Sterowanie wyświetlaczem graficznym z kontrolerem firmy Toshiba T6963C – kod programu.

Edgar Ostrowski
Jan Kędzierski

www.konar.ict.pwr.wroc.pl

Wrocław, 28.01.2007

Spis treści

Niniejszy raport przedstawia listing wszystkich funkcji jakie zostały napisane dla wyświetlaczy graficznych z kontrolerem firmy Toshiba dla mikrokontrolera formy FreeScale MC68332..

1 Wstęp

Niezwykle istotne jest to aby pomiary z różnych czujników przedstawiać w odpowiedni łatwy do ogarnięcia okiem sposób. Taką możliwość dają nam wyświetlacze graficzne. Sporo dostępnych na rynku modeli posiadają już wbudowany kontroler grafiki a co niektóre nawet generator znaków. Jednym z najpopularniejszych jest kontroler firmy Toshiba T6963C. W raporcie tym zrezygnowano z opisu poszczególnych rejestrów. Skupiono się głównie na kodzie napisanym dla mikrokontrolera firmy FreeScale MC68332. Wadą modułów z wyświetlaczami jest brak normy na wyprowadzenia tak jak w przypadku wyświetlaczy tekstowych. Zaletą jednak może być fakt, że niemal każdy moduł wyposażony jest w generator napięcia ujemnego.

2 Linki

Do dokładnego zrozumienia zasady działania niektórych funkcji niezbędne będzie zapoznanie się z dokumentacją kontrolera firmy Toshiba T6963C.

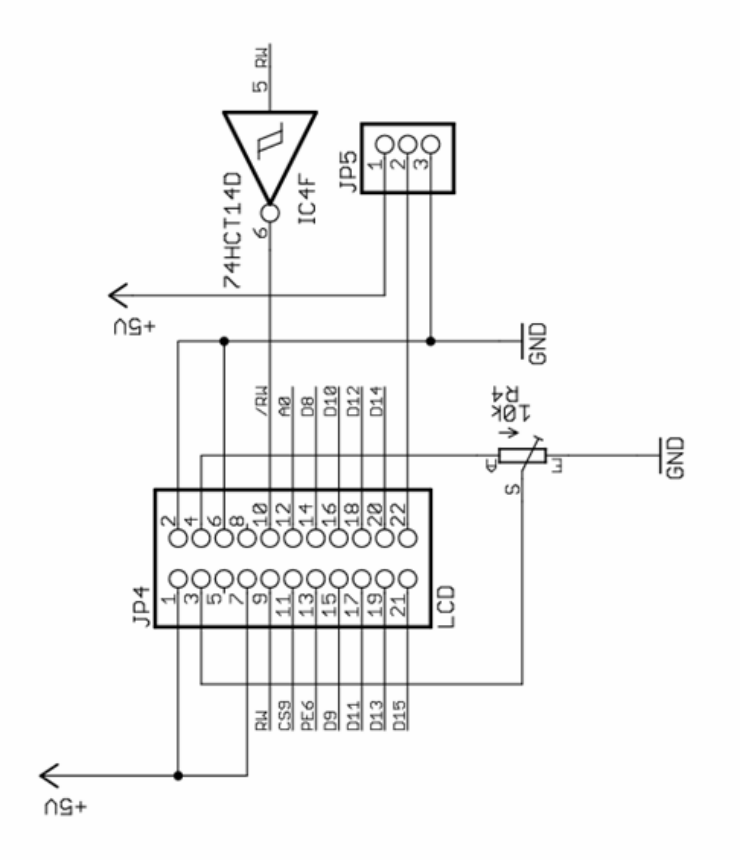
http://konar.ict.pwr.wroc.pl/module.php?op=download&cmd=click&id=71

http://www.lcdinfo.com/lcd/datasheets/e007211.pdf

http://www.skippari.net/lcd/t6963c.pdf

http://homepages.tesco.net/~steve.lawther/steve/t6963c.htm

3 Schemat połączeń



No.	Symbol	Function
1	A	Backlight supply voltage(+5V)
2	K	Backlight supply voltage(-)
3	VOUT	Negative voltage output
4	V0	Negative voltage input
5	PD	Control pin for oscillation
6	GND	VSS(0V)
7	VDD	Power supply for Logic(+5.0V)
8	VEE	LCD supply voltage
9	/WR	Write signal
10	/RD	Read signal
11	/CE	Chip selection signal
12	C/D	Data or Instruction select signal(H: data register L:instruction register)
13	/RST	Reset signal
14-21	DB0~DB7	Data bus line
22	FS	Font selection terminal

4 Listing kodu

LCD.H

```
//////////////////////////////////////////

//  Jan Kedzierski

//       2006


//////////////////////////////////////////


#define LCD_DR 0xeff800             /* adres rejestru danych LCD */
#define LCD_IR LCD_DR+1             /* adres rejestru sterujacego LCD */


volatile char LcdIr @LCD_IR;
volatile char LcdDr @LCD_DR;
void lcd_data(char, unsigned char);
void lcd_cmd_2(char, unsigned int);
void lcd_cmd(char);
void Lcd_Init();
void lcd_logo();
void Lcd_Put_Text(int,int,unsigned char *) ;
void Lcd_Put_Graphic(int,int,int,unsigned char *);
void Lcd_clear_graph();
void Lcd_clear_text();
void Lcd_pixel(int, int ,char );
void Lcd_line(int,int,int,int,unsigned char);
void Lcd_circle(int,int, int, unsigned char);
void Lcd_box(int,int,int,int,unsigned char);
void Lcd_circle_segment(int,int,int,int,int,unsigned char);
void Lcd_box_filled (int,int,int,int,unsigned char );
void Lcd_V_line (int,int,int, unsigned char);
void Lcd_H_line (int,int,int, unsigned char);
void Lcd_window (int,int,int,int,char *, unsigned char);



unsigned char fontgen_char_table[];
```

LCD.C
```
/*************************************************************/
#include <stdio.h>
#include <string.h>
#include "Lcd.h"
#include "sim.h"                                /* definicje rejestrow SIM
dla 68332 */
#include "logo.h";
//////////////////////////////////////////

//  Jan Kedzierski

//       2006


//////////////////////////////////////////
#define lcd_FONT_SELECT 1                       // Jaka trzcionka? 1 = FS
HIGH = waska (6x8) / 0 = FS LOW = normalna (8x8)
#define LCD_WIDTH  240
#define LCD_HEIGHT 128
#define PIN_RESET 2
//////////////////////////////////////////


#if lcd_FONT_SELECT                             // waska trzcionka
      #define FONT_WIDTH 6                 // czerokosc trzcionki/"bajtu"
      #define BYTES_PER_ROW LCD_WIDTH/FONT_WIDTH           // ilosc bajtow
w jednej lini
      #define lcd_G_BASE 0x0400                // adres pamieci grafiki
#else                                           // zwykla trzcionka
      #define FONT_WIDTH 8                   // zerokosc trzcionki/"bajtu"
      #define BYTES_PER_ROW LCD_WIDTH/FONT_WIDTH               // ilosc
bajtow w jednej lini
    #define lcd_G_BASE 0x0200            // adres pamieci grafiki
#endif

#define lcd_T_BASE  0x0000                    // adres pamieci grafiki

#define lcd_XMAX LCD_WIDTH        // limity (x,y) graficznego pola
wyswietlacza
#define lcd_XMIN 0
#define lcd_YMAX LCD_HEIGHT
#define lcd_YMIN 0


#define SET_TEXT_ADDR_START(addr)     lcd_cmd_2(0x40, addr)            //
definicja zapisu adresu pola textowego
#define SET_TEXT_LINE_WIDTH(bytes)    lcd_cmd_2(0x41, bytes)     //
definicja ustawienia szerokosc lini w polu textowym

#define SET_GRAPH_ADDR_START(addr)    lcd_cmd_2(0x42, addr)            //
definicja zapisu adresu pola graficznego
#define SET_GRAPH_LINE_WIDTH(bytes)   lcd_cmd_2(0x43, bytes)     //
definicja ustawienia szerokosc lini w polu graficznym

#define SET_ADDR_PTR(ptr)             lcd_cmd_2(0x24, ptr)       //
definicja ustawienia adresu (text or graph)
#define SET_CURS_PTR(ptr)             lcd_cmd_2(0x21, ptr)       //
definicja ustawienia kursora
```

```
#define SET_GRAPH_TEXT_XOR()          lcd_cmd(0x81)                    //
definicja trubu pracy przy nakladaniu sie textu i grafiki typ: XOR
#define SET_GRAPH_TEXT_OR()           lcd_cmd(0x80)                    //
definicja trubu pracy przy nakladaniu sie textu i grafiki typ: OR
#define SET_GRAPH_TEXT_AND)           lcd_cmd(0x83)                    //
definicja trubu pracy przy nakladaniu sie textu i grafiki typ: AND

#define SET_DISPLAY_MODE(mode)        lcd_cmd(mode)                    //
definicja mode (grafika on/off,text on/of, cursor on/off/blink)


#define AUTO_WRITE_ON()                lcd_cmd(0xB0)             //
definicja zalaczenia trybu auto (nie trzeba ustawiac adresu)
#define AUTO_WRITE_OFF()               lcd_cmd(0xB2)             //
definicja zalaczenia trybu auto (jak off to trzeba zawsze ustawic addr_ptr)

#define SET_CURSOR_PATTERN(pattern)    lcd_cmd(0xA0|pattern)     //
definicja wzoru kursora (8 trybow) cienki....guby

#define lcd_busy_wait()      while(((LcdIr & 0x01) != 0x01)|((LcdIr & 0x02)
!= 0x02))   //sprawdza statuc wyswietlacza na ST1 i ST2



// zapis instrukcji i danych jednobajtowych
void lcd_data(char cmd, unsigned char parm)
{
  lcd_busy_wait();
  LcdDr=(parm);
  lcd_busy_wait();
  LcdIr=cmd;
}

// zapis instrukcji i danych dwubajtowych
void lcd_cmd_2(char cmd, unsigned int parm)
{
  lcd_busy_wait();
  LcdDr=(parm&0xFF);
  lcd_busy_wait();
  LcdDr=(parm>>8);
  lcd_busy_wait();
  LcdIr=cmd;
}

// zapis tylko instrukcji bez danych
void lcd_cmd(char cmd)
{
  lcd_busy_wait();
  LcdIr=cmd;
}

// Reset i Inicjacja wyswietlacza
void Lcd_Init()
{
  unsigned int i;

  //reset
  DDRE|= (1<<PIN_RESET);                              // 2-ga linia portu E
jako wyjscie
  PORTE1 &= ~(1<<PIN_RESET);
  for(i=0; i<32000; i++);                             // Delay
```

```
  PORTE1 |= (1<<PIN_RESET);
  // end reset

  SET_GRAPH_ADDR_START(lcd_G_BASE);             // ustaw zdefiniowany wyzej
adres pola grafiki
  SET_GRAPH_LINE_WIDTH(BYTES_PER_ROW);          // ustaw zdefiniowana wyzej
szerokosc lini pola grafiki

  SET_TEXT_ADDR_START(lcd_T_BASE);              // ustaw zdefiniowany wyzej
adres pola textu
  SET_TEXT_LINE_WIDTH(BYTES_PER_ROW);           // ustaw zdefiniowana wyzej
szerokosc lini pola textu


  SET_ADDR_PTR(0);                                          // ustaw na
poczatku pola textowego
  SET_GRAPH_TEXT_XOR();                                 // ustaw zdefiniowany
powyzej tryb pracy textu i grafiki
  SET_DISPLAY_MODE(0x9c);                               // Graph and text ON,
Cursor OFF, Blinking cursor OFF
                                                                  // 1  0
      0     1     1     1     0     0
                                                                  //
                          |     |     |     \
                                                                  //
                          |     |     |      blink off
                                                                  //
                          |     |     \ cursor off
                                                                  //
                          |     \ text on
                                                                  //
                          \ grafika on
    Lcd_clear_text();                                   // czysci pole textowe
    Lcd_clear_graph();                                  // czysci pole grafiki
}


// funkcja czyszczaca pole grafiki, zapisuje 0 do kazdego adresu
void Lcd_clear_graph()              // clear graphics memory of LCD
{
  int i;

  SET_ADDR_PTR(lcd_G_BASE);         // ustaw adres poczatku pola grafiki
  lcd_busy_wait();                        // sprawdz gotowosc
  AUTO_WRITE_ON();                        // tryb autozapisu on
  for (i=0;i<BYTES_PER_ROW*LCD_HEIGHT;i++) {lcd_busy_wait();
LcdDr=(char)0;    }     // zapisz dana na linie danych
  lcd_busy_wait();                        // sprawdz gotowosc
  AUTO_WRITE_OFF();                       // tryb autozapisu off
}

// funkcja czyszczaca pole textu, zapisuje 0 do kazdego adresu
void Lcd_clear_text()
{
  int i;

  SET_ADDR_PTR(lcd_T_BASE);         // ustaw adres poczatku pola textu
  lcd_busy_wait();                        // sprawdz gotowosc
  AUTO_WRITE_ON();                        // tryb autozapisu on
  for (i=0;i<BYTES_PER_ROW*LCD_HEIGHT/8;i++) {lcd_busy_wait();
LcdDr=(char)0;  }
```

```
  lcd_busy_wait();                          // sprawdz gotowosc
  AUTO_WRITE_OFF();                         // tryb autozapisu off
}

// funkcja zpisujaca text na wyswietlaczu
// zadaje sie pozycje textu a nie pixeli,  x - linia textu, y - kolumna
textu oraz wskaznik na stringa
void Lcd_Put_Text(int x,int y,unsigned char *string)  // send string of
characters to LCD
{
  int i;
  int addr;

  addr = lcd_T_BASE + (y * BYTES_PER_ROW) + x; // wylicz adres i dodaj go
do poczatku adresu pola tekstowego
  SET_ADDR_PTR(addr);                                          // ustaw
adres w polu textowym na zadanej pozycji
  AUTO_WRITE_ON();
  lcd_busy_wait();
  while(*string){lcd_busy_wait();  LcdDr=(char)((*string++)-0x20); }   //
konwertuj znaki aby generator textu je zrozumial
  lcd_busy_wait();
  AUTO_WRITE_OFF();
}

// funkcja zpisujaca tablice grafiki na wyswietlaczu
// zadaje sie pozycje  lcd row - od ktorej lini na wyswietlaczu ma zaczac
rysowac,
// offset_row - od ktorej lini w tablicy zadanej ma rysowac (jesli nie
potrzebujemy wszystkiego rysowac),
// size - do ktorego elementu zadanej tablicy grafiki na rysowac np
sizeof(tablica[]),
// oraz wskaznik na tablice z zadana grafika
void Lcd_Put_Graphic(int lcd_row,int offset_row,int size,unsigned char
*graph_table)
{
  int i;
  int addr;
  addr=lcd_G_BASE + (lcd_row)*BYTES_PER_ROW;         // wylicz adres i
dodaj go do poczatku adresu pola grafiki
  SET_ADDR_PTR(addr);                                          // ustaw
adres w polu textowym na zadanej pozycji
  AUTO_WRITE_ON();
  lcd_busy_wait();
  for(i=offset_row*BYTES_PER_ROW;i<size;i++) {lcd_busy_wait();
LcdDr=(char)(graph_table[i]); } //zapis wg zadanych parametrow
  lcd_busy_wait();
  AUTO_WRITE_OFF();
}


// zaswieca lub gasi pojedynczy pixel na wyswietlaczu, show=1 zaswiec,
show=0 zgas
// zadaje sie pozycje pixela (x,y)
void Lcd_pixel(int column, int row, char show)
{
  int addr;
  if( (column>=lcd_XMAX) || (row>=lcd_YMAX) )return;       // sprawdzamy
czy miesci sie w grancach wyswietlacza
  addr=lcd_G_BASE + (row*BYTES_PER_ROW)  + (column/FONT_WIDTH);
  SET_ADDR_PTR(addr);    // set LCD addr. pointer
```

```
  if(show)
    lcd_cmd(0xf8 | ((FONT_WIDTH-1-column%FONT_WIDTH)) );  // ustawia
pojedynczy bit w bajcie 0xf8 - komenda zapalenia
  else
            // pojedynczego pixela w ustawionym adresie
    lcd_cmd(0xf0 | ((FONT_WIDTH-1-column%FONT_WIDTH)) );  // ustawia
pojedynczy bit w bajcie 0xf0 - komenda zgaszenia
  // pojedynczego pixela w ustawionym adresie
}
// funkcja wyswietla linie laczaca dwa zadane punkty
// zadaje sie punkt1(x,y) i punkt2(x,y) oraz show=1 zaswiec, show=0 zgas
void Lcd_line(int x1, int y1, int x2, int y2, unsigned char show)
{
  int dy ;
  int dx ;
  int stepx, stepy, fraction;
  dy = y2 - y1;
  dx = x2 - x1;
  if (dy < 0)
    {
      dy = -dy;
      stepy = -1;
    }
  else
    {
      stepy = 1;
    }
  if (dx < 0)
    {
      dx = -dx;
      stepx = -1;
    }
  else
    {
      stepx = 1;
    }
  dy <<= 1;
  dx <<= 1;
  Lcd_pixel(x1,y1,show);
  if (dx > dy)
    {
      fraction = dy - (dx >> 1);
      while (x1 != x2)
      {
        if (fraction >= 0)
          {
            y1 += stepy;
            fraction -= dx;
          }
        x1 += stepx;
        fraction += dy;
        Lcd_pixel(x1,y1,show);
      }
    }
  else
    {
      fraction = dx - (dy >> 1);
      while (y1 != y2)
      {
        if (fraction >= 0)
          {
```

```
            x1 += stepx;
            fraction -= dy;
          }
        y1 += stepy;
        fraction += dx;
        Lcd_pixel(x1,y1,show);
      }
    }
}

// funkcja wyswietla okrag a wyswietlaczu
// zadaje sie srodek okregu w pixelach(x,y), promien oraz show=1 zaswiec,
show=0 zgas
void Lcd_circle(int x, int y, int radius, unsigned char show)
{
  int xc = 0;
  int yc ;
  int p ;
  yc=radius;
  p = 3 - (radius<<1);
  while (xc <= yc)
    {
      Lcd_pixel(x + xc, y + yc, show);
      Lcd_pixel(x + xc, y - yc, show);
      Lcd_pixel(x - xc, y + yc, show);
      Lcd_pixel(x - xc, y - yc, show);
      Lcd_pixel(x + yc, y + xc, show);
      Lcd_pixel(x + yc, y - xc, show);
      Lcd_pixel(x - yc, y + xc, show);
      Lcd_pixel(x - yc, y - xc, show);
      if (p < 0)
      p += (xc++ << 2) + 6;
      else
      p += ((xc++ - yc--)<<2) + 10;
    }
}

// funkcja wyswietla prostokat na wyswietlaczu
// zadaje sie wierzcholek lewy-gorny(x,y) i prawy dolny(x,y) oraz show=1
zaswiec, show=0 zgas
void Lcd_box(int x1, int y1, int x2, int y2, unsigned char show)
{
  Lcd_H_line(x1,x2,y1,show);  // up
  Lcd_H_line(x1,x2,y2,show);  // down
  Lcd_V_line(x1,y1,y2,show);  // right
  Lcd_V_line(x2,y1,y2,show);  // left
}



#if lcd_FONT_SELECT                // Jaka trzcionka? 1 = FS HIGH = waska
(6x8) / 0 = FS LOW = normalna (8x8)
      #define MASK 0x20
#else                              // definiuje maske potrzebna w kolejnej
fukkcji 00100000  "1" na 6 miejscu
      #define MASK 0x80       // definiuje maske potrzebna w kolejnej
fukkcji 10000000  "1" na 8 miejscu
#endif

// funkcja wyswietla zamalowany prostokat na wyswietlaczu
```

```
// zadaje sie wierzcholek lewy-gorny(x,y) i prawy dolny(x,y) oraz show=1
zaswiec, show=0 zgas
void Lcd_box_filled (int x1, int y1, int x2, int y2, unsigned char show)
{
  int i,j,add=0;
  int addr1,addr2;
  int tmp1=0,tmp2=0;

  for(i=0;i<=FONT_WIDTH-(x1%FONT_WIDTH)-1;i++) { tmp1=tmp1<<1;  tmp1+=1; }
  for (i=0;i<=x2%FONT_WIDTH;i++)  {  tmp2=tmp2>>1;  tmp2+=MASK;  }

  for (j=0;j<=y2-y1;j++)
    {
      addr1=lcd_G_BASE + ((y1+j)*BYTES_PER_ROW) + (x1/FONT_WIDTH);
      addr2=lcd_G_BASE + ((y1+j)*BYTES_PER_ROW) + (x2/FONT_WIDTH);
      SET_ADDR_PTR(addr1);
      if (addr1==addr2) {if (show)  lcd_data(0xc0, tmp1&tmp2); else
lcd_data(0xc0, 0x00);
      }
      else
      {
        AUTO_WRITE_ON();
        lcd_busy_wait();
        if (show)  LcdDr=(char)tmp1; else LcdDr=0x00;
        for (i=1;i<=addr2-addr1-1;i++)
          {
            lcd_busy_wait();
            if (show) LcdDr=(char)0xff; else LcdDr=(char)0x00;
          }
        lcd_busy_wait();
        if (show) LcdDr=(char)tmp2; else  LcdDr=0x00;
        lcd_busy_wait();
        AUTO_WRITE_OFF();
      }
    }
}


#if lcd_FONT_SELECT                              // Jaka trzcionka? 1 = FS
HIGH = waska (6x8) / 0 = FS LOW = normalna (8x8)
      #define MASK 0x20                    // definiuje maske potrzebna w
kolejnej fukkcji 00100000  "1" na 6 miejscu
#else
      #define MASK 0x80                  // definiuje maske potrzebna w
kolejnej fukkcji 10000000  "1" na 8 miejscu
#endif

// funkcja wyswietla pozioma linie
// zadaje sie poczatek(x), koniec(x) i wiersz(y) w pixelach oraz show=1
zaswiec, show=0 zgas
void Lcd_H_line (int x1, int x2, int y, unsigned char show)
{
  int i;
  int addr1,addr2;
  int tmp1=0,tmp2=0;

  for(i=0;i<=FONT_WIDTH-(x1%FONT_WIDTH)-1;i++) { tmp1=tmp1<<1;  tmp1+=1; }
  for (i=0;i<=x2%FONT_WIDTH;i++)  {  tmp2=tmp2>>1;  tmp2+=MASK;  }

  addr1=lcd_G_BASE + ((y)*BYTES_PER_ROW) + (x1/FONT_WIDTH);
  addr2=lcd_G_BASE + ((y)*BYTES_PER_ROW) + (x2/FONT_WIDTH);
```

```
  SET_ADDR_PTR(addr1);
  if (addr1==addr2) {if (show) lcd_data(0xc0, tmp1&tmp2); else
lcd_data(0xc0, 0x00);
  }
  else
    {
      AUTO_WRITE_ON();
      lcd_busy_wait();
      if (show)  LcdDr=(char)tmp1; else LcdDr=0x00;
      for (i=1;i<=addr2-addr1-1;i++)     {
      lcd_busy_wait();
      if (show) LcdDr=(char)0xff; else LcdDr=(char)0x00;
      }
      lcd_busy_wait();
      if (show) LcdDr=(char)tmp2; else  LcdDr=0x00;
      lcd_busy_wait();
      AUTO_WRITE_OFF();
    }
}

// funkcja wyswietla pionowa linie
// zadaje sie poczatek(y), koniec(y) i kolumne(x) w pixelach oraz show=1
zaswiec, show=0 zgas
void Lcd_V_line (int x, int y1,int y2, unsigned char show)
{
  int i;
  int addr;

  for (i=y1;i<=y2;i++) {
    addr=lcd_G_BASE + ((i)*BYTES_PER_ROW) + (x/FONT_WIDTH);
    SET_ADDR_PTR(addr);
    if(show)
      lcd_cmd(0xf8 | ((FONT_WIDTH-1-x%FONT_WIDTH)) );  // set bit-within-
byte command
    else
      lcd_cmd(0xf0 | ((FONT_WIDTH-1-x%FONT_WIDTH)) );  // set bit-within-
byte command
  }
}

/*
  void Lcd_box_filled (int x1, int y1, int x2, int y2, unsigned char show)
  {
  int i;

  for (i = y1; i <= y2; i++)
  Lcd_line(x1, i, x2, i, show);
  }
*/

// funkcja ta wyswietla okno na wyswietlaczu wraz z podnym tytulem
// zadajemy lewy-gorny wierzcholek (kolumna,wiersz), prawy-dolny
wierzcholek (kolumna,wiersz), tytul (string)
// oraz oraz show=1 zaswiec, show=0 zgas
// UWAGA!!!! pozycje zadajemy tak jak przy zapisywaniu textu czyli kolumne
i wiersz pola textowego
// a nie grafiki w pixelach
void Lcd_window (int row_x1,int column_y1 ,int row_x2,int column_y2,char
*title, unsigned char show)
{
  char buf[40];
```

```
  int tmp=0;
  int i=0;
  if (column_y1>0)  tmp=1;
  Lcd_box_filled(row_x1*FONT_WIDTH+2,column_y1*8-tmp,row_x2*FONT_WIDTH+2,
column_y2*8+8-tmp, 0);
  Lcd_box_filled(row_x1*FONT_WIDTH+2,column_y1*8-tmp,row_x2*FONT_WIDTH+2,
column_y1*8+8-tmp, show);
  Lcd_box(row_x1*FONT_WIDTH+2, column_y1*8-tmp ,row_x2*FONT_WIDTH+2,
column_y2*8-tmp, show);
  if (show){
    Lcd_Put_Text(row_x1+1,column_y1,title);
    Lcd_Put_Text(row_x2-1,column_y1,"X");
  }
  else
    {
      for(i=0;i<row_x2-row_x1-2;i++) buf[i]=*" ";
      Lcd_Put_Text(row_x1+1,column_y1,buf);
    }
}

// funkcje te mozna uzyc do testow wyswietlacza
// Uwaga poprawnei wyswietlona grafika LOGO mozliwa bedzie tylko gdy FS=1
6x8
void lcd_logo()
{
  Lcd_Put_Graphic(0,0,sizeof(logo_char_table),logo_char_table);
}
```

Tak wygląda zapisane w tablicy **`logo_char_table[]`** logo KoNaR.

LOGO.H

```
//////////////////////////////////////////

//  Jan Kedzierski

//       2006


//////////////////////////////////////////

unsigned static char logo_char_table[] = {
            0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,0x3C,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x3F,
0x3F,0x00,0x00,0x00,0x00,0x18,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x07,0x01,0x3F,0x3F,0x3F,0x20,0x00,0x00,0x0F,0x3E,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3A,0x00,0x3F,
0x3F,0x3F,0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1E,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x23,0x3F,0x3F,0x3F,0x3F,0x20,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x30,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3E,0x0F,0x3F,0x3F,0x3F,0x3F,
0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x07,0x3C,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x38,0x0F,
0x3F,0x3F,0x20,0x0E,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x0F,0x3E,0x00,0x03,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x0E,0x03,0x20,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3C,0x01,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3E,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x1F,0x3F,0x3F,0x07,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x0F,0x3F,0x2F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x27,0x3F,0x3F,0x3F,0x3F,0x20,0x01,0x20,0x20,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1F,0x3F,0x3F,0x3F,
0x3F,0x38,0x00,0x3F,0x3F,0x3F,0x31,0x3F,0x3E,0x0F,0x3F,0x20,
0x07,0x39,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3C,0x00,0x1F,0x3F,0x3F,0x3C,0x3F,
0x38,0x07,0x3E,0x00,0x0F,0x3F,0x30,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x07,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x20,0x00,0x3F,
0x3F,0x3F,0x3C,0x0F,0x20,0x1F,0x3E,0x00,0x1F,0x3F,0x3F,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x20,0x03,0x3F,0x3F,0x3F,0x3F,0x0E,0x00,0x3F,0x3C,0x00,
0x3F,0x01,0x3F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x30,0x07,0x3F,0x3F,0x3F,0x3F,0x3C,
0x03,0x3F,0x00,0x01,0x3F,0x00,0x3F,0x30,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x30,0x07,0x3C,0x03,0x01,0x3E,0x00,0x17,0x20,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x07,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x0F,0x3F,0x20,0x0F,0x20,0x0F,0x21,
0x00,0x01,0x3F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x37,0x2F,0x3E,0x00,
0x1E,0x00,0x3F,0x00,0x00,0x03,0x3C,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x23,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3E,0x00,0x3E,0x00,0x3E,0x00,0x00,0x07,0x30,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x03,0x3F,0x3F,0x3F,0x3F,0x3F,0x3E,0x00,0x3F,0x30,0x00,0x07,
0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3C,
0x00,0x1F,0x3F,0x3F,0x01,0x3F,0x3F,0x3F,0x30,0x3F,0x3E,0x07,
0x3F,0x30,0x00,0x03,0x3F,0x38,0x00,0x07,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x3F,0x3F,0x31,0x38,0x3F,0x3F,0x38,0x1F,0x3F,0x3F,0x3F,
0x3D,0x3F,0x3E,0x0F,0x3F,0x3E,0x00,0x03,0x3F,0x38,0x00,0x1F,
0x3F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x38,0x1F,0x3F,0x3E,0x0F,0x31,
0x3F,0x00,0x00,0x3F,0x3F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x38,0x3F,
0x3F,0x3C,0x0F,0x30,0x00,0x00,0x01,0x3F,0x3F,0x20,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3E,0x00,0x3F,0x3F,0x38,0x00,0x03,0x20,0x00,0x03,0x3C,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x20,0x00,0x1F,0x3F,0x30,0x00,0x07,
0x20,0x00,0x07,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x00,0x07,
0x3F,0x00,0x00,0x00,0x00,0x00,0x1F,0x20,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,
0x3F,0x3F,0x3F,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3E,0x00,0x00,0x1F,0x30,0x00,0x0E,0x00,0x00,0x01,0x3F,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x3F,0x3F,0x3F,0x3F,0x07,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3E,0x0E,0x00,0x3F,0x00,0x00,0x1F,0x20,
0x00,0x03,0x3F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3E,0x1F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x20,0x1F,0x21,0x38,
0x00,0x00,0x03,0x00,0x00,0x03,0x23,0x20,0x00,0x3C,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,
0x3F,0x3C,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x21,0x3F,0x23,0x30,0x0F,0x00,0x00,0x00,0x00,0x1F,0x37,0x20,
0x03,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x38,0x03,0x21,0x3F,0x00,0x00,0x00,
0x01,0x3F,0x3F,0x3C,0x0F,0x3C,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x30,0x00,0x01,
0x38,0x00,0x00,0x00,0x07,0x26,0x3F,0x3F,0x3F,0x30,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,
```

```
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x20,0x00,0x01,0x30,0x00,0x00,0x00,0x1E,0x1C,0x01,0x3F,
0x3F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x1C,0x00,0x01,0x00,0x00,0x00,0x01,
0x39,0x38,0x00,0x0F,0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,
0x3F,0x3F,0x38,0x03,0x3F,0x3F,0x3F,0x3F,0x3E,0x3C,0x3C,0x00,
0x00,0x00,0x00,0x07,0x27,0x20,0x00,0x01,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3C,0x01,0x3F,0x3F,0x3F,0x3E,
0x00,0x3F,0x3C,0x00,0x00,0x00,0x00,0x1E,0x1E,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x07,0x3F,0x3F,0x3F,0x3F,0x3F,0x27,0x3F,0x3F,0x3E,0x00,
0x3F,0x3F,0x3F,0x20,0x07,0x3F,0x00,0x00,0x00,0x00,0x00,0x39,
0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x07,0x3F,0x3F,0x37,0x3F,0x3F,0x07,
0x3F,0x3F,0x3E,0x01,0x3F,0x3F,0x3F,0x30,0x07,0x23,0x00,0x00,
0x00,0x00,0x1F,0x37,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x3F,0x3F,
0x21,0x3F,0x32,0x07,0x3E,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x30,
0x00,0x1F,0x30,0x00,0x00,0x00,0x3F,0x3E,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x07,0x3F,0x3F,0x01,0x3F,0x30,0x01,0x38,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x20,0x00,0x3F,0x39,0x3F,0x3F,0x31,0x3F,0x38,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,0x3F,0x3F,0x38,0x00,
0x01,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x20,0x00,0x3D,0x3F,0x3F,
0x3F,0x3F,0x31,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x3F,0x3F,
0x3F,0x3F,0x3C,0x00,0x01,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x37,
0x3D,0x38,0x3E,0x00,0x00,0x1F,0x31,0x38,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x18,0x3F,0x3F,0x38,0x00,0x00,0x3F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x3F,0x3C,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x30,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x3F,0x00,0x00,
0x00,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x20,0x3F,0x39,0x3F,
0x3F,0x3D,0x3F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x01,0x38,0x00,0x00,0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3F,0x38,
0x00,0x0F,0x30,0x00,0x00,0x00,0x1F,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,
0x3F,0x3F,0x3F,0x20,0x01,0x30,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x03,0x20,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x3F,0x3F,0x3F,0x3F,0x3E,0x1F,0x20,
0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,
0x3F,0x20,0x07,0x30,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x07,0x3F,0x3F,0x20,0x00,0x01,0x3F,0x3E,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x00,0x00,0x00,0x00,0x3F,
```

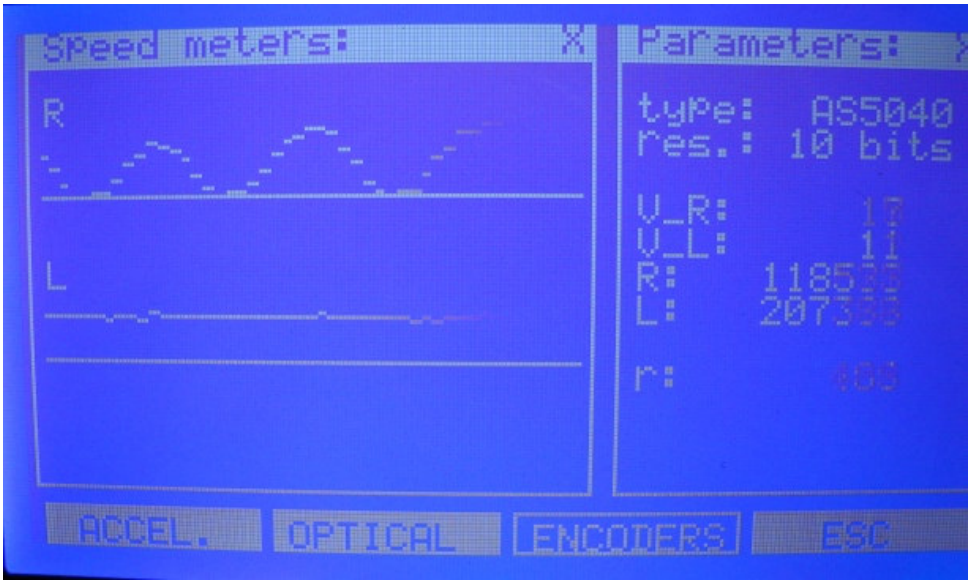
```
0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x0F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x38,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x38,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3E,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,
0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x0F,0x00,0x00,0x00,0x03,0x30,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3E,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x1F,0x3F,0x3F,0x3F,0x3E,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x1F,0x00,0x00,0x00,0x07,0x38,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x3F,0x3F,0x3F,0x3E,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x3F,0x3F,0x3F,0x3F,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0x00,
0x0F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x3F,0x3F,
0x3F,0x3F,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,
0x3F,0x3F,0x3F,0x3F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x0F,0x38,0x00,0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x03,0x3E,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,
0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,
0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,
0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x01,0x3F,0x3F,0x3F,0x20,0x00,0x03,0x3F,
0x00,0x00,0x00,0x1F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
```

```
0x3F,0x1F,0x3F,0x3F,0x3F,0x20,0x00,0x0F,0x3F,0x3F,0x3E,0x00,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x01,0x3F,0x3F,0x3F,
0x30,0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x00,0x0F,
0x3F,0x3F,0x3F,0x00,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x01,0x3F,0x3F,0x3F,0x38,0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3E,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x3F,0x3F,0x3F,
0x3E,0x00,0x00,0x0F,0x38,0x00,0x3F,0x20,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x01,0x3F,0x00,0x07,0x3C,0x00,0x03,0x3F,
0x3F,0x3F,0x3F,0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x00,0x0F,0x38,0x00,0x3F,0x30,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x01,0x3F,0x00,0x07,
0x3E,0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3E,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x3F,0x3F,0x3F,0x3F,0x20,0x00,0x0F,
0x38,0x00,0x3F,0x30,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x01,0x3F,0x00,0x07,0x3E,0x00,0x03,0x3F,0x3F,0x3F,0x3F,0x3F,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x30,0x00,0x0F,0x38,0x00,0x3F,0x30,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x01,0x3F,0x00,0x07,0x3E,0x00,0x03,0x3E,
0x00,0x00,0x00,0x1F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x0F,0x38,0x00,0x3F,0x30,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x01,0x3F,0x00,0x07,
0x3E,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x0F,
0x38,0x00,0x3F,0x30,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x01,0x3F,0x00,0x07,0x3E,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x38,0x00,0x0F,0x38,0x00,0x3F,0x30,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x01,0x3F,0x00,0x07,0x3E,0x00,0x03,0x3F,
0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x0F,0x38,0x00,0x3F,0x30,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x01,0x3F,0x00,0x07,
0x3E,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x0F,
0x38,0x00,0x3F,0x30,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x01,0x3F,0x3F,0x3F,0x3E,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,
0x1F,0x38,0x00,0x0F,0x38,0x00,0x3F,0x30,0x00,0x1F,0x38,0x00,
0x00,0x03,0x3E,0x00,0x01,0x3F,0x3F,0x3F,0x3E,0x00,0x03,0x3F,
0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3F,0x00,0x00,0x00,0x1F,0x38,0x00,0x0F,0x38,0x00,0x3F,0x30,
0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,0x01,0x3F,0x0F,0x3F,
0x3E,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x3F,0x00,0x00,0x00,0x0F,0x38,0x00,0x0F,
0x38,0x00,0x1F,0x30,0x00,0x1F,0x38,0x00,0x00,0x03,0x3E,0x00,
0x01,0x3F,0x00,0x03,0x3E,0x00,0x03,0x3F,0x00,0x00,0x00,0x1F,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x3E,0x00,0x00,0x00,
0x0F,0x38,0x00,0x07,0x3C,0x00,0x1F,0x30,0x00,0x1F,0x30,0x00,
0x00,0x01,0x3E,0x00,0x00,0x3F,0x00,0x03,0x3E,0x00,0x03,0x3E,
0x00,0x00,0x00,0x0F,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0x3C,0x00,0x00,0x00,0x07,0x38,0x00,0x03,0x3F,0x3F,0x3F,0x30,
```

```
0x00,0x1F,0x20,0x00,0x00,0x00,0x3E,0x00,0x00,0x1F,0x00,0x01,
0x3E,0x00,0x03,0x3C,0x00,0x00,0x00,0x07,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x38,0x00,0x00,0x00,0x03,0x38,0x00,0x01,
0x3F,0x3F,0x3F,0x30,0x00,0x1F,0x00,0x00,0x00,0x00,0x0E,0x00,
0x00,0x0F,0x00,0x00,0x3C,0x00,0x03,0x38,0x00,0x00,0x00,0x03,
0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x06,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x0F,0x00,0x00,0x00,0x39,0x3D,0x3F,0x39,0x20,0x07,0x3F,0x20,
0x0F,0x03,0x07,0x3C,0x30,0x19,0x27,0x23,0x3F,0x33,0x07,0x06,
0x1F,0x38,0x00,0x3F,0x39,0x3F,0x39,0x3F,0x33,0x3F,0x37,0x3F,
0x0E,0x1C,0x33,0x31,0x3F,0x39,0x23,0x23,0x3F,0x01,0x20,0x1D,
0x28,0x06,0x01,0x20,0x0F,0x33,0x0E,0x0C,0x30,0x19,0x3C,0x03,
0x00,0x31,0x2D,0x2C,0x1E,0x00,0x00,0x3C,0x19,0x20,0x19,0x3F,
0x37,0x00,0x30,0x38,0x03,0x30,0x3F,0x01,0x20,0x18,0x37,0x36,
0x3B,0x31,0x30,0x1D,0x30,0x06,0x03,0x20,0x0C,0x1F,0x0C,0x1C,
0x38,0x19,0x27,0x23,0x20,0x30,0x38,0x38,0x1D,0x00,0x00,0x3F,
0x31,0x20,0x39,0x21,0x33,0x01,0x30,0x38,0x01,0x20,0x3B,0x31,
0x30,0x18,0x1C,0x1C,0x30,0x1C,0x3F,0x39,0x3F,0x27,0x3F,0x00,
0x0C,0x07,0x0F,0x34,0x1F,0x39,0x21,0x31,0x3F,0x30,0x10,0x30,
0x1F,0x3C,0x00,0x3C,0x19,0x3F,0x31,0x3F,0x23,0x3F,0x20,0x10,
0x01,0x20,0x30,0x18,0x3F,0x38,0x08,0x08
};
```

5 Przykłady



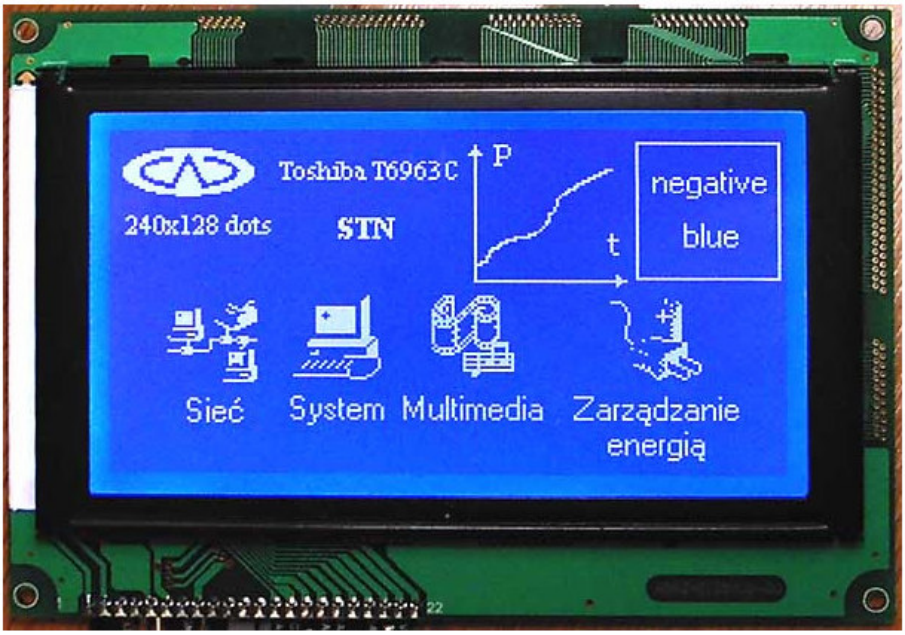
Fot. 1 – Wizualizacja przebiegów czasowych.

Fot. 2 – Wektor.

Fot. 3 – Tekst w oknach.

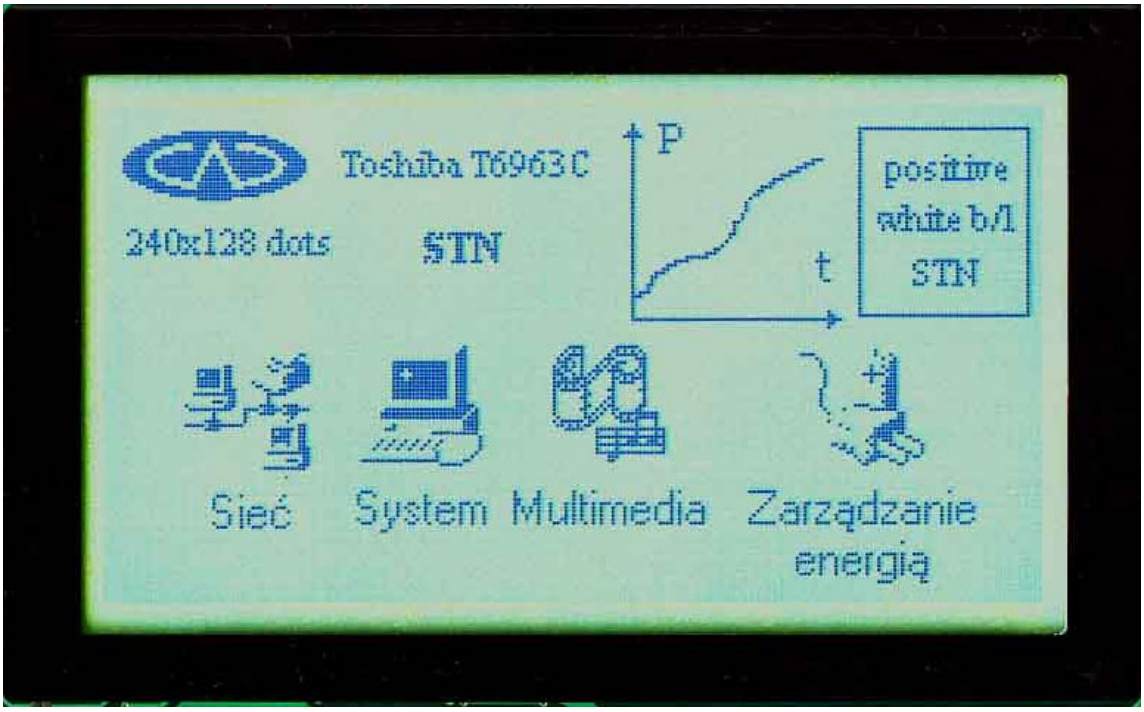
Fot. 4 – Menu

6 Przykładowe wyprowadzenia



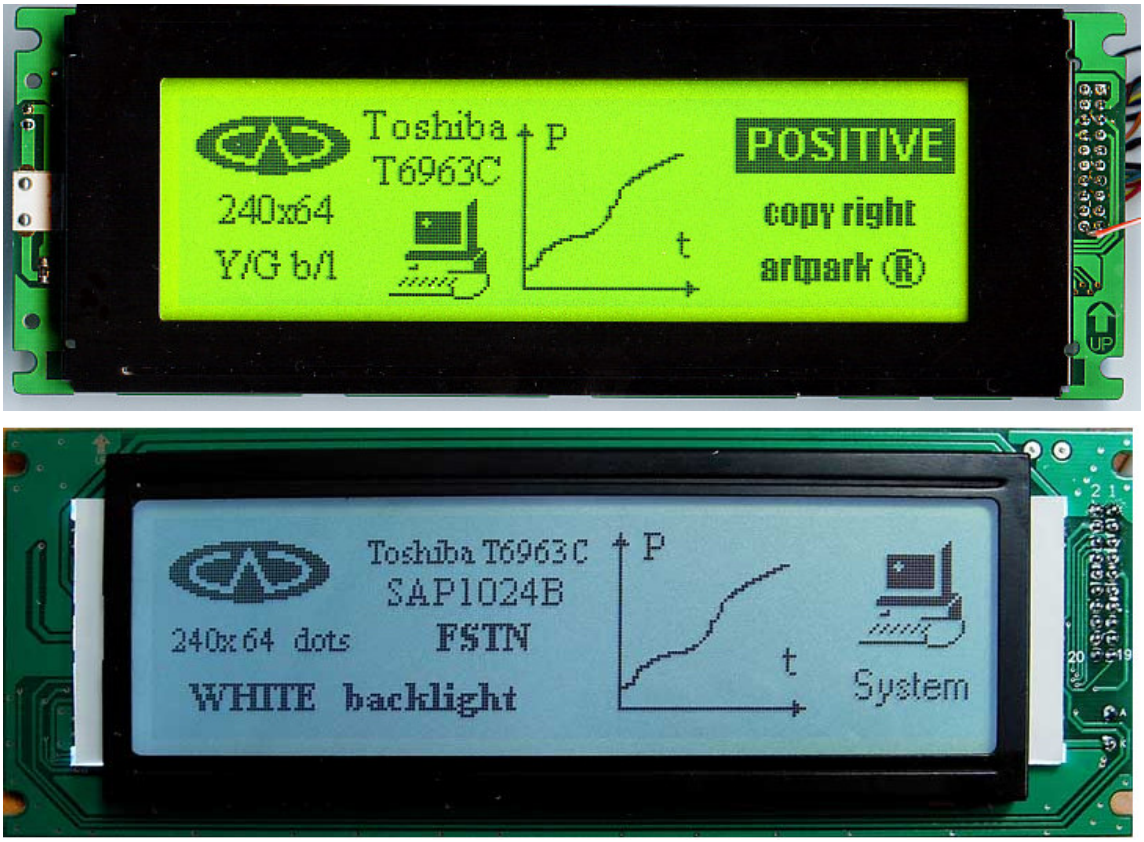
Fot. 5 – Widok wyświetlacza.

No.	Symbol	Function
1	A	Backlight supply voltage(+5V)
2	K	Backlight supply voltage(-)
3	VOUT	Negative voltage output
4	V0	Negative voltage input
5	PD	Control pin for oscillation
6	GND	VSS(0V)
7	VDD	Power supply for Logic(+5.0V)
8	VEE	LCD supply voltage
9	/WR	Write signal
10	/RD	Read signal
11	/CE	Chip selection signal
12	C/D	Data or Instruction select signal(H: data register L:instruction register)
13	/RST	Reset signal
14-21	DB0~DB7	Data bus line
22	FS	Font selection terminal



Fot. 6 – Widok wyświetlacza.

Pin No.	Symbol	Level	Description
1	FG	0V	Frame ground
2	V_{SS}	0V	Ground Chip select signal
3	V_{DD}	5.0V	Supply voltage for logic and LCD
4	V_O	-	Operating voltage for LCD(variable)
5	WR	L	Write signal
6	RD	L	Read signal
7	CE	L	Chip enable signal
8	C/D	H/L	Data or Instruction select signal
9	RST	L	Reset
10	DB0	H/L	Data bit0
11	DB1	H/L	Data bit1
12	DB2	H/L	Data bit2
13	DB3	H/L	Data bit3
14	DB4	H/L	Data bit4
15	DB5	H/L	Data bit5
16	DB6	H/L	Data bit6
17	DB7	H/L	Data bit7
18	FS	H/L	Font selection
19	NC	LED A	power supply for backlight (+)
20	NC	VEE	negative voltage output



Fot. 7 – Widok wyświetlaczy.

Pin No	Symbol	Level	Description
1	FG	--	Frame Ground
2	VSS	--	Ground for Logic
3	VDD	--	Power supply for Logic
4	Vo	--	Power supply for LCD drive
5	/WR	L	Write signal
6	/RD	L	Read signal
7	/CE	L	Chip enable signal
8	C/D	H/L	Register select signal (H:data register, L:instruction reg
9	NC	--	No connection
10	/RST	L	Reset signal
11-18	DB0-DB7	H/L	Data Bus lines
19	FS	H/L	Font selection
20	VEE	--	Negative voltage output