# Wbo2 2A0 Test Procedure

***Revision 0.6, 22st August 2005***

**We appreciate your feedback with suggestions on how to improve the structure of this document.** This is best be done by sending a short email to

peter@techedge.com.au … with subject = "2A0 test proc (rev 0.5) updates"

Please ensure you have the latest version of this document before suggesting changes. The latest version is available from this page

http://wbo2.com/2a0/kit.htm

# Section 1.0 Introduction

This document outlines the troubleshooting sequence for the DIY WBo2 **2A0** unit. The test sequence is divided into five sections.

Section 1. Vital as well as general background information
Section 2. Power supply and Derived Voltages
Section 3. CPU and Host Communications
Section 4. Analog Interface Circuitry
Section 5. Sensor Interface
Section 6. Functional Checks
Section 7. Calibration and Setup
Appendix A. Change History

We assume that the WBo2 2A0 unit has been assembled by the person reading this document and that they are familiar with the naming of all components. A basic understanding of electronics and personal computers is also assumed. The tests and actions described are to be performed in sequence, and, unless otherwise stated, all tests require the **sensor to be disconnected** from the WBo2 unit.

If a test procedure fails or doesn't produce the desired measurement or result (a negative result), then this is a pointer or clue to an underlying build problem or component fault that must be investigated and rectified before proceeding. In many cases, an indication of where to look for the fault will be given, but this does not mean the fault you have will be readily diagnosed. Note also that a negative result may also be an indication that the test has not been performed correctly, and all negative tests should be checked to confirm that the test procedure has been understood and followed correctly.

Remember too that the schematics page **http://wbo2.com/2a0/schem.htm** is an excellent source for debugging should you find your measurements do not match the results presented here.

## *1.0 Possible Problems You May Encounter*

As much as we'd like everyone to be able to build and have a functioning Wbo2 unit in the shortest timeframe, our experience has shown that maybe 5% of constructors will not have well enough developed skills to fault find Wbo2. The most common causes of faults on WBo2 are as follows:

1. Poor soldering leading to shorts and open circuits, or worse, "cold solder" joints.
2. Incorrect, missing, and swapped placement of components.
3. Reversing polarised components (diodes, electrolytic capacitors, ICs, etc.)
4. Poor static electricity minimisation resulting in damaged ICs.
5. Badly or incorrectly constructed cable from Wbo2 to the sensor.
6. Using an inadequate power supply (must be able to supply **3 Amps** at **14 Volts**)
7. Testing with a multimeter or DVM with poor or semi-flat batteries.

If there is any misunderstanding about the required steps to perform then we suggest you do **NOT** proceed, but rather, contact the **oz-diy-wb** Yahoo list (preferably) for assistance from other like minded constructors, or contact Tech Edge for clarification and help in proceeding (*Note : Tech Edge cannot guarantee it will always reply in a prompt fashion*). Please remember that there may be errors in the steps below and we rely on your feedback to correct these errors.

In the case of a unit that cannot be debugged, refer to the **Warranty** link on the home page to find out how to receive an **RMA** (Return To Manufacturer's authority) that can be used to send your unit back to us for fixing. We will charge according to how long it takes us to repair your unit. In extreme cases we may deem your unit uneconomic to repair.

## *1.1 Basic Tools Required*

The most essential tool is your own understanding of electronics, and a willingness to troubleshoot. Also essential are the following basic test items you should already have.

- **DVM**: (Digital Voltmeter) 3.5 digit, 10 M ohm input impedance (test clips highly recommended).
- **Power Supply**: 12-15V (13.8 nominal) @ 2A (always check the voltage under full load). A good unit also has a selectable current limit, and shows voltage and current. Note that one of the tests calls for an adjustable power supply with variable output from below 10 Volts to above 20 Volts.
- Wire **jumpers** with test clips.

- PC with **terminal emulation** (serial comms) program and functional serial port (like *Hyperterminal*). See the note below about USB serial adaptors.

It is assumed you also have the following parts supplied with all kits (you are required to solder the DB9 connector for DIY kits).

- WBo2 serial cable (RJ45 to 9 pin "D" female)
- WBo2 power cable

## *1.2 Basic Assembly*

PCB assembly is described elsewhere (see the Wbo2 web site), however it is recommended that the testing process begins with all discrete components and IC sockets installed, but with **NO** ICs installed except the **DAC7612** (which is pre-installed by Tech Edge) and the **5V voltage** regulator **U10**.

During the test procedure instructions will be given to install ICs; this MUST be done with the power removed from the WBo2 unit, and power subsequently re-applied as required by the instructions.

## *1.3 Voltage measurements*

When an instruction is given to ***measure*** a voltage, this assumes a positive DC voltage with respect to ground, unless specifically stated. For example, "measure +5V" implies connecting the negative DVM input to ground, the positive DVM input to the +5V signal, and setting the DVM to a suitable DC voltage range, generally the lowest range setting that will encompass the expected voltage. We recommend you turn off the auto ranging DVM features if present on your meter. After the measurement instruction, the expected result will be given in parenthesis, for example (4.95, **5**, 5.05), the three values being the lowest acceptable result, the nominal voltage, and the highest expected result.

As well as confirming that the actual result is within the limits given, it is expected that the result will be recorded. Here's a complete example:

- measure +5V (4.95, **5**, 5.05)

When a measurement is to be taken with respect to a reference voltage other than ground, the instruction will read "measure CAL wrt VGnd", in which case the method above is modified to use the second signal as the DVM negative input connection. Signals may be described in two different formats:

- By name (eg. +5V)
- By location (eg. U7p3, Y1p8) – where U7p3 means IC U7 pin 3.

In either case, locate the signal on both the schematics and the PCB. Refer to the appropriate image referred to in the on-line parts list. Another measurement option is indicated using a single test within parenthesis, as in the following examples :

- measure MUXA (>4.5V) ensure voltage at MUXA is greater than 4.5V
- measure MUXB (<0.5V) ensure voltage at MUXB is less than 0.5V

As it is very easy to allow the DVM probes to slip, it is recommended that good quality test clips are used, especially for the ground reference, which is quite close to the fuse (left hand pin of D2).

## *1.4 Connections and jumpers*

An instruction to **connect** two signals implies that a wire jumper needs to be connected between the signals, for example "connect DACV to Vs" requires:

- Identifying the source (first) signal on the schematics, and then on the PCB. The most convenient connection points are either testpoints, jumper pins and the wire lead of resistors which connect to the appropriate signal.
- Identifying the target (second) signal, as above.
- Using a wire jumper with test clips to firmly connect the two signals together. Care MUST be taken to ensure no short circuits occur.

Other instructions requires that a resistor be connected between two signals, for example

- *connect 100 ohm resistor between Ip and VGnd.*

After identifying the connection points, as above, two test leads are used (one for each signal), the other end of each lead then being connected to the leads of the resistor specified. Also noted will be a request to *jumper two pins on IC sockets*, for example:

- *jumper U1p2 to U1p3*.

This is best accomplished using a lead cut from one of the installed resistors, cut and bent to the required length and then very carefully inserted into the IC socket. When connections must be made between pins on different IC sockets, use a short length of resistor lead inserted into each IC socket, and connect these together using a wire jumper with test clips. Take great care not to deform the IC contact when inserting and removing these connections.

## *1.5 Serial Ports & User Interface*

The test method relies heavily on using a PC terminal emulator program to communicate with the WBo2 unit's user interface. One such PC program is **Hyperterminal**. The required setting are listed below but first ensure the port is off-line with *{Call, Disconnect}* then select *{file properties, Configure…}* and set the following

| | |
|---|---|
| **speed** | 19,200 bps |
| **bits** | 8 |
| **parity** | none |
| **stop** | 1 bit |
| **flow** | ALL flow control options disabled (*note : very important*) |

Lastly, set the following parameter to see your commands as they are entered

- **echo** local, use *{File Properties, setting, ASCII setup…}.*

Another contender for talking to WBo2 is the WBo2 **config** program (*downloadable from Tech Edge via **WBo2.com/sw/conf.htm***). Use the ***Comms*** tab screen to set the COMM port and then enter commands and see their results.

It is possible to use a **USB to serial converter** to either add the first, or another serial port to your PC. We don't recommend any particular brand but certainly early model units seem to have some problems.

**Serial Loop-back Test :** The majority of PCs use a 9 pin male "D" connector (known as a DB9) for the serial port, and a quick check of both the terminal emulator software and the serial port hardware may be performed by shorting pins 2 and 3 of the DB9 connector together and following this serial loop-back test :

Typing any alphabetic character on the keyboard should result in that character being displayed twice on the screen (one from the local echo and the second from the loop-back receive data).Once the terminal emulator is connected to a WBo2 unit and is working, **commands** can be sent to WBo2, and it should provide responses. The commands to type are described here using angle brackets as delimiters; for example :

<**i02**>

… is an instruction to type the key sequence **i02 (ie.** letter **i**, number zero, and number **2** followed by the **Enter** key. Note : do not type the angled brackets). Also note that the commands are case sensitive (ie. make sure the **Caps Lock** key is not depressed). The commands comprise a command character ("**i**" in the example above) plus parameters specific to each command; these will be described in a later section, but are all in hexadecimal format, and may take the following forms:

| | | |
|---|---|---|
| nibble | (single character) | eg. <**m0**> |
| byte | (two characters) | eg. <**i02**> |
| w | (four characters) | eg. <**d2200**> |

The majority of commands will respond with a hexadecimal response, for example the <v> command will return the response **"> 0042**", which in this case is simply the software version number. Invalid commands (eg. <**vv**>) will return a "**Err vv**" string, indicating that the WB user interface software (UI) cannot process the command given, which is echoed back to the user. If the response is to be recorded, the command string will be followed with a pair of braces; for example

**<i01> []**

… implies that the response code should be noted, whereas :

**<m0>**

… indicates that the response is not of significance. Only the four hex digits need to be recorded.

When the WBo2 unit is first powered on, it will display a sign-on string such as:

**tec2a0r0042**

… which identifies the version of firmware installed in the unit. It is important that only firmware for the 2A0 product is used. Any 2A0 version is suitable for this test procedure, and it is recommended that the firmware is NOT upgraded to a later version until all steps have been completed.

Two seconds after the firmware version string is displayed, the WBo2 unit may start sending logging data, which may appear as random rubbish on the PC screen. The <**tf**> command will stop this data being sent, allowing debug commands to be entered. To prevent logging transmission after subsequent power cycles, use the following sequence of commands:

| | |
|---|---|
| **<s0302>** | turn off auto-logging, and select CAL mode for the Tx frame type |
| **<w> []** | write config data to EEPROM, [ensure result is NOT 0x8000] |

## *1.6 Test Modes*

The User Interface can be used to set the WBo2 unit into different operating modes. The unit defaults to mode 1 (operational) which is indicated by the (red) status LED flashing at a 2.5Hz rate (2.5 cycles per sec, 0.2 seconds on, 0.2 seconds off). Prior to the majority of the tests to be taken, the WBo2 must be placed in mode 2 (config mode), using the following commands :

| | |
|---|---|
| **<m0>** | "off" mode, indicated by slow (0.5Hz) LED pattern |
| **<m2>** | "config" mode, indicated by double flash LED pattern. |

result codes returned by WBo2 can be ignored for both of these steps.

## *1.7 Test Commands Available*

Once in config mode, a number of UI commands will be used to generate test voltages and read the CPUs analog to digital converter channels, amongst others.

These are some of the commands, and required parameters, which are:

- ***n*** nibble (single hex character)
- ***b*** byte (two hex characters)
- ***w*** w (four hex characters)

| cmd | param | resp | Action |
|---|---|---|---|
| **a** | ***w*** | data | Read EEPROM or RAM - depends on MSB of ***w***<br>MSB of ***w***=0 read byte from EEPROM<br>MSB of ***w***=1 read word from RAM |
| **e** | ***w b*** | code | write b to EE, ram or registers |
| **m** | ***n*** | code | controls WBo2 main state machine |
| **d** | ***w*** | code | sets DAC (address embedded)<br>0xxx: both DACs<br>2xxx: DAC A (DACV)<br>3xxx: DAC B (WBVO) |
| **p** | ***w*** | code | sets PWM (address embedded)<br>0xxx: PWM A (**SVout**)<br>1xxx: PWM B (NBsim) |
| **i** | ***n1 n2*** | data | reads ADC channel ***n1*** with mux set to ***n2*** |
| **t** | ***n*** | data | serial logging control, n=0, 1 .. e, f |
| | ***n*** = 0 | … | Send a single Tx Frame (only if Tx auto gen is disabled) |
| | ***n*** = f | … | Stop Tx Frames being sent (sets Tx auto gen OFF) |
| | ***n*** =? | … | Start Tx Frames sending (sets Tx auto gen ON *if* **0<n<f**) |
| **s** | ***b1 b2*** | code | set config byte ***b1*** to value ***b2*** |
| **g** | ***b*** | data | get config byte ***b*** |
| | ***ff*** | … | get all config data bytes used |
| **g** | ***b*** | data | get config byte ***b*** |
| **l** | **-** | code | (ie. Letter *ell*) reload conf-RAM based data from EEPROM |
| **w** | **-** | code | write conf-RAM based config data to EEPROM |

| | | | |
|---|---|---|---|
| **r** | **-** | code | reload conf-RAM from ROM (*note:* command may change) |
| **u** | **-** | code | update system params from conf-RAM |
| **v** | **-** | code | display firmware version |
| **x** | ***n*** | ptr | stop/start logging to on-board log memory (ie. log-RAM)<br>***n*** = 0, stop logging<br>***n*** = 1, start logging<br>else, returned value is current storage pointer |
| **y** | ***w*** | code | set log-RAM storage pointer to ***w*** |
| **z** | **-** | -- | dump logged data stored in log-RAM |

## *1.8 Analog Tests*

Analog stimulation voltages generated by the DAC7612 **DAC** and taken from **U7p12** but a buffered version of this signal is also available at **Test Pin 2**. A commonly used test is referred to as a "DAC test", where a test voltage is generated, connected to a specified signal and a voltage measurement is performed. These all require the unit to be in config mode (**mode 2**, see section 1.6).An example of using the DACV signal would be:

<**d27d0**> measure DACV (1.99, **2.0**, 2.01)

Another set of tests uses the CPU's analog to digital converter (ADC) to sample a signal. In a similar fashion to the voltage measurements (section 1.3), a decision is made on the basis of a range of expected values; in the following, the ADC is used to measure a signal <**i02**>, the results are recorded ([]) and the result compared with the range given in parentheses.

**<i02>** [] (0x0189, 0x019A, 0x01AA)

## *1.9 Converting from Hex to Decimal Formats*

It can be useful to use a pocket calculator to convert the hex numbers to decimal, It may be easier to use Windows **calculator** (find it under Accessories **->** calculator, or Start **->** Run... **->** "**calc**"). When you find it, switch to "**Scientific mode**" (View **->** Scientific). At the left of the screen, right under the display, there is a **Hex** and a **Dec** radio button. By default the **Dec** (for Decimal) mode is selected, so click on the **Hex** radio button.

To convert **hex** to decimal simply enter a **hex** number (when in hex mode) and switch to Dec mode. You always know the mode because decimal mode always has a decimal point in the display, and Hex mode doesn't.

Remember that you copy and paste to/from Calc by using the **Ctrl+C** and **Ctrl+V** keys - so you don't have to re-type numbers you can already see with the terminal program.

The hex numbers are 0 through 9 & A, B, C, D, E, F. Hex is also commonly written with either a leading $ symbol or 0x, so $10 = 0x10 = 16. (16. Is in decimal format, not the decimal point).

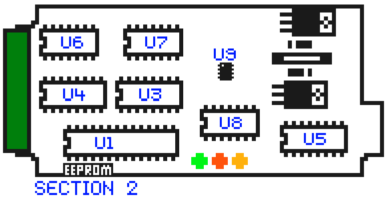
## *1.10 PCB Orientation, Test Points & Rev number*

The descriptions below assume you are looking at the PCB so that the majority of the white silkscreen's writing reads normally (ie. left to right). The green logger connector is on the left and the black RJ45 connector is on the lower right. The 3 LEDs are at the bottom.

Test points **TP1** to **TP6** are available on the board. **TP1**, **TP2**, **TP6** & **TP5** are located under the **TEST** area at the centre of the board, **TP3** & **TP4** are at top right just above the **J1** shunt.

The PCB version or **Rev-** (ie. revision) number is printed on the solder side of the PCB right under U5 (bottom right of the PCB). Earlier boards were Rev-2 and Rev-3, and Rev-4 is current at the start of 2005. The tests will refer to these Rev- numbers only if they are relevant to that test.

# Section 2. Power Supply and Derived Voltage Tests



At this point, the WBo2 unit should be fully assembled with all discrete components and IC sockets, but no ICs should be loaded at this stage, except the DAC7612 (it comes pre-soldered to the PCB) and the 3 terminal 7058 voltage regulator. The power supply cable should be connected to a power supply (switched off), the sensor cable NOT connected.

## 2.1 Power Supply.

Connect the DVM's –ve (Black) lead to GND (a suitable point is the left end of D2, just below the fuse). Connect the DVM's +ve lead (red) to U10p3 (the lower leg of U10, which is the TO-220 7805 part). This connection is to measure the +5V power supply voltage.

Briefly turn on the power supply and confirm the +5V supply. If other than 5V is present, turn off the supply immediately and investigate.

## 2.2 +5V rail

Measure +5V (4.90, **5**, 5.10) {_ _ _ _}

If no voltage is present then

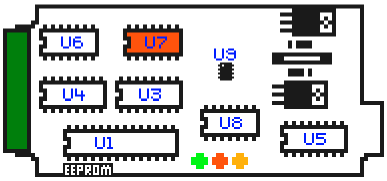
1. confirm there is power to the unit – measure the voltage at the top pin of Y4 (Call it VBatt)
2. Confirm that D1 has the band to the left and has power supply voltage on the right pin and on the left pin is at VBatt less a few 10<sup>th</sup>s of a Volt.
3. Confirm the fuse is not blown – if so, check for shorts in the area of the fuse, rectify, and then replace the fuse. If it blows again then you still have a short or D2 has been reversed (its band to the right).

## 2.3 Power LED

The green LED should be glowing. {_ _ _ _}

If not then confirm that the flat of the LED (the flat part at the base of the LED) matches the flat on the silk screen (the LEDs flat should be at the top).

## 2.4 Reference divider for 2.0 Volts (VGND)



SECTION 2.4

Install the **LMC6484** in position **U7** – note, it may be convenient to remove it later.

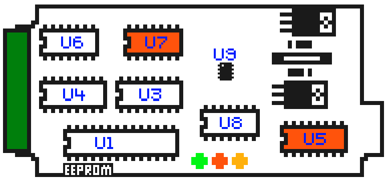
Measure U7p3 (1.92, **2.0**, 2.08) {_ _ _ _}

If not within these limits then check that R5 and R6 are correctly installed, and there are no shorts.

Measure U7p1 (1.92, **2.0**, 2.08) {_ _ _ _}

If not within these limits then check that R7 is 100R and there are no shorts either side of R7.

## 2.5 RS232 positive supply



SECTION 2.5

Install the ST202 into U5.

Measure U5p2 (9.0, 9.5,10.0) {_ _ _ _}

If not within these limits then check for shorts around U5, and that the capacitors C301 to C305 are installed and there are no shorts across them.

## 2.6 RS232 negative supply

Measure U5p6 (-9.0, **-9.5**, -10) {_ _ _ _}

## 2.7 Serial loopback test

Jumper U1p2 to U1p3

Perform the loopback test described in section 1.5. {_ _ _ _}

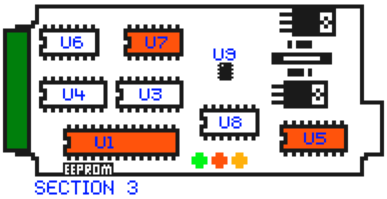
Remove the U1p2 to U1p3 jumper at completion of the test.

If no loopback is present then verify that loopback is possible by checking the DB9 directly by jumpering pins 2 and 3 of the RS232 cable's DB9 connector.

### 2.8 Auxiliary display supply (power on the RJ45 connector)

Measure voltage at Y1p8 (top pin closest to Y3) This should be approximately at the supply voltage less 0.4V.
{_ _ _ _}

## Section 3. CPU and Host Communications Tests



### 3.1 CPU startup, Status (Red) LED Operation

Install the CPU at U1. Power on the unit.

Verify the red status LED is flashing at a steady 2.5Hz rate

(0.2 seconds on 0.2 seconds off, etc.) {_ _ _ _}

If no red LED activity then check the correct connection of the LED against the silkscreen layout. If you have a logic probe, test to see if the G* (U1p5) LED control pin is pulsing at 2.5 Hz. If not then either

- the CPU (U1) has not programmed properly (unlikely), or has no power,
- CPU has been zapped by static,
- there's a short on another of the CPU pins forcing it into a "strange" mode,
- the crystal (X1) is defective, or shorted, or open circuit, or,
- there's a short across C307 or C308 (the 15 pF capacitors), or capacitors are the wrong value.

### 3.2 Heater (Amber) LED operation

The amber heater LED should have a dim but fast flash of about 10 times/second however at this stage there are many causes for the LED to be inactive. {_ _ _ _}

If no activity, check the LED's connection and wait until we do further checks below.

### 3.3 Firmware version, RS232 operation

Ensure that power to Wbo2 is turned off. Connect the RS232 cable and start the terminal emulator (see section 1.5). When WBo2 is powered up and running properly it continually (and blindly) pumps out RS232 logging data frames.

Now turn ON the power to Wbo2 and observe the default binary logging data stream. This will appear mostly as strange symbols. If you do not see any data from WBo2 then first check that Hyperterminal has been set up properly (see section 1.5), and then check the complete data path through RS232 chip U5 to the CPU U1.

We don't want this to intrude on our debugging so we'll turn off:

**<tf>** turn off logging frames [we expect response "> 400F"]

**<s0302>** turn off auto-logging, and select CAL mode for the Tx frame type

**<w> []** write config data to EEPROM, [ensure result is NOT 0x8000]

Now turn Wbo2 off, wait two seconds, and turn it back on. Ensure that there is a response such as "tec2a0r0037" which is made up of "tec", "2a0" & "r0nxx". {_ _ _ _}

Further confirm communication from the PC to WBo2 is working by typing **V** to return the WBo2 unit's current firmware version number {_ _ _ _}

Check there is a response then, depending on the firmware WBo2 has been flashed to :

for **lsu** **<v> []** Ensure first two digits of response code are 00 {_ _ _ _}

for **ntk** **<v> []** Ensure first two digits of response code are 01 {_ _ _ _}

### 3.4 Logger Pushbutton open

**<m0> <m2>** Ensure red LED flashes with two short **ON** flashes and a longer **OFF** pattern. This flash sequence is called the "*double LED pattern*" below. {_ _ _ _}

**<a8036> []** Ensure third digit of response is 3 (eg. 2E30), indicating that the logger switch is **NOT** permanently turned ON {_ _ _ _}

### 3.5 Logger Pushbutton closed

Continuously hold down PB1 (hint : type **a8036**, hold down PB1, then press the **Enter** key)

**<a8036> []** Ensure the third digit of response is 2 (eg. 2E20), indicating the switched closed condition of PB1 can be detected {_ _ _ _}

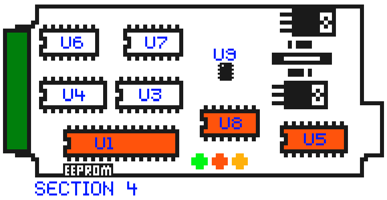
### 3.6 *Emergency boot (Force re-flash function)*

Power down unit. Insert a jumper shunt between **Y5p2** and **Y5p3** (*Note:* use a temporary piece of wire if you don't have any pins installed on the Y5 area of the PCB).

Power on the WBo2unit. Ensure terminal window shows string "**AlaR**" {_ _ _ _}

Note, some random characters may appear prior to the required string, due to noise on serial line during power on. After testing, remove the Y5p2/Y5p3 shunt (or temporary jumper)

## Section 4. Analog Interface Circuitry Tests



### *4.1 SVout*

Power down the unit. Install **LMC6484** at **U8** (the right of the CPU). Power on unit. Note: **SVout** is available at **TP5** as well as **U8p9** and at **Y1p1**, which is the RJ45 connector at the right of the PCB. **p1** is at the bottom of **Y1** (PCB view) or left (connector view). Ensure the **TP5** signal is routed to the **Y1p1** output too.

**<m0> <m2>** Ensure the double LED pattern flashes

**<p0000>** Measure **SVout** (0.0, **0.0**, 0.020) {_ _ _ _}

**<p0200>** Measure **SVout** (2.47, **2.5**, 2.53) {_ _ _ _}

**<p03ec>** Measure **SVout** (4.85, **4.9**, 4.95) {_ _ _ _}

### 4.2 *NBsim*

Note **NBsim** is available at **TP6** as well as **U8p13** and at **Y1p6** which is the RJ45 connector at the right of the PCB. **p6** is third from the top (PCB view) or right (connector view). Ensure the **TP6** signal is routed to the **Y1p6** output too. Note that

**<m0> <m2>** Ensure double LED pattern flashes

**<p1000>** Measure NBsim (0.0, **0.0**, 0.020) {_ _ _ _}

**<p1200>** Measure NBsim (2.47, **2.5**, 2.53) {_ _ _ _}

**<p13ec>** Measure NBsim (4.85, **4.9**, 4.95) {_ _ _ _}

### 4.3 *Tacho Input*

Remove jumper **J4** (bottom left) and connect **J4p3** to **U3p9** (this uses an internally generated TTL signal **MUXB** to "tickle" the RPM circuitry).

**<m0> <m1>** (note **M1**, not **M2** this time)

**<s0301>** Ensure the red LED is flashing at 2.5Hz

<**t0**> Ensure second last field (RPM) contains an entry between 0x160 and 0x170 {_ _ _ _}

Remove the connection **J4p3** to **U3p9**, Install jumper **J4** per intended application.

### 4.4 *SPI (internal bus) and DAC (U9 = DAC7612)*

The following tests that the CPU can communicate with the DAC via the internal SPI bus, and that the DAC's output buffers are working.

<**m0**> <**m2**> Ensure double LED pattern.

<**d2000**> Measure DACV **U7p12** (0.0, **0.0**, 0.01) {_ _ _ _}

<**d27d0**> Measure DACV (1.99, **2.0**, 2.01) {_ _ _ _}

<**d2fff**> Measure DACV (4.075, **4.095**, 4.115) {_ _ _ _}

<**d3000**> Measure WBVO **U6p12** (0.0, **0.0**, 0.01) {_ _ _ _}

<**d37d0**> Measure WBVO (1.99, **2.0**, 2.01) {_ _ _ _}

<**d3fff**> Measure WBVO (4.075, **4.095**, 4.115) {_ _ _ _}

If both tests fail then the DAC, or communications from the CPU to the DAC (via SPI) are not working properly. If one but not the other fails, then the relevant output circuitry must be checked.

### 4.5.1 ADC Channel 0

Jumper **U7p12** to **U4p13.** This is the DACV signal from the U9 DAC7612 to the CPU's ADC0 input. ***Reminder :*** Note that the **74HC4052** mux **U4** and the **LMC6484** op-amp **U7** must **not** be present for this series of tests.

<**m0**> <**m2**> Ensure double LED pattern
<**d2000**> <**i00**> [] (0x0000, **0x0000**, 0x0000) {_ _ _ _}
<**d27d0**> <**i00**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}
<**d2fff**> <**i00**> [] (0x0337, **0x0347**, 0x0357) {_ _ _ _}
remove the **U4p13** end of the jumper ONLY for use in the next test …

### 4.5.2 ADC Channel 1

jumper **U7p12** to **U4p3**

<**m0**> <**m2**> Ensure double LED pattern
<**d2000**> <**i10**> [] (0x0000, **0x0000**, 0x0000) {_ _ _ _}
<**d27d0**> <**i10**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}
<**d2fff**> <**i10**> [] (0x0337, **0x0347**, 0x0357) {_ _ _ _}
remove the **U4p3** end of the jumper ONLY for use in the next test …

### 4.5.3 ADC Channel 2

jumper **U7p12** to **U6p1** (we have moved from U4 to **U6** in this and the next test). ***Reminder :*** Note that the 74HC4052 mux **U3** must **not** be present for this test.

<**m0**> <**m2**> Ensure double LED pattern
<**d2000**> <**i20**> [] (0x0000, **0x0000**, 0x0000) {_ _ _ _}
<**d27d0**> <**i20**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}
<**d2fff**> <**i20**> [] (0x0337, **0x0347**, 0x0357) {_ _ _ _}
remove the **U6p1** end of the jumper ONLY for use in the next test …

### 4.5.4 ADC Channel 3

jumper **U7p12** to **U3p3**

<**m0**> <**m2**> Ensure double LED pattern
<**d2000**> <**i30**> [] (0x0000, **0x0000**, 0x0000) {_ _ _ _}
<**d27d0**> <**i30**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}
<**d2fff**> <**i30**> [] (0x0337, **0x0347**, 0x0357) {_ _ _ _}
remove jumper **U7p12** to **U3p3.**

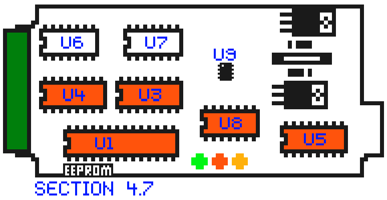
***note*** for tests 4.5.1 to 4.5.4 : If any of the above 4 ADC channel tests fail then the path from the ADCn pin to the CPU should be checked for solder bridges or other shorts.

## *4.6 Multiplexer Control Logic*

This confirms that the mux control signals from the CPU (that control routing of signals to the CPU's A/D inputs) are functioning correctly. MUXA & B are found on pins 10 (MUXA) and 9 of U3 & U4.

<**m0**> <**m2**> Ensure double LED pattern
**<i01>** Measure MUXA (>4.8V) {_ _ _ _}
Measure MUXB (<0.2V) {_ _ _ _}
**<i02>** Measure MUXA (<0.2V) {_ _ _ _}
Measure MUXB (>4.8V) {_ _ _ _}

## *4.7 Multiplexers, User inputs*



The following tests verify the operation of the user inputs. First power down the unit, then

- install two x **74HC4052** at **U3** and **U4**
- Install **LMC6484** at **U8**

Return power to the unit and perform the following tests 4.7.n test

### 4.7.1 User **1** Input

jumper **U7p12** to **Y2p6** – this routes the DACV voltage to the user input

<**m0**> <**m2**> Ensure double LED pattern

<**d2000**> <**i31**> [] (0x0000, **0x0000**, 0x0002) {_ _ _ _}

<**d27d0**> <**i31**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}

<**d2fff**> <**i31**> [] (0x0337, **0x0347**, 0x0357) {_ _ _ _}

remove the **Y2p6 end** of the jumper ONLY for the next test …

### 4.7.2 User **2** Input

jumper **U7p12** to **Y2p7**

<**m0**> <**m2**> Ensure double LED pattern

<**d2000**> <**i32**> [] (0x0000, **0x0000**, 0x0002) {_ _ _ _}

<**d27d0**> <**i32**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}

<**d2fff**> <**i32**> [] (0x0337, **0x034**7, 0x0357) {_ _ _ _}

remove the **Y2p7 end** of the jumper ONLY for the next test …

### 4.7.3 User **3** Input

jumper **U7p12** to **Y2p8**

<**m0**> <**m2**> Ensure double LED pattern

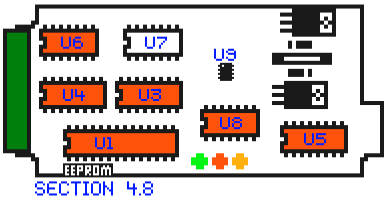
<**d2000**> <**i33**> [] (0x0000, **0x0000**, 0x0002) {_ _ _ _}

<**d27d0**> <**i33**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}

<**d2fff**> <**i33**> [] (0x0337, **0x0347**, 0x0357) {_ _ _ _}

remove jumper **U7p12** to **Y2p8**

## *4.8 Thermocouple Amp and Multiplexer (Mux)*



Note, due to the high gain (101 times), only a very small input voltage is required to produce a large output voltage, these thermocouple tests are really only a guide to the functionality of the circuits involved, and hence much looser tolerances are given. The additional <i> commands are to preset the multiplexers, due to the long time constant associated with the TC amp lowpass filtering.

*Note : Check that C402 is 100 pF as very early kits were supplied with a much larger incorrect value.*

### 4.8.1 TC**1**

Install a **LMC6484** in position **U6** (U6A is the thermocouple amp), then …

jumper **U7p12** to **Y2p1**

<**m0**> <**m2**> Ensure double LED pattern

<**i20**> <**d2000**> <**i20**> [] (0x0000, **0x0000**, 0x0020) {_ _ _ _}

<**i20**> <**d2016**> <**i20**> [] (0x01AC, **0x01C3**, 0x01D9) {_ _ _ _}

<**i20**> <**d202D**><**i20**> [] (0x036C, **0x039A**, 0x03C8) {_ _ _ _}

remove the **Y2p1 end** of the jumper ONLY for the next test …

### 4.8.2 TC**2**

jumper **U7p12** to **Y2p2**

<**m0**> <**m2**> Ensure double LED pattern

<**i21**> <**d2000**> <**i21**> [] (0x0000, **0x0000**, 0x0020) {_ _ _ _}

<**i21**> <**d2016**> <**i21**> [] (0x01AC, **0x01C3**, 0x01D9) {_ _ _ _}

<**i21**> <**d202D**><**i21**> [] (0x036C, **0x039A**, 0x03C8) {_ _ _ _}

remove the **Y2p1 end** of the jumper ONLY for the next test …

### 4.8.3 TC**3**

Jumper **U7p12** to **Y2p3**

<**m0**> <**m2**> Ensure double LED pattern

<**i22**> <**d2000**> <**i22**> [] (0x0000, **0x0000**, 0x0020) {_ _ _ _}

<**i22**> <**d2016**> <**i22**> [] (0x01AC, **0x01C3**, 0x01D9) {_ _ _ _}

<**i22**> <**d202D**><**i22**> [] (0x036C, **0x039A**, 0x03C8) {_ _ _ _}

Remove jumper **U7p12** to **Y2p3**

### *4.9 WBlin Output*

The WBlin output is available at the TP3 test point, but the output to the outside world is at **Y1p4** (Y1 is the RJ45, p4 is the 4th pin from the bottom/left)

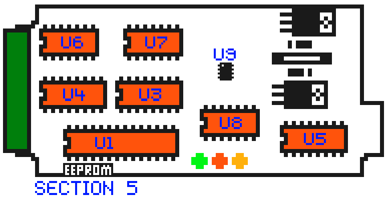
<**m0**> <**m2**> Ensure double LED pattern

<**d3000**> Measure WBlin (0.0, **0.0**, 0.01) {_ _ _ _}

<**d37d0**> Measure WBlin (2.39, **2.44**, 2.49) {_ _ _ _}

<**d3fff**> Measure WBlin (4.90, **5.00**, 5.10) {_ _ _ _}

## Section 5. Sensor interface



Power down the unit and install **LMC6484** at **U7**

Power the unit back ON.

### *5.1 Vref (VGND, 2.0 Volt Virtual Ground Reference)*

Measure Vref at **J1p1**, ie. right under **TP3** (1.92, **2.0**, 2.08) {_ _ _ _}

### *5.2 Vref Monitor*

Ensure **J1** is installed (right below **TP3** & **TP4**, top left of PCB)

<**m0**> <**m2**> Ensure double LED pattern

<**i03**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}

### *5.3 Vs amp (Sense Cell Voltage, VsX1 & VsX5 inputs)*

Connect DACV at **U7p12,** to Vs at **Y3p1**. **Y3** is the circular 8 pin connector at the right of the PCB and **p1** is at the top of **Y3**. There are two outputs to be tested, a **VsX1** and a **VsX5** signal.

<**m0**> <**m2**> Ensure double LED pattern

<**d27d0**> <**i13**> [] (0x0191, **0x019A**, 0x01A2) {_ _ _ _}

<**i10**> [] (0x0000, **0x0000**, 0x0006) {_ _ _ _}

<**d29c4**> <**i13**> [] (0x01F6, **0x0200**, 0x020A) {_ _ _ _}

<**i10**> [] (0x01F6, **0x0200**, 0x020A) {_ _ _ _}

Remove connection DACV to Vs

### *5.4 Ip Drive Circuit & Ip Sense Circuit*

This test checks that the **Ip Sense** circuitry **U7**C correctly records the sense current that is made to flow through a temporarily attached 100 ohm resistor (this resistor is supplied with recent kits).

Connect the spare 100 ohm ¼ resistor from Ip to VGnd. The Ip signal is on **Y3p3** and **VGnd** is on **Y3p2**. It may be easiest to connect the 100 ohm resistor directly to pins 2 and 3 of the circular 8 pin connector at the right end of the PCB (in the 1 and 3 o'Clock positions for a correctly assembled unit).

A DVM may optionally (test marked as * below) be used to directly measure the current through the 100 ohm resistor.

Note: The <**i00**> command reads the current as determined by the differential amplifier circuitry around U7C. The actual Ip drive circuitry is from U7D and the DAC we have already tested.

<**m0**> <**m2**> Ensure double LED pattern

<**d27d0**> * Measure **Ip** wrt **VGnd** (-0.02, **0**, 0.02) {_ _ _ _}

<**i00**> [] (0x0189, **0x019A**, 0x01AA) {_ _ _ _}

<**d24e2**> * Measure **Ip** wrt **VGnd** (-0.295, **-0.286**, -0.277) {_ _ _ _}

<**i00**> [] (0x0045, **0x0048**, 0x004B) {_ _ _ _}

<**d2bb8**> * Measure **Ip** wrt **VGnd** (0.371, **0.382**, 0.393) {_ _ _ _}

<**i00**> [] (0x0339, **0x0352**, 0x036C) {_ _ _ _}

Remove resistor from **Ip** to **VGnd**

## *5.5 H+ (vBatt) monitor*

H+ is the voltage at the "High", or positive (+ve) side of the sensor's heater drive circuitry – this is also the full battery voltage vBatt. Circuitry monitors ¼ of this battery voltage. This test compares the value vBatt, measured directly using a multimeter, with the value sensed by the CPU.

<**m0**> <**m2**>  Ensure double LED pattern

Measure vBatt  []  {_ _ _ _} *1*

<**i30**>  [] call this value **ADC** – (it is in hex)  {_ _ _ _}

convert **ADC** value to decimal (**ADCd**) with a calculator  {_ _ _ _}

convert **ADCd** to voltage (Voltage **V** = **ADCd** / 51.2)  {_ _ _ _} *2*

compare **V** (*1* above) with the calculated value (*2*)  {_ _ _ _}

## *5.6 H- monitor*

H- is like H+ but measures the voltage on the other side of the heater. This side of the heater is switched to GND via the FET under control of the CPU. This test compares the value vBatt of measured with a multimeter with the value senses by the CPU. The 100 ohm test resistor should be connected to pins 6 and 7 of the circular 8 pin sensor connector Y3.

Connect **H+** (**Y3p7**) to H- (**Y3p6**) using the 100 ohm test resistor

<**m0**> <**m2**>  Ensure double LED pattern

Measure vBatt  []  {_ _ _ _} *1*

<**i11**>  [] call this value ADC – (it is in hex)  {_ _ _ _}

convert ADC value to decimal (ADCd) with a calculator  {_ _ _ _}

convert ADCd to voltage (Voltage = ADCd / 51.2)  {_ _ _ _} *2*

compare V (*1* above) with the previous value (*2*)  {_ _ _ _}

unless immediately going on with the next test, remove resistor from H+ to H-

## *5.7 Heater current monitor*

Note that due to the small currents involved, this test is only a rough guide to the functionality of the circuits involved.

<**m0**> <**m2**>  Ensure double LED pattern

If not already connected, connect **H+** to **H-** using 100 ohm resistor. (reads the Ih signal …)

<**i12**>  [] (0x0000, **0x0000**, 0x002)  {_ _ _ _}

Connect junction of **R113** and **Q4** to ground, this turns on the FET

Confirm the heater LED (AMBER) turns **ON**  {_ _ _ _}

<**i12**>  [] (0x0008, **0x001F**, 0x0030)  {_ _ _ _}

remove the resistor from H+ to H-

## *5.8 Vs Sensing*

This test confirms that the sensor 's Vs signal pickup circuitry is correctly wired (also see previous section 5.3 that checks that the Vs amplifier is working properly). There are different procedures for the LSU (ie. **6066** & **7057**) and the NTK (ie. **L1H1** & **L2H2**) sensor versions.

First, and most importantly, ensure that :

- for Rev-**2** or Rev-**3** PCBs, the correct **R10**4 is installed (**910 ohms** for LSU, and **82**k for NTK), or …
- For the Rev-**4** PCB, that the NTK jumper shunt is in the correct position (NTK shunt is **installed** for **LSU**, but is **absent** for **NTK** – *the shunt's name is misleading*).

For the **LSU** test, note that **Vs** measurements are performed with a DVM between **GND** and **Vs**, but the additional 1k resistor (not supplied in the kit) is connected between **VGnd** and **Vs.** The 1k resistor is best connected to the circular 8 pin connector Y3 (connect between pins 1 and 2).

**LSU version** only test

Ensure that R104 is

<**m0**> <**m2**>  Ensure double LED pattern

connect an additional 1k resistor from **Vs** (**Y3p1**) to **VGnd** (**Y3p2**)

<**e8031fc**>

<**e803234**> Measure Vs with DVM (3.4, **3.45**, 3.5) {_ _ _ _}

<**e803214**> Measure Vs (0.85, **0.95**, 1.05) {_ _ _ _}

<**e8031dc**> Measure Vs (1.95, **2.0**, 2.05) {_ _ _ _}

remove the resistor from **Vs** to **VGnd**

**NTK version** only test

<**m0**> <**m2**> Ensure double LED pattern

<**e8031fc**>

<**e803234**> Measure Vs with DVM (3.4, **3.45**, 3.5) {_ _ _ _}

<**e803214**> Measure Vs (0.0, **0.0**, 0.2) {_ _ _ _}

# Section 6. Functional checks

## *6.1 Sensor Status Tests*

if already switched **ON**, power **down** the unit for at least 10 seconds.

**Important** - ensure your sensor is **NOT** connected in the following tests.

Power unit on

Confirm the **Amber** Heater LED is flashing at 10Hz rate (very dim flicker) {_ _ _ _}

<**s0381**> Set frame generation to ASCII/HEX mode

<**s0432**> Set frame delay to 0x32 (40 decimal)

<**w**> Save frame parameters to EEPROM (sets them as the default)

<**t1**> Start auto Tx frame generation

Tx Frames generation should begin and this will show on the terminal emulator screen.

In the following steps, the **status word** referred to is the **final** hexadecimal word shown.

Also note that the RED status LED should be flashing the 0.5Hz even pattern in these test.

### 6.1.2 VBatt low status threshold

Adjust the supply voltage down to **10V**.

Confirm status word indicates : battery low (**xxx2**) {_ _ _ _}

Note that at some point, as the supply voltage is increased, we expect that at about 10.5 Volts the status word should change to **NOT** *Low battery Voltage* status.

### 6.1.3 VBatt high status threshold

Adjust supply up to **19.5**V

Confirm status word: battery high status (**xxx1**) {_ _ _ _}

Note that at some point, as the supply voltage is decreased, we expect that at about 18.8 Volts the status word should change to **NOT** *High battery Voltage* status.

### 6.1.4 Heater circuit open status test

Adjust supply to **13.8**V

Confirm status word: heater open status (**xxx4**) {_ _ _ _}

### 6.1.5 Heater circuit shorted status test

Connect H+ to H- (**Y3p6** to **Y3p7** on the circular 8 pin connector)

Confirm status word: heater short status: (**xxx3**) {_ _ _ _}

Remove connection H+ to H-

### 6.1.7 FET Open

Temporarily short the heater driver FET (Q3) Gate to Source pin. Do this by connecting the outer most two pins of the Q3 (top right TO220 package).

Confirm status word: FET Failure status: (**xxx5**) {_ _ _ _}

Remove connection across FET

### 6.1.8 Preliminary Cable Test

As a preliminary test of the Wbo2 to Sensor cable, the two recent tests **6.1.4** & **6.1.5** and the earlier **5.7** test should be repeated, but with the sensor cable now connected (but **NO** sensor actually connected) and the **H+** and **H-** points at the sensor end of the cable. Note for the LSU sensor, the **BLUE** wire is the **H+** lead, and the **WHITE** wire is the **H-** lead. Remember, maybe 25% of faulty DIY-kit units we get back here have badly constructed cables. It is vitally important you have followed the cable construction guide (on the web site at **http://wbo2.com/cable/default.htm** and are confident your cable is 100%.

## *6.2 Heater Warmup – First test with Sensor Connected*

Power down the unit and connect the sensor via the sensor cable. Ensure the sensor is in free air (this means it is open to the air and not exposed to other than the atmosphere). Be extra careful to place the sensor on a non-flammable surface as the sensor body rapidly gets too hot to touch – we suggest a ceramic surface such as a saucer or a small floor tile. A thick low sided jar also works well (make sure the table doesn't burn UNDER the container). If everything is operating correctly, and if the power supply can maintain the 13.8 Volts it should be set to, then the heater will rapidly heat and internally it will glow a cherry red in close to 20 seconds. Remember, a large percentage of people who experience problems at this point have a power supply that cannot maintain the required sensor heater current (up to 3 Amps) and at least 12 Volts AT THE SAME TIME – possible symptoms of include a continually resetting processor (version string may be repeatedly sent).

Power unit **ON**. System current should start at near 1A (25°C sensor) {_ _ _ _}

Supply current should ramp to 2A over a 10 second period {_ _ _ _}

Confirm status word: **xxx0** {_ _ _ _}

While heating, confirm LED indicator shows a 1Hz (slow) ON/OFF {_ _ _ _}

Current drops to under 1A (value depends on VBatt) after 20 to 25 seconds {_ _ _ _}

After the heating cycle, confirm LED is now on solidly, with no flashing {_ _ _ _}

## *6.3 HPID lock*

Confirm status word: xx00 {_ _ _ _}

## *6.4 Vslp PID lock*

Confirm status word: 0300 {_ _ _ _}

Confirm LED indicator on {_ _ _ _}

## *6.5 Ipx check*

Ipx is the third field in the broadcast frames. Until calibration is performed, this can vary considerably between sensors, hence the wide range given here:

Ipx between 0x1E00 and 0x2400 {_ _ _ _}

Ipx delta (difference) between samples should not exceed 0x0020 {_ _ _ _}

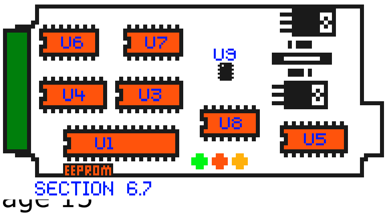
## *6.6 Use WBCAL program and confirm cal within range*

7057 sensor between (low) and (high) TBD

6066 sensor between (low) and (high) TBD

NTK sensor between (low) and (high) TBD {_ _ _ _}

## *6.7 On-Board (EEPROM) logging*



Start by inserting the 8 pin **U2** chip part (ST) **M24256**.

With a sensor connected, power on the unit and check that the logging button control the operation of on-board (or EEPROM) logging

<**tf**> Stop logging Tx Frame generation

<**s0381**> Set ASCII/HEX logging Mode 2.0

<**s0432**> Set frame delay to be 400 mSec (2.5 frame/sec)
<**w**> Save current logging mode (Tx frames off)

Now Press the black logging button on the top of the case for 5 secs to start on-board logging.
Ensure ***triple led flash***, followed by ***slow flash*** {_ _ _ _}
<**x2**> [] Logging Pointer returned should be 0x0030 {_ _ _ _}
Press button for 1 sec. Ensure double led flash {_ _ _ _}
Delay 10 seconds, push button for 1 sec, ensure **slow led flash** {_ _ _ _}
push button for 2 seconds, ensure LED is on solid, **no flashing** {_ _ _ _}
use **z** command to dump stored data (approximately 12 lines) {_ _ _ _}

after this test, return the WB02 to "normal" logging mode by issuing the
<**tf**> Stop logging Tx Frame generation
<**s03e1**> Set Binary logging Mode 2.0
<**w**> Save current logging mode (Tx frames off)

Note, after calibration

# Section 7. Calibration and Setup

## 7.1 Free Air Calibration

The **CONF**.exe software, available from the **software** links at Wbo2.com should be use to calibrate the sensor. The sensor should be sitting in free air, and should be given a few minutes to heat up before running the calibration procedure.

Note that this calibration procedure relies on the sensor seeing the average "free air" oxygen concentration normally deemed to be 20.9%. If you blow on the sensor then the carbon dioxide in your breath will be seen by the sensor (ie. percent O2 will go down). If you Leave the sensor sitting still for a long time then it will pump oxygen and the air around the sensor element will not have a 20.9% concentration. For best accuracy we suggest that you somehow organise a very low velocity supply of fresh air past the sensor while free-air calibrating. Perhaps slowly the waving the sensor element is an easy technique.

We appreciate any feedback you may provide on any obvious errors you found in the document.

# Appendix A. Change History

| | | |
|---|---|---|
| 0.6 | Sect 1.9 | Add Hex calculator information 22 August '05. |
| 0.5 | *Various*<br>Sect 1.7 | Add images indicating ICs required at various steps<br>Add ***gff*** command to dump config EEPROM memory |
| 0.4 | Page 1 | Add Section 7 "setup" & Appendix A, "Change history" |
| | Sect. 4.8.x | Change all <**i20**> <**d2045**><**i20**> to <**i20**> <**d202D**><**i20**>,<br>The 45 should have been converted from decimal to hex – 0x02d = 45. |
| | Sect. 6.7 | Add reference to U2 required for this step. |