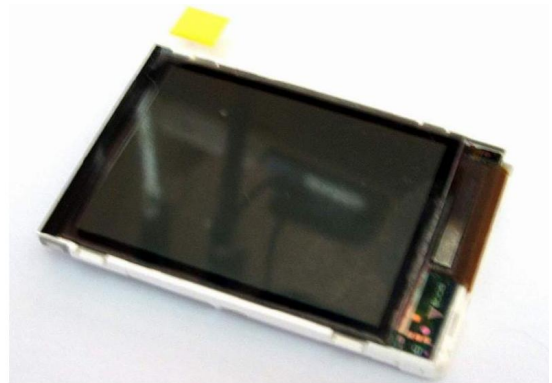


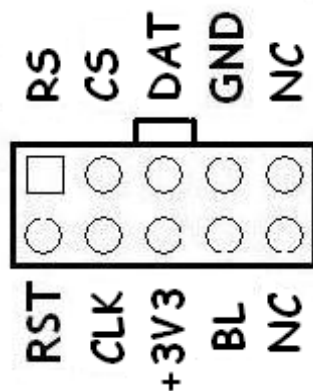
# Moduł wyświetlacza graficznego

ARE0055

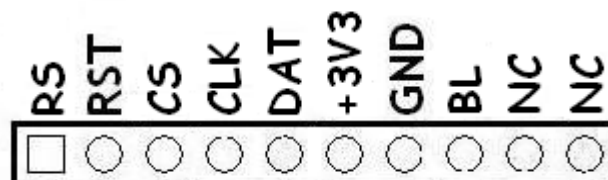
Doskonałe rozszerzenie zestawu edukacyjnego opartego o mikrokontrolery z rodziny LPC210x. Pozwala na wyświetlanie różnego rodzaju grafik, wykresów oraz prostych animacji.



Moduł wyświetlacza graficznego oparty o matrycę z telefonu komórkowego Siemens S65. Posiada rozdzielczość 132 x 176 pikseli oraz paletę 65536 kolorów. Jest sterowany przez interfejs SPI. Na rysunku 1 przedstawiono opis wyprowadzeń modułu.



A.



B.

Rysunek 1. Widok złącza modułu wyświetlacza:  
a. widok wtyczki z przodu,  
b. listwa kołkowa typu goldpin

RST - RESET  
RS - Data/Command  
CS - Chip Select  
CLK - Clock  
BL - Backlight  
DAT - Data  
NC - Not connected

Wszystkie wyprowadzenia modułu obsługują standard LVTTTL (3V3). Aktywacja podświetlenia modułu odbywa się stanem wysokim. W module zastosowano wyświetlacz ze sterownikiem LS020xxx firmy Sharp.

Poniżej zamieszczono poszczególne funkcje umożliwiające komunikację z wyświetlaczem. Zadaniem użytkownika jest zdefiniowanie stałych w nich użytych.

```
// definiowanie wyrazen dla mikrokontrolera LPC210x
//ustawienie wybranego pinu
#define RS_set IOSET=(1<<11);
#define RST_set IOSET=(1<<12);
#define CS_set IOSET=(1<<13);
#define CLK_set IOSET=(1<<14);
#define DAT_set IOSET=(1<<15);
// wyzerowanie wybranego pinu
#define RS_clr IOCLR=(1<<11);
#define RST_clr IOCLR=(1<<12);
#define CS_clr IOCLR=(1<<13);
#define CLK_clr IOCLR=(1<<14);
#define DAT_clr IOCLR=(1<<15);

// Wysyłanie danych
void S65_send(int data)
{
    int i;

    for(i=0;i<16;i++)
    {
        if((data<<i)&(1<<15))
        {
            DAT_set;
            CLK_clr;
            CLK_set;
        }
        else
        {
            DAT_clr;
            CLK_clr;
            CLK_set;
        }
    }
}

// Inicjalizacja.
// First, Second, Third, Fourth - to sekwencje inicjalizujące,
// które muszą występować po soboe w ściśle określonych odstępach czasowych.
void S65_init(void)
{
    CS_set;
    CLK_set;
    DAT_set;
    RS_set;
    RST_clr;
    RST_set;

    RS_set;
```

```
//First
CS_clr;
S65_send(0xfdfd);
S65_send(0xfdfd);
CS_set;
wait_ms(100);

//Second
CS_clr;
S65_send(0xef00);
CS_set;

CS_clr;
S65_send(0xee04);
S65_send(0x1b04);
CS_set;

CS_clr;
S65_send(0xfefe);
S65_send(0xfefe);
CS_set;

CS_clr;
S65_send(0xef90);
S65_send(0x4a04);
S65_send(0x7f3f);
S65_send(0xee04);
S65_send(0x4306);
CS_set;
wait_ms(7);

//Third
CS_clr;
S65_send(0xEF90);
S65_send(0x0983);
S65_send(0x0800);
S65_send(0x0BAF);
S65_send(0x0A00);
S65_send(0x0500);
S65_send(0x0600);
S65_send(0x0700);
CS_set;

CS_clr;
S65_send(0xEF00);
CS_set;

CS_clr;
S65_send(0xEE0C);
CS_set;

CS_clr;
S65_send(0xEF90);
S65_send(0x0080);
CS_set;

CS_clr;
S65_send(0EFB0);
S65_send(0x4902);
CS_set;
```

```

CS_clr;
S65_send(0xEF00);
CS_set;

CS_clr;
S65_send(0x7F01);
S65_send(0xE181);
CS_set;

CS_clr;
S65_send(0xE202);
CS_set;

CS_clr;
S65_send(0xE276);
CS_set;

CS_clr;
S65_send(0xE183);
CS_set;
wait_ms(100);

//Fourth
CS_clr;
S65_send(0x8001);
CS_set;

CS_clr;
S65_send(0xEF90);
S65_send(0x0000);
CS_set;
}

void S65_settings(int orient, int X, int Y)
{
    RS_set;

    CS_clr;
    S65_send(0xef90);
    S65_send(0x0500+orient);
    S65_send(0x0600+Y);
    S65_send(0x0700+X);
    CS_set;
}

// Zestaw funkcji, jakie nalezy wywolac, aby uruchomic LCD i nadac mu kolor niebieski
S65_init();
S65_settings(0x04,0x00,0x00);

RS_clr;
CS_clr;
for(i=0;i<23232;i++)
{
    S65_send(0x041f);
    wait_us(100);
    //S65_send(are_logo[i]);
}
CS_set;

```