

# Pavlin.si -> same zanimive stvari

Šola, MTB, potepanja, elektronika, foto

Višinski diagram	Manjka GPX zapis.
------------------	-------------------

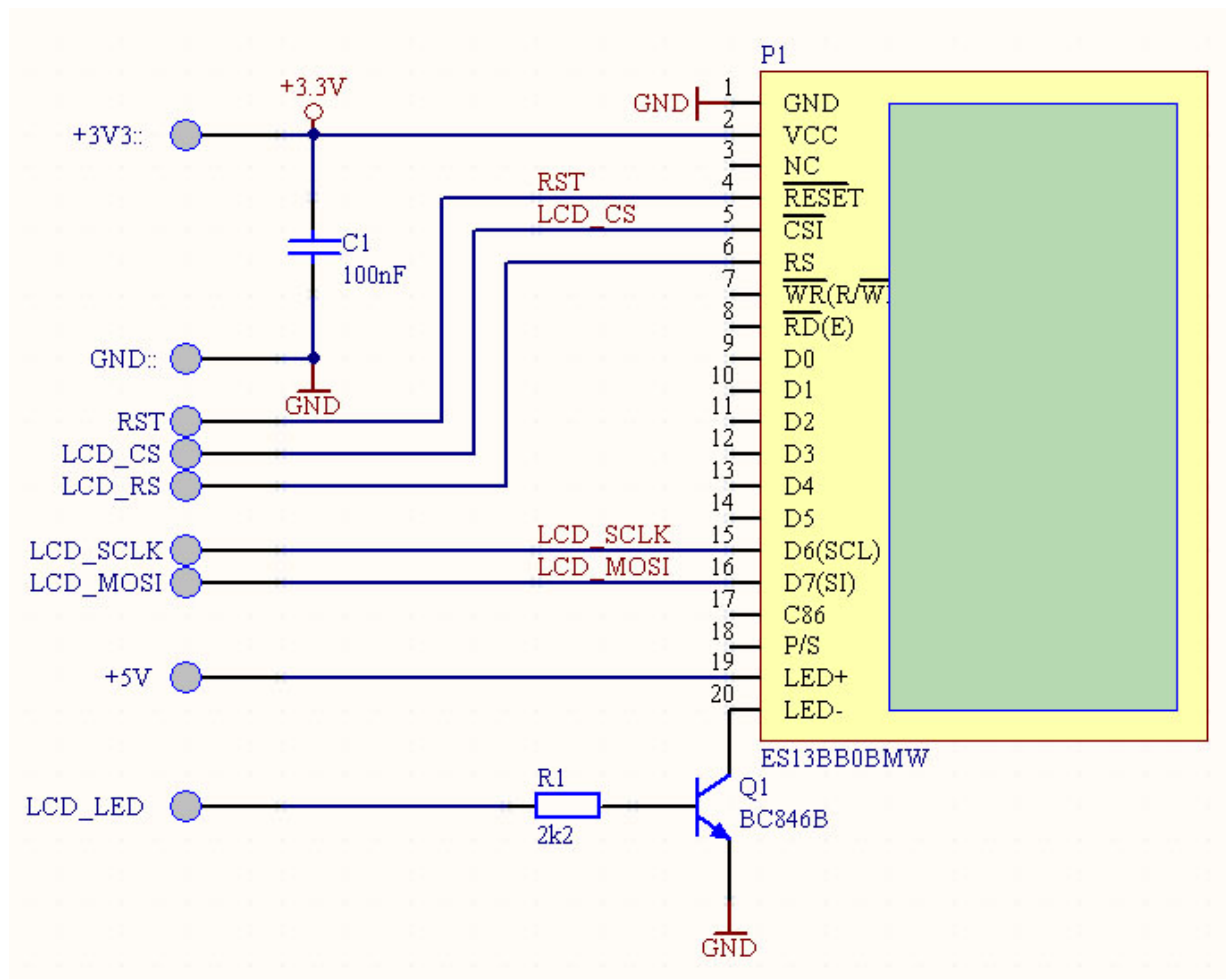
## Grafični LCD

Velikokrat moramo svojemu projektu dodati tudi možnost prikaza različnih stvari. Ena od možnosti je LED prikazovalnik s 7 segmentnimi prikazovalniki, druga možnost je alfanumerični, "klasični" LCD velikosti od 1×8 do 4×40 znakov in vse vmes, tretja možnost so preprosti grafični LCD prikazovalniki (kot je opisan v nadaljevanju), "najlepšejšo" sliko pa dobimo z uporabo barvnih grafičnih LCDjev (npr. QVGA 320×240 TFT barvni LCD). Žal pa slednji potrebujejo tudi precej dodatne elektroniko in so zelo zahtevni s stališča procesiranja. Zaradi tega se mi zdi nekje najbolj optimalna izbira monokromatski (dvobarvni) LCD prikazovalnik ločljivosti 128×64 točk. Ta omogoča nekaj več kot običajni 2×16 alfanumerični, istočasno pa za krmiljenje ne potrebujemo nekega "hudega" programa. Vmesnik je "samo" SPI, kar zmanjša število potrebnih linij na minimum. Dobro je tudi, da je prikazovalnik dobavljiv. Jaz sem izbral model proizvajalca EDT z oznako ES13BB0BMW.

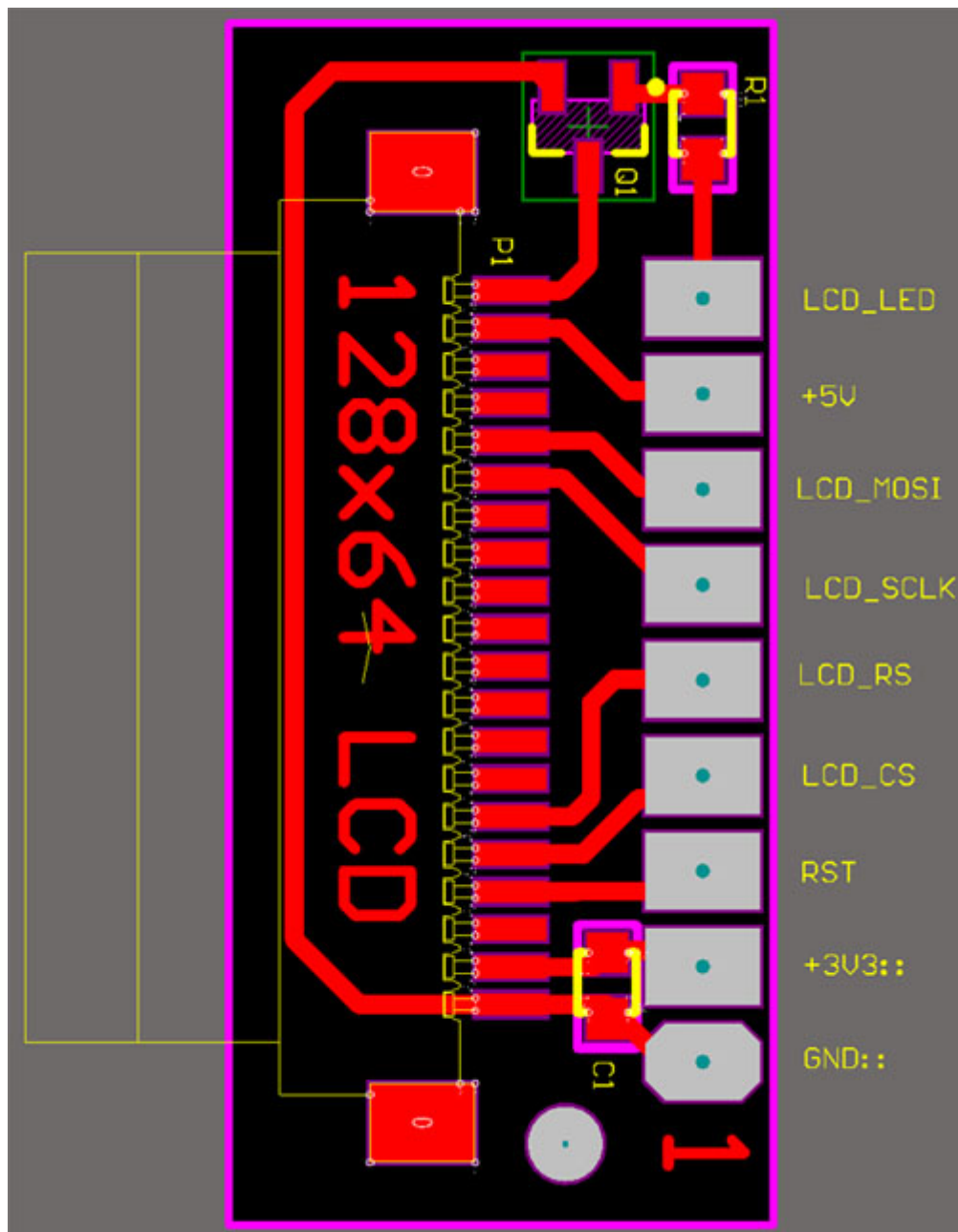
### Monokromatski LCD ločljivosti 128×64 točk z osvetlitvijo

LCD proizvajalca [Emerging Display Technologies](#) ima oznako [ES13BB0BMW](#) ima diagonalo 73mm in je v modro/sivi kombinaciji. Obstajajo tudi kombinacije v zeleni, rumeni barvi. Poganja ga sitronixov krmilnik [ST7565P](#), tako da je tale opis primeren za vse LCD prikazovalnike s tem krmilnikom brez sprememb ali zu minimalnimi prilagoditvami v inicializaciji. Za napajanje potrebujemo le 3,3V. Digitalne linije pa so SPI vhodne linije (SCK, MOSI in CS) ter ena linija za izbiro vpisa ukaza ali podatka (RS). LCD ima tudi osvetlitev, čeprav je viden tudi ob dnevni svetlobi brez osvetlitve.

Shema priklopa je prikazana na sliki:



LCD ima ploščat fleksibilen priključek, za katerega potrebujemo konektor. Izdelava vmesne ploščice je skoraj trivialna, saj v glavnem služi za prilagoditev ozkega rastra na LCD priključku na raster 2,54mm:



Tule je arhiv projekta za altium.

### Izbira materiala

**Konektor FCC** Molex 528521970

**LCD 128×64 mono** ES13BB0BMW

Oboje lahko dobavi IC.

### Priklop na mikrokrmilnik

LCD sem uspešno priklopil na LPC2138 in LPC2378. Lahko pa uporabimo katerikoli mikrokrmilnik, ki ima SPI in dodatna 2 pina za signal R/S ter vklop osvetlitve. Osvetlitev lahko tudi direktno vežemo na napajanje.

Spodaj je koda za preizkus delovanja. Edina datoteka, ki jo je potrebno prilagoditi lastnemu vezju je glcdhw.h, kjer so vse definicije hardverskih povezav med

mikrokontrolerjem in LCD-jem.

Ker grafični LCD nima lastnih pisav, moramo sami poskrbeti za to. Sam sem uporabil kar kodo, ki je na voljo v [tem projektu](#).

Povezave sem izbral takole:

- \* P0.9 - MOSI 1
- \* P0.7 - SCK1
- \* P0.6 - LCD\_CS
- \* P0.5 - Backlight
- \* P0.4 - LCD\_RS

Koda je razdeljena na naslednje datoteke:

- glcd.c** Funkcije za krmiljenje grafičnega LCD in risanje osnovnih gradnikov
- fonts.c** Zbirka pisav
- glcdhw.c** Hardverske povezave med LCD in mikrokontrolerjem
- lcdtest.c** Testni program

Celotna izvorna koda je na voljo [tule](#). V bistvu je to arhiv celotnega keilovega uVision projekta. Koda je pripravljena tako, da jo je zelo enostavno uporabiti kjerkoli drugje. Spremeniti moramo le funkcije v `glcdhw.c`.

Če LCD priklopimo na LPC2378 po zgoraj opisanem razporedu, je za prikaz napisa in enega krogca dovolj taka koda:

```
#include "glcd.h"

int main(void)
{
    glcd_Init();
    glcd_ClrScr(1);
    glcd_backlight(1);

    glcd_Text(1, 32, 3, "(c)Mare 2010" );
    glcd_Circle(100, 50, 10, GLCD_MODE_SET);
    glcd_Text(1, 50, 1, "Ready." );

    glcd_UpdateAll();

    while(1);
}
```

## Opis funkcij

**Grafični LCD krmilimo z naslednjimi funkcijami:**

```

/*****
*
* Function:          glcd_Init(void)
* PreCondition:     SPI and RTOS initialized
* Input:            -
* Output:           -
* Overview:         Initialize LCD module
* Note:             Reset pause is implemented with os_dly_wait()
*****
/
void glcd_Init(void);
/*****
*
* Function:          glcd_SetInvert(unsigned char mode)
* PreCondition:     LCD initialized
* Input:            mode (GLCD_MODE_INVERT or GLCD_MODE_NORMAL)
* Output:           -
* Overview:         Set writing mode
* Note:             -
*****
/
void glcd_SetInvert(unsigned char mode);
/*****
*
* Function:          glcd_Update(unsigned int top, unsigned int bottom
)
* PreCondition:     LCD initialized
* Input:            top      top line of area to update.
*                  bottom  bottom line of area to update.
* Output:           -
* Overview:         Copy buffer-RAM to display-RAM
* Note:             -
*****
/
void glcd_Update(unsigned int top, unsigned int bottom);
#define glcd_UpdateAll() glcd_Update(GLCD_TOP, GLCD_BOTTOM);
/*****
*
* Function:          glcd_ClrScr(unsigned char bufferAndDisplay)
* PreCondition:     LCD initialized
* Input:            bufferAndDisplay (boolean), if 1 = clear LCD too
* Output:           -
* Overview:         clear LCD buffer and display
* Note:             -
*****
/
void glcd_ClrScr(unsigned char bufferAndDisplay);
/*****

```

```

*
* Function:          glcd_Dot(unsigned char x, unsigned char y, unsign
ed char mode)
* PreCondition:     LCD initialized
* Input:            x,y - dot coordinate
*                   mode -
GLCD_MODE_SET, GLCD_MODE_CLEAR, GLCD_MODE_XOR
* Output:           -
* Overview:         Set a single dot within LCD buffer
* Note:             -
*****
/
void glcd_Dot(unsigned char x, unsigned char y, unsigned char mode);
#define glcd_SetDot(x,y) glcd_Dot(x, y, GLCD_MODE_SET)
#define glcd_ClearDot(x,y) glcd_Dot(x, y, GLCD_MODE_CLEAR);
/*****
*
* Function:          void glcd_Line( unsigned char x1, unsigned char y
1,
                    unsigned char x2, unsigned char y2,
                    unsigned char mode )
* PreCondition:     LCD initialized
* Input:            x1,y1 - line start coordinate
*                   x2,y2 - line end coordinate
*                   mode -
GLCD_MODE_SET, GLCD_MODE_CLEAR, GLCD_MODE_XOR
* Output:           -
* Overview:         Draw a line
* Note:             -
*****
/
void glcd_Line(unsigned char x1, unsigned char y1,
              unsigned char x2, unsigned char y2, unsigned char mode);
/*****
*
* void glcd_Circle(unsigned char xCenter, unsigned char yCenter,
                  unsigned char radius, unsigned char mode)
* PreCondition:     LCD initialized
* Input:            xCenter,yCenter - circle center
*                   radius - circle radius
*                   mode -
GLCD_MODE_SET, GLCD_MODE_CLEAR, GLCD_MODE_XOR
* Output:           -
* Overview:         Draw a circle
* Note:             -
*****
/
void glcd_Circle(unsigned char xCenter, unsigned char yCenter,

```

```

        unsigned char radius, unsigned char mode);
/*****
*
* void glcd_Rect(unsigned char x, unsigned char y,
                unsigned char width, unsigned char height,
                unsigned char mode)
* PreCondition:   LCD initialized
* Input:         x,y - top left corner
*               width, height -
width and height of the rectangle
*               mode -
GLCD_MODE_SET, GLCD_MODE_CLEAR, GLCD_MODE_XOR
* Output:        -
* Overview:      Draw a rectangle
* Note:          -
*****/
/
void glcd_Rect(unsigned char x, unsigned char y, unsigned char width,
              unsigned char height,
              unsigned char mode);
/*****
*
* void glcd_Box(unsigned char x, unsigned char y,
               unsigned char width, unsigned char height,
               unsigned char mode)
* PreCondition:   LCD initialized
* Input:         x,y - top left corner
*               width, height - width and height of the box
*               mode -
GLCD_MODE_SET, GLCD_MODE_CLEAR, GLCD_MODE_XOR
* Output:        -
* Overview:      Draw a solid box
* Note:          -
*****/
/
void glcd_Box(unsigned char x, unsigned char y, unsigned char width,
             unsigned char height,
             unsigned char mode);
/*****
*
* glcd_Text(unsigned char left, unsigned char top,
            unsigned char font, const char *str)
* PreCondition:   LCD initialized
* Input:         left      coordinate of left start of string.
                top       coordinate of top of string.
                font      font number to use for display (see fo
nts.h)
                str       text string to display (null-

```

```

terminated)
* Output:          -
* Overview:        Prints the given string at location x,y in the sp
ecified font.
                    Prints each character given via calls to lcd_glyp
h.
* Note:            The entry string is null terminated and
                    non 0x20->0x7e characters are ignored.
*****
/
void glcd_Text(unsigned char left, unsigned char top, unsigned char f
ont, const char *str);
/*****
*
* Function:        glcd_WriteCommand(unsigned char byte)
* PreCondition:    LCD initialized
* Input:           command byte
* Output:          -
* Overview:        Send command to LCD module
* Note:            None
*****
/
void glcd_WriteCommand(unsigned char byte);
/*****
*
* Function:        glcd_backlight(BOOL state)
* PreCondition:    -
* Input:           LED status
* Output:          -
* Overview:        Switch backlight ON/OFF
* Note:            None
*****
/
void glcd_backlight(int state);

```