacza służy także instrukcja *Config Graphlcd*, której pierwszy parametr, będący rozdzielczością, powinien być bez dodatkowych znaków SED. Dodatkowo w tej instrukcji, prócz linii mikrokontrolera, do których został dołączony wyświetlacz, należy podać wartość parametru *MODE*. Parametr *MODE* określa liczbę kolumn dla wyświetlanego tekstu. W przykładzie podano wartość 8, czyli uzyska się 128/8=16 kolumn.

Jedna linia tekstu będzie zatem mieć 16 znaków. Instrukcja *Config Clock* konfiguruje zegar czasu rzeczywistego (RTC), natomiast instrukcja *Config Date* konfiguruje format daty. Procedura *Rys_zeg* rysuje tarcze analogowego zegara, procedury *Sekundy*, *Minuty*, *Godziny* obliczają pozycje wskazówek (oraz je wyświetlają) odpowiednio sekund, minut oraz godzin. Procedura *Obl_f_tr* oblicza wartość funkcji trygonometrycznych sin oraz cos w oparciu, o które są wyliczane pozycje wskazówek. Procedura *Wysw_dat* wyświetla na wyświetlaczu datę w sposób cyfrowy. Do zmiennych *Time\$* oraz *Date\$* są zapisane domyślny czas oraz data. W dalszej części programu zostały odblokowane przerwania, co umož-



Rys. 6. Schemat blokowy wyświetlacza ze sterownikiem T6963C

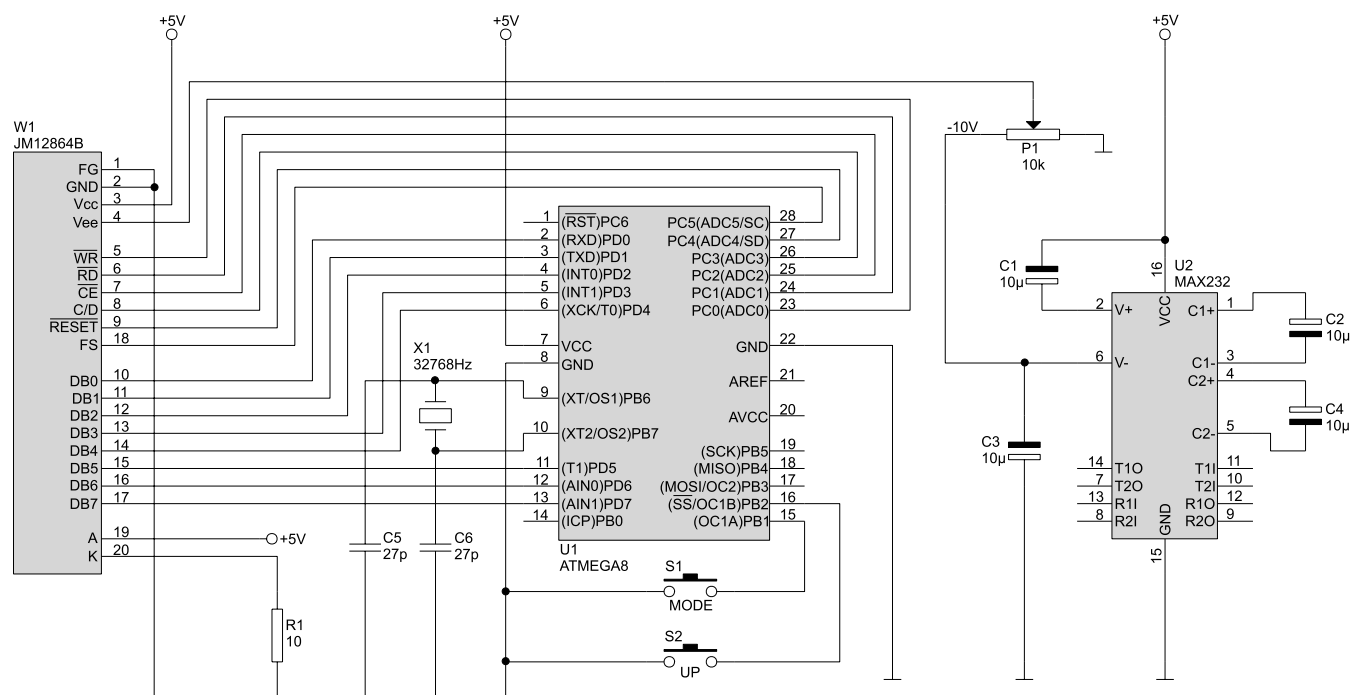
```
List. 2.
'Przykład realizacji analogowego zegara z datownikiem
'Wykorzystany został wyświetlacz graficzny JM12864B (128*64) z
kontrolerem TC6963C
'Podczas włączania zegara przyciśnięcie przycisku S1 umożliwia
wejście do trybu ustawiania zegara w którym przycisk S1 umożliwia
wybor zmienianej wartosci, a S2 Zwiększanie Wybranej Wartosci.
'Do obliczania pozycji wskazówek wykorzystano funkcje trygonome-
tryczne sin oraz cos
'Należy ustawić Fusebity by mikrokontroler był taktowany wewnętr-
znym oscylatorem o częstotliwości 8 MHz.
'Marcin Wiazania
'marcin.wiazania@ep.com.pl

$lib „glcd.lib”
laczenie biblioteki obsługi wyświetlacza graficznego z kontrolerem
T6963C
$regfile = „M8def.dat”
cza definicje rejestrów mikrokontrolera
$crystal = 8000000
Const Srodekx = 64
sla x srodka okregu
Const Srodeky = 32
sla y srodka okregu
Config Pinb.1 = Input
PB1 jako wejsciowa
Config Pinb.2 = Input
PB2 jako wejsciowa
Config Graphlcd = 128 * 64 , Dataport = Portd , Controlport = Portc
, Ce = 2 , Cd = 3 , Wr = 0 , Rd = 1 , Reset = 4 , Fs = 5 , Mode =
8
'konfiguracja graficznego LCD
Config Clock = Soft
'konfi-
guracja zegara czasu rzeczywistego
Config Date = Dmy , Separator = -
'konfi-
guracja formatu daty
Declare Sub Rys zeg
'pro-
cedura rysowania zegara
Declare Sub Sekundy
'pro-
cedura obsługi sekund
Declare Sub Minuty
'pro-
cedura obsługi minut
Declare Sub Godziny
'pro-
cedura obsługi godzin
Declare Sub Obl_f tr(pom As Single)
'pro-
cedura zamiany stopni na radiany i obliczania funkcji trygonome-
trycznych
Declare Sub Wysw dat
'pro-
cedura wyświetlenia daty
Dim Poczx As Single
'zmien-
na wskazujaca początkowa wartosc x
Dim Pocyx As Single
'zmien-
na wskazujaca początkowa wartosc y
Dim Kony As Single
'zmien-
na wskazujaca koncowa wartosc y
Dim Konx As Single
'zmien-
na wskazujaca koncowa wartosc x
Dim Sek As Byte
'zmienna sekund
Dim Minut As Byte
'zmienna minut
Dim Godz As Byte
'zmienna godzin
Dim Sek y As Single
'okre-
sla pozycje y wskazowki sekund
Dim Sek x As Single
'okre-
sla pozycje x wskazowki sekund
Dim Kat_dod As Single
'prze-
chowuje wartosc dodatkowego kata
Dim Kat_cos As Single
'obli-
czony kat cos
Dim Kat_sin As Single
'obli-
czony kat sin
Dim Kat_h As Single
'prze-
chowuje wartosc dodatkowego kata dla godzin
Dim K_r As Single
'zmien-
na pomocnicza przechowujaca kat
Dim Pom As Single
'zmien-
na pomocnicza
Dim X As Byte
'zmienna wskazujaca wartosc pozycji x
Dim Y As Byte
'zmienna wskazujaca wartosc pozycji y
Dim Kx As Byte
'zmienna przechowujaca dodatkowa pozycje x
Dim Ky As Byte
'zmienna przechowujaca dodatkowa pozycje y
Dim J As Integer
'zmien-
na licznikowa
Dim I As Byte
'zmienna licznikowa
Dim K As Byte
'zmienna licznikowa
S1 Alias Pinb.1
'przy-
pisanie do linii PB1 aliasu S1
S2 Alias Pinb.2
'przy-
pisanie do linii PB2 aliasu S2
Set Portb.1
'wla-
czenie rezystora podciagajacego linii PB1
Set Portb.2
'wla-
czenie rezystora podciagajacego linii PB2
Time$ = „00:00:00”
'po-
czatkowy czas
Date$ = „01/01/01”
'po-
czatkowa data
Enable Interrupts
'od-
blokowanie globalnego systemu przerw
Cursor Off , Noblink
'wyla-
czenie kursora i jego migotania
Cls
'czy-
sci LCD
If S1 = 0 Then
'jesli
S1 przycisnieto to
Bitwait S1 , Set
'cze-
kaj az zostanie puszczone S1 - podprogram ustawiania zegara
K = 0
'wyze-
rowanie zmiennej K
```

```
Locate 1 , 7
'okre-
slenie pozycji kursora
Lcd „Time”
'wy-
swietlenie tekstu
Locate 5 , 7
'okre-
slenie pozycji kursora
Lcd „Date”
'wy-
swietlenie tekstu
Do
'po-
czatek petli do-loop
Sek = _sec
'przy-
pisanie zmiennej sec wartosci sekund
Locate 7 , 5
'okre-
slenie pozycji kursora
Lcd Date$
'wy-
swietlenie daty
Locate 3 , 5
'okre-
slenie pozycji kursora
Lcd Time$
'wy-
swietlenie czasu
Debounce S1 , 0 , Obs_s1 , Sub
'jesli
S1 nacinety ponownie to skok do podprogramu Obs_s1
If S2 = 0 Then
'jesli
S2 przycisnieto to
Select Case K
'jesli
k rowne
'zero
to
Incr _hour
'zwiększenie o jeden wartosci godzin
If _hour = 24 Then _hour = 0
'jesli
wartosci godzin rowna 24 to zerowanie wartosci godzin
Case 1:
'jeden
to
Incr _min
'zwiększenie o jeden wartosci minut
If _min = 60 Then _min = 0
'jesli
wartosci minut rowna 60 to zerowanie wartosci minut
Case 2:
'dwa
to
_sec = 0
'zero-
wanie wartosci sekund
Case 3:
'trzy
to
Incr _day
'zwiększenie o jeden wartosci dni
If _day = 32 Then _day = 1
'jesli
wartosci dni rowna 32 to ustawienie wartosci dni na 1
Case 4:
'cztery
to
Incr _month
'zwiększenie o jeden wartosci miesiecy
If _month = 13 Then _month = 1
'jesli
wartosci miesiecy rowna 13 to ustawienie wartosci miesiecy na 1
Case 5:
'piec
to
Incr _year
'zwiększenie o jeden wartosci lat
If _year = 20 Then _year = 0
'jesli
wartosci lat rowna 20 to wyzerowanie wartosci lat
End Select
'ko-
niec instrukcji wyboru
Waitms 200
'cze-
kaj 200 ms
End If
'petla
Loop Until K = 6
wykonywana az K=6
End If
'czy-
sci LCD
Call Rys zeg
'wywo-
lanie podprogramu rysowania analogowego zegara
Locate 8 , 12
'okre-
slenie pozycji kursora
Lcd „EP-MW”
'wy-
swietlenie tekstu
Locate 1 , 1
'okre-
slenie pozycji kursora
Lcd „Dzien”
'wy-
swietlenie tekstu
Locate 7 , 1
'okre-
slenie pozycji kursora
Lcd „Mies”
'wy-
swietlenie tekstu
Locate 1 , 14
'okre-
slenie pozycji kursora
Lcd „Rok”
'wy-
swietlenie tekstu
Call Wysw dat
'wywo-
lanie procedury wyświetlenia daty
Do
'po-
czatek nieskonczonej petli do-loop - program glowny
If Sek <> sec Then
'jesli
sec rozne od _sec to
Sek = _sec
'zapi-
sanie do sec wartosci _sec
Minut = _min
'zapi-
sanie do minut wartosci _min
Godz = _hour
'zapi-
sanie do godz wartosci _hour
Call Sekundy
'wywo-
lanie procedury wyświetlajacej sekundy
Call Minuty
'wywo-
lanie procedury wyświetlajacej minuty
Call Godziny
'wywo-
lanie procedury wyświetlajacej godziny
If _hour = 0 Then Call Wysw dat
'jesli
_hour (godziny) rowne 0 to wywołanie procedury wyświetlajacej date
End If
Loop
'ko-
niec programu
Obs_s1:
'pod-
program obsługi przycisku S1
Incr K
'zwiększenie o jeden zmiennej k
Return
'po-
```

```
wrot z podprogramu
Sub Wysw_dat
cedury wyswietlenia daty
  Locate 2 , 3
  slenie pozycji kursora
  I = Makebcd( day)
  miana wartosci dni na kod BCD
  Lcd Bcd(i)
  swietlenie zamienionej wartosci BCD na tekst dni
  Locate 8 , 3
  slenie pozycji kursora
  I = Makebcd( month)
  miana wartosci miesiecy na kod BCD
  Lcd Bcd(i)
  swietlenie zamienionej wartosci BCD na tekst miesiecy
  Locate 2 , 15
  slenie pozycji kursora
  I = Makebcd( year)
  miana wartosci lat na kod BCD
  Lcd Bcd(i)
  swietlenie zamienionej wartosci BCD na tekst lat
End Sub
niec procedury
Sub Obl_f_tr(pom As Single)
cedura Zamienia kat na radiany i oblicza wartosc funkcji sin i cos
  K_r = 90 - Pom
  czenie wartosci kata
  K_r = Deg2rad(k_r)
  miana wartosci kata na radiany
  Kat_sin = Sin(k_r)
  czenie wartosci funkcji sin
  Kat_cos = Cos(k_r)
  czenie wartosci funkcji cos
End Sub
niec procedury
Sub Sekundy
cedura obslugujaca sekundy
  For I = 0 To 1
  FOR wykonywana dwa razy
  If I = 0 Then
  I=0 to
    Pom = Sek - 1
  sanie do pom wartosci o jedna sekunde mniej
  J = 0
  rowanie J
  Else
  przeciwnym razie
  Pom = Sek
  sanie do pom wartosci sekund
  J = 255
  sanie do J wartosci 255
  End If
  Pom = Pom * 6
  mnozenie wartosci pom przez 6
  Call Obl_f_tr(pom)
  lanie procedury obliczajacej funkcje sin i cos
  Pom = 27 * Kat sin
  liczenie wartosci funkcji sin (obliczana dlugosc wskazowki - y)
  Sek_y = Srodeky - Pom
  czenie wspolrzednej y wskazowki
  Pom = 27 * Kat cos
  liczenie wartosci funkcji cos (obliczana dlugosc wskazowki - x)
  Sek_x = Srodekk + Pom
  czenie wspolrzednej x wskazowki
  Sek_x = Round(sek_x)
  okraglenie pozycji x wskazowki
  Sek_y = Round(sek_y)
  okraglenie pozycji y wskazowki
  X = Int(sek_x)
  sanie w x wartosci calkowitej pozycji wskazowki x
  Y = Int(sek_y)
  sanie w x wartosci calkowitej pozycji wskazowki y
  Line(srodekk , Srodeky) -(x , Y) , J
  wanie wskazowki sekund
  Circle(srodekk , Srodeky) , 0 , 255
  swierzenie srodkowego punku zegara
  Circle(srodekk , Srodeky) , 1 , 255
  swierzenie srodkowego punku zegara
  Next I
  'zwiekszenie o jeden wartosci I
End Sub
niec procedury
Sub Minuty
cedura obslugujaca minuty
  For I = 0 To 1
  FOR wykonywana dwa razy
  If I = 0 Then
  I=0 to
    Pom = Minut - 1
  sanie do pom wartosci o jedna minucie mniej
  J = 0
  rowanie J
  Else
  przeciwnym razie
  Pom = Minut
  sanie do pom wartosci minut
  J = 255
  sanie do J wartosci 255
  End If
  Pom = Pom * 6
  mnozenie wartosci pom przez 6
  Call Obl_f_tr(pom)
  lanie procedury obliczajacej funkcje sin i cos
  Pom = 21 * Kat sin
  liczenie wartosci funkcji sin (obliczana dlugosc wskazowki - y)
  Sek_y = Srodeky - Pom
  czenie wspolrzednej y wskazowki
  Pom = 21 * Kat cos
  liczenie wartosci funkcji cos (obliczana dlugosc wskazowki - x)
  Sek_x = Srodekk + Pom
  czenie wspolrzednej x wskazowki
  Sek_x = Round(sek_x)
  okraglenie pozycji x wskazowki
  Sek_y = Round(sek_y)
  okraglenie pozycji y wskazowki
  X = Int(sek_x)
  sanie w x wartosci calkowitej pozycji wskazowki x
```

```
Y = Int(sek_y)
  sanie w x wartosci calkowitej pozycji wskazowki y
  Line(srodekk , Srodeky) -(x , Y) , J
  wanie wskazowki minut
  Circle(srodekk , Srodeky) , 0 , 255
  swierzenie srodkowego punku zegara
  Circle(srodekk , Srodeky) , 1 , 255
  swierzenie srodkowego punku zegara
  Next I
  'zwiekszenie o jeden wartosci I
End Sub
niec procedury
Sub Godziny
cedura obslugujaca godziny
  For I = 0 To 1
  FOR wykonywana dwa razy
  If I = 0 Then
  I=0 to
    J = 0
  rowanie J
  Else
  przeciwnym razie
  J = 255
  sanie do J wartosci 255
  Kat h = Godz * 30
  sanie wartosci godzin pomnozonych o 30
  Kat dod = Minut
  sanie do Kat dod wartosci minut
  Kat dod = Kat dod * 0.5
  mnozenie wartosci w_kat_dod przez wartosc 0.5
  End If
  Pom = Kat h + Kat dod
  sanie do pom sumy kat_h i kat_dod
  Call Obl_f_tr(pom)
  lanie procedury obliczajacej funkcje sin i cos
  Pom = 15 * Kat sin
  liczenie wartosci funkcji sin (obliczana dlugosc wskazowki - y)
  Sek_y = Srodeky - Pom
  czenie wspolrzednej y wskazowki
  Pom = 15 * Kat cos
  liczenie wartosci funkcji cos (obliczana dlugosc wskazowki - x)
  Sek_x = Srodekk + Pom
  czenie wspolrzednej x wskazowki
  Sek_x = Round(sek_x)
  okraglenie pozycji x wskazowki
  Sek_y = Round(sek_y)
  okraglenie pozycji y wskazowki
  X = Int(sek_x)
  sanie w x wartosci calkowitej pozycji wskazowki x
  Y = Int(sek_y)
  sanie w x wartosci calkowitej pozycji wskazowki y
  Line(srodekk , Srodeky) -(x , Y) , J
  wanie wskazowki godzin
  Circle(srodekk , Srodeky) , 0 , 255
  swierzenie srodkowego punku zegara
  Circle(srodekk , Srodeky) , 1 , 255
  swierzenie srodkowego punku zegara
  Next I
  'zwiekszenie o jeden wartosci I
End Sub
niec procedury
Sub Rys_zeg
cedura rysujaca zegar
  Circle(srodekk , Srodeky) , 31 , 255
  wanie zewnetrznej tarczy zegara
  Circle(srodekk , Srodeky) , 0 , 255
  wanie srodkowego punku zegara
  Circle(srodekk , Srodeky) , 1 , 255
  wanie srodkowego punku zegara
  For J = 0 To 360 Step 30
  wykonywana od j=0 do 360 z krokiem 30
  Pom = J
  sanie do pom wartosci j
  Call Obl_f_tr(pom)
  lanie procedury obliczajacej funkcje sin i cos
  Pom = 31 * Kat sin
  czenia pozycji poczatkowej y podzialki
  Poczy = Srodeky - Pom
  sze obliczenie punktu poczatkowego y
  Pom = 31 * Kat cos
  czenia pozycji poczatkowej x podzialki
  Poczx = Srodekk + Pom
  sze obliczenie punktu poczatkowego x
  Pom = 28 * Kat sin
  czenia pozycji koncowej y podzialki
  Kony = Srodeky - Pom
  sze obliczenie punktu koncowego y
  Pom = 28 * Kat cos
  czenia pozycji koncowej x podzialki
  Konx = Srodekk + Pom
  sze obliczenie punktu koncowego x
  Poczx = Round(poczx)
  okraglenie pozycji poczatkowej x podzialki
  Poczy = Round(poczy)
  okraglenie pozycji poczatkowej y podzialki
  X = Int(poczx)
  sanie w x wartosci calkowitej pozycji poczatkowej podzialki x
  Y = Int(poczy)
  sanie w x wartosci calkowitej pozycji poczatkowej podzialki y
  Konx = Round(konx)
  okraglenie pozycji koncowej x podzialki
  Kony = Round(kony)
  okraglenie pozycji koncowej y podzialki
  Kx = Int(konx)
  sanie w Kx wartosci calkowitej pozycji koncowej podzialki x
  Ky = Int(kony)
  sanie w Ky wartosci calkowitej pozycji koncowej podzialki y
  Line(x , Y) -(kx , Ky) , 255
  wanie podzialki zegara
  Next J
  'zwiekszenie o jeden wartosci J
End Sub
niec procedury
```

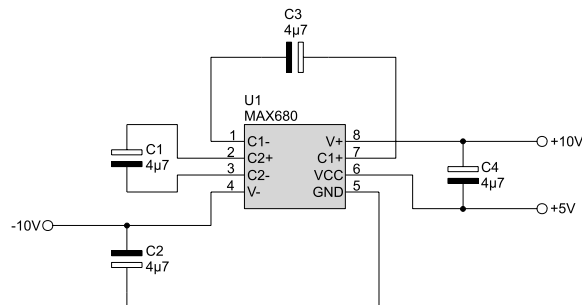


Rys. 7. Schemat elektryczny ilustrujący sposób dołączenia wyświetlacza JM12864b do mikrokontrolera

liwia działanie zegara RTC. Kolejna instrukcja *Cursor* z dwoma parametrami odnosi się do wyświetlacza LCD. Wyświetlacze z kontrolerem T6963C mają możliwość włączenia lub wyłączenia kursora, co jest związane z wbudowanym generatorem znaków. Instrukcją *Cursor* można skonfigurować kursor. Pierwszy parametr określa czy ma być włączony, natomiast drugi określa czy ma migać. W przykładzie wyłączono kursor oraz możliwość jego migotania. Kolejna instrukcja *Cls* tak, jak w pierwszym przykładzie, służy do czyszczenia ekranu wyświetlacza. W przypadku wyświetlaczy ze sterownikiem T6963C, może ona mieć dodatkowy parametr. Instrukcja *Cls* z parametrem *Text* czyści tylko dane tekstowe, a z parametrem *Graph* – dane graficzne. W przypadku instrukcji *Cls* bez parametru czyszczony jest zarówno tekst, jak i grafika. Instrukcje w warunkach *If S1=0...* realizują funkcję ustawiania czasu oraz daty. Przyciskiem S2 wybiera się ustawianą wartość czasu lub daty, natomiast przyciskiem S1 zwiększa się wybraną wartość. Przycisk S1 obsługiwany jest z wykorzystaniem instrukcji *Debounce* i związanego z nią podprogramu. W tym warunku zawarte zostały dwie instrukcje dotyczące LCD. Instrukcja *Locate* ustawia kursor w wybranym położeniu ekranu wyświetlacza. Jej parametrami są odpowiednio wiersz oraz kolumna.

Instrukcja *Lcd* umożliwia wyświetlenia na wyświetlaczu, w wybranym instrukcją *Locate* miejscu, tekstu lub wartości zmiennej. Przy programowaniu czasu oraz daty, czas i data wyświetlane są w formie cyfrowej. Aby wejść do trybu ustawiania zegara, należy podczas jego włączania przycisnąć przycisk S1. Po ustawieniu czasu oraz daty, LCD jest czyszczony oraz wywołana zostaje procedura *Rys zeg*, w której rysowana jest tarcza zegara. Okrąg tarczy rysowany jest instrukcją *Circle*, która ma taką samą składnię jak dla wyświetlaczy z kontrolerem KSx, z tym, że w tym przypadku kolor o wartości 1...255 oznacza zapalony piksel, a 0 zgaszony. Rysowane są 3 okręgi, z których jeden reprezentuje zewnętrzną tarczę zegara, a dwa jego punkty środkowe. W pętli *For J=0...* są obliczane (oraz rysowane) współrzędne podziałek zegara. Do tego celu wykorzystano funkcje trygonometryczne obliczane w procedurze *Obl_f_tr*, której parametrem jest wartość wyrażona w stopniach. W tej procedurze, za pomocą funkcji *Der2rad*, stopnie są zamieniane na radiany oraz obliczane są wartości funkcji *sin* oraz *cos*. Obliczone wartości funkcji *sin* i *cos* są skalowane, zaokrąglane przez funkcję $5 V \rightarrow \pm 10 V$

Round oraz zamieniane na wartość całkowitą przez funkcję *Int*. Obliczane są wartości początku oraz końca podziałki, które są następnie rysowane za pomocą instrukcji *Line*, mającej identyczną składnię jak dla wyświetlaczy z kontrolerem KSx. Tu także, by linia była reprezentowana przez zapalone piksele, kolor powinien mieć wartość z zakresu 1...255. Po wykonaniu procedury rysującej zegar na LCD, zostają wyświetlone w określonych pozycjach nazwy składników daty, pod którymi będą wyświetlane wartości daty. Następnie program przechodzi do wykonywania nieskończonej pętli, w której jeśli wartość sekund będzie różna od poprzedniej to następuje przepisanie wartości czasu do zmiennych pomocniczych, po czym zostają wywołane procedury obliczające i wyświetlające nowe pozycje wskazówek zegara. Jeśli wartość godzin równa jest 0, to wywołana zostanie procedura wy-



Rys. 8. Alternatywne rozwiązanie przetwornicy $5 V \rightarrow \pm 10 V$

światlająca (aktualizująca) datę. Do wyświetlenia daty wykorzystano instrukcje *Locate*, *Lcd* oraz funkcje *Makebcd* oraz *Bcd*. Dzięki temu przy wartości elementów daty poniżej 10, wyświetlane są one jako 05, a nie 5. Bardzo ważnymi elementami są procedury obliczające pozycje i rysujące wskazówki. W procedurze *Sekundy*, której zawartość wykonywana jest dwukrotnie w pętli *For*, następuje obliczenie współrzędnych wskazówki sekund w zależności od wartości zmiennej *Sek*. Do wyznaczenia położenia wskazówki, także tym przypadku wykorzystano funkcje *sin* oraz *cos*, obliczane w procedurze *Obl_f_tr*. Po obliczeniu, wartości funkcji są skalowane, zaokrąglane i zamieniane na wartości całkowite. Rysowanie wskazówki odbywa się za pomocą instrukcji *Line*. W tej procedurze, odbywa się także, za pomocą instrukcji *Circle*, odświeżanie środkowego punktu tarczy zegara. Zawartość procedury *Sekundy* wykonywana jest dwukrotnie w pętli *For*. Pierwsze wykonanie pętli *For* powoduje wymazanie poprzedniego położenia wskazówki (kolor w instrukcji *Line* przyjmuje wartość 0), natomiast w drugim jej wykonaniu rysowana jest wskazówka

w nowym położeniu. Podobną budowę ma procedura *Minuty*, która dotyczy wskazówki minut. Trochę inaczej jest zbudowana procedura *Godziny* obliczająca i wyświetlająca wskazówkę godzin. Ma ona większą rozdzielczość niż wskazówka sekund i minut, których rozdzielczość wynosi 60. Położenie wskazówki godzin dodatkowo zostało uzależnione od wartości minut. Chodzi o to by wskazówka godzin zmieniała położenie stopniowo (z większą rozdzielczością niż 12) w zależności od minut (tak jak jest w normalnych zegarkach), a nie skokowo, przy zmianie godziny na nową. Oczywiście dla odróżnienia wskazówek sekund, minut i godzin mają one różne wartości określające długość. Wskazówka sekund ma długość 27, minut – 21, a godzin – 15. Wydaje mi się, że program i algorytm działania analogowego zegara jest zrozumiałe. Można go we własnym zakresie rozbudować oraz zaimplementować jako dodatek we własnych układach. Wyświetlenie pliku graficznego na wyświetlaczu z kontrolerem T6963C przebiega identycznie jak dla wyświetlaczy z kontrolerem KSx i SED. Odbywa się więc przy użyciu dyrektywy *\$bgf* oraz

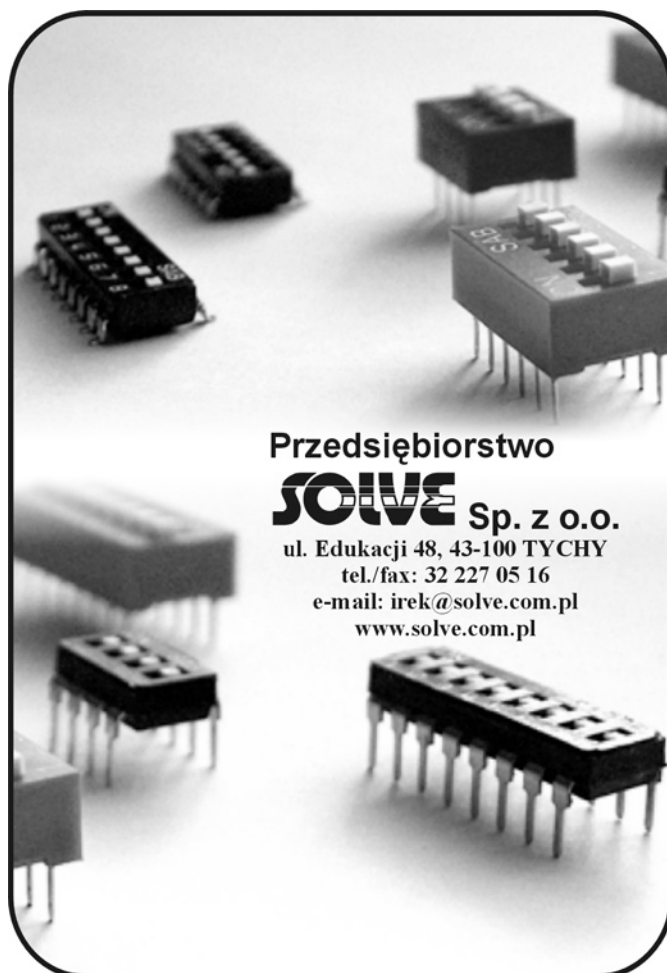
instrukcji *Showpic* oraz *Showpice*, jeśli grafika jest zapisywana w pamięci EEPROM. Dla wyświetlaczy z kontrolerem T6963, dodatkowo jest dostępna instrukcja *Pset*, która umożliwia zapalenie/zgaszanie jednego piksela w wybranym miejscu ekranu. Jej pierwsze dwa parametry określają pozycję x, y modyfikowanego piksela. Ostatni parametr określa kolor. Przy wartości koloru 0, piksel jest gaszony, a przy 1...255 zapalany. Na pewno dużym atutem wyświetlaczy z kontrolerem T6963C, jest wbudowany generator znaków, którego nie mają wyświetlacze z kontrolerem KSx czy SED.

Podsumowanie

Zaprezentowane w artykule wyświetlacze z kontrolerem KSx i T6963C musiały mieć obwody

regulacji kontrastu polarizowane ujemnym napięciem, do którego wytworzenia użyto znanego układu MAX232. Można także wykorzystać inne układy przetwornic pojemnościowych, jak np. MAX680, którego schemat aplikacyjny przedstawiono na **rys. 8**. Do poprawnego działania potrzebuje jedynie czterech kondensatorów. Układ ten dodatkowo wytwarza oprócz napięcia -10 V , napięcie $+10\text{ V}$. Dzięki zaimplementowanym w Bascomie instrukcji przeznaczonych dla wyświetlaczy graficznych, ich obsługa jest niezwykle prosta. Co ważne, wykorzystywanie coraz tańszych wyświetlaczy graficznych, podnosi znacznie atrakcyjności i komfort obsługi konstruowanych urządzeń.

Marcin Wiązania, EP
marcin.wiazania@ep.com.pl



Przedsiębiorstwo
SOLVE Sp. z o.o.
ul. Edukacji 48, 43-100 TYCHY
tel./fax: 32 227 05 16
e-mail: irek@solve.com.pl
www.solve.com.pl



PRECYZYJNE
REZYSTORY METALIZOWANE

Rezystancje od $0,3\ \Omega$ do $10\ \text{M}\Omega$
Tolerancje od 0,01% do 0,5%

elpod
POLSKI
PRODUCENT

31-416 Kraków
ul. Dobrego Pasterza 120
tel. (012) 410-25-50 do 51
fax (012) 410-25-52

http://www.elpod.com.pl e-mail: biuro@elpod.com.pl

Oferujemy ponadto: Rezystory SMD 0805 oraz 1206 $10\ \Omega$ do $1\ \text{M}\Omega$
Tolerancje 0,1%; 0,25%; 0,5%; 1%
TWR 10, 25, 50 ppm/K