CCS does not monitor this forum on a regular basis.

Please do not post bug Reports on this forum. Send them to support@ccsinfo.com

A Driver for the XPT2046 or TI TSC2046 Touch Controllers

Author | Message

cbarberis **A Driver for the XPT2046 or TI TSC2046 Touch Controllers**

Posted: Mon Aug 10, 2020 3:47 pm quote

Joined: 01 Oct 2003
Posts: 172
Location: Punta Gorda, Florida USA

profile pm email

This is a functional driver you can use on the XPT2046 or the TI TSC2046, it may also work with the ADS7843 but I cannot attest to that particular device.
My testing was done on a cheap ILI9341 (240 X 320) TFT display that has this touch screen the module is excellent and only costs $10 from Amazon. I did not include here the driver and libraries for the TFT display this is just for the touch controller and you can easily test it via the serial port.

Code:

```
////////////////////////////////////////////////////////////////////////////////
//  Interfacing dsPIC33FJ128GP802 microcontroller touch display using the XPT2046
//  Touch Controller which is the same as TSC2046 from TI,This quick test does not
//  include the ILI9341 TFT LCD display just a basic functionality test using the
//  serial port to report X,Y and Z touch values
//  Carlos Barberis
//  August 10, 2020
////////////////////////////////////////////////////////////////////////////////
#include <33FJ128GP802.h>
#device ADC=12
#device ICSP=1
//#device ICD=TRUE

#FUSES NOWDT                    //No Watch Dog Timer
//#FUSES WDT //be careful using WDT make sure is timed correctly otherwise it will never work
#FUSES WPRES32                  //Watch Dog Timer PreScalar 1:32
#FUSES WPOSTS10                 //Watch Dog Timer PostScalar 1:512
#FUSES CKSFSM                   //Clock Switching is enabled, fail Safe clock monitor is enabled
#FUSES NODEBUG
//#FUSES DEBUG
#use delay(internal = 64MHz)


////////////////////// Connector Wiring for Touch Module ///////////////////////
// TOUCH-MISO -->   PIN_B4
// TOUCH-IRQ  -->   PIN_B7
// TOUCH-SCK  -->   PIN_B6
// TOUCH-MOSI -->   PIN_B5
// TOUCH-CS   -->   PIN_B8
////////////////////////////////////////////////////////////////////////////////
// define XPT2046 Touch module pin connections for this particular setup
#define TOUCH_CS    PIN_B8   // chip select pin
#define TOUCH_IRQ   PIN_B7   // Interrupt request
#pin_select SDI2=PIN_B4
#pin_select SDO2=PIN_B5
#pin_select SCK2OUT=PIN_B6
#pin_select U1TX=PIN_B14
#pin_select U1RX=PIN_B15

#use spi(MASTER, SPI2, MODE=0, BITS=8, baud=2000000,stream = TOUCH)
#use rs232(UART1, baud=9600,restart_wdt, errors,stream= USART1) // connected to USB serial



 /// Function Prototypes ///////
 void ext0_isr(void);
```

```
 void ReadTouch_XYZData(void);
 void ReadTouch_XData(void);
 void ReadTouch_YData(void);


 unsigned int X_Val,Y_Val;
 unsigned int Z_Val,Z1,Z2;
 unsigned int xraw,yraw;
 unsigned int ScreenRotation = 1; //screen rotation
 int1 TOUCH_FLAG = 0;

 #define X_COMMAND_BYTE  0xD0  // get X values
 #define Y_COMMAND_BYTE  0x90  // get Y values
 #define Z1_COMMAND_BYTE 0xB1 // get Z1 value
 #define Z2_COMMAND_BYTE 0xC1 // get Z2 value
 #define MSB_BIT_MASK    0x7F // Bit mask for MSB byte
 #define Z_THRESHOLD     400




#int_ext0
void ext0_isr() {
   TOUCH_FLAG = 1;
}




void ReadTouch_XData()  {     // read 12 bit data from SPI bus after sending a command byte

  unsigned int8 MSB =0,LSB =0;

   output_low(PIN_B8);
   delay_us(2);
   SPI_XFER(TOUCH,X_COMMAND_BYTE); // send command byte twice to dump first reading
   SPI_XFER(TOUCH,X_COMMAND_BYTE); // send command byte
   MSB = SPI_XFER(TOUCH,0);
   LSB = SPI_XFER(TOUCH,0);
   output_high(PIN_B8);
   xraw = (((MSB & MSB_BIT_MASK) << 8)+ LSB) >> 3;

  }



void ReadTouch_YData()  {     // read 12 bit data from SPI bus after sending a command byte

  unsigned int8 MSB =0,LSB =0;

   output_low(PIN_B8);
   delay_us(2);
   SPI_XFER(TOUCH,Y_COMMAND_BYTE); // send command byte
   MSB = SPI_XFER(TOUCH,0);
   LSB = SPI_XFER(TOUCH,0);
   output_high(PIN_B8);
   yraw = (((MSB & MSB_BIT_MASK) << 8)+ LSB) >> 3;

  }



void ReadTouch_XYZData()  {  // read 12 bit data from SPI bus after sending a command byte for
X,Y,Z1&Z2 measurements


  unsigned int8 MSB =0,LSB =0;


   output_low(PIN_B8);
   delay_us(2);
   ////////////////////Gather Z pressure readings////////////////////////////////////
   SPI_XFER(TOUCH,Z1_COMMAND_BYTE); // send command byte
   MSB = SPI_XFER(TOUCH,0);
   LSB = SPI_XFER(TOUCH,0);
   Z1 = (((MSB & MSB_BIT_MASK) << 8)+ LSB) >> 3;

   SPI_XFER(TOUCH,Z2_COMMAND_BYTE); // send command byte
   MSB = SPI_XFER(TOUCH,0);
   LSB = SPI_XFER(TOUCH,0);
   Z2 = (((MSB & MSB_BIT_MASK) << 8)+ LSB) >> 3;
   Z_Val =  Z1 + 4095;
   Z_Val -= Z2;
   ////////////////////////////////////////////////////////
   if(Z_Val >= Z_THRESHOLD) {  // if we have more than the pressure threshold then read X&Y data
```

```
   SPI_XFER(TOUCH,X_COMMAND_BYTE); // send command byte
   MSB = SPI_XFER(TOUCH,0);
   LSB = SPI_XFER(TOUCH,0);
   xraw = (((MSB & MSB_BIT_MASK) << 8)+ LSB) >> 3;


   SPI_XFER(TOUCH,Y_COMMAND_BYTE); // send command byte
   MSB = SPI_XFER(TOUCH,0);
   LSB = SPI_XFER(TOUCH,0);;
   output_high(PIN_B8);
   yraw = (((MSB & MSB_BIT_MASK) << 8)+ LSB) >> 3;

 }
      output_high(PIN_B8); // in case Z_Val did not qualify lets bring CS high

      switch (ScreenRotation) {     /// We need to know the display rotation to compensate for
X,Y locations
      case 1:
      X_Val = 4095 - yraw;
      Y_Val = 4095 - xraw;
      break;
      case 2:
      X_Val = xraw;
      Y_Val = yraw;
      break;
      case 3:
      X_Val = yraw;
      Y_Val = 4095 - xraw;
      break;
      default: // 4
      X_Val = 4095 - xraw;
      Y_Val = 4095 - yraw;
    }

  }


////////////////////////////////// TEST  PROGRAM
/////////////////////////////////////////////////


void main() {

 set_tris_b (0x9090);
 set_pullup(true, PIN_B7);
 enable_interrupts(INT_EXT0);
 ext_int_edge( H_TO_L );
 enable_interrupts(INTR_GLOBAL);
 ReadTouch_XData(); // just to init once may not be really needed

while(1) {

 if(TOUCH_FLAG)
   {
   while(!input(PIN_B7))  {
   ReadTouch_XYZData();
   delay_ms(100);
   printf("XposRaw value = %lu , ",xraw);
   printf("YposRaw value = %lu , ",yraw);
   printf("Xpos value = %lu , ",X_Val);
   printf("Ypos value = %lu , ",Y_Val);
   printf("Z value = %lu \r\n",Z_Val);
   }
   TOUCH_FLAG = 0;}

   delay_ms(100);

    }
}

//////////////////////////////////////////  END OF FILE
 /////////////////////////////////////////////////////
```

Enjoy!!!

Display posts from previous: All Posts | Oldest First | Go

newtopic postreply CCS Forum Index -> Code Library

All times are GMT - 6 Hours

Page 1 of 1

Jump to: Code Library | Go

You **cannot** post new topics in this forum
You **cannot** reply to topics in this forum
You **cannot** edit your posts in this forum

You **cannot** delete your posts in this forum
You **cannot** vote in polls in this forum