An NXT cable you can cut up

2 x 83K Ohm resistors (pull-ups)

Some spare thin wire to connect stuff

Some copper strip board, like this : http://www.bitsbox.co.uk/images/stripboard.jpg

0.1" header strip like this: http://www.bitsbox.co.uk/images/connect/headerstrip.jpg

It would be fairly trivial to make an adapter board with these but a bit tricky to explain it you. Here's a pinout of both the NXT cable and the sensor:



The 4.3V on the NXT is really more like 4.8 and should be good enough for the sensor. Connect the matching pins from the sensor with the corresponding NXT wires. Ignore the servo pins on the sensor. Make sure you add the 83KOhm resistors on both SCL and SDA lines to the +4.8V of the NXT, one on each line. Make sure you connect the NXT cable to the copper strip board and add some tension relief on it so the solder connections don't get damaged.

I would recommend you cut the header pins down to 5 length and solder it to the sensor first, plastic strip on the bottom of the sensor. Then solder those pins onto the copper strip board, but leave this until last when all the resistors and wires have been soldered onto the copper strip board.

**Due to the NXT HW setup, your pullups should be 82k instead of 1.8k. I personally have been able to use pullups in the range of 47k to 100k, and even 10k with varying degrees of success. However, 1.8k is far too strong. Other than that, your HW setup sounds fine.** **//**

**READ TPA81 (http://sourceforge.net/apps/phpbb/mindboards/viewtopic.php?f=3&t=1447&p=13682&hilit=tpa+81#p13682)**

**#define TPA81\_PORT IN\_1**

**#define TPA81\_ADDR 0xD0**

**safecall void MyI2CRead(byte port, byte addr, byte reg, byte cnt, byte& outbuf[])**

**{**

**byte cmdbuf[]; // The buffer of data that will be sent**

**byte nByteReady; // required for getting the status**

**ArrayBuild(cmdbuf, addr, reg); // Build the output buffer**

**while (I2CStatus(port, nByteReady) == STAT\_COMM\_PENDING){Yield();} // Wait until bus is free**

**I2CBytes(port, cmdbuf, cnt, outbuf) //Read the registers**

**while (I2CStatus(port, nByteReady) == STAT\_COMM\_PENDING){Yield();} // Wait until bus is free**

**}**

**task main()**

**{**

**SetSensorLowspeed(TPA81\_PORT);**

**byte inbuf[10];**

**while(true)**

**{**

**MyI2CRead(TPA81\_PORT, TPA81\_ADDR, 0, 10, inbuf);**

**ClearScreen();**

**for (int i = 0; i < 8; i ++){**

**NumOut(0, LCD\_LINE1 - (i\*8), inbuf[i+2]);**

**}**

**Wait(50);**

**}**

**}**