

Interface to a SCP1000-D01 Pressure Sensor

V1.1

Once I received the SCP1000-D01 from Spark Fun Electronics I started my research. I couldn't find much but the links to the manufacturer's website lead me to the documents that I needed.

There are two versions of the SCP1000. The SCP1000-D11 uses an I2C interface and the SCP1000-D01 uses an SPI interface. We will be looking at the SCP1000-D01 in this article. I am not going to get into the actual protocol as that is covered in the manufacturer's documentation. I built a complete DiosPro library so I will be using that in my examples.

Hook Up

The SCP1000 is a 3.3V device so if you are running the DiosPro at 5V you will need to use a 3.3V voltage regulator to power the sensor and a couple resistors to interface to the IO leads as shown in Schematic 1.

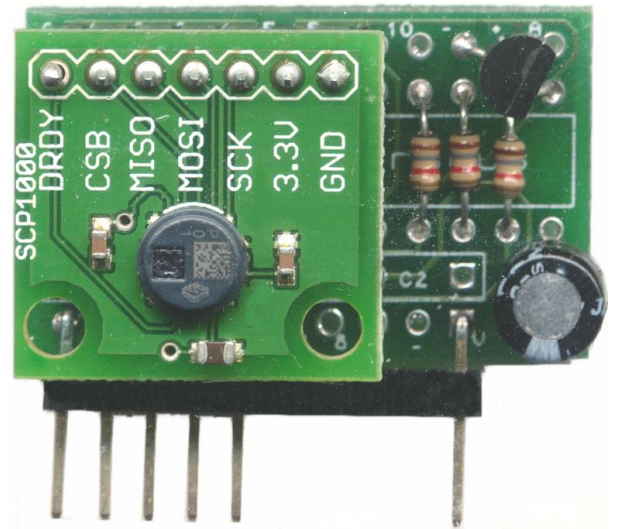


Figure 2

To make it easy you can use the Perseus Carrier Board and to create an inexpensive adapter as shown in Figure 2. The following is the step by step procedure on how to do this.

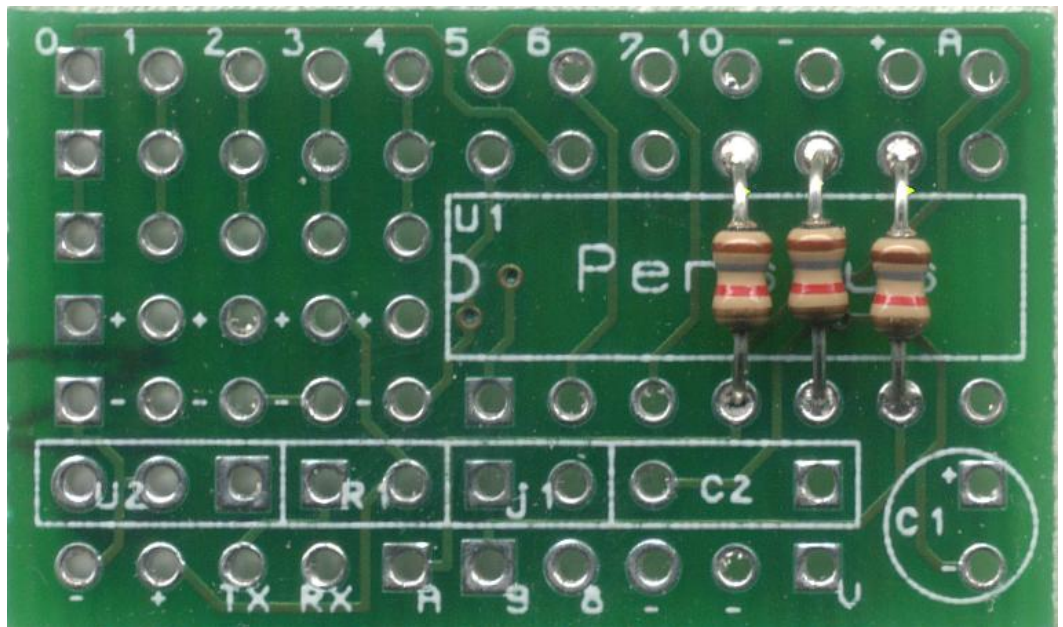
You will need the following components

- Perseus PCB
- 4, 1.8K resistors (1/8W)
- 3, 3.3K resistors (1/8w)
- 100UF Capacitor
- .1UF Capacitor
- 3.3V T092 Voltage regulator
- Female Header (36-Pin, Snapable)
- Male Header (40-Pin, Snapable)

Kronos Robotics sells a universal kit that contains the needed components and is listed at the end of this document.

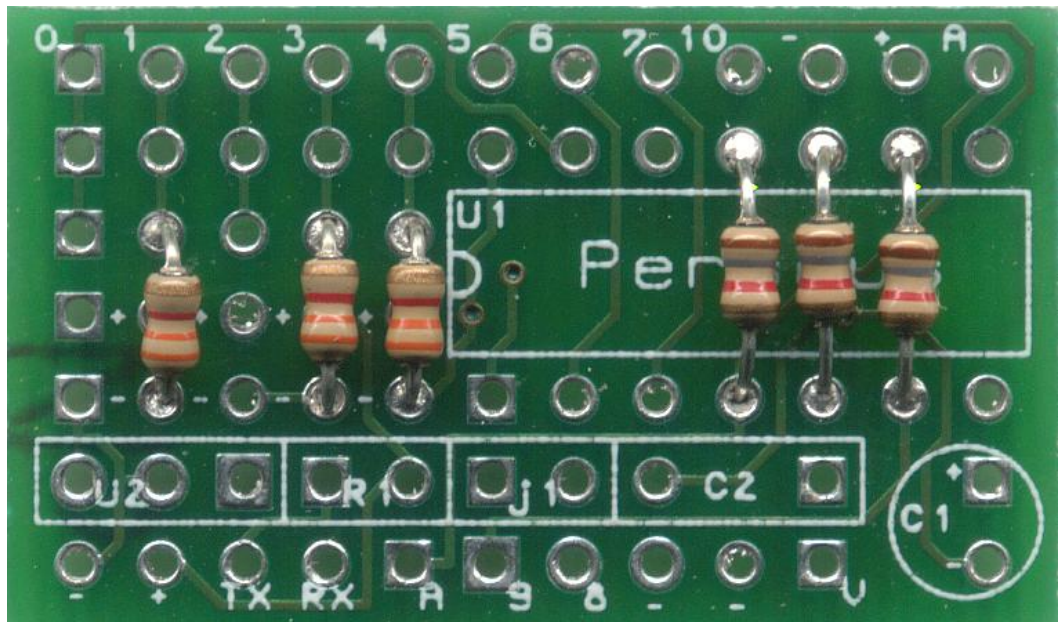
Step 1

Insert and solder three 1.8K resistors (Brown, Grey, Red) into the positions shown.



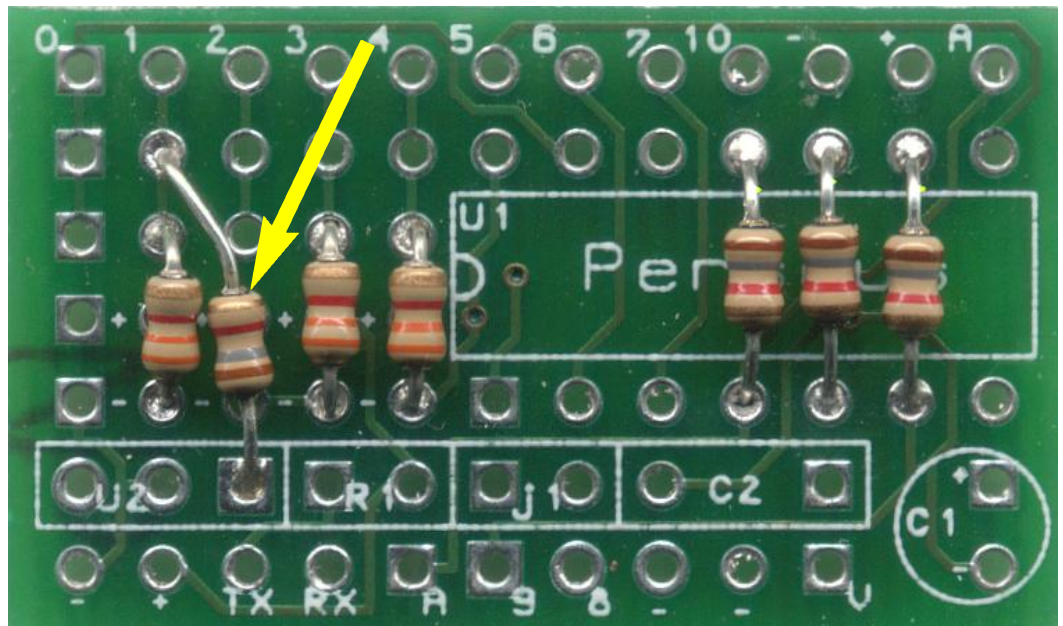
Step 2

Insert and solder three 3.3K resistors (Orange, Orange, Red) into the positions shown.



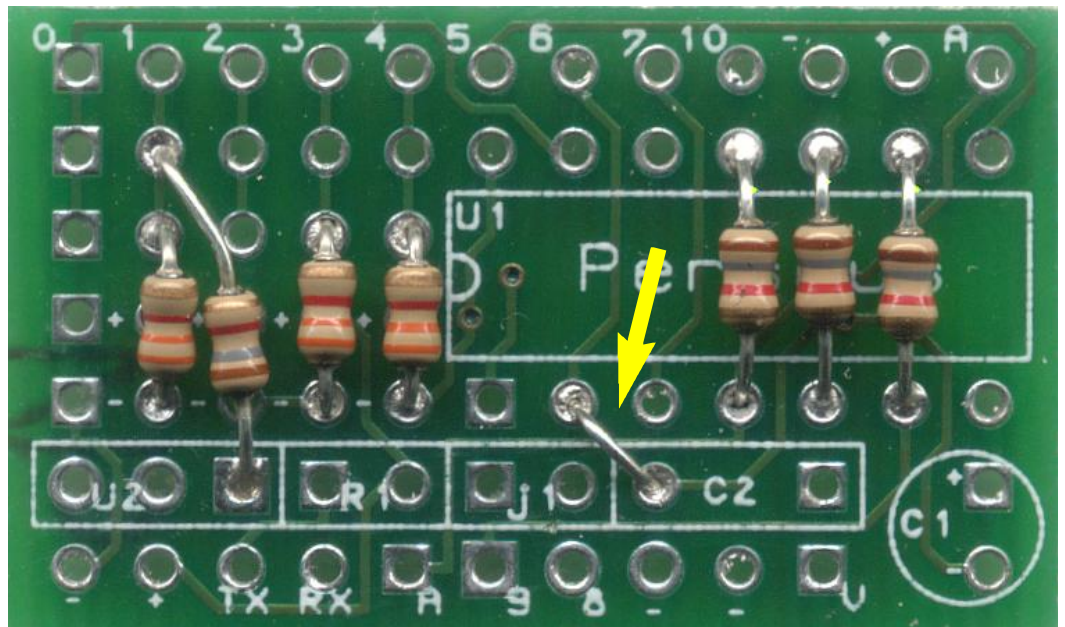
Step 3

Insert and solder a 1.8K resistor (Brown, Grey, Red) into the position shown.



Step 4

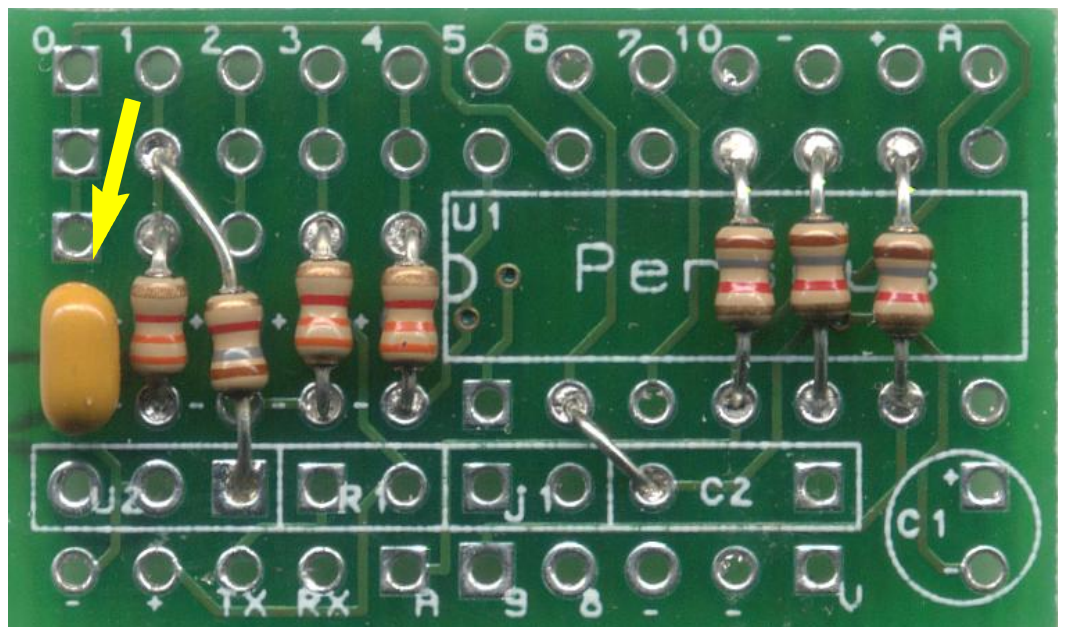
Insert and solder a small jumper into the position shown. You can use one of the leads cut off from a resistor from one of the previous steps.



Step 5

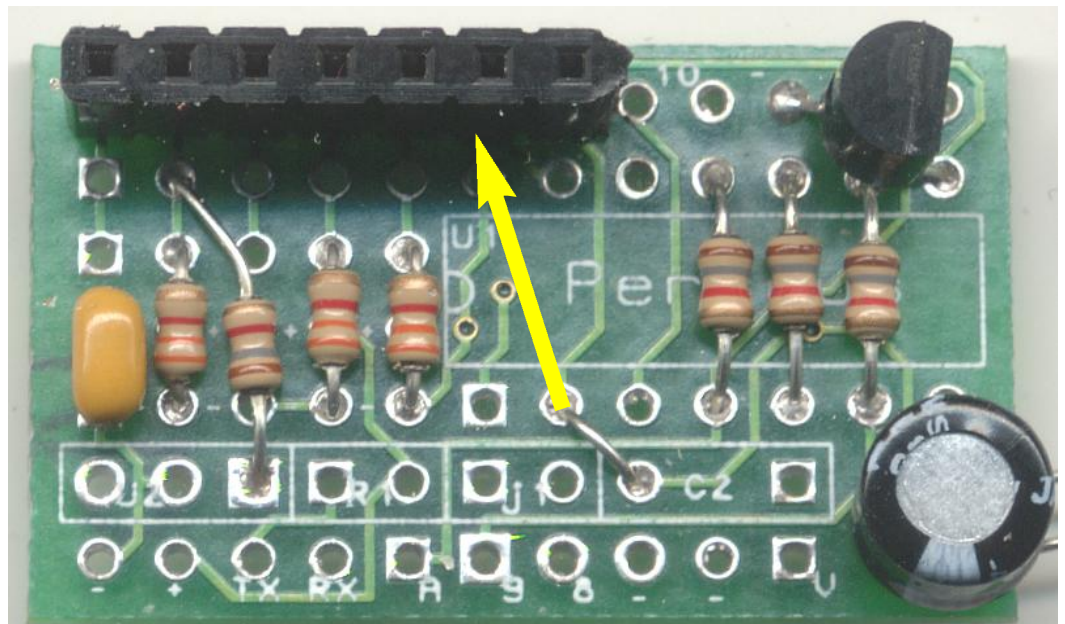
Insert and solder a .1UF capacitor into the position shown.

This will be a tight fit so proceed slowly. Gently pull the leads from the bottom with a pair of needle nose pliers.



Step 6

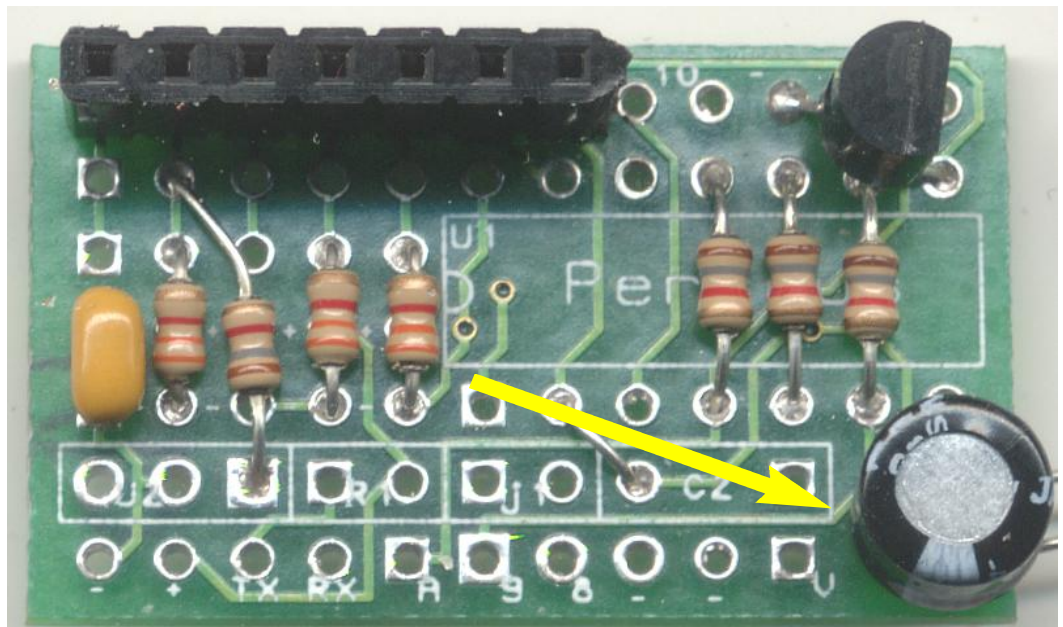
Snap off and solder a 7-pin female header into the position shown



Step 7

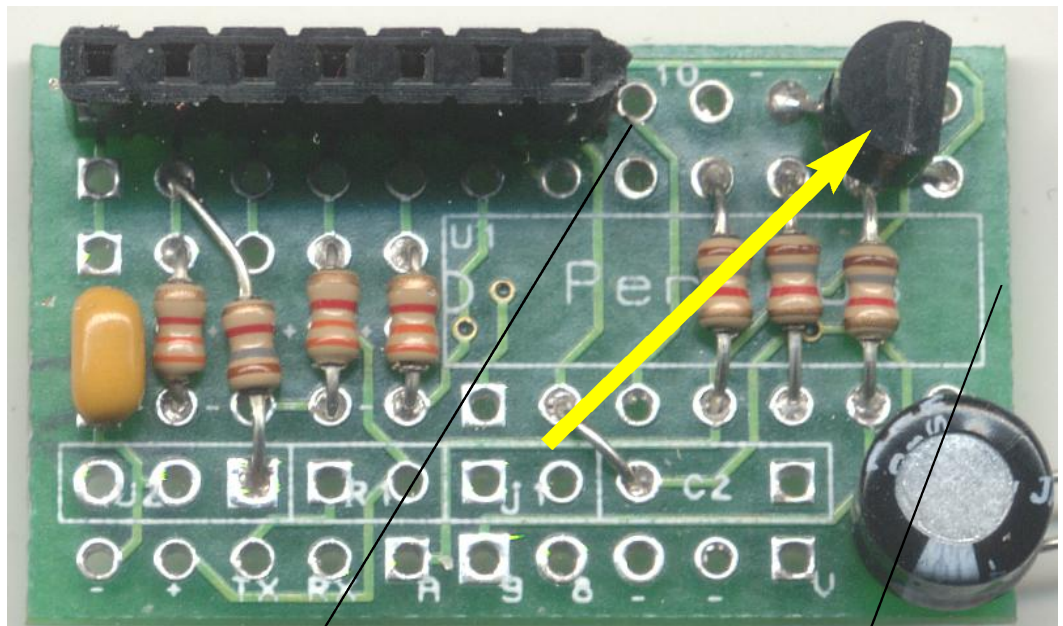
Install a 100UF capacitor into the position marked C1 on the board as shown. Make sure the negative side (Band) is down.

Solder in place.



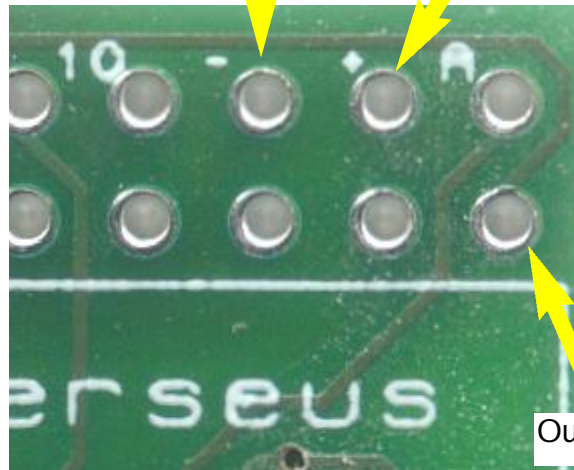
Step 8

Install the 3.3V regulator as shown. Study the inset to make sure you get the pins correctly oriented.



Gnd (Center Lead)

Input (Pin 3)



Output (Pin 1)

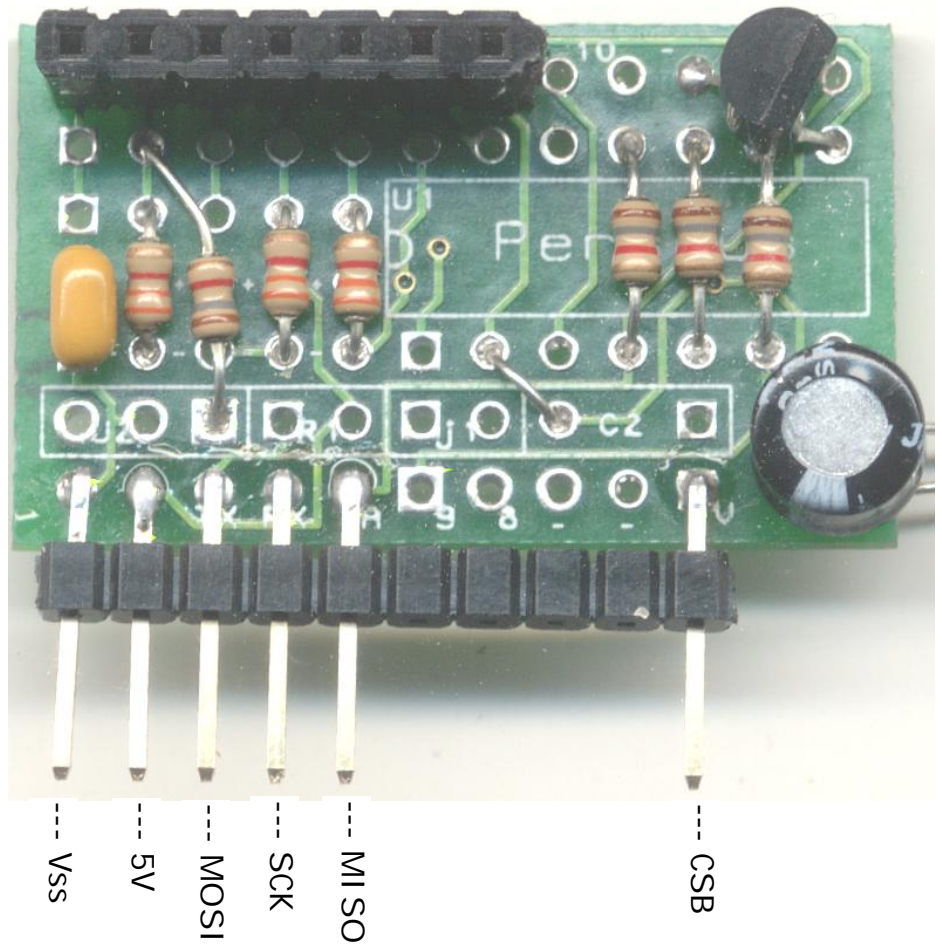
Step 9

Snap off a 10-pin male header and attach it to the board.

The connection header can be connected in a variety of ways. You can connect it so that the pins are located on the bottom of the board or out the side as shown here.

Note that 4 of the pins on the header are not used.

All that's needed now is to place a 7-pin header on the SCP1000 and slip into place as shown in Figure 2.



Notice that we are not using the DRDY lead on the sensor. This is because we are using the spi interface to pole the the DRDY state before each reading call.

The Interface

When I first started the library had problems talking to the sensor. I found the problem to be the fact that the built-in shiftin and shiftout commands were much to fast for the sensor. What I did was to create a function called **SCPspi** that handled the actual byte send and recieve operation. This command is called by all the other routines.

The first command in the library that you must issue is the **SCPinitD01**. This command sets up the ports and initalizes the SCP1000. It also puts the sensor into low noise mode.

Program 1 is a simple program that starts the SCP1000 and prints outs the readings in pascal.

The program makes a call to **SCPsetmodeHR** which puts the sensor into high resolution mode.

The **SCPreadpressure** function takes a reading in pascal units which are then printed out.

If you call the **SCPreadpressureMB** funtion the reading will be returned in millibars. And likewise with **SCPreadpressureIN** they will be returned in inches of mercury.

```
'SCP1000-D01 Test
func main()

    dim pressure as float
    SCPinitD01(0,1,2,3)

    SCPsetmodeHR()

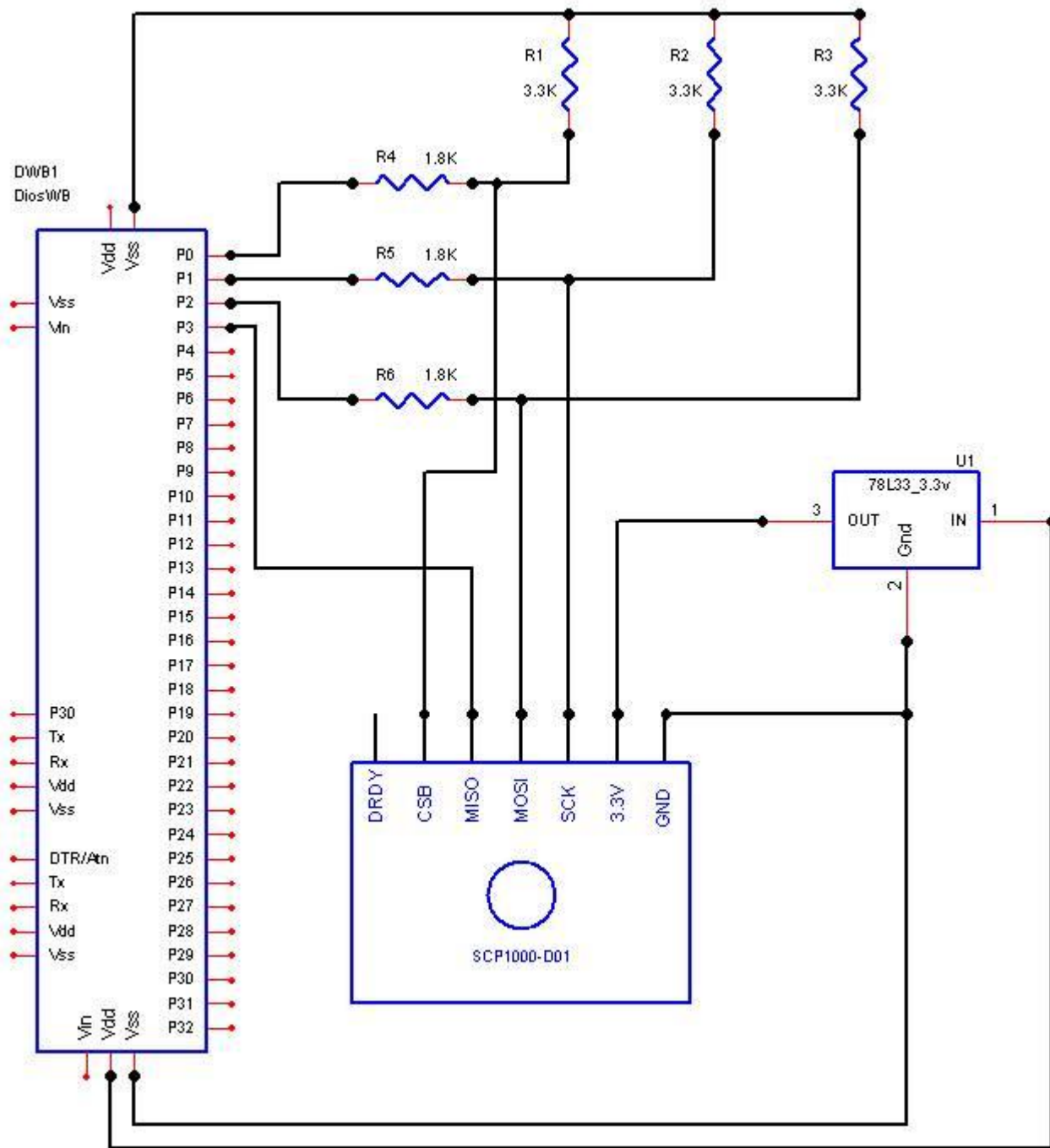
loop:

    pressure = SCPreadpressure()
    print {7.0} pressure
    goto loop

endfunc

include \lib\SCP1000.lib
```

Program 1



Schematic 1

Take a look at Program 2. I have added a call to the function **SCPreadtempF**. This function will return the temperature of the sensor in Fahrenheit. There is also a function called **SCPreadtempC** that will return the temperature in Celsius.

```
'SCP1000-D01 Test
func main()

    dim pressure as float
    dim temperature as float

    SCPinitD01(0,1,2,3)

    SCPsetmodeHR()

loop:
    temperature = SCPreadtempF()
    pressure = SCPreadpressureIN()
    print {5.1} temperature,"F ",{5.2} pressure,"in"
    goto loop

endfunc

include \lib\SCP1000.lib
```

Program 2

In Program 3 we added a couple of LCD commands so that we can display the readings.

```
'SCP1000-D01 LCD Test
func main()
  dim dat as integer
  dim temperature as float
  dim pressureI N as float

  lcdinit 23,25,24,29,28,27,26 'RS, E, RW, D0,D1,D2,D3

  SCPinitD01(0,1,2,3)
  SCPsetmodeHR()

loop2:
  temperature = SCPreadtempF()
  pressureI N = SCPreadpressureI N()

  lcdgoto 1,1
  lcdwrite {5.2} dec temperature," ",{5.3} dec pressureI N

  goto loop2

endfunc

include \lib\SCP1000.lib
```

Program 3

The library is included with the free DiosPro compiler. It also has a help file the that explains each command.

Parts

Spark Fun Electronics

SCP1000

http://www.sparkfun.com/commerce/product_info.php?products_id=8161

Kronos Robotics

<http://www.kronosrobotics.com>

DiosWorkboard Deluxe

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16452>

DiosPro 40

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16428>

Universal 3.3v to 5v Interface kit (Best Deal)

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16537>

contains the following as well as headers.

3.3v Regulator

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16528>

3.3K Resistors

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16530>

1.8K Resistors

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16529>

Persious PCB

<http://www.kronosrobotics.com/xcart/customer/product.php?productid=16389>