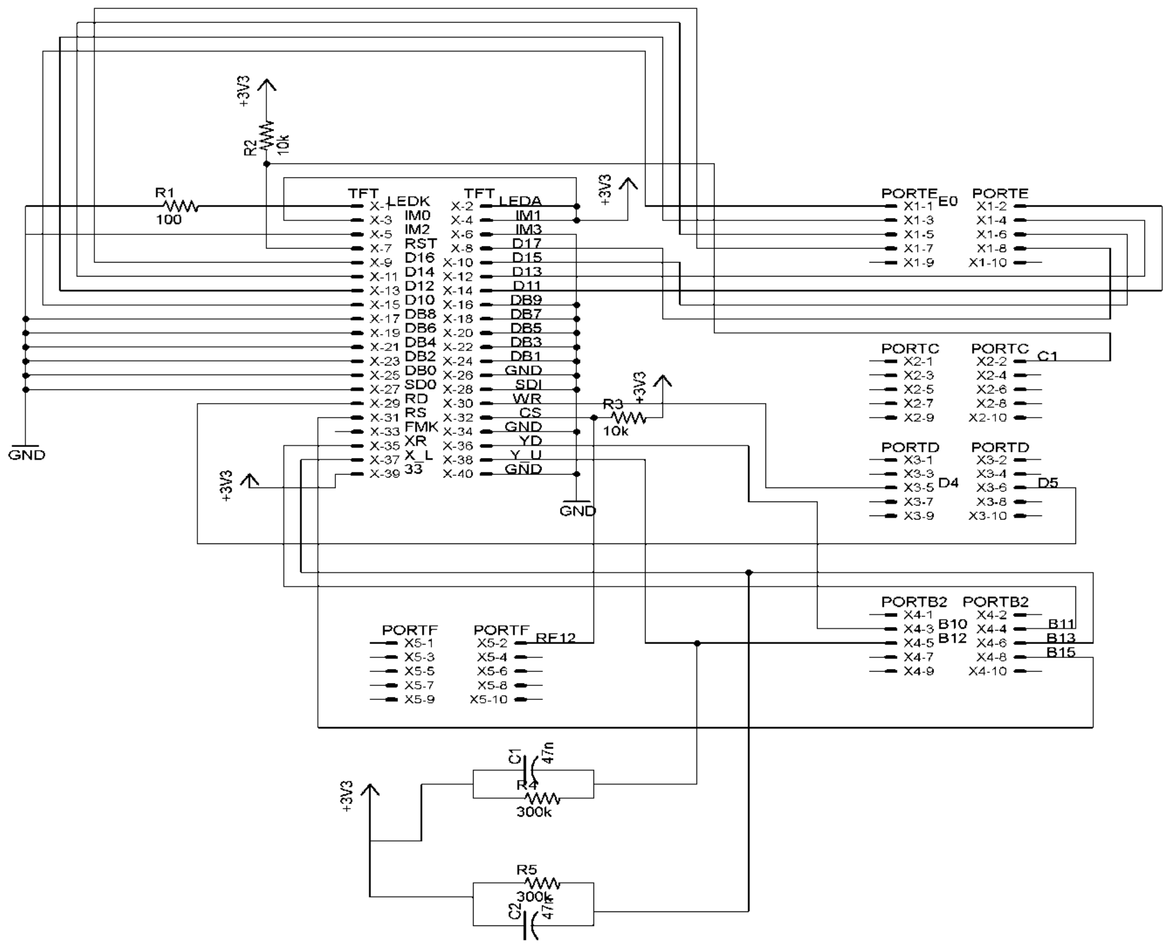
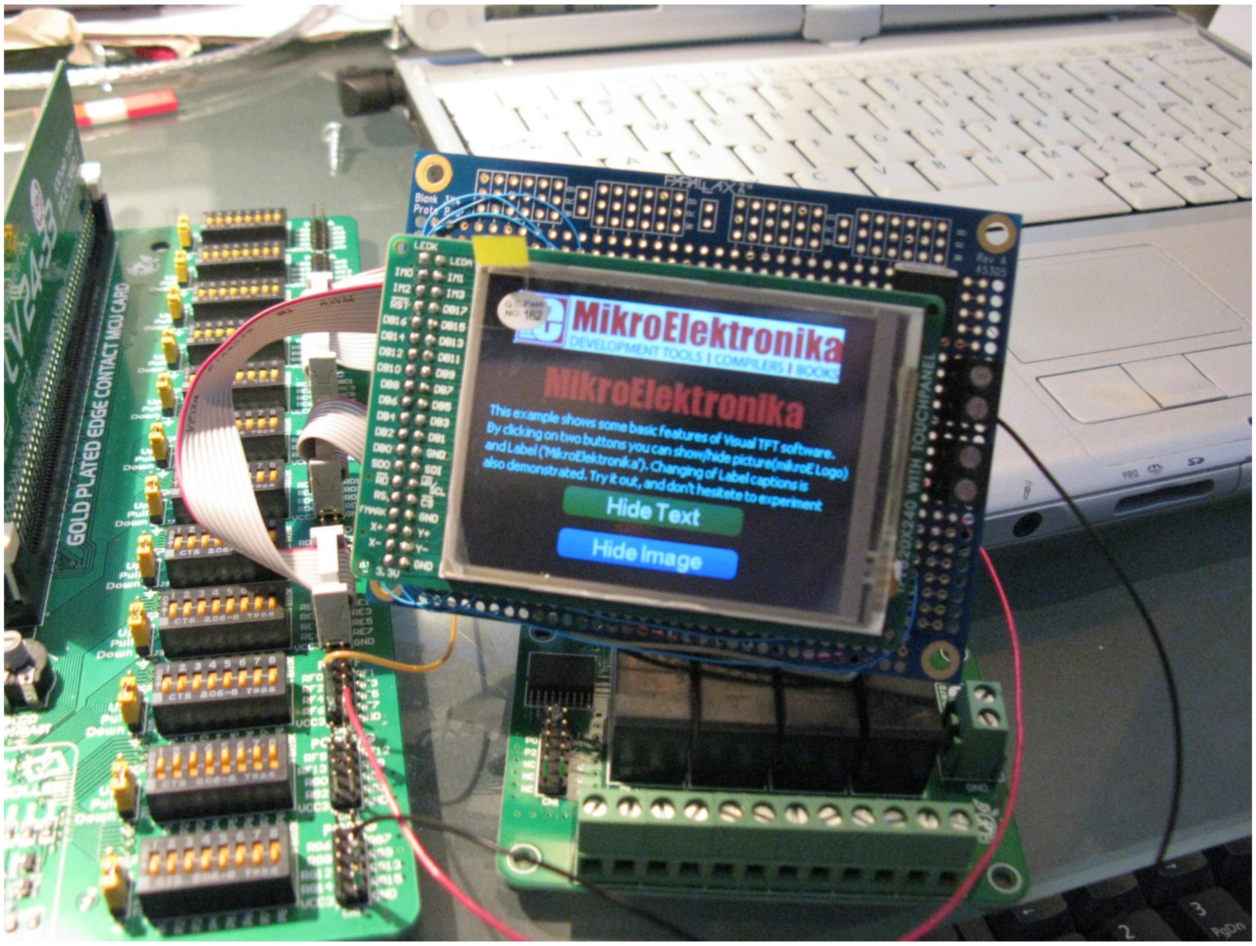
Using the TFT proto board with the LV24-33 development board and Visual TFT

1. Solder a header to the TFT proto board (male or female). You can desolder one from an obsolete PC card if you are stuck.
2. Obtain a blank 3”x4” prototype or perf board. Ideally, get something with plated through holes and solder mask to make things easy to solder.
3. Look at the schematic for the MikroMMB_for_PIC24. The goal is to wire up the perf board so that you can cable it to the ports on the LV24-33 board using ribbon connectors. The connections required are shown below, along with the capacitors needed for the touch controller. The advantage of doing this is that the preset for the MikroMMB_for_PIC24 board can be used in Visual TFT.
4. Look carefully at the LV24-33 board and disconnect any peripheral that is using the same ports. Turn off the monitoring LED’s on port B if you want the touch pad to work. The other port LED’s don’t seem to be a problem to the operation of the TFT.



The picture below shows the proto board connected to a Parallax 3”x4” proto board. On the back of the proto board are five 10 pin male headers just like the ones on the LV24-33 board. This allows the use of straight-through 10 conductor ribbon connectors which is a lot neater than a rat’s nest of single wires. I ran out of ribbon cables and used a few wires with female Molex connectors soldered to both ends.

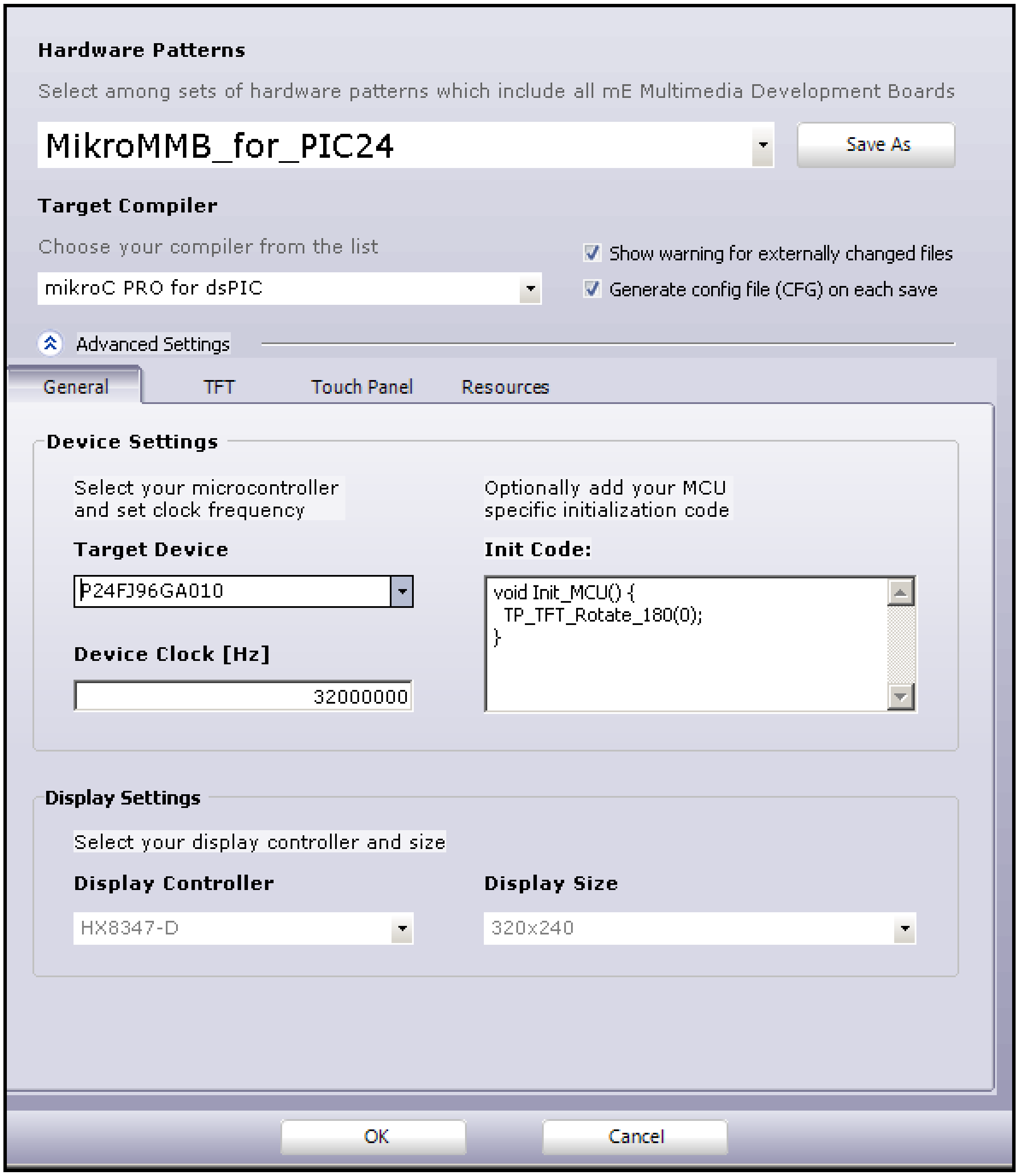

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS
MikroElektronika
This example shows some basic features of Visual TFT software. By clicking on two buttons you can show/hide picture(mikroE Logo) and Label ("MikroElektronika"). Changing of Label captions is also demonstrated. Try it out, and don't hesitate to experiment
Hide Text
Hide Image
GOLD PLATED EDGE CONTACT MCU CARD
LV24-33
PARALLAX

For power, you can use any of the VCC3 connections on the LV24-33 board. The board above has a 3.3 volt regulator and screw connector terminals so that it can be powered independently by a 5 volt or greater DC source.

Once you connect the TFT board and power it up via a USB connection (via the LV24-33), the back light should come on, however, there will be nothing on the display.

Visual TFT set up

1. Run Visual TFT and open a sample program for the MikroMMB_for_PIC24.
2. You will have to comment out AD1PCFGL = 0xCFFF; in the generated code
3. Change ADC threshold to 700. Otherwise the touch pad will not work.
4. Change the target to P24FJ96GA010 if that is your target CPU (you may have a dsPIC).


Hardware Patterns

Select among sets of hardware patterns which include all mE Multimedia Development Boards

MikroMMB_for_PIC24

Target Compiler

Choose your compiler from the list

mikroC PRO for dsPIC

✓ Show warning for externally changed files

✓ Generate config file (CFG) on each save

Device Settings

Select your microcontroller and set clock frequency

Optionally add your MCU specific initialization code

Target Device

P24FJ96GA010

Init Code:

```
void Init_MCU() {
  TP_TFT_Rotate_180(0);
}
```

Device Clock [Hz]

32000000

Display Settings

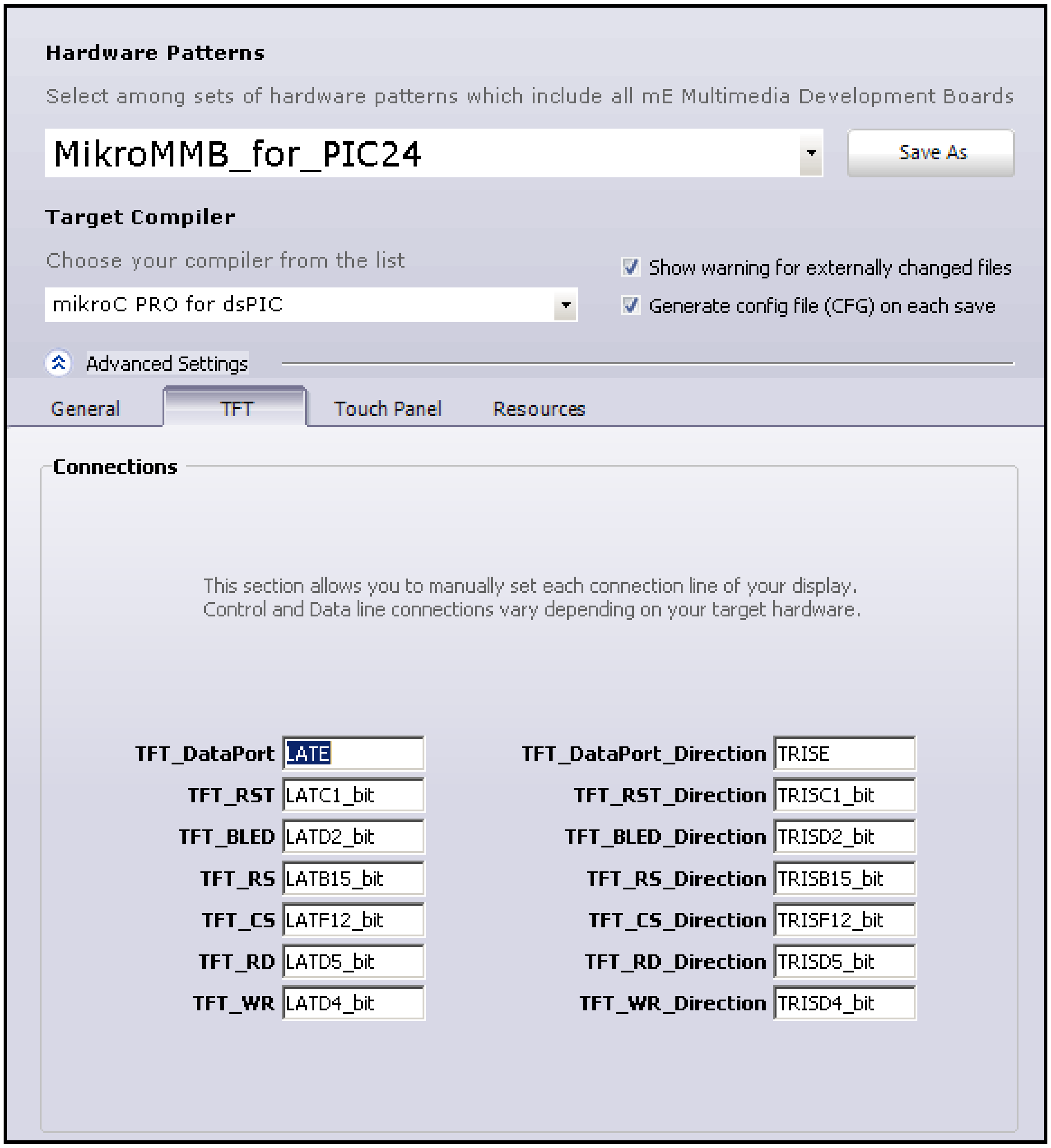
Select your display controller and size

Display Controller

HX8347-D

Display Size

320x240



Hardware Patterns

Select among sets of hardware patterns which include all mE Multimedia Development Boards

MikroMMB_for_PIC24

Save As

Target Compiler

Choose your compiler from the list

mikroC PRO for dsPIC

✓ Show warning for externally changed files

✓ Generate config file (CFG) on each save

Advanced Settings

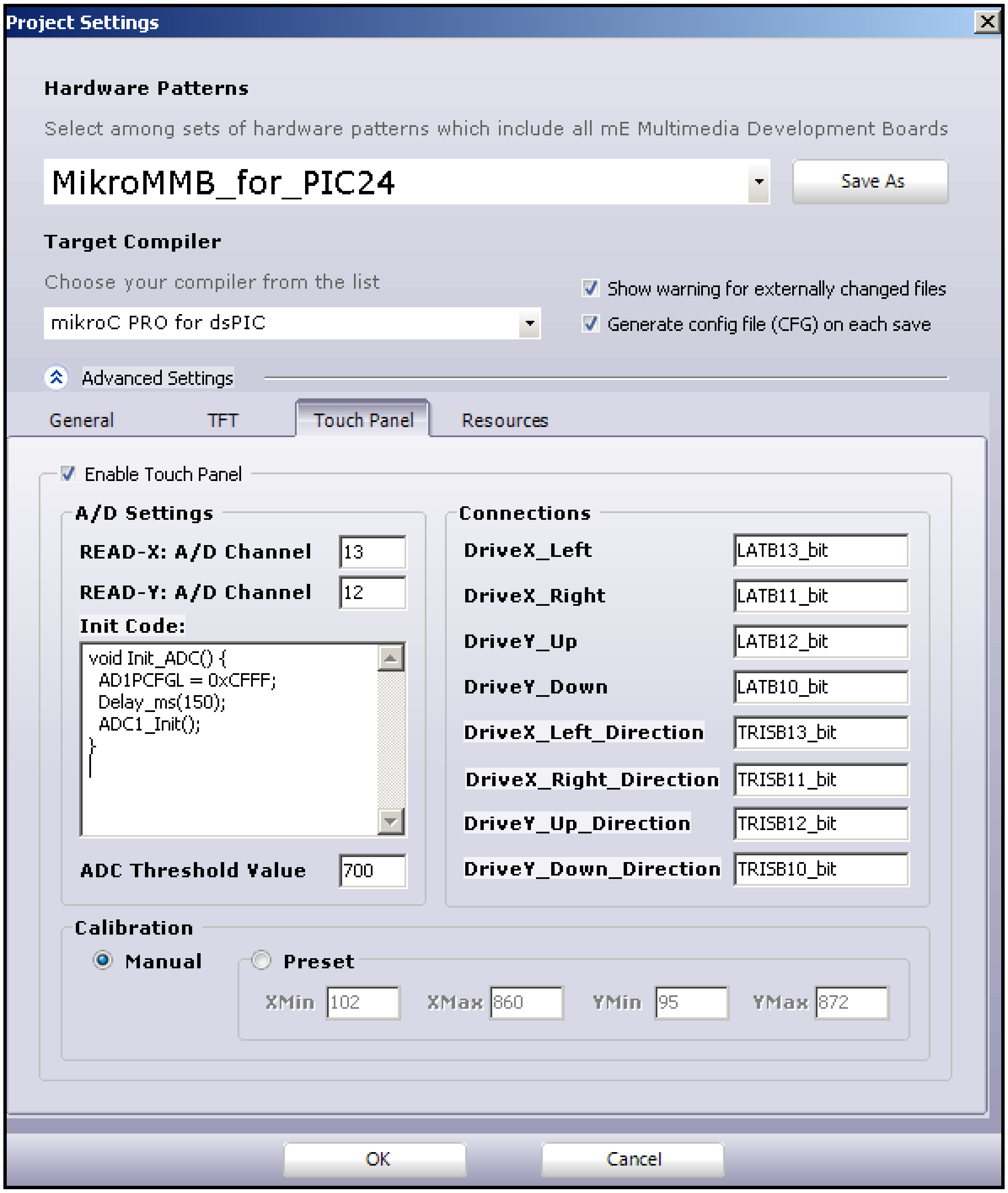
General | TFT | Touch Panel | Resources

Connections

This section allows you to manually set each connection line of your display.
Control and Data line connections vary depending on your target hardware.

TFT_DataPort	LATE	**TFT_DataPort_Direction**	TRISE
TFT_RST	LATC1_bit	**TFT_RST_Direction**	TRISC1_bit
TFT_BLED	LATD2_bit	**TFT_BLED_Direction**	TRISD2_bit
TFT_RS	LATB15_bit	**TFT_RS_Direction**	TRISB15_bit
TFT_CS	LATF12_bit	**TFT_CS_Direction**	TRISF12_bit
TFT_RD	LATD5_bit	**TFT_RD_Direction**	TRISD5_bit
TFT_WR	LATD4_bit	**TFT_WR_Direction**	TRISD4_bit

Comment out the AD1PCFGL reference here or change it in the generated code.



Hardware Patterns

Select among sets of hardware patterns which include all mE Multimedia Development Boards

MikroMMB_for_PIC24 [Save As]

Target Compiler

Choose your compiler from the list

mikroC PRO for dsPIC

[✓] Show warning for externally changed files

[✓] Generate config file (CFG) on each save

Advanced Settings

General | TFT | Touch Panel | Resources

[✓] Enable Touch Panel

A/D Settings

READ-X: A/D Channel	13
READ-Y: A/D Channel	12

Init Code:

```
void Init_ADC() {
  AD1PCFGL = 0xCFFF;
  Delay_ms(150);
  ADC1_Init();
}
```

ADC Threshold Value: 700

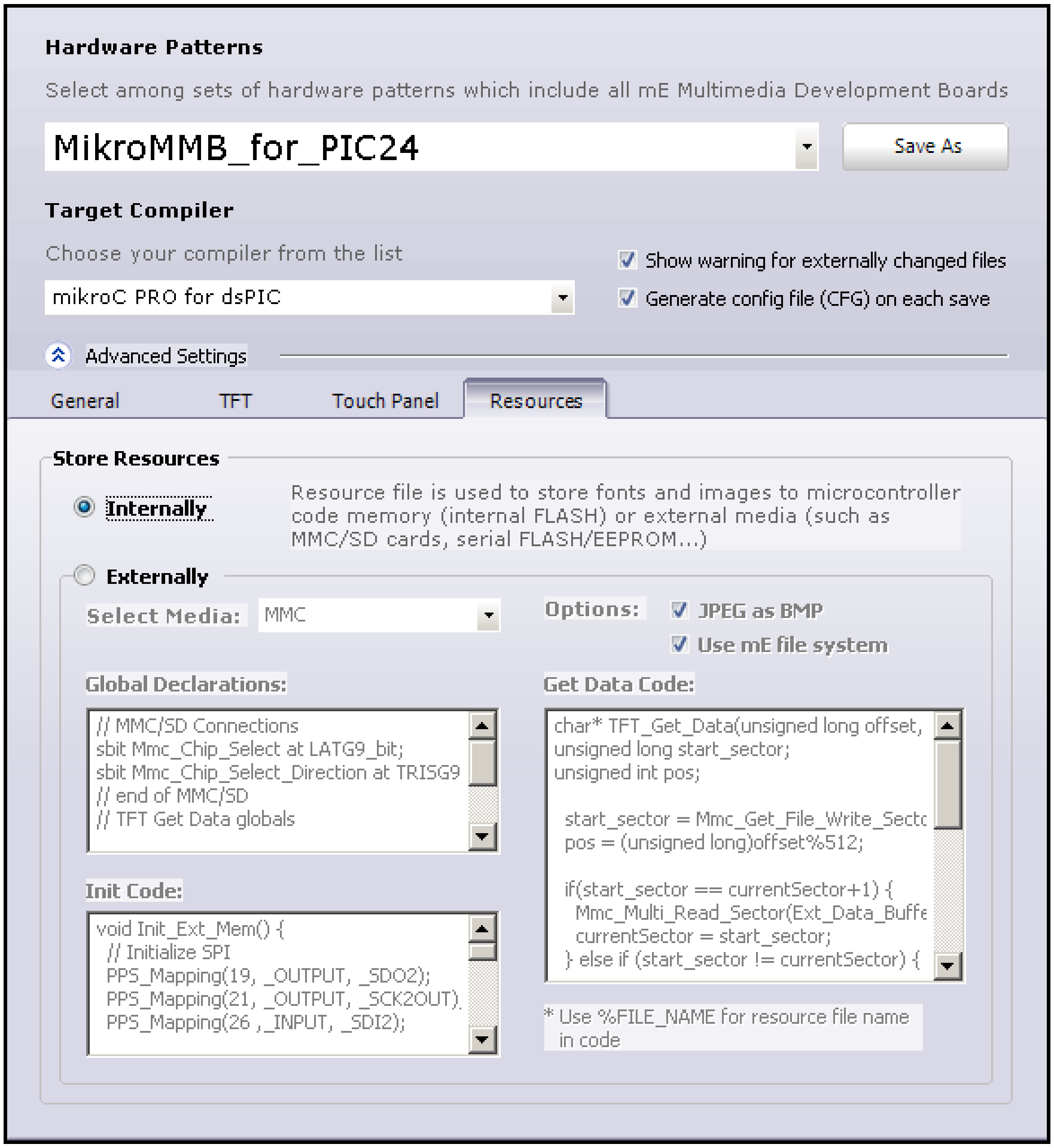
Connections

DriveX_Left	LATB13_bit
DriveX_Right	LATB11_bit
DriveY_Up	LATB12_bit
DriveY_Down	LATB10_bit
DriveX_Left_Direction	TRISB13_bit
DriveX_Right_Direction	TRISB11_bit
DriveY_Up_Direction	TRISB12_bit
DriveY_Down_Direction	TRISB10_bit

Calibration

[✓ Manual] Preset

XMin	XMax	YMin	YMax
102	860	95	872



Hardware Patterns

Select among sets of hardware patterns which include all mE Multimedia Development Boards

MikroMMB_for_PIC24

Save As

Target Compiler

Choose your compiler from the list

mikroC PRO for dsPIC

☑ Show warning for externally changed files

☑ Generate config file (CFG) on each save

Advanced Settings

General | TFT | Touch Panel | Resources

Store Resources

[✓ Internally] Externally

Resource file is used to store fonts and images to microcontroller code memory (internal FLASH) or external media (such as MMC/SD cards, serial FLASH/EEPROM...)

Externally

Select Media: MMC

Options: ☑ JPEG as BMP ☑ Use mE file system

Global Declarations:

```
// MMC/SD Connections
sbit Mmc_Chip_Select at LATG9_bit;
sbit Mmc_Chip_Select_Direction at TRISG9
// end of MMC/SD
// TFT Get Data globals
```

Init Code:

```
void Init_Ext_Mem() {
  // Initialize SPI
  PPS_Mapping(19, _OUTPUT, _SDO2);
  PPS_Mapping(21, _OUTPUT, _SCK2OUT)
  PPS_Mapping(26 , _INPUT, _SDI2);
```

Get Data Code:

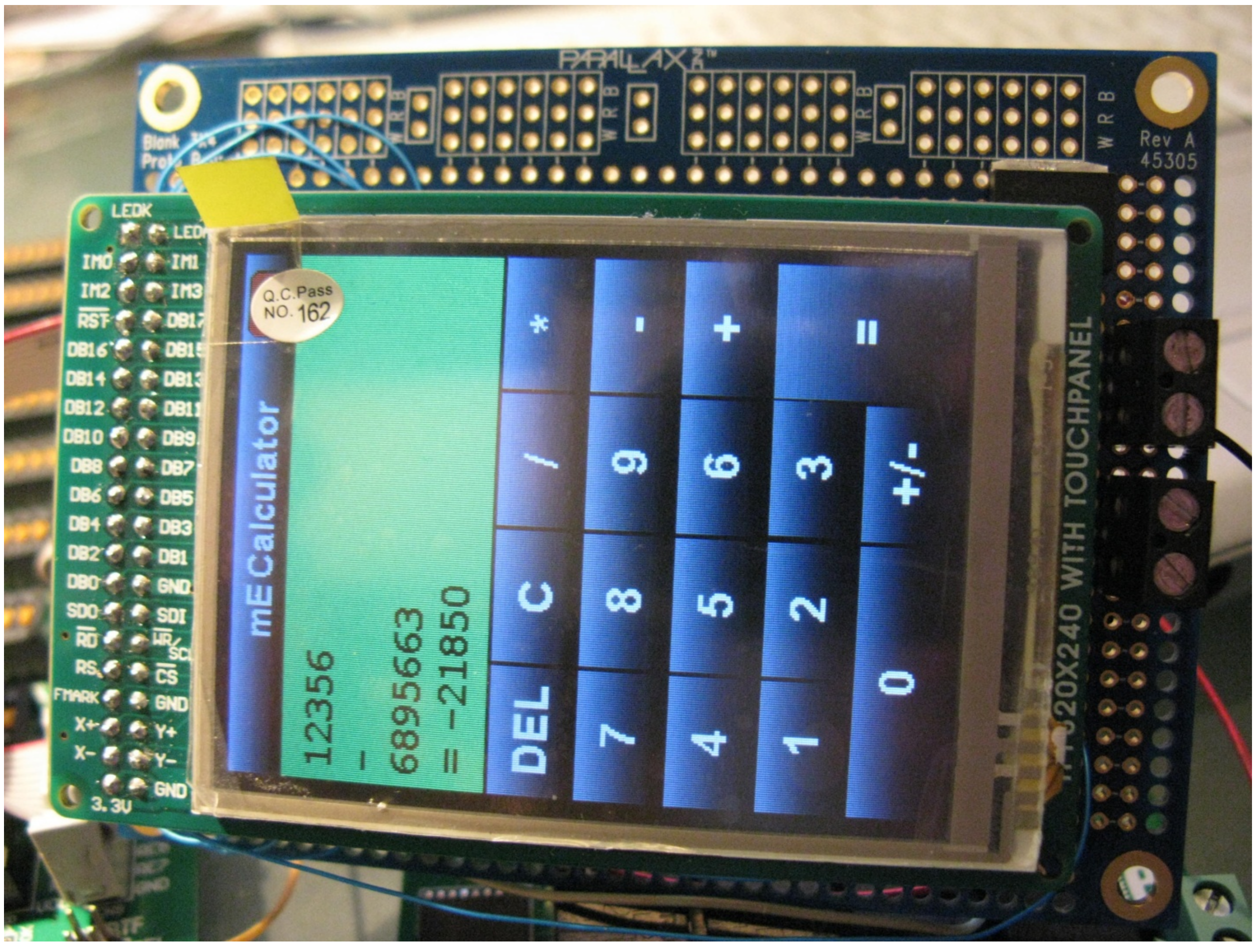
```
char* TFT_Get_Data(unsigned long offset,
unsigned long start_sector;
unsigned int pos;

  start_sector = Mmc_Get_File_Write_Sectc
  pos = (unsigned long)offset%512;

  if(start_sector == currentSector+1) {
    Mmc_Multi_Read_Sector(Ext_Data_Buffe
    currentSector = start_sector;
  } else if (start_sector != currentSector) {
```

* Use %FILE_NAME for resource file name in code

Success.



If there are any errors in this document, leave a message at www.nlcpr.com and I will fix it. I'd really hate to burn up anyone's TFT or development board.