XYZ Interactive Technologies, Using the ZX SparkFun sensor for Arduino

Introduction

XYZ Interactive Technologies has developed introductory ZX Sensor (in conjunction with SparkFun Electronics Inc.) herein referred to as the ZX SF sensor that can be used for rapid proof of concept or for sensor related applications. This document describes the setup and application of this sensor for the Arduino UNO platform and using the I2C communication protocol.

Getting Arduino

Arduino is a common platform for electronic enthusiasts interested in an inexpensive but easy-to-program environment. The simplicity of Arduino and limit-less projects available make it the most popular brand for enthusiast projects. Arduino software can be obtained at www.arduino.cc. The Arduino software is free and may be operated without restrictions. Arduino has many products but this document will focus on Arduino UNO as a starting application and use the I2C interface to get any user started with the sensor.

Getting the ZX SparkFun sensor

XYZ Interactive has developed the ZX SF sensor for many applications that include robotics and toy concept development. The sensor uses the trademark GestureSense and has a distinct and patented method of determining proximity of objects close to the sensor (within 40 cm), and can also be used to determine gestures done by human hand-waving in proximity to the sensor.

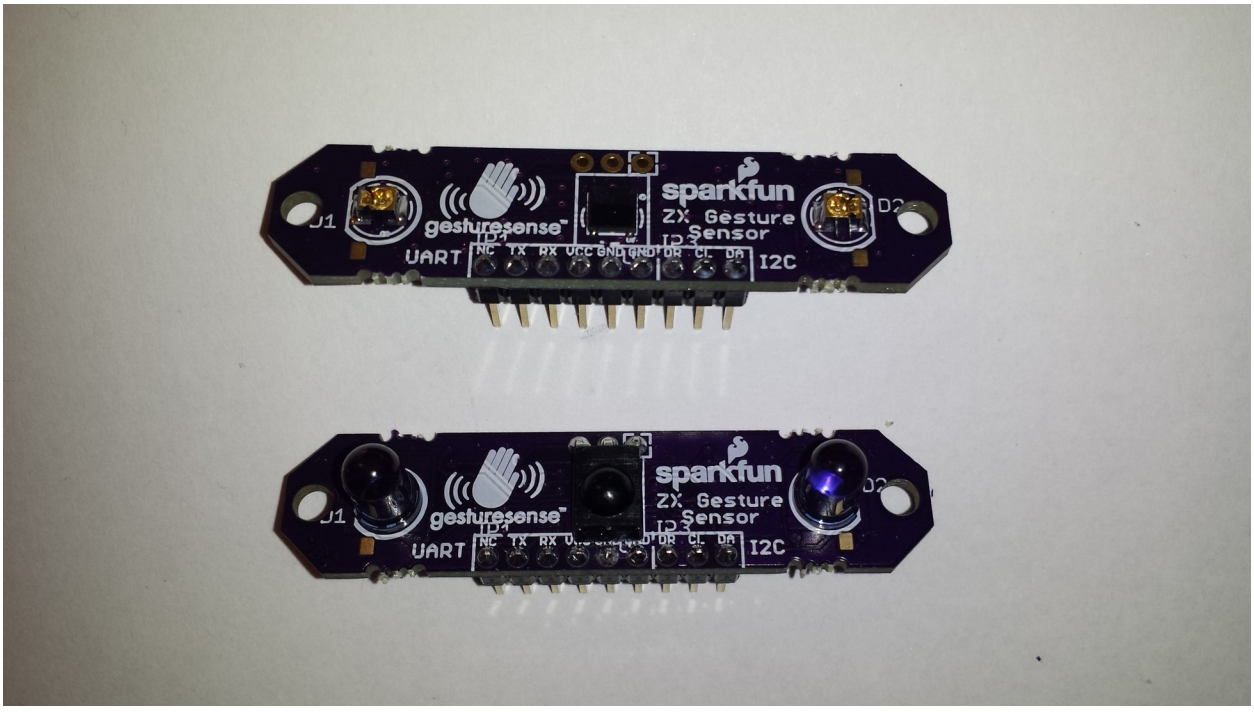

Figure 1: The ZX SF sensor as SMT (surface-mount component) and a TH (through-hole component) designs.

The ZXSF sensor is not yet available for sale, but will be available at SparkFun when production is ready. However, the sensor is available from XYZ Interactive as a prototype sensor for purchase.

The ZX SF sensor pin-out

Looking at Figure 2 illustrates the pin-out of the sensor as a combination of UART and I2C interface pins. This pin-out was chosen to allow multiple application to operate with a demonstrator program and also have the possibility of interfacing directly with an embedded application. Data is output simultaneously for both sensors and there are no physical or electronic restrictions for the use of each interface protocol (either UART or I2C).

Figure 2: The ZX SF front and pin-out.

The following summarizes the pinout:

1) Pin 1 : NC
2) Pin 2 : TX (UART transmission line)
3) Pin 3 : RX (UART receiving line)
4) Pin 4 : VCC (positive voltage unregulated), should ideally be 5 volts, but 3.3 volts will work.
5) Pin 5 : GND ground.
6) Pin 6 : GND ground.
7) Pin 7 : DR (I2C data ready signal).
8) Pin 8 : CL (or I2C SCL clock signal).
9) Pin 9 : DA (or I2C SDA data signal).

How to use the ZX SF sensor

The ZX SF sensor has a few details about its use for I2C that need to be mentioned.

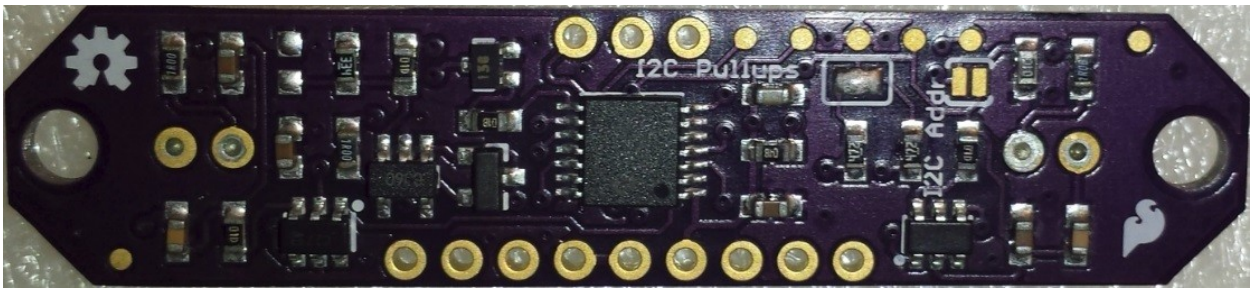

Figure 3: The ZX SF sensor back side.

By observing Figure 3, there are two solder plugs on the board. The solder plugs are for:

1) I2C line pull-ups. There are 4.7 K ohm resistors on the back side of the sensor to allow for voltage pull-ups in case they are needed. Typically Arduino UNO has pull-ups already on the board so the solder plug is not really needed in this case.

2) I2C address change jumper. This jumper allows the I2C slave address to change from a base address of 0x20 when OPEN, to another address of 0x22 when the solder jumper is CLOSED.

Item 2) is useful for sensor configurations that require more than one sensor configured with the Arduino.

Wiring up the Arduino for I2C

Refer to Figure 4 as the wiring diagram for the Arduino to the sensor.

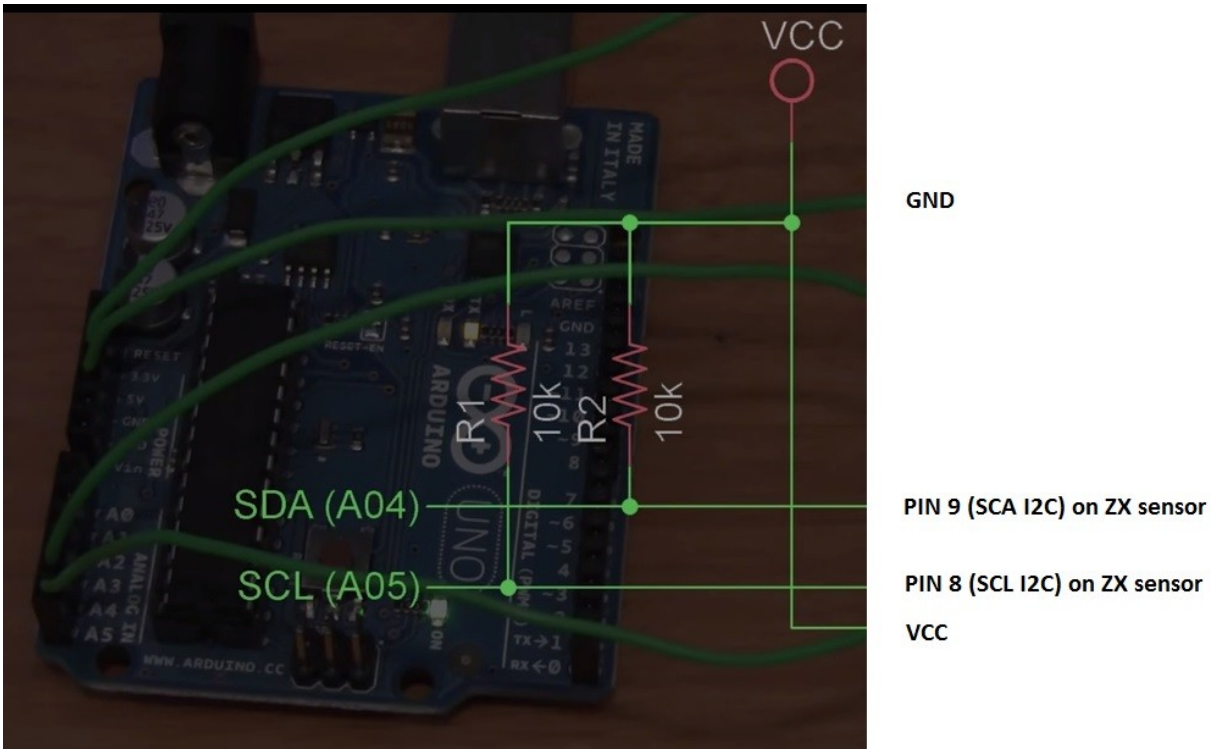

Figure 4: Wiring the Arduino to the ZX SF sensor

The user can therefore connect the SCA and SCL lines directly to the sensor with just the added VCC and GND lines. The user can also use the VCC pull-ups for the I2C lines if necessary, but they are not required for the Arduino UNO.

Creating an Arduino script

Creating an Arduino script for the ZX SF sensor is easy and based on following the steps to add Arduino software to your computer system. Once done, simply start a sketch and begin coding. An example of code to operate with the I2C interface we just wired to the Arduino, is as follows:

```
#include <Wire.h>
void setup()
{
  Wire.begin();
  Serial.begin(9600);
}
void loop(void)
{
  byte x,z,g;
  Wire.beginTransmission(0x10);
  Wire.write(0x08);
  Wire.endTransmission();
  Wire.requestFrom(0x10, 1, true);
  if (Wire.available()) {
    x = Wire.read();
  }
  Wire.beginTransmission(0x10);
  Wire.write(0x0a);
  Wire.endTransmission();
```

```
  Wire.requestFrom(0x10, 1, true);
  if (Wire.available()) {
    z = Wire.read();
  }
  Wire.beginTransmission(0x10);
  Wire.write(0x04);
  Wire.endTransmission();
  Wire.requestFrom(0x10, 1, true);
  if (Wire.available()) {
    g = Wire.read();
  }
  Serial.print("X = ");
  Serial.print(x);
  Serial.print(", Z = ");
  Serial.print(z);
  if (g != 0) {
    Serial.print(" gesture ");
    Serial.print(g);
  }
  Serial.println(".");
  delay(100);  // change this value “100” to different values
}
```

Just copy this script into an Arduino sketch and compile/download into the Arduino directly. This script allows you to read the I2C lines for X, Z, and gesture data to be presented on the host computer interface.

Connecting and operating the sensor

Once connected, the user may turn on the Arduino power by plugging the USB cable into a computer, or powering via the external power jack.

The Arduino will report that the data interface with the sensor is working when it samples and flashes an LED indicating that data sampling using I2C is successful. See Figure 5.

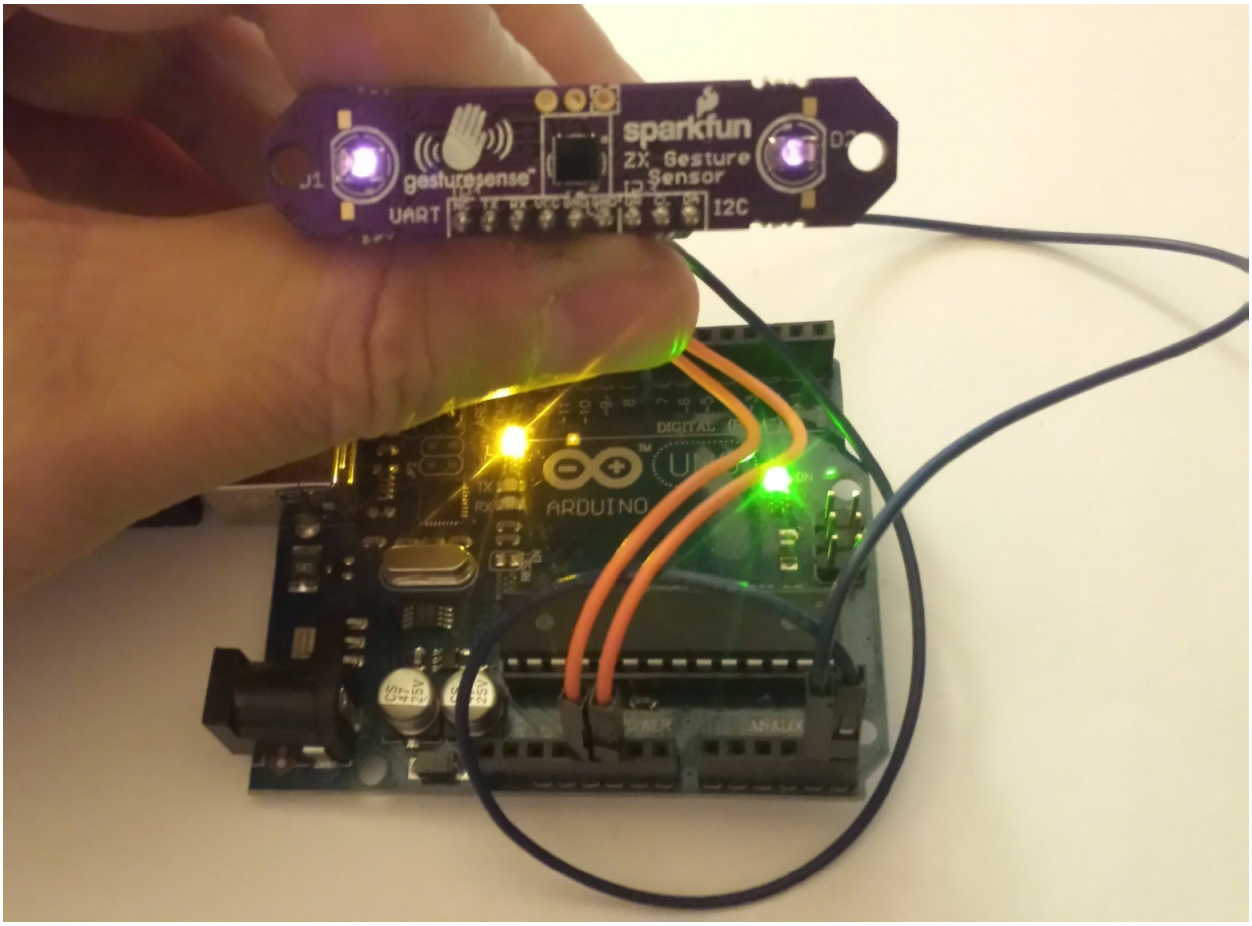

Figure 5 : Arduino rigged and wired for the ZX SF sensor.

Finally, the output of the Arduino sketch will produce a screen of X and Z numbers, and the occasional gestures depending how you swipe across the front face of the sensor. See Figure 6.

In this Arduino sketch, the I2C interface is polled at every 100 milliseconds. The X and Z data is presented even if there is no update to these values in the sensor. However, the gesture data is presented whenever there is a gesture ready to be displayed. This highlights the benefit of I2C as a data interface method as the data is presented at a rate acceptable to the user, and in different forms also required by the user.

You are DONE !!

Good luck in your testing of the ZX SF sensor.

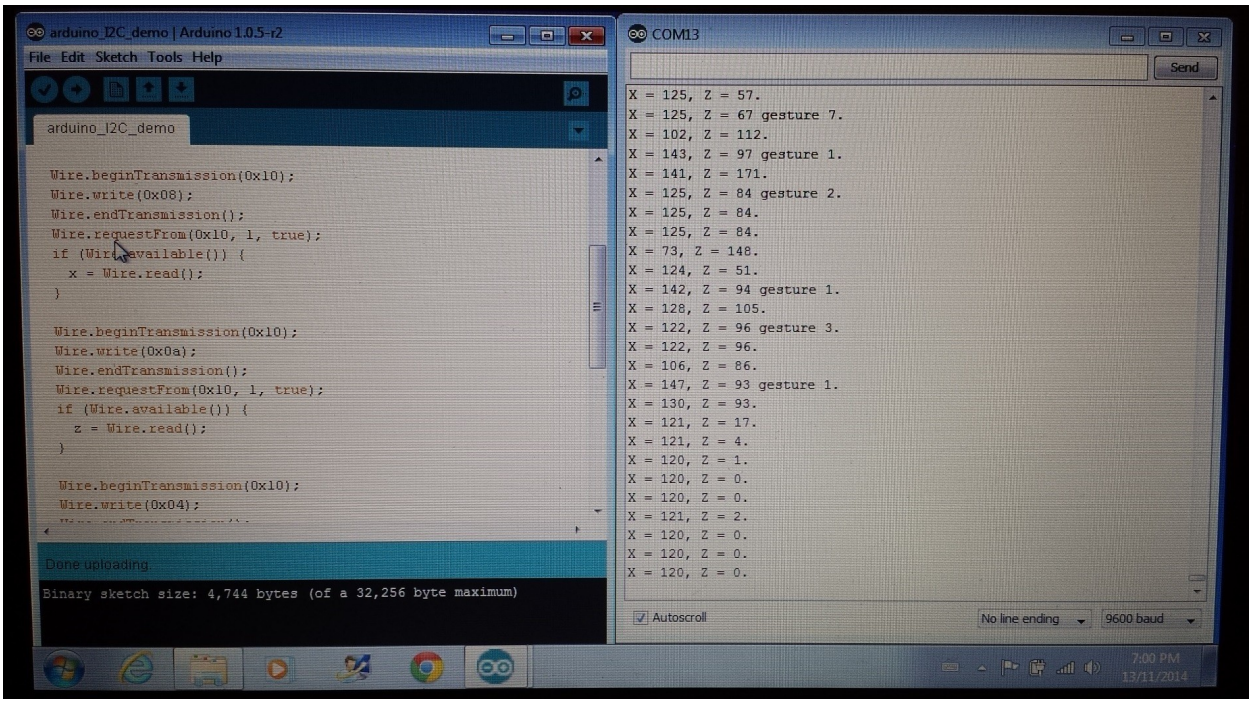

Figure 6 : Arduino sketch output using the I2C data polling method.

INTELLECTUAL PROPERTY CONTACT AND SUPPORT

Physical and Intellectual property:

Please note that this document and ZX SF sensor information kit should be considered confidential materials as per our signed non-disclosure agreement (NDA). If you have received this kit in error, or are not bound by the NDA with XYZ Interactive Technologies Inc., please call +1-416-898-0820 or send an email to demo@xyzinteractive.com. Please also note that the demo kit hardware remains property of XYZ Interactive Technologies and must be returned upon completion of evaluation or at any time if requested by XYZ Interactive Technologies Inc.

External Support

Thank you for this opportunity to bring XYZ Interactive sensor solutions to your company or organisation. If you have any support questions regarding this ZX Sensor Demo Kit or the document, don't hesitate to email me at Andrew@xyzinteractive.com, or contact me directly at +1-519-498-2189.

Andrew Lohbihler, CTO, XYZ Interactive Technologies Inc.