Tech Edge

Display Configuration

Technical Information

Covers LD02, LA1 & LX1 (and other unreleased hardware)

*Version **136a** (14 Apr 2007)*
Version 136 (05 Nov 2006) See change history at end of document.

The features described here relate to the LA1, LD02, LX1, and as of the date of this document, the unreleased displays LS1 and LD3. Applicable firmware versions are indicated above. A detailed firmware and documentation history appears in the appendices.

Where a feature is relevant to one display type only, this will be indicated accordingly.

This document was originally prepared for the LD02 display, and any references to this display will be with the text "LD2". Where the description applies to all (or most) displays, the reference is simply to the "display unit".

1 Display Overview

Information to be displayed on the LA1/LD2/LD3/LX1/LS1 may be derived from a serial connection to a Tech Edge WBo2, from sources local to the display unit (analog input, VSS input), or any combination of these. From now on, I'll refer to these as "source" data.

Data may be displayed in a range of numeric formats ("digital" display), or on a linear (LD2/3 and LX1) or angular (LA1) "bargraph", where the source value is represented by a "dot position" within the bargraph.

A combination of digital and bargraph displays is considered to be a "view", and there may be up to four such views configured, the "active" view being selected using a front-panel pushbutton.

A second pushbutton may be used to control logging on the connected Tech Edge WBo2 unit; for the LD3 unit, logging controls affect local storage.

2 Display Operation

References will be made to the units' front panel pushbuttons, where the LA1 has "left" and "right" pushbuttons and the LD2 and LD3 have "top" and "bottom" pushbuttons. In this document "left" and "top" may be used interchangeably, as may "right" and "bottom".

Four possible data display "views" are available, selected using the top pushbutton. These are labelled 0 to 3, and whenever the top button is depressed, the "next" view will be selected, becoming active when the button is released. If the button is held down for more than 4 seconds, the indicated display will revert to the "current" view, and once the button is released, this display will become the default view when the unit is next powered on. Should the user prefer a lesser number of views, there is a global configuration item to reduce the number of views which may be selected.

For each of the four display views which may be active, the user can provision separately the source and formatting of both the digital display and the bargraph display; as an example, it is possible to have a display view in which the digital display shows current RPM, while the bargraph display continues to indicate lambda.

The right-hand or lower pushbutton is used to control logging on the attached TE WB unit. Whenever the button is pressed, the next logging state will be displayed, and on release the WB unit will have its logging state set accordingly. Should the pushbutton be held depressed for more than 4 seconds, the display will indicate that the next operation will be to clear the logging memory, again this occurring once the button is released. In all cases, the new state will remain displayed for about 1 second after the button is released.

When logging is active on the WB unit, the right-most decimal point will flash at a 2Hz rate as a reminder to the user.

Here are the possible state transitions, where:

"-"	no button activity
"Tp"	top pushbutton is pushed
"Tr"	top button released
"Bp"	lower button is pushed
"Br"	lower button is released
"N"	data view "n" is active
"n"	next display is selected ("n ")
"w"	current display will be set as default ("n---")
"L"	logging status is displayed ("L on", "Loff" or "Lclr")

The period of time the button is held is shown as "s" for less than 4 seconds, and "4" for greater than this interval.

Note that the top button takes priority over the lower button.

State	Action	Period	New State	Activity
N	-	-	N	default operating condition
N	Tp	-	n	select next (n) data display
n	Tr	s	N	"n" becomes active display
n	Tp	4	w	current display set as default when button released
w	Tr	-	N	new default display set
N	Bp	-	L	next logging state displayed
L	Br	s	L	logging changed, display held for 1 second
L	Bp	s	L	next logging state displayed
L	Bp	4	L	next logging state becomes "Lclr"
L	Br	4	L	logging memory cleared
L	-	1sec	N	After delay, revert to normal data display

During normal data displays, if the active data source is from the WB unit (via the serial interface), the display will show " nd" (no data) when valid serial frames cannot be detected, or " crc" if any corruption is found in the serial frame from the WBo2 unit. If the current view is displaying lambda or AFR from the WBo2 unit, the operational status of the WBo2 will be displayed if the sensor is not in a fully operational state.

During initial power-up, the pushbuttons may be used to invoke test and configuration modes as follows. To return to normal operation, remove and re-apply power to the display.

No PB pressed: Normal startup

Top: "Config" mode is established, allowing re-configuration and code downloads.

Bottom: "Test" mode. The unit is configured with the following views, used to check hardware functionality. The button needs to be held until "tESt" is displayed, and should then be released. As a reminder to the user, this mode causes the RH decimal point to flash at a rapid rate.

View 0: Vbatt
View 1: Analog voltage
View 2: Vss
View 3: LDR % (v0132)

Both: "DTEST" mode. If both PBs are held for more than 4 seconds, a display test is run, allowing the display hardware to be checked. Note that the setpoint outputs are activated as part of this test cycle.

The display uses the right-most decimal point to indicate additional status information:

logging active:	2.5Hz even pattern reminds the user that logging is active.
ipid error:	"double" flash indicates sensor has experienced Ipid error.
hpid error:	single flash indicates sensor has experienced heater error.
test mode	5Hz flash reminds user that test mode is active.

3 Configuration

Source data are stored in memory in a "source array"; configuration variables are used to control the following operations to map source data to a display view:

* For each view's digital display:
    * Selecting source data from the source array
    * Optionally scaling the source data
    * Applying a conversion function to the data
    * Optionally setting the display limits

* For each view's bargraph display:
    * Selecting the source data
    * Optionally scaling the source data
    * Applying a conversion function to the data
    * Setting bargraph display range
    * Setting optional bargraph features

Additional configuration values also manage the following:

* Mapping serial input data to the source array.

* Managing source data filtering

* Controlling the output "set-points"; these are hardware signals set when certain input conditions have been satisfied, and may be used for alarms, shift-point indicators, over-rev indicators, etc.

* Controlling certain "global" options, such as display brightness.

The LA1 and LD2 display units are configured using a PC running a serial terminal program (19k2, 1 stop bit, no parity, CR terminates lines, local echo enabled) and requires a breakout adaptor enabling connection to the PC and provision of power to the LD2. If the LD2 is powered up with the top button depressed (LA1 uses the left button), configuration mode is established, indicated by the prompt "TE LA01 v.01.34" being displayed on the terminal, and the firmware version being displayed on the digital display for one second, followed by the message "conF", indicating configuration mode.

All configuration values are saved in EE memory, and may be divided into two categories.

"Configuration Data" uses config memory (same model as WBo2) using the "g" and "s" commands to retrieve and change values, and the "w" command to save the config data and checksum. The "g" and "s" operate on config data stored in RAM, and the "w" command will commit a copy of that data to EEPROM. The "l" command will over-write the RAM copy with data from EEPROM. A default (backup) copy of config data is also stored in program memory, and may be retrieved to RAM using the "r" command.

"Table Data" is stored only in EE memory using the "e" command to write the data and the "a" command to retrieve data. Table data is used for the two user programmed lookup tables only, and as this access is directly to EE, no "w" command is required to finalise changes.

Here are some examples:

```
> g54           retrieves config value 0x54
> s54aa         set config value 0x54 to 0xAA
> r             retrieve config values from ROM to RAM (system default)
> l             retrieve config values from EE to RAM
> w             write current config values from RAM to EE (active on next restart)
> a0100         retrieve EE table data from offset 0x0100
```

```
> e010055       write EE table data at offset 0x00 to 0x55
```

Version 0130 has an additional "gff" command to dump all config variables.

It's important to note that the "e" command should only be used for writing table data, and not configuration variables. This is because config values are saved with a checksum which is calculated when the "w" command is used to save the configuration. Using the "e" command will render the checksum invalid, causing the display unit to fall back on the default configuration table which is stored in ROM.

This document uses the convention of writing hexadecimal numbers in upper case eg. 0xABCD, and the display unit requires all lower case characters in its input stream, as in the examples above. All configuration values will be described using the following format, which outlines the name of the config value, its offset for use with the "g" and "s" commands, its default value and a brief description, expanded upon in the text as required. All configuration variables are a single byte (8 bits).

name	offset	value	description
frame_cnt	0x01	0x1A	byte count of frame size, not counting header bytes

The LA/LD unit may be switched from config mode to normal frame mode by issuing an "xaa" command, and will switch from frame to config mode if it receives a valid frame with the three thermocouple values all set to 0xAAAA.

4 Source Data management

4.1 Serial Data Extraction

The first source of LD2 display data is the serial logging stream from the WB unit, and the first item to be configured is the mapping of this serial data to LD2 internal variables. The serial frame is accumulated into a buffer, and once a frame is assembled, configuration values are used to copy from this buffer to display memory. The "0x5AA5" header is not accumulated and should not be considered when calculating the offsets. Note the configuration values are split into two distinct address regions for historical reasons.

name	offset	value	destination	description
frame_cnt	0x01	0x1A		byte count of frame size, not counting header
ser_offset_0	0x02	0x05	ser_source_0	offset to serial var 0: default= Ipx
ser_offset_1	0x03	0x07	ser_source_1	offset to serial var 1: default= User_1
ser_offset_2	0x04	0x09	ser_source_2	offset to serial var 2: default= User_2
ser_offset_3	0x05	0x15	ser_source_3	offset to serial var 3: default= RPM
ser_offset_4	0x2A	0x93	thermistor	offset to serial var 4: default= thermistor
ser_offset_5	0x2B	0x17	wb_status	offset to serial var 5: default= WB status
ser_offset_6	0x2C	0x03	alt_source_0	offset to serial var 6: default= Ipx_0
ser_offset_7	0x2D	0x05	alt_source_1	offset to serial var 7: default= Ipx_1

The values are the offsets into serial input buffer after the logging frame has been assembled, and once each serial frame is assembled, data is copied from the serial buffer to the display source buffer destinations shown. The 3 MSBs may be used to transform the serial data to a new data format when it is copied to the source array.

The defaults assume the connected WB unit is configured to produce the default frame type of 0xE1, as detailed in the relevant WBo2 product pages, which also provides details of the serial stream's contents. Here is a summary of the expected frame types and offsets, noting that any model not mentioned below will be compatible with the 2A0.

Entry	Offset	2A0	2J0	3H1
0	0x01	dtime	dtime	dtime
1	0x03	wb_lambda	wb_lambda	Ipx_0
2	0x05	Ipx	Ipx	Ipx_1
3	0x07	usr1	usr1	usr1
4	0x09	usr2	usr2	usr2
5	0x0B	usr3	usr3	usr3
6	0x0D	tc1	tc1	tc1
7	0x0F	tc2	tc2	tc2
8	0x11	tc3	tc3	tc3
9	0x13	therm	vss	therm
10	0x15	rpm	rpm	rpm
11	0x17	htr_status	htr_status	c_status_l

If valid serial frames are not detected within one second, any view using a serial frame value as its source will display "nd" (no data); if frames are detected with an invalid checksum, the view will indicate "crc".

The 3 MSBs of the serial offset variable are used to translate the serial input data to the 13 bit input format required by the display conversion functions. Currently only one such translation function is provided, and is used to transform the thermistor values to 13 bit format. This conversion is also required if the thermocouple (TC) sources are to be displayed, and also for the user (USR) inputs if alternative WB frame formats are selected; please refer to the WBo2 documentation for details of this.

	Code	Function
	--------	--------
bits	7654 3210	
	0--s ssss	no translation required
	1--s ssss	10 bit input data is transformed to 13 bits
	-xxs ssss	2 bits reserved for additional translation codes

4.2 Display Source Data

The source data to be displayed is maintained in an 16 entry source array. The serial data extracted above are stored in the first four of these locations. As well as serial WB data, the LA1 has a local ADC input, and as of version 0130 will also support a local VSS/RPM input (with optional adaptor unit) and local battery voltage. The "size" column assumes the default serial configuration is used. Note the alt_source fields cannot be selected for direct display purposes, but are included in the table to show their reserved location in memory; these may be used as inputs to the EWMA filters only.

offset	variable	size**	description
0x00	ser_source_0	14b	first serial source
0x01	ser_source_1	13b	second serial source
0x02	ser_source_2	13b	third serial source
0x03	ser_source_3	16b	fourth serial source
0x04	ewma_filt_00	13b	first filtered source (digital)
0x05	ewma_filt_01	13b	second filtered source (bargraph)
0x06	ewma_filt_10	13b	third filtered source (digital)
0x07	ewma_filt_11	13b	fourth filtered source (bargraph)
0x08	Analog_source	13b	raw ADC input
0x09	Analog_filt_0	13b	ADC filtered (digital display)
0x0A	Analog_filt_1	13b	ADC filtered (bargraph)
0x0B	thermistor	13b	thermistor field in serial frame
0x0C	wb_status	16b	wideband status word
0x0D	Local VSS/RPM	16b	VSS/RPM local period
0x0E	Battery voltage	13b	VBATT filtered
0x0F	LDR	10b	LDR filtered (byte only)
0x10	alt_source_0	13b	EWMA alternate source
0x11	alt_source_1	13b	EWMA alternate source

** Default variable sizes shown

All of the entries in the first half of the array are derived from the WBo2 serial stream, and conversion functions will display an indication if this serial stream is not available.

4.3 EWMA Filtering

Because raw data may be under-filtered and contain jitter, the following config values are used to control the EWMA filtering, the results being placed in the source array (above) in offsets 0x04 to 0x07. The EWMA filters may be used on any of the input sources (local or serial), but default to filtering IPX_0 and IPX_1.

In the default case, Ipx is the first value retrieved from the serial stream, the filter source is set to zero, this being the offset in the Display Source Data (above) for WBo2 Ipx. The second EWMA filter is set to use one of the alternate input variables which has IPX_1 as its source (valid only for dual-channel WB units).

name	offset	value	dest	description
ewma_src_0	0x0E	0x00		filter input source from source array
ewma_00	0x0F	0xF0	ewma_filt_00	Ipx display filter coefficient
ewma_01	0x10	0xC0	ewma_filt_01	Ipx bargraph filter coefficient
ewma_src_1	0x11	0x10		filter input source from source array
ewma_10	0x12	0xF0	ewma_filt_10	Ipx1 display filter coefficient
ewma_11	0x13	0xC0	ewma_filt_11	Ipx1 bargraph filter coefficient

The filter used is an exponentially weighted moving average (EWMA) with a sample rate of 30.5Hz. This may be compared with a single-pole RC filter with the following time constants, noting that larger values of the coefficient correspond with longer time constants, and hence more filtering of the input value.

A(hex)	Tf
0xFF	8.36S
0xFE	4.16S
0xFD	2.76S
0xFC	2.06S
0xFA	1.37S
0xF8	1.02S
0xF4	666mS
0xF0	492mS
0xE0	229mS
0xD0	142mS
0xC0	98mS
0xA0	72mS
0x80	33mS

0x00	0		no filtering applied

4.4 ADC Filtering

The analog input also has 2 filters supplied, however there is no need to select the data source in this case. Note that the 10 bit analog input has been scaled to 13 bits prior to being written to the Source Array. Filter coefficient values are as per the Ipx filter above.

name	offset	value	dest	description
ana_ewma_d	0x1C	0xC0	Analog_filt_0	ADC display filter coefficient
ana_ewma_b	0x1D	0x80	Analog_filt_1	ADC bargraph filter coefficient

5 Digital Display

5.1 Digital Display Sources

There are four display "views" which may be configured, the "active" view being selected using the front-panel buttons. Each display view has associated with it a pair of config variable which selects the data source and the display processing used. The first group of control variables select the data source to be used for each view, and are subsequently referred to as the "source controls". These are the defaults:

name	offset	value	description	default
view_src_0	0x06	0x04	view 0: data source	filtered Ipx (digital)
view_src_1	0x07	0x04	view 1: data source	filtered Ipx (digital)
view_src_2	0x08	0x01	view 2: data source	user 1
view_src_3	0x09	0x03	view 3: data source	RPM

The values may be viewed as two hex nibbles (0xXY); generally the X nibble will be cleared and the Y nibble converted and displayed, it being used as the offset into the source array (section 4.2 above). It's value will be retrieved and processed per the algorithm selected in the next section.

If the X value is specified, it is also used as an offset into the source array, which will also be retrieved and converted. The result of conversion of both the X and Y source will then be averaged prior to being displayed; this provides a "hybrid" AFR, Lambda or temperature display to be generated (mainly for dual-channel WB units).

5.2 Digital Display Processing

Once the data has been selected (and scaled as needed), a second group of config values converts from raw numeric format to a format suitable for the display, and are referred to as the "conversion controls".

name	offset	value	description	default
view_fmt_0	0x0A	0x00	view 0: display format	Ipx>afr.2
view_fmt_1	0x0B	0x02	view 1: display format	Ipx>L.2
view_fmt_2	0x0C	0x85	view 2: display format	%.1
view_fmt_3	0x0D	0x84	view 3: display format	>RPM

The stored value may be viewed as a hex "dual" (a,b), where "b" is used to format the display, and "a" may be used by some of the conversion utilities for additional control. The "a" code is split into two 2-bit fields; "al" comprises bits 4 and 5 of the conversion code, while "ah" uses bits 6 and 7.

The "al" field may be used to control the number of decimal places displayed, although not all conversion functions require this.

The "ah" field is used to select how the view indicates that the value to be displayed exceeds programmed limits, described in a later section.

Note that the "b" values in the following are hex digits, and the "size" field is the expected variable size passed to the conversion function.

	"b"	type	size	#dp	output range		notes
	0	Ipx>afr.2	13b	2	9.00	99.99	Ipx to AFR
*	1	L16>afr.2	16b	2	9.00	99.99	Lambda_16 to AFR
	2	Ipx>L.2	13b	2	0.61	3.00	Ipx to Lambda
*	3	L16>L.2	16b	2	0.5	99.99	Lambda_16 to Lambda
	4	>RPM	16b	al	0	9999	RPM interval to RPM
	5	%.1	13b	1	0.0	100.0	User i/p to %
	6	V.2	13b	2	0.00	5.00	User i/p to Voltage
	7	>Therm	13b	0	**	**	Thermistor to dF/dC
	8	>TC	13b	0	**	**	Thermocouple to dF/dC
	9	>LIN	13b	al	0	9999	linear scaling, #dp = al
	A	>lut_1	13b	al	0	9999	user LUT 1, #dp = al
	B	>lut_2	13b	al	0	9999	user LUT 2, #dp = al
	C	>batt	13b	1	0.0	19.9	ADC to Vbatt

D	>VSS	16b	al	0	9999	VSS/RPM interval to RPM
--	more as required					

Notes:

* These limits are set by the display, not the L_16 data format.
** See section on temperature conversions for details of conversion ranges.

If the conversion code requires that the decimal point position be specified, the "al" code is interpreted as follows:

	al Code	#DP	
	-------	---	
bits	7654 3210		
	--00 bbbb	0	no DP displayed
	--01 bbbb	1	eg 21.3
	--10 bbbb	2	eg 2.13
	--11 bbbb	3	eg 1.123

The "strings" presented when conversion values for a view exceed programmed limits are selected using the "ah" bits:

	"ah" Code	Low limit	High limit
	---------	---------	----------
bits	7654 3210		
	00-- bbbb	"rich"	"LEAN"
	01-- bbbb	" r"	" L"
	10-- bbbb	" LO"	" HI"
	11-- bbbb	prog	prog

5.3 AFR and Lambda Displays

The Ipx>L.2 conversion code converts Ipx to Lambda, whilst the Ipx>afr.2 conversion code converts Ipx to AFR, in both cases using 2 decimal places. Both functions use a common lookup table in ROM, which is designed for use with the Bosch LSU 4.2 sensor only. WBo2 units using L2H2 or LSU 4.9 sensors should use the following function for the conversion for additional accuracy.

The L16>afr.2 and L16>L.2 conversion functions accept the Tech Edge "lambda_16" format (referred to as L16 in this document) from the incoming serial stream, and converts this to lambda or AFR as above. The lambda_16 format is an unsigned 16 bit number (n) representing lambda in a dual-slope transfer function:

```
If lambda is below 5.0:
        n = ( L - 0.5 ) * 8192
        L = ( n / 8192 ) + 0.5

If lambda is equal to or above 5.0:
        n = ( L - 5.0 ) * 128 + 36864
        L = ( n - 36864 ) / 128 + 5.0

Both representations provide for n = 36864 at L = 5.0.
```

The conversion to AFR requires that the fuel's stoichiometric ratio be known, and this is configurable using the following config entries:

name	offset	value	description
afr_conv_lo	0x24	0x66	AFR conversion factor low
afr_conv_hi	0x25	0x1D	AFR conversion factor high

The combined value of the above (0x1D66) is the required stoich AFR multiplied by 512, the result then being converted to hex; the default values are for a stoich point of 14.7.

If the WBo2 unit is not in an operational state, the WB status will be displayed instead:

WB state	Display
Off	" OFF"
Cal mode	" CAL"
Sense, battery Low	"H bL"
Sense, battery high	"H bH"
Sense, short	"H hS"
Sense, open	"H hO"
Sense, drive error	"H hF"
Warmup	" .."
Operating	normal lambda or AFR display

5.4 RPM Display

The >RPM conversion also uses config values to convert from the WBo2 5uS sampling units to RPM, the default being for two sparks per revolution:

name	offset	value	description
rpm_div_0	0x20	0x80	RPM dividend byte 0
rpm_div_1	0x21	0x8D	RPM dividend byte 1
rpm_div_2	0x22	0x5B	RPM dividend byte 2
rpm_div_3	0x23	0x00	RPM dividend byte 3

The config value may be calculated as:

```
200,000 * 60 / (events/rpm), where
        200,000         = WBo2 sampling rate (5uS)
        60              = Hz to RPM factor
        event/rpm       = number of spark events per rev
```

events/rpm	dividend	hex	notes
1	12,000,000	0x00B71B00	200kHz * 60
0.5	24,000,000	0x016E3600	cam rate (single cylinder)
2	6,000,000	0x005B8D80	4 cyl wasted spark (default)
etc.			

The final hex value in stored in little-endian format, ie. rpm_div_0 contains the least significant byte of the final dividend.

As the WBo2 RPM input may be used for functions other than RPM (eg. speed inputs), the number of decimal places may also be specified using the "al" bits in the conversion specifier.

5.5 VSS Display

The LA1, LD3 and LC1 (and future LD2) features a digital input which may be used for a local RPM or VSS (vehicle speed sensor) input, via an optional adaptor unit. This input is referred to as the VSS signal to differentiate it from the RPM signal from the WBo2 unit, but functions in a similar fashion. The VSS input variable is the number of 5uS periods within one period of the incoming waveform, however its value will be zero in the absence of an incoming waveform. The input values are "2-point" averaged to remove jitter.

The >VSS conversion function uses the same algorithm as the >RPM function, described above, but uses a different set of dividend values:

name	offset	value	description
vss_div_0	0x82	0x80	VSS dividend byte 0
vss_div_1	0x83	0x8D	VSS dividend byte 1
vss_div_2	0x84	0x5B	VSS dividend byte 2
vss_div_3	0x85	0x00	VSS dividend byte 3

As with the >RPM conversion function, the number of decimal places displayed may be configured using the "al" bits in the conversion specifier.

An additional configuration value is used to set the "dead time" between incoming events, allowing high frequency noise to be rejected from the incoming signal, eg. ringing on the coil input signal. The hysteresis value (dead-time) should be set to the maximum value compatible with the incoming data rate.

name	offset	value	description
rpm_dtime	0x1E	0x02	RPM dead-time

The dead-time is 512uS units, hence the default is approximately 1mS.

5.6 User Lookup Tables

Two 65 word lookup tables reside in EE at offsets 0x00FC and 0x017E. Any selected data source may be used as an input to these tables, although data must be converted to 13 bit format prior to interpolation, which may be accomplished using the view source code's "x" nibble if required.

Data in the tables are in raw numeric format, eg. a table entry of "662" will be displayed as 662, with leading zeroes suppressed, and are stored LS Byte first. These decimal values must be converted to hexadecimal prior to being transferred to the LD2. Here is an example of writing the first entry in the second table:

```
0662 => 0x029C          convert to hex: hi byte = 0x02, low byte = 0x9c
e017e9c                 write LS digit to low byte in table 2, entry 0
e017f02                 write MS digit to high byte in table 2, entry 0
```

Data is recovered from the table by retrieving the pair of values closest to the input variable, and then performing linear interpolation on these two values to find the resulting display value.

The LUT code requires that the decimal point position is specified as the upper nibble of the conversion code, in the following format:

"al" Code	#DP

```
bits    7654 3210
        --00 bbbb       0       no DP displayed
        --01 bbbb       1       eg 21.3
        --10 bbbb       2       eg 2.13
        --11 bbbb       3       eg 1.123
```

The LA1/LD2 is shipped with the following LUTs as defaults:

```
LUT1    maps local ADC input to AFR, using DIY-WB as signal source
LUT2    maps local ADC input to Lambda, using DIY-WB as signal source
```

5.7 Linear Conversions

A number of linear conversions are provided, all assuming 13 bit source data, which is the default for the WBo2 serial stream and local ADC inputs. All of these displays are unsigned.

```
%.1     Displays the source data as a percentage, with one decimal place.
---     Assumes 0-5V input range applied to either local or WBo2 user input.
        Min: 0.0%       Max: 100.0%

V.2     Displays the source data as a voltage, with two decimal places.
---     Assumes 0-5V input range applied to either local or WBo2 user input.
        Min: 0.00V      Max: 5.00V

>batt   Displays the source data as a voltage, with one decimal place. Assumes
-----   that the data has been sourced from the local battery ADC channel.
        Min: 0.0V       Max: 20.0V

>LIN    Displays the source data after a general linear transfer function is applied.
----    Display = (Input + Offset) * scale / 2**16. The following config values are used:
```

name	offset	value	description
lin_c_l	0x7E	0x00	lin offset low
lin_c_h	0x7F	0x00	lin offset high
lin_m_l	0x80	0x40	lin scale low
lin_m_h	0x81	0x1F	lin scale high

The offset is a signed (twos-complement) number which is first added to the input variable. The result of this is then scaled by multiplying by the scale value (unsigned), then dividing by 2**16. The default values perform the equivalent function to the %.1 conversion above. The number of decimal places is controlled using the "al" bits in the view conversion code. The offset word may be signed (two's complement) however the scale factor is interpreted as an unsigned number.

5.8 Temperature Conversions

>Therm converts thermistor data to degrees C or F, using a lookup table in ROM to provide the conversion. The thermistor field is extracted from the serial frame and stored in the source array at offset 0x0B. The conversion routine expects 13 bit input data, hence the source data should be selected with the (default) 13 bit conversion code selected (section 4.1).

The temperature units displayed (dF or dC) is globally selected for all temperature displays using a bit in the tri_modes control variable (section 8.4). Clearing tri_modes bit 6 (default) will display temperature in dC, in the range -59dC to 209dC; setting the bit will display temperatures in the range -74dF to 408dF.

>TC converts thermocouple data to degrees C or F, again using a ROM lookup table. The thermocouple data needs to be extracted from the serial stream into one of the applicable source array fields reserved for serial data. 13 bit input data is expected, so again the 10b->13b scaling option should be chosen (section 4.1).

An additional option applies to the thermocouple conversion function, also controlled with a bit in tri_modes (section 8.4). Clearing bit 5 causes the thermistor value in the source array to be used for cold-junction compensation (CJC), while setting it disables CJC.

This is best explained by example. Let's setup view 2 to display thermocouple 1 in dF, assuming a 2A0 WBo2 unit, with thermistor (cold-junction) compensation.

```
> s048d         derive ser_source_2 from thermocouple_1 fields in serial stream
                MSB is set to cause 10b->13b conversion
> s0802         view 2 data source is ser_source_2
> s0c88         Lower nibble (0x8) sets conversion function to >TC
                "al" disables all DPs
                "ah" selects LO/HI as limit text (if limits are used)
> s2e4e         turn on dF bit, clear CJC bit, leave other bits intact
```

5.9 Digital Display Limits

Each digital view has associated with it optional limits, such that if the display value (after the conversion function) is below the lower limit, or above the upper limit, the display will indicate this to the user by substituting a "text" message on the display. One of four messages can be selected using the "ah" field of the view's conversion code.

	"ah" Code	Low limit	High limit
bits	7654 3210		
	00-- bbbb	"rich"	"LEAN"
	01-- bbbb	" r"	" L"
	10-- bbbb	" LO"	" HI"
	11-- bbbb	prog	prog

For each view there is a low and a high limit available. Setting either limit to zero will cause that limit to be ignored. The following config values specify the default limits:

name	offset	value	description
0_lim_lo_l	0x3C	0xE8	view 0 lower limit, low byte
0_lim_lo_h	0x3D	0x03	view 0 lower limit, high byte
0_lim_hi_l	0x3E	0xB8	view 0 upper limit, low byte
0_lim_hi_h	0x3F	0x0B	view 0 upper limit, high byte
1_lim_lo_l	0x40	0x46	view 1 lower limit, low byte
1_lim_lo_h	0x41	0x00	view 1 lower limit, high byte
1_lim_hi_l	0x42	0xC8	view 1 upper limit, low byte
1_lim_hi_h	0x43	0x00	view 1 upper limit, high byte
2_lim_lo_l	0x44	0x00	view 2 lower limit, low byte
2_lim_lo_h	0x45	0x00	view 2 lower limit, high byte
2_lim_hi_l	0x46	0x00	view 2 upper limit, low byte
2_lim_hi_h	0x47	0x00	view 2 upper limit, high byte
3_lim_lo_l	0x48	0x00	view 3 lower limit, low byte
3_lim_lo_h	0x49	0x00	view 3 lower limit, high byte
3_lim_hi_l	0x4A	0x00	view 3 upper limit, low byte
3_lim_hi_h	0x4B	0x00	view 3 upper limit, high byte

The limits are applied independent of the number of decimal places applied, so from the above we can summarise the defaults as:

View	Default View	Low limit	High Limit
0:	AFR	10.00 (0x03E8)	30.00 (0x0BB8)
1:	Lambda	0.70 (0x0046)	2.00 (0x00C8)
2:	User_1	--	--
3:	RPM	--	--

5.10 Limit Strings

As well as the in built limit strings, the user may configure the display by using the following configuration values and setting the conversion code "ah" field to "11"

name	offset	value	description	
u_text_lo_8	0x4C	0x50	clamp 3 lower text,	"r___"
u_text_lo_4	0x4D	0x10	clamp 3 lower text,	"_i__"
u_text_lo_2	0x4E	0x58	clamp 3 lower text,	"__c_"
u_text_lo_1	0x4F	0x74	clamp 3 lower text,	"___h"
u_text_hi_8	0x50	0x38	clamp 3 upper text,	"L___"
u_text_hi_4	0x51	0x79	clamp 3 upper text,	"_E__"
u_text_hi_2	0x52	0x77	clamp 3 upper text,	"__A_"
u_text_hi_1	0x53	0x37	clamp 3 upper text,	"___n"

The "text" values are raw digit codes applied to the 7 segment displays:

b7:	decimal point	
b6:	G segment	middle bar
b5:	F segment	top left
b4:	E segment	bottom left
b3:	D segment	bottom
b2:	C segment	bottom right
b1:	B segment	top right
b0:	A segment	top

6.0 Bargraph Controls

6.1 Bargraph Data Sources

For each display view, there is an associated bargraph view, although the LD2 and LA1 default to a common Lambda bargraph for all views. Each bargraph view is configured using 8 config variables containing bit fields which control the data source and the data transformation applied to each bargraph type.

name	offset	value	description	source	Scale

bar_src_0	0x14	0x05	view 0: bargraph src	filtered Ipx	none
bar_src_1	0x15	0x05	view 1: bargraph src	filtered Ipx	none
bar_src_2	0x16	0x05	view 2: bargraph src	filtered Ipx	none
bar_src_3	0x17	0x05	view 3: bargraph src	filtered Ipx	none

The values may be viewed as two hex nibbles (0xCD); generally the C nibble will be cleared and the D nibble converted and displayed, it being used as the offset into the source array (section 4.2 above). It's value will be retrieved and processed per the algorithm selected in the next section.

If the C value is specified, it is also used as an offset into the source array, which will also be retrieved and converted. Depending on the bargraph options chosen, the result of conversion of both the C and D source may then be averaged prior to being displayed; this provides a "hybrid" AFR, Lambda or temperature display to be generated (mainly for dual-channel WB units). Alternatively, the LA1 may be configured to display the converted C and D values as different coloured dot positions in the bargraph, this being discussed in a later section.

6.2 Bargraph Conversions

An additional set of config values control the transformation applied to the scaled data prior to the final bargraph conversion:

name	offset	value	description	Conversion
bar_fmt_0	0x18	0x00	view 0: bargraph fmt	Ipx>L, L->R, dot mode
bar_fmt_1	0x19	0x00	view 1: bargraph fmt	Ipx>L, L->R, dot mode
bar_fmt_2	0x1A	0x00	view 2: bargraph fmt	Ipx>L, L->R, dot mode
bar_fmt_3	0x1B	0x00	view 3: bargraph fmt	Ipx>L, L->R, dot mode

The format codes are split into 2 fields (E,F), with the "F" value defining the intermediate transformation. The converted units need to be understood so that the following section on setting the bargraph limit and scale can be interpreted. The "E" nibble is used to enable various bargraph options, described in a later section.

	fmt Code	"F"	Conversion
bits	7654 3210		
	---- 0000	0	Ipx>Lambda
	---- 0001	1	>lut_1
	---- 0010	2	>lut_2
	---- 0011	3	>rpm
	---- 0100	4	>vss
	---- 0101	5	>therm
	---- 0110	6	>tc
	---- 0111	7	L16>Lambda
	---- 1000	8	no translation
			others as required

The conversion functions are a subset of those used in the digital display conversion, and they are summarised here.

Function	Input	Output	Min	Max
Ipx>Lambda	14b	Lambda * 100 * 128	7796	65535
>lut_1	13b	set by user	0	65535
>lut_1	13b	set by user	0	65535
>rpm	16b	RPM	0	65535
>vss	16b	RPM	0	65535
>therm	13b	dC	-59	209
		dF	-74	408
>tc	13b	dC	-59	1427
		dF	-74	2600
L16>Lambda	16b	Lambda * 100 * 128	6400	65535

The controls for the temperature conversions are identical to those used in the digital display; please refer to sections 5.8 and 8.4 for details.

6.3 Bargraph Limit/Scale

Once the source data has been scaled and converted, the bargraph is configured using a low (rich) setpoint, and a scale value, the required values being dependent on the range of values which are returned from the conversion code chosen.

name	offset	value	description	
b0_lo_l	0x54	0x00	bargraph 0:	lower limit, low byte
b0_lo_h	0x55	0x28	bargraph 0:	lower limit, high byte
b0_sc_l	0x56	0xF3	bargraph 0:	scale, low byte
b0_sc_h	0x57	0x00	bargraph 0:	scale, high byte
b1_lo_l	0x58	0x00	bargraph 1:	lower limit, low byte
b1_lo_h	0x59	0x28	bargraph 1:	lower limit, high byte
b1_sc_l	0x5A	0xF3	bargraph 1:	scale, low byte

```
b1_sc_h         0x5B    0x00    bargraph 1:     scale, high byte

b2_lo_l         0x5C    0x00    bargraph 2:     lower limit, low byte
b2_lo_h         0x5D    0x28    bargraph 2:     lower limit, high byte
b2_sc_l         0x5E    0xF3    bargraph 2:     scale, low byte
b2_sc_h         0x5F    0x00    bargraph 2:     scale, high byte

b3_lo_l         0x60    0x00    bargraph 3:     lower limit, low byte
b3_lo_h         0x61    0x28    bargraph 3:     lower limit, high byte
b3_sc_l         0x62    0xF3    bargraph 3:     scale, low byte
b3_sc_h         0x63    0x00    bargraph 3:     scale, high byte
```

The value from the conversion function has the lower limit subtracted from it, and should an overflow occur, a low indication given (equivalent to a rich indication). The result is then divided by 2**16/scale_value, the resultant value being the bargraph bit which will be illuminated, with a high (lean) indication being given if the range of the bargraph range is exceeded. Obviously, the magnitude of the limit and scale values are determined by the magnitude of the values returned from the chosen conversion function.

As an example, let's examine the (default) values for the Ipx>Lambda conversion function. Data from the Ipx>Lambda converter is in the format of Lambda * 100 * 128. The lower (rich) limit is therefore:

```
limit   = Limit(r) * 100 * 128
        = 0.8  * 100 * 128
        = 0x2800
```

Once this value is subtracted from the lambda value, the result is still in units of Lambda * 100 * 128. This value is multiplied by the scale value and then divided by 2**16. If the result exceeds the number of bargraph digits, a lean indication is given, otherwise the bargraph will display the resulting value. In the following the "num_intervals" value is one less than the number of segments in the bargraph, this being due to the lower limit being indicated with the left-most segment illuminated.

```
scale   = num_intervals * 2**16 / (Range(L) * 100 * 128 )
        = 19 * 2**16 / (0.4 *100 * 128) (LD23)
        = 0xF3 (for LD2/3)
        OR
        = 29 * 2**16 / (0.4 * 100 * 128) (LA1)
        = 0x173 (for LA1)
        OR
        = 24 * 2**16 / (0.4 *100 *128) (LC1)
        = 0x133 (for LX1)
```

Data from all other conversion functions are in "raw display units" ie. in the format in which they would be displayed in a digital view, ignoring decimal places As an example, a display value of 50.1 from one of the lookup tables (1 dp), has a raw value of 501 or 0x01F5.

As another example, if we wanted an RPM display from 0 to 10000 RPM, the lower limit would be set at 0, and the scale factor set to:

```
scale   = num_segments * 2**16 / Range(RPM)
        = 0xBE (for LA1)
        = 0x7D (for LD2/3)
        = 0x9D (for LX1)
```

6.4 Bargraph Options

The "E" bits of the bargraph conversion codes are used to select certain bargraph mode options. As the various displays have different hardware capabilities, some of the options are available only on the LA1, being ignored if selected on the LD2, LD3 or LC1.

The simplest mode of operation is "dot" mode, where a single segment is illuminated, the position of the segment reflecting the input value. The bargraph may be configured as "L->R", with lower values at the left of the display, or as "R->L" where the direction is reversed.

If the selected source is beyond the programmed bargraph limits, the 2 segments at that end of the bargraph will be illuminated.

```
                "E" Code        action
                --------        ------
        bits    7654 3210
                0000 ----       single L->R dot mode
                0001 ----       single R->L dot mode
        *       0010 ----       dual   L->R dot mode
        *       0011 ----       dual   R->L dot mode
        **      01x0 ----       single L->R bar mode
        **      01x1 ----       single R->L bar mode
        *       1xxx ----       unused

*       LA1 only
**      LA1 only
```

Reverse mode is selected when bit 4 is set, otherwise L->R direction is established.

When bit 5 is cleared, if multiple sources are selected (see section 6.1) these will be averaged prior to being displayed. For the LA1 only, if bit 5 is set and multiple sources are selected, the second source (C nibble) will be display in red, while the primary value (D nibble) will be displayed in green.

Dot mode is selected when bit 6 is cleared. Setting this bit has no effect on the LD2/3, and will select solid "bar" mode on the LA1 and LC1. Note that "dual" and "bar" modes are mutually exclusive.

6.5 Colour "highlights" (LA1 only)

The LA1 has bi-colour bargraph LEDs. By default the green display is used, with the option to "highlight" high and low values by displaying these as red. To this end, for each bargraph view there is a low and high colour "limit" value. If the displayed segment is lower than the lower limit, or greater than the upper limit, the segment will be displayed in red. The default values leave the limits disabled.

If dual dot mode, colour highlights will be disabled.

In solid bar mode, when colour highlights are exceeded, all illuminated LEDs will have the highlight applied.

The units used are bargraph dot "positions" (0-29 for LA1).

```
name            offset  value   description
----            ------  -----   -----------
b0_lim_l        0x64    0x00    bargraph 0:     lower colour clamp
b0_lim_h        0x65    0x20    bargraph 0:     upper colour clamp
b1_lim_l        0x66    0x00    bargraph 1:     lower colour clamp
b1_lim_h        0x67    0x20    bargraph 1:     upper colour clamp
b2_lim_l        0x68    0x00    bargraph 2:     lower colour clamp
b2_lim_h        0x69    0x20    bargraph 2:     upper colour clamp
b3_lim_l        0x6A    0x00    bargraph 3:     lower colour clamp
b3_lim_h        0x6B    0x20    bargraph 3:     upper colour clamp
```

6.6 Reference indicators (LA1 only)

LA1 has the option to flash one of the bargraph LEDs to indicate the reference (or stoich) point of the bargraph. Setting the control variable to zero causes this feature to be disabled.

```
name            offset  value   description
----            ------  -----   -----------
b0_ref          0x86    0x00    bargraph 0:     reference indicator
b1_ref          0x87    0x00    bargraph 0:     reference indicator
b2_ref          0x88    0x00    bargraph 1:     reference indicator
b3_ref          0x89    0x00    bargraph 1:     reference indicator
```

The 5 LSBs are used to indicate the segment to be used as the reference indicator (1..29).

In "dot" or "dual" modes, if the MSB is cleared, this segment will be illuminated in addition to the normal display, while if the MSB is set, the reference segment will flash. If bit 6 is set, the illuminated segment will be red, otherwise it will be green.

In "bar" mode the state of the selected segment will be exclusive-ORed with the bargraph display; setting the MSB of the control variable will flash the selected segment.

6.7 LX1 Display controls

LX1 has additional LCD segments which can be enabled for each view, controlled by the following configuration values.

```
name            offset  value   description
----            ------  -----   -----------
v0_lbl_l        0x64    0x00    view 0:         aux digit controls
v0_lbl_h        0x65    0x20    view 0:         aux segments
v1_lbl_l        0x66    0x00    view 1:         aux digit controls
v1_lbl_h        0x67    0x20    view 1:         aux segments
v2_lbl_l        0x68    0x00    view 2:         aux digit controls
v2_lbl_h        0x69    0x20    view 2:         aux segments
v3_lbl_l        0x6A    0x00    view 3:         aux digit controls
v3_lbl_h        0x6B    0x20    view 4:         aux segments
```

The digit values are raw digit codes applied to the 7 auxiliary segment display:

```
b7:     unused
b6:     G segment       middle bar
b5:     F segment       top left
b4:     E segment       bottom left
b3:     D segment       bottom
b2:     C segment       bottom right
b1:     B segment       top right
b0:     A segment       top
```

The aux segments enable the following:

b3:	Decimal point (duplicate method)
b2:	bar direction indicators
b1:	! symbol
b0:	degree symbol

Note that under the following conditions, the programmed display will be overridden, and the "!" symbol will be flashing at 1Hz.

* When logging is enabled, the aux digit will change to "L".

* When a heater PID error is encountered in the serial stream, the aux digit will change to "H".

* When a WB PID error occurs, the aux digit will change to a "P".

7 Setpoint Outputs

The LA1 (and future versions of the LD2) contain three, active-low, open-drain outputs which may be asserted when certain input conditions are satisfied. The LD3 and LC1 have a single setpoint output, so references to SP2 and SP3 may be ignored for these displays.

Each output can be asserted when one input condition alone is satisfied, when either of two input conditions are met ("or" function), or when both input conditions are met ("and" function).

The outputs are controlled using config values below.

name	offset	value	description
sp1_ctl_a	0x6C	0x00	Setpoint 1 "A" source and control byte
sp1_tst_a_l	0x6D	0x00	Setpoint 1 "A" test value low
sp1_tst_a_h	0x6E	0x00	Setpoint 1 "A" test value high
sp1_ctl_b	0x6F	0x00	Setpoint 1 "B" source and control byte
sp1_tst_b_l	0x70	0x00	Setpoint 1 "B" test value low
sp1_tst_b_h	0x71	0x00	Setpoint 1 "B" test value high
sp2_ctl_a	0x72	0x00	Setpoint 2 "A" source and control byte
sp2_tst_a_l	0x73	0x00	Setpoint 2 "A" test value low
sp2_tst_a_h	0x74	0x00	Setpoint 2 "A" test value high
sp2_ctl_b	0x75	0x00	Setpoint 2 "B" source and control byte
sp2_tst_b_l	0x76	0x00	Setpoint 2 "B" test value low
sp2_tst_b_h	0x77	0x00	Setpoint 2 "B" test value high
sp3_ctl_a	0x78	0x00	Setpoint 3 "A" source and control byte
sp3_tst_a_l	0x79	0x00	Setpoint 3 "A" test value low
sp3_tst_a_h	0x7A	0x00	Setpoint 3 "A" test value high
sp3_ctl_b	0x7B	0x00	Setpoint 3 "B" source and control byte
sp3_tst_b_l	0x7C	0x00	Setpoint 3 "B" test value low
sp3_tst_b_h	0x7D	0x00	Setpoint 3 "B" test value high

The setpoint source and control bytes are formed as two hex nibbles (g,h), where "h" is used to select the data source (word offset into source array) and the "g" code determines how the test values (16 bit word) are interpreted. The test values are in the same native format as the selected source variable.

bit	"A" test	"B" test
7:	A test enable	B test enable
6:		
5:		"AND" logic
4:	A "GT" comparison	B "GT" comparison

If the "A" test enable bit (A7) is cleared, no further action is taken, and the setpoint output is deactivated.

If the "A" test enable bit is set, the "A" source data is retrieved and compared with the "A" setpoint test value, this comparison being controlled by the A4 bit.

If the A4 bit is set, and the source data is larger than the "A" test value, the test is deemed to be true. If the A4 bit is cleared and the source data is less than the "A" test value, the "A" test result is also true.

Once the "A" test is completed, the "B" test enable bit (B7) is examined. If cleared, the "A" test result is used to control the output, a true result causing the output to be asserted.

If the B7 bit is true, the "B" test is performed is the same fashion as the "A" test. The "B" result is then ANDed with the "A" result if the B5 bit is set, or ORed with the "A" result if B5 is cleared. The final result is then used to control the output.

As both tests have individual sources, it is possible to create tests such as:

Assert alarm if Ipx is above "A" limit AND if boost is above "B" limit

```
        Setup "A" threshold
        Set "A" source = Ipx
        Set A7 and A4 to enable "A" test (GT)
        Setup "B" threshold
        Set "B" source = U1
        Set B7, B4 and B5 to enable "B" test (GT) and "AND" function

Assert alarm if Ipx is below "A" limit or above "B" limit
        Setup "A" threshold
        Set "A" source = Ipx
        Set A7 and A4 to enable "A" test (GT)
        Setup "B" threshold
        Set "B" source = Ipx
        Set B7 to enable "B" test (LT) and "OR" function

Assert alarm if RPM is above "A" limit
        Setup "A" threshold
        Set "A" source = RPM (5uS units)
        Set A7 to enable "A" test (LT, as units are period)
        Clear B7 to disable "B" test
```

8 Global Options

8.1 Display Update Intervals

The update period of both the digital and bar display may be controlled using the following entries. The bargraph delay is the number of 10mS "timer ticks" between iterations; the main display period is two more than the number of 50mS "semaphore ticks" between updates.

name	offset	value	description
disp_dly	0x3A	0x03	250mS main display update rate
bar_dly	0x3B	0x02	20mS bargraph update rate

8.2 Default View

The default display view may be selected. The default view can also be changed and saved using the front panel buttons.

name	offset	value	description
def_view	0x2F	0x00	defaults to display view 0

8.3 View Maximum

Users may configure the LD2/LA1 to use less than the four possible display views by setting the maximum view allowable. The programmed value is used to clamp the sequence available for front panel changes to the selected view:

name	offset	value	available views
max_view	0x1F	0x03	0, 1, 2, 3
		0x02	0, 1, 2
		0x01	0, 1
		0x00	0

8.4 Misc Operating Options

A number of operating modes may be controlled using the tri_modes config value:

name	offset	value	description		
tri_modes	0x2E	0x0E	B7:	1 = config (pc) mode	0 = operating mode
			B6:	1 = dF	0 = dC
			B5:	0 = CJC	1 = no correction
			B4:	1 = Lss Tx (1200 baud)	0 = 19200 baud
			B3:	1 = decr on store	0 = no decr
			B2:	1 = enable WB status	0 = no WB status
			B1:	1 = enable PID display	0 = no PID display

Setting B7 forces the LD2 to start in configuration mode, in which the CIUI is run instead of the usual serial extraction code. This bit must be cleared and conf written to EE prior to normal startup.

When cleared, B6 selects degrees Celsius as the default temperature conversion unit; set this bit for dF.

When cleared, B5 enables cold junction compensation; setting this bit disables CJC.

Setting B4 starts the serial extraction code at 1200 baud for use with the Lss Tx stream.

B3 controls how the current display will be written to EE when the top button is held down. When the button is first pushed, the "next" display is selected; keeping the button depressed for more than 4 seconds causes the unit prepares to write the default display to EE for future restarts. When B3 is set, the display will revert to the previous selected display; when B3 is cleared, the current display will be written, requiring the user to cycle through all displays to write the correct one.

B2, when set, causes the lambda and AFR format codes to display the sensor status, as follows:

WB state	Display
Off	" OFF"
Cal mode	" CAL"
Sense, battery Low	"H bL"
Sense, battery high	"H bH"
Sense, short	"H hS"
Sense, open	"H hO"
Sense, drive error	"H hF"
Warmup	" .."
Operating	normal lambda or AFR display

B1 controls whether the right-most decimal point will flash during WB PID unlock conditions. Any error or "unlock" in the heater control loop will cause a single flash of the right-most decimal point, while a double flash reports an error in the Ip control loop.

8.5 LDR Table

Users may wish to operate the display unit in differing ambient light conditions, to which end a lookup table is provided to map between ADC readings of the LDR and output brightness levels. This table may also be adjusted when using different LED display components.

During high ambient light levels, the LDR ADC reading will be low, hence the following table should be read from "bright" to "dark". Higher configuration values will cause a higher display brightness.

name	offset	value	description
ldr_y_0	0x30	0x80	disp bright factor 0 (high ambient levels)
ldr_y_1	0x31	0x40	disp bright factor 1
ldr_y_2	0x32	0x38	disp bright factor 2
ldr_y_3	0x33	0x30	disp bright factor 3
ldr_y_4	0x34	0x28	disp bright factor 4
ldr_y_5	0x35	0x20	disp bright factor 5
ldr_y_6	0x36	0x18	disp bright factor 6
ldr_y_7	0x37	0x10	disp bright factor 7
ldr_y_8	0x38	0x10	disp bright factor 8 (low ambient levels)

Note that successive value pairs cannot differ by more than 128 (0x80).

8.6 Bargraph Brightness Controls

Due to the brightness mismatches between the 7 segment displays and the bargraph LEDs, config values are supplied to scale the pulse widths computed and used when the green or red bargraph segments are used. The values are 4+4 (int+fraction), so the defaults are effectively 2.5 and 5.0. Due to the very long turn off time of the Micrel high-side driver, the clamp value is required to prevent segment "bleed through" at low ambient light levels. As this value is hardware dependent, it is recommended that the default value be used.

The scaling values are not used on the LD2, as the display and bargraph LEDs are matched.

The seg_pw is used as the default pulse width when the display unit is powered up in configuration mode, and in operational mode is used prior to the LDR filtering being established.

name	offset	value	description
seg_pw	0x26	0x40	default digit pulse width
led_sc_red	0x27	0x28	red bargraph pulse width scale factor
led_sc_green	0x28	0x50	green bargraph pulse width scale factor
seg_pw_clamp	0x29	0x10	LED pulse width clamp

8.7 LDR Filter

To prevent "strobing" effects on the display due to variations in ambient light levels, a filter is applied to the LDR ADC values. Larger coefficient values will cause a greater smoothing effect.

name	offset	value	description
ldr_ewma	0x39	0xF0	LDR filter coefficient

9 Appendices

9.1 Config Offsets (by view)

		Offs	Def	Sec	Notes
View 0:	Dig Source	0x06	0x04	5.1	ipx_filt_0, no scaling
	Dig Conv	0x0A	0x00	5.2	Ipx>AFR.2, limit text = rich/LEAN
	Limit Low	0x3C,3D	0x03E8	5.9	10.00
	Limit High	0x3E,3F	0x0BB8	5.9	30.00
	Bar Source	0x14	0x05	6.1	ipx_filt_1, no scaling
	Bar Conv	0x18	0x00	6.2	Ipx>L
	Bar Low	0x54,55	0x2800	6.3	0.8
	Bar Scale	0x56,57	0x0100	6.3	0.4
View 1:	Dig Source	0x07	0x04	5.1	ipx_filt_0, no scaling
	Dig Conv	0x0B	0x02	5.2	Ipx>Lambda.2, limit text = rich/LEAN
	Limit Low	0x40,41	0x0046	5.9	0.70
	Limit High	0x42,43	0x00C8	5.9	2.00
	Bar Source	0x15	0x05	6.1	ipx_filt_1, no scaling
	Bar Conv	0x19	0x00	6.2	Ipx>L
	Bar Low	0x58,59	0x2800	6.3	0.8
	Bar Scale	0x5A,5B	0x0100	6.3	0.4
View 2:	Dig Source	0x08	0x01	5.1	User 1, no scaling
	Dig Conv	0x0C	0x85	5.2	%.1, limit text = LO/HI
	Limit Low	0x44,45	0x0000	5.9	--
	Limit High	0x46,47	0x0000	5.9	--
	Bar Source	0x16	0x05	6.1	ipx_filt_1, no scaling
	Bar Conv	0x1A	0x00	6.2	Ipx>L
	Bar Low	0x5C,5D	0x2800	6.3	0.8
	Bar Scale	0x5E,5F	0x0100	6.3	0.4
View 3:	Dig Source	0x09	0x03	5.1	RPM, no scaling
	Dig Conv	0x0D	0x84	5.2	>RPM, limit text = LO/HI
	Limit Low	0x48,49	0x0000	5.9	--
	Limit High	0x4A,4B	0x0000	5.9	--
	Bar Source	0x17	0x05	6.1	ipx_filt_1, no scaling
	Bar Conv	0x1B	0x00	6.2	Ipx>L
	Bar Low	0x60,61	0x2800	6.3	0.8
	Bar Scale	0x62,63	0x0100	6.3	0.4

Change History - Tech Edge Pty Ltd

136a	13 Apr 07	spell-check, PDF doc changes	(v0136)
136	5 Nov 06	LX1 added	(v0136)
134	9 Sep 06	L-16 added, LD3 and LX1	(v0134)
133a	7 Jul 06	clean up	(v0133)
133	3 Jun 06	Dual channel changes	(v0133)
132	23 Oct 05	startup controls	(v0132)
131a	16 Oct 05	clean up, param address changes	(v0131)
131	14 Oct 05	display limits changed	(v0131)
130b	17 Sep 05	Bargraph feature extensions	(v0130)
130a	11 Sep 05	Additional conversion types	(v0130)
130	13 Aug 05	Linear conversion	(v0130)
129	25 Jun 05	Programmed outputs	(v0129)
128	22 Jun 05	Version number changes	(v0128)
14	16 Jun 05	clamp "strings" in flash	(v0128)
13	9 Apr 05	logging controls updated	(v0126)
12	25 Mar 05	PID indicator controls	(v0125)
11	20 Mar 05	Display options described	(v0125)
10	16 Mar 05	Bargraph enhancements	(v0124)
9a	13 Mar 05	Clamping controls enhanced	(v0123)
9	13 Mar 05	Clamping controls	(v0123)
8	12 Mar 05	WB status controls	(v0123)
7	11 Mar 05	therm, tc added	(v0123)
6	6 Mar 05	disp_dly changed, LUTs added	(v0123)
5	3 Mar 05	tri_modes added	(v0122)
4	27 Feb 05	minor tweaks	(v0121)
3	26 Feb 05	Major changes	(v0120)
2	26 Feb 05	minor updates	(v0119)
1	27 Jan 05	first draft	