Adafruit TSC2046 SPI Resistive Touch Screen Controller

Created by Liz Clark


Last updated on 2023-08-29 04:57:44 PM EDT
Table of Contents

Overview 3

Pinouts 6
• Power Pins
• Input Pins
• SPI Pins
• Interrupt Pin
• ADC Pins
• VRef Pin

Arduino 8
• Wiring
• Library Installation
• Load Example

Arduino Docs 12

Downloads 12
• Files
• Schematic and Fab Print
Overview

Getting touchy performance with your screen's touch screen? Resistive touch screens are incredibly popular as overlays to TFT and LCD displays. Only problem is they require a bunch of analog pins and you have to keep polling them since the overlays themselves are basically just big potentiometers. If your microcontroller doesn't have analog inputs, or maybe you want just a way more elegant controller, the TSC2046 is a nice way to solve that problem.

This breakout board features the TSC2046, which has an easy-to-use SPI interface available. There is also an interrupt pin that you can use to indicate when a touch has
been detected to your microcontroller or microcomputer. It can be powered from 3-5V, so it's safe to use with 3V or 5V logic. It's a nicely designed chip and has very stable precise readings. We found it's also a lot faster than trying to do all the readings on an Arduino.

For the screens that have 1mm pitch FPC cables, you can plug the cable right into the connector. The majority of medium/large touchscreens have that kind of connector. If you have another kind of touch screen, the four X/Y contacts are available on 0.1" pitch breakouts so you can hand-solder or wire them.

Getting started is super easy with our simple TSC2046 Arduino library (). Plug any 1mm-pitch 4-wire resistive touchscreen to the on-board FPC connector, then use the
library example to read touch points with X, Y and Z (pressure) results returned instantaneously. There's an IRQ pin that will drop low when a touch is detected, you can use that to reduce the SPI polling.

We also break out the Busy pin, that lets you know when the chip is still reading, and the Vref pin if you want to read or change the default 2.5V reference voltage. There's also two extra ADC pins you can read, Aux (0V - VRef) and VBat (0V - 2*VRef).
Pinouts

Power Pins

- **Vin** - This is the power pin. It can be powered from 3-5V, so it's safe to use with 3V or 5V logic. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 3V microcontroller like a Feather RP2040, use 3V, or for a 5V microcontroller like Arduino, use 5V.
- **GND** - This is common ground for power and logic.
Input Pins

- **FPC Connector** - FPC socket for a 1mm-pitch 4-wire resistive touchscreen located at the top of the board. This reads the X/Y contacts directly from a 4-wire FPC cable.
- **Y-, X-, Y+, X+** - These are the X/Y contacts available on 0.1" pitch breakouts. If you have another kind of touch screen that does not have a 1mm pitch FPC cable then you can solder directly to these points.

SPI Pins

- **MOSI** - This is the SPI MOSI (Microcontroller Out / Serial In) pin. It is used to send data from the microcontroller to the TSC2046.
- **MISO** - This is the SPI MISO (Microcontroller In / Serial Out) pin. It's used for sending data from the TSC2046 to the microcontroller.
- **SCK** - This is the SPI clock input pin.
- **CS** - This is the Chip Select pin.
- **BUSY** - This is the SPI Busy pin. It lets you know when the TSC2046 is still reading.

Interrupt Pin

- **IRQ** - This is the interrupt output pin. It will drop low when a touch is detected. You can use this to reduce the SPI polling.

ADC Pins

- **Vbat** - The Vbat pin can be used to measure some other voltage external to the touchscreen. You can measure voltages (inclusively) between 0V - 2*VRef.
- **AUX** - The AUX pin can be used to measure some other voltage external to the touchscreen. Unlike Vbat, however, the maximum voltage this pin can measure is 0V - VRef.

VRef Pin

- **VRef** - The VRef pin can be used to override the "reference voltage" that the TSC2046 uses to measure other voltages. The TSC2046 has an internal reference voltage of 2.5V that by default is used when measuring temperature, Vbat, and AUX voltages. Connecting a 5V supply to the VRef pin instead will
allow you to use that as the reference voltage, which will increase the accuracy of voltage reads for temperature, Vbat, and AUX; and will also increase the range of voltage reads for AUX.

However, if you connect to the VRef pin, you must connect VRef to the same voltage you connected Vin to.

If you connect to the VRef pin, you must connect VRef to the same voltage you connected Vin to.

Arduino

Using the TSC2046 with Arduino involves wiring up the controller to your Arduino-compatible microcontroller, installing the Adafruit TSC2046 library and running the provided example code.

Wiring

Wire as shown for a 5V board like an Uno. If you are using a 3V board, like an Adafruit Feather, wire the board's 3V pin to the TSC2046 VIN.

Here is an Adafruit Metro wired up to the TSC2046:

- Board 5V to breakout Vin (red wire)
- Board GND to breakout GND (black wire)
- Board pin 13 to breakout SCK (blue wire)
- Board pin 12 to breakout MISO (green wire)
- Board pin 11 to breakout MOSI (yellow wire)
- Board pin 10 to breakout CS (purple wire)

Library Installation

You can install the TSC2046 library for Arduino using the Library Manager in the Arduino IDE.
Click the Manage Libraries ... menu item, search for TSC2046, and select the Adafruit TSC2046 library:

If asked about dependencies, click "Install all".

If the "Dependencies" window does not come up, then you already have the dependencies installed.

If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

Load Example

Open up File -> Examples -> Adafruit TSC2046 -> touchscreendemo and upload to your Arduino wired to the breakout.
#include <Adafruit_TSC2046.h>

/* To use this library you will also need the [Adafruit BusIO]
library and include it in your libraries folder.
Instructions for how to install libraries in the Arduino IDE can be found at
https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-installing-a-library

Connections
==========
Connect board SCK to breakout SCK.
Connect board CIPO (MISO) to breakout MOSI.
Connect board COPI (MOSI) to breakout MOSI.
Connect board SS (CS) to breakout CS
You can instead use a different digital pin, but you must pass the pin
number as the third argument to Adafruit_TSC2046::begin().
IF you have a board with 5V-logic:
Connect board 5V to breakout Vin
IF you have a board with 3.3V-logic:
Connect board 3V3 to breakout Vin
Connect board GND to breakout GND

Measurements
============
Set your multimeter to measure resistance, place one probe on the pin-hole
labeled "X-" on the TSC2046 breakout, and the other probe on the pin-hole
labeled "X+". Your multimeter should show you a number in ohms (Ω), the unit
for resistance. Pass that number as the first argument to
Adafruit_TSC2046::begin().

This example uses a 400Ω X-resistance.
*/

Adafruit_TSC2046 touchscreen;
#define TSC_CS         10  // chip select pin
#define TS_RESISTANCE 400  // For a 400Ω resistance across X- and X+

void setup() {
  Serial.begin(115200);
  // Wait till serial connection opens for native USB devices
  while (!Serial) {
    delay(10);
  }
  Serial.println("Adafruit TSC2046 touchscreen demo");
  touchscreen.begin(TSC_CS, &SPI, TS_RESISTANCE);
  touchscreen.enableInterrupts(true);
}

// Prints the three coordinates in nice columns.
void displayTouchPoint(TSPoint point) {
  Serial.print(point.x);
  Serial.print("    ");
  Serial.print(point.y);
  Serial.print("    ");
  // Z measures the pressure; the value for Z *decreases* as the physical
  // pressure *increases*.
  Serial.print(point.z, 2);
  Serial.print(" ohms\n");
}

void loop() {
  delay(50); // Add delay to avoid overloading the serial monitor
// Check that the touchscreen is being touched at all before getting
// coordinates. Otherwise the coordinate values we get will be garbage.
if (touchscreen.isTouched()) {
    TSPoint point = touchscreen.getPoint();
displayTouchPoint(point);
} else {
    // We can also monitor the ambient temperature around the chip,
    // in case you want to do something like check for overheating.
    // So if the touchscreen isn't being touched right now,
    // let's measure the temperature and print it over serial.
    float tempC = touchscreen.readTemperatureC();
    Serial.print(tempC);
    Serial.print(" C\t\t");
    Serial.print("Aux: ");
    Serial.print(touchscreen.readAuxiliaryVoltage());
    Serial.print(" V\t");
    Serial.print("Bat: ");
    Serial.print(touchscreen.readBatteryVoltage());
    Serial.println(" V\n");
}

Connect a resistive touchscreen to the TSC2046. Upload the sketch to your board
and open up the Serial Monitor (Tools -> Serial Monitor) at 115200 baud. If you touch
the touchscreen, you should see the values read from the touchscreen being printed
out.

The first number is the X coordinate, the second number is the Y coordinate and the
last number is the Z resistance measurement in ohms. This lets you know how much
pressure is being exerted on the touchscreen.

Additionally, the ambient temperature around the TSC2046 chip, voltage reading on
the AUX pin and voltage reading from the Vbat pin are also printed out.
Downloads

Files

- TSC2046 Datasheet ()
- EagleCAD PCB files on GitHub ()
- Fritzing object in the Adafruit Fritzing Library ()

Schematic and Fab Print