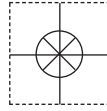


**1 YEAR**  
WARRANTY

**OMEGA CARE**<sup>SM</sup>  
Extended Warranty  
Program



# User's Guide

*Shop online at*

**omega.com**<sup>®</sup>

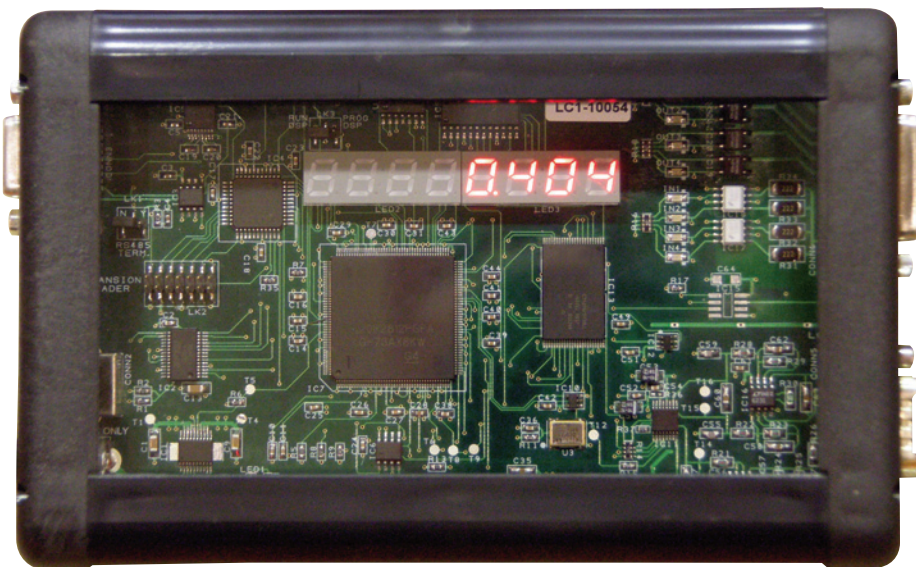
Ω OMEGA<sup>®</sup>

*omega.com*

*e-mail: info@omega.com*

*For latest product manuals:*

*omegamanual.info*



**ISO 9001**  
CERTIFIED  
CORPORATE QUALITY

STAMFORD, CT

**ISO 9001**  
CERTIFIED  
CORPORATE QUALITY

MANCHESTER, UK

## **LCIC-WIM-BEN** **High Speed** **Load Cell Interface Card**



OMEGAnet® Online Service  
omega.com

Internet e-mail  
info@omega.com

### **Servicing North America:**

**U.S.A.:**  
ISO 9001 Certified  
One Omega Drive, P.O. Box 4047  
Stamford, CT 06907-0047  
TEL: (203) 359-1660  
FAX: (203) 359-7700  
e-mail: info@omega.com

**Canada:**  
976 Bergar  
Laval (Quebec) H7L 5A1, Canada  
TEL: (514) 856-6928  
FAX: (514) 856-6886  
e-mail: info@omega.ca

### **For immediate technical or application assistance:**

**U.S.A. and Canada:** Sales Service: 1-800-826-6342/1-800-TC-OMEGA®  
Customer Service: 1-800-622-2378/1-800-622-BEST®  
Engineering Service: 1-800-872-9436/1-800-USA-WHEN®

**Mexico:**  
En Español: (001) 203-359-7803  
FAX: (001) 203-359-7807  
e-mail: espanol@omega.com  
info@omega.com.mx

### **Servicing Europe:**

**Czech Republic:** Frystatska 184, 733 01 Karviná, Czech Republic  
TEL: +420 (0)59 6311899  
FAX: +420 (0)59 6311114  
Toll Free: 0800-1-66342  
e-mail: info@omegashop.cz

**Germany/Austria:** Daimlerstrasse 26, D-75392 Deckenpfronn, Germany  
TEL: +49 (0)7056 9398-0  
FAX: +49 (0)7056 9398-29  
Toll Free in Germany: 0800 639 7678  
e-mail: info@omega.de

**United Kingdom:**  
ISO 9001 Certified  
One Omega Drive, River Bend Technology Centre  
Northbank, Irlam, Manchester  
M44 5BD United Kingdom  
TEL: +44 (0)161 777 6611  
FAX: +44 (0)161 777 6622  
Toll Free in United Kingdom: 0800-488-488  
e-mail: sales@omega.co.uk

---

It is the policy of OMEGA Engineering, Inc. to comply with all worldwide safety and EMC/EMI regulations that apply. OMEGA is constantly pursuing certification of its products to the European New Approach Directives. OMEGA will add the CE mark to every appropriate device upon certification.

The information contained in this document is believed to be correct, but OMEGA accepts no liability for any errors it contains, and reserves the right to alter specifications without notice.

**WARNING:** These products are not designed for use in, and should not be used for, human applications.

# **Table of Contents**

## **1. Introduction**

- 1.1 General Description
- 1.2 Typical Applications

## **2. Installing the board in the PC**

## **3. Utilities**

- 3.1 Setup & Running
- 3.2 The Calibration Utility
  - 3.2.1 General
  - 3.2.2 The Calibrations Library
  - 3.2.3 Parameters
  - 3.2.4 Calibration Efficiency (CE)
- 3.3 The Settings Utility
  - 3.3.1 The Menu Bar
    - 3.3.1.1 Tools / Analog Output
    - 3.3.1.2 Tools / Baud Rate for SCI port
    - 3.3.1.3 Tools / General Setpoints
  - 3.3.2 Parameters
    - 3.3.2.1 Communication
      - 3.3.2.1.1 Port
      - 3.3.2.1.2 Baud Rate
      - 3.3.2.1.3 RS485 Address
      - 3.3.2.1.4 Get results immediately
    - 3.3.2.2 Auto Zero
      - 3.3.2.2.1 Activate
      - 3.3.2.2.2 Max Zero
      - 3.3.2.2.3 Min Zero
      - 3.3.2.2.4 Time limit
    - 3.3.2.3 Start Fill-mode
      - 3.3.2.3.1 Fill-mode starts automatically upon card reset
    - 3.3.2.4 Filtering
    - 3.3.2.5 Filling Definition

## **4. Programming your Application**

- 4.1 Commands
- 4.2 Parameters
- 4.3 LCIC-WIM ActiveX
  - 4.3.1 Start/Stop Communication
  - 4.3.2 Variables
  - 4.3.3 Filters
  - 4.3.4 Fast Mode
  - 4.3.5 Misc.

## **Appendices**

### **A. I/O & the LED Display**

- A.1 General Notes about the I/O
- A.2 Connecting External Devices to the Inputs
- A.3 LCIC-WIM I/O & Analog Out
- A.4 The LED Display

### **B. Scaling the Load Cell Input**

### **C. Load Cell Connections**

### **D. USB, RS232 & RS485**

- D.1 Communication Parameters
- D.2 Baud Rate
- D.3 General RS232 Terminal
- D.4 Serial Communication & PC Power On/Off
- D.5 RS485

### **E. Fill Mode**

- E.1 Introduction
- E.2 Hardware Inputs
- E.3 Hardware Outputs
- E.4 Filling Parameters
  - E.4.1 Filling By = Weight
    - E.4.1.1 Filling by Weight Parameters
    - E.4.1.2 Auto Tare
    - E.4.1.3 Valid Results Limits
    - E.4.1.4 Stabilization Criterion (Tare & Stop)
    - E.4.1.5 Lazy Filling
  - E.4.2 Filling By = Time
    - E.4.2.1 Filling by Time Parameters
    - E.4.2.2 Stabilization Criterion (Stop)
  - E.4.3 Fast Speed Config
  - E.4.4 The Filling Configurations Library
- E.5 LED Display Notations
- E.6 Commands
- E.7 Error Codes

### **F. Specifications**

- F.1 Load Cell Input
- F.2 A/D
- F.3 Digital Inputs
- F.4 Digital & Analog Outputs
- F.5 Standard Interfaces
- F.6 Software

F.7 Dimensions (mm)

F.8 Misc.

## **G. Trouble-shooting**

G.1 Card does not respond after PC power-on

## **H. Zero & Tare**

H.1 The *Zero* function

H.2 The *Auto-Tare* function

# 1. Introduction

## 1.1 General Description

The model LCIC-WIM is a very High Speed, Intelligent Load Cell Interface Card with USB/RS232/RS485. Besides its basic mode – named below as the **general mode** – the board includes an integral Fill Mode supplying an independent filling control. The board is intelligent and powerful enough for OEM customers – it is ready to accept piggy-back modules and/or embedded applications for OEM special requirements.

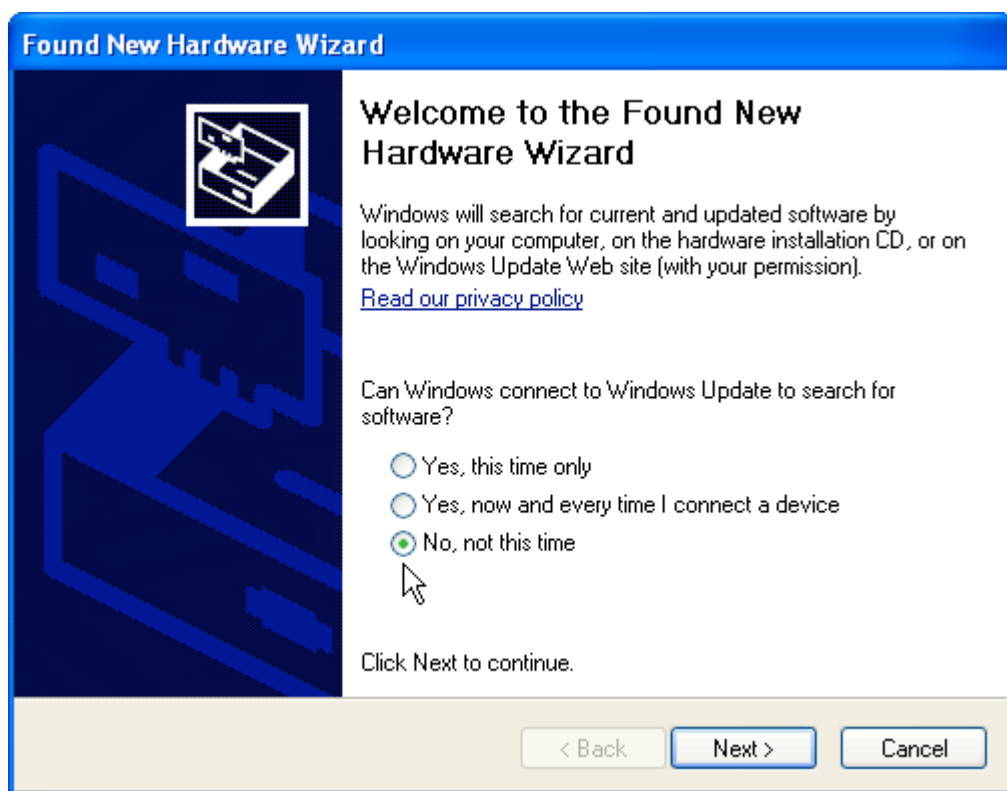
## 1.2 Typical Applications

- Dynamic weighing – vehicles, livestock
- Dynamic force measurement
- High speed checkweighing
- High speed filling / batching
- Beltweighing
- Force measurement / Press Machines
- WIM-Monitor for analysis of dynamic systems

## 2. Installing the board in the PC

(The following description refers to Windows XP. Obviously, on another operating system it might be different.)

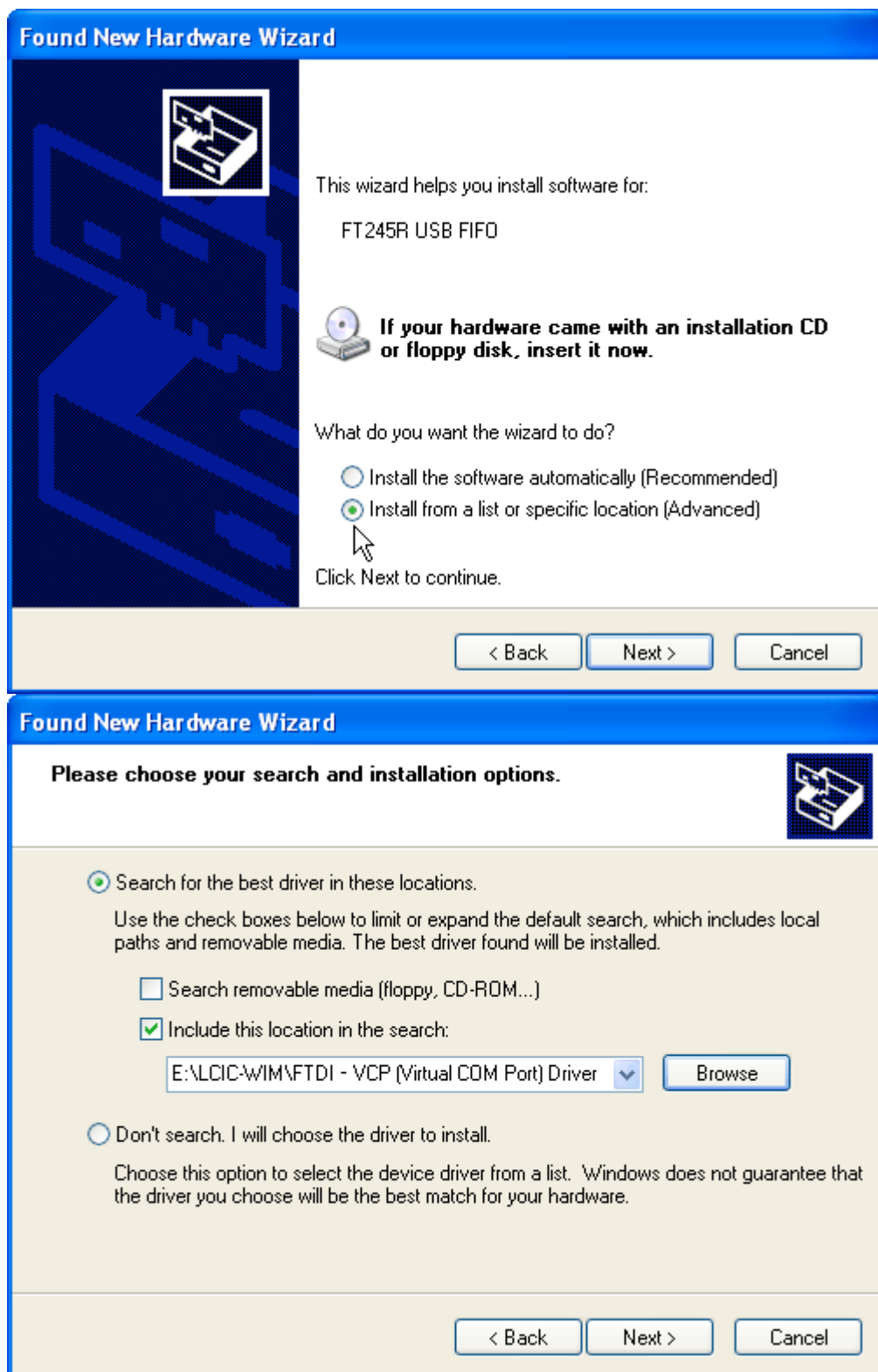
1. Make sure that all installation files have been copied to your hard disk to a new folder, say, LCIC-WIM.
2. Connect the LCIC-WIM board to your PC.
3. The 'Found New Hardware Wizard' appears. Select the last option like this:



Click 'Next'.

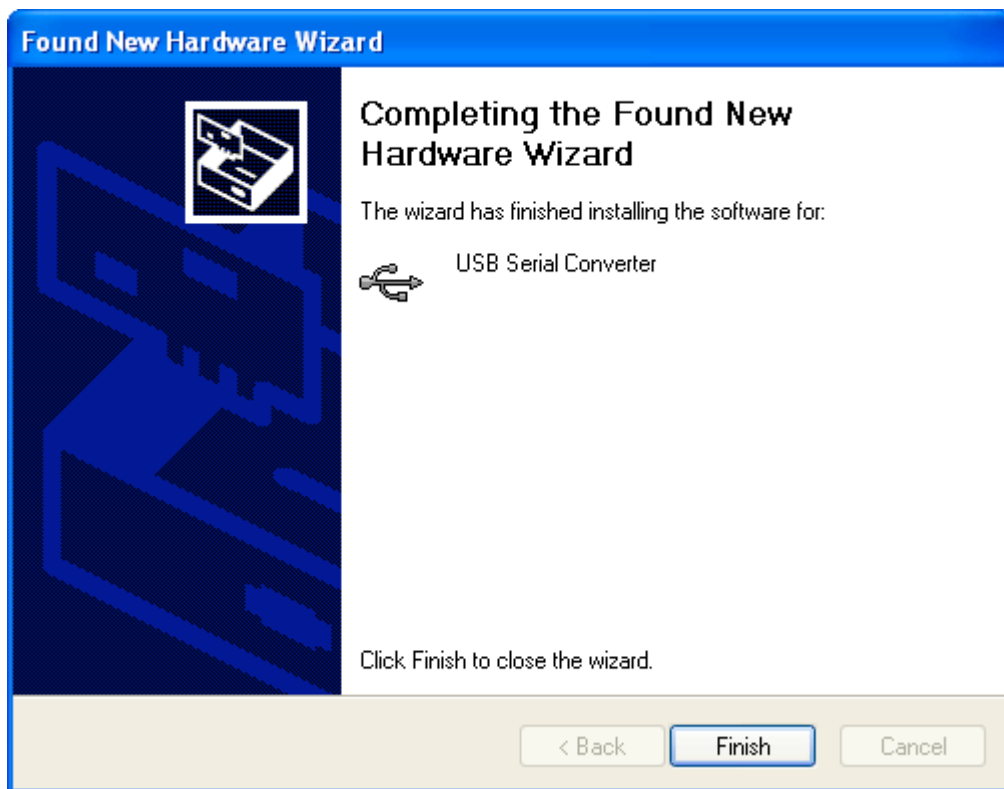


4. Select the second option, click 'Next' and browse to the "FTDI - VCP (Virtual COM Port) Driver" folder (under the folder where you copied the installation files in step 1).



Click 'Next'.

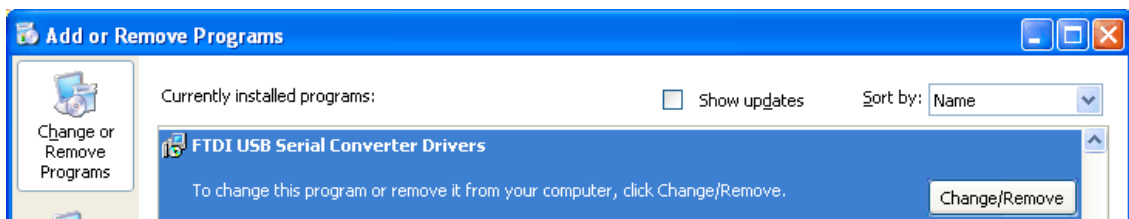
5. After a while, you'll have this display:



Click 'Finish'.

### Notes

1. *It might occur that the wizard will return to step 3, requiring to repeat the process. This is normal, just repeat steps 3-5.*
2. *You may watch the new driver in 'Add/Remove programs':*



## **3. Utilities**

### **3.1 Setup & Running**

**1. Run the setup(s) of the LCIC-WIM utilities in the folders:**

- \* LCIC-WIM-CALIBRATION
- \* LCIC-WIM-SETTINGS

**2. Run a utility:**

- \* If the utility reports that .Net Framework is not installed, then run "dotnetfxV1.1.4322.exe" in the "Microsoft Net Framework" folder on your CD.
- \* If the utility reports "LCIC driver is not installed", then refer to the previous section ("Installing the board in the PC").

## **3.2 The Calibration Utility**

### **3.2.1 General**

The calibration utility (LCIC-WIM-CALIBRATION) enables to calibrate the LCIC-WIM board adjusting it to your own system.

The utility is straightforward and is in the form of a Windows wizard. It includes three main stages carried out in five steps.

#### **The three stages are:**

##### **1. Show Data (step 1)**

(Described below under ‘Step 1’.)

##### **2. Pseudo Calibration (steps 2-3-4)**

The calibration is called ‘pseudo’ as it won’t be saved in the board unless the user confirms it in the next stage. As long as the user did not confirm the new (pseudo) calibration, the previous calibration remains in effect.

(The details are described below under ‘Step 2-3-4’.)

##### **3. Save or Quit (step 5)**

(Described below under ‘Step 5’.)

**The five calibration steps are:**

**Step 1 – Show Data**

This step introduces both the **parameters** and the **current readings**, as received from the board. The step is passive in the sense that it only *shows* data passed by the board, but it does not make any *change* in the board. Some **additional parameters** may be displayed in the bottom of the “Current Board Calibration” box. Type <Ctrl>/<I> and – as shown in the screenshot below – the additional parameters will appear, hiding the lower original parameters (Load Cell Output, Full Load Cell(s) Capacity, Maximum Applied Capacity and Display Resolution). In order to hide the additional parameters and return to all the original ones, type <Ctrl>/<I> again.

Current Board Calibration	
Calibration Name	IMS
Calibration Date...	October 25, 2006
Calibration Time..	14:21
Calibration Counter...	2
Zero A/D .....	1630491
Gain .....	424
Calibration Efficiency	20.22 %

} Additional Parameters

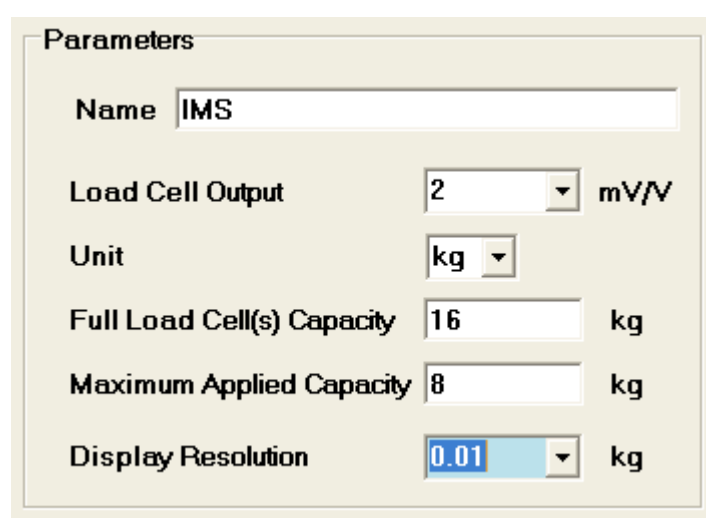
## Step 2 – Pseudo Calibration / Parameters

This step starts the calibration procedure. It enables to change calibration parameters. Whether you changed the parameters or not, you may proceed to the next step by pressing the 'Next' button.

### Library issues:

1. Alternatively, you may click 'Library' in order to access the library, as described in the 'Calibrations Library' section. (If there are still no calibrations in the library, the 'Library' button will be inactive.)
2. Normally, upon confirming a new calibration it will be saved also to the calibrations library. However, if you want to save it only in the board, uncheck the 'Save to Library' box.

In the example below the user changed the Display Resolution parameter:



The screenshot shows a 'Parameters' dialog box with the following fields:

Parameter	Value	Unit
Name	IMS	
Load Cell Output	2	mV/V
Unit	kg	
Full Load Cell(s) Capacity	16	kg
Maximum Applied Capacity	8	kg
Display Resolution	0.01	kg

The 'Display Resolution' field is highlighted with a blue background.

### Notes

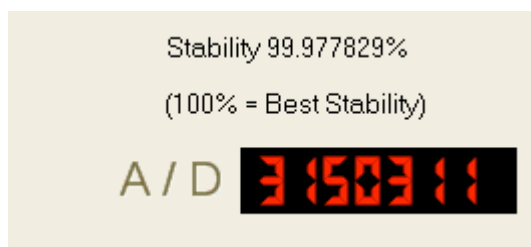
1. The changed parameter (Display Resolution) is displayed with blue background.
2. The options list of the Display Resolution depends on the value of Maximum Applied Capacity.

### Step 3 – Pseudo Calibration / Zero

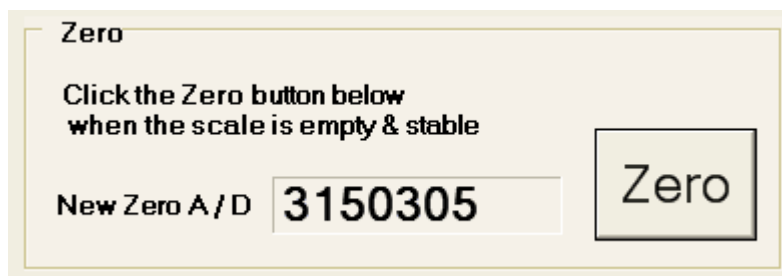
This step enables to redefine the ‘zero’ level. Click ‘Skip’ if you are satisfied with the previous definition of the ‘zero’ level. Otherwise, when the scale is empty and stable (see note), click ‘Zero’ to sample another ‘zero’ level. Once you clicked ‘Zero’, you may either confirm the new ‘zero’ level by pressing ‘Next’, or redefine it by clicking again the ‘Zero’ button, or leave it out by clicking ‘Skip’. Please note that even if you do confirm the new ‘zero’ level by pressing ‘Next’, its effect is limited to the ‘pseudo calibration’ stage. It will be stored in the board only if the new calibration is confirmed in step 5.

#### Note

In order to know the **stability**, watch the Stability indicator above the A/D reading (e.g., 99.977829% in the screenshot below).



In the example below the user clicked ‘Zero’:



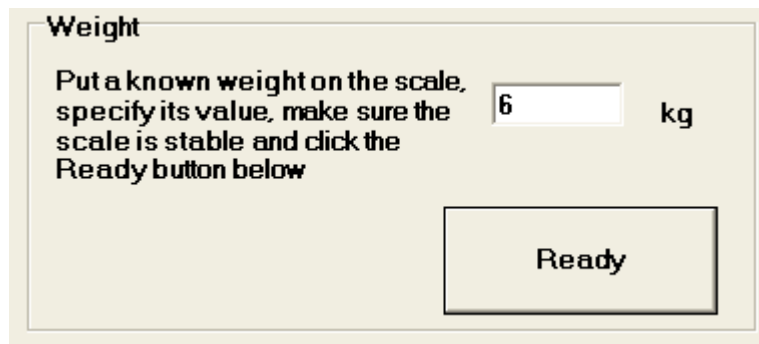
Now the user has three options:

1. Confirm this ‘zero’ level (click ‘Next’);
2. Redefine the ‘zero’ level (click ‘Zero’ once more);
3. Leave out this new ‘zero’ level staying with the previous ‘zero’ adjustment (click ‘Skip’).

#### Step 4 – Pseudo Calibration / Weight

This step enables to redefine the ‘weight’ level. Click ‘Skip’ if the previous ‘weight’ level was OK. Even if there was a fixed shift in the weight (which you probably corrected in step 3), you *don’t have* to redefine the ‘weight’ level – just click ‘Skip’. Otherwise, specify the value of the weight, and when the weight is stable on the scale, click ‘Ready’ to sample another ‘weight’ level. Once you clicked ‘Ready’, you may either confirm the new ‘weight’ level by pressing ‘Next’, or redefine it by clicking again the ‘Ready’ button, or leave it out by clicking ‘Skip’. Please note that even if you do confirm the new ‘weight’ level by pressing ‘Next’, its effect is limited to the ‘pseudo calibration’ stage. It will be stored in the board only if the new calibration is confirmed in step 5.

In the example below the user specified the value of the weight (6 kg) and clicked ‘Ready’:



**Weight**

Put a known weight on the scale, specify its value, make sure the scale is stable and click the Ready button below

6 kg

Ready

Now the user has three options:

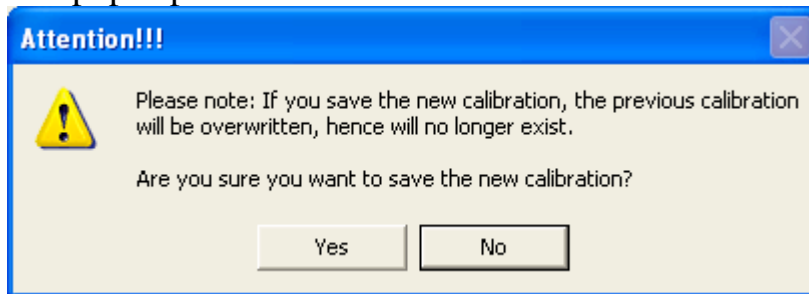
1. Confirm this ‘weight’ level (click ‘Next’);
2. Redefine the ‘weight’ level (click ‘Ready’ once more);
3. Leave out this new ‘weight’ level staying with the previous ‘weight’ adjustment (click ‘Skip’).



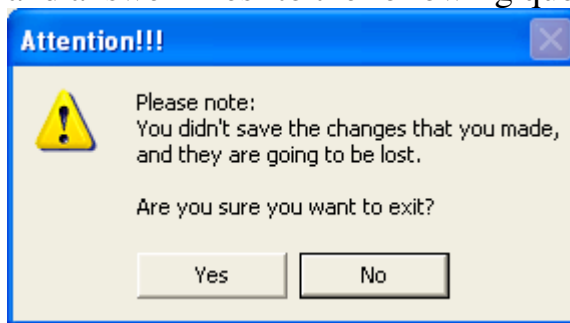
### Step 5 – Save or Quit

This is the final step – here you decide whether to confirm the pseudo calibration, or leave it out. Before you decide, you may watch the current readings examining whether they are satisfactory.

- In case you do want to confirm the new (pseudo) calibration (overwriting the previous calibration), press the ‘Save to Board’ button and answer ‘Yes’ to the following question that pops up:



- Otherwise – that is, you want to stay with the previous calibration leaving out the ‘pseudo calibration’ – click ‘Exit’ and answer ‘Yes’ to the following question that pops up:



Please note that after saving the calibration to the board you still may remain in the utility, which will show now board's response after the calibration, which is now ‘real’ and not ‘pseudo’ any more.

### 3.2.2 The Calibrations Library

Each calibration that the user applies may be **saved** in the 'Calibrations Library'. Later on, the user may use that library as a short cut in order to **restore** a previous calibration quickly and reliably.

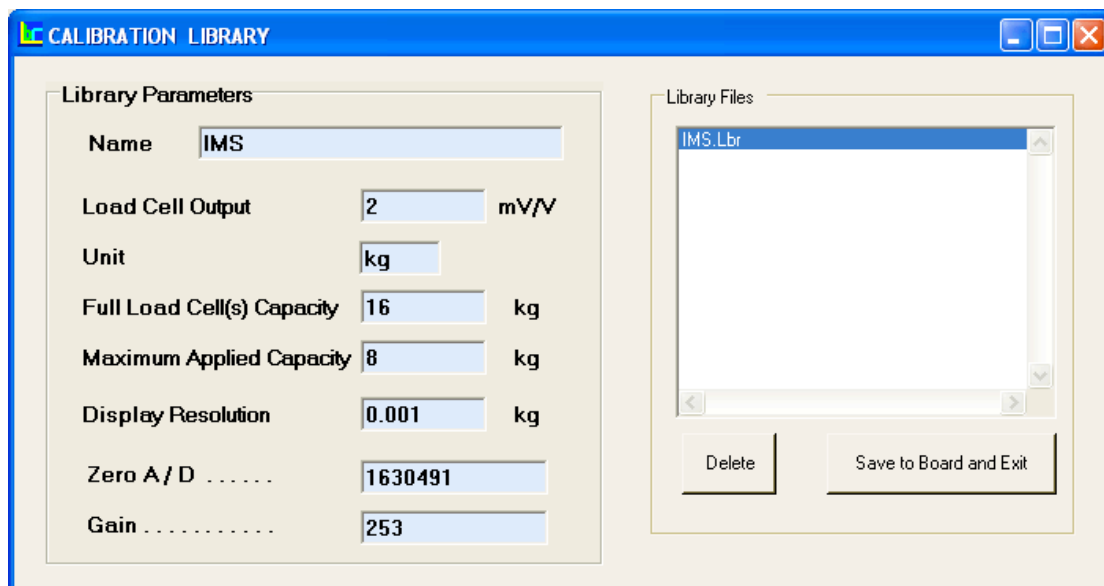
The procedure is very simple:

#### Save

Unless the 'Save to Library' box is unchecked, each calibration is automatically saved to the library upon its saving to the board in Step 5. Its name is the 'Calibration Name' parameter.

#### Restore

In order to restore a calibration from the library to the board, press 'Library' in Step 2, and select the required calibration file (e.g., IMS.Lbr in the example below). You may watch (but not change) the selected calibration's parameters. Clicking 'Save to Board and Exit' will restore the selected calibration to the board. Unnecessary calibration may be erased by clicking the 'Delete' button.



### 3.2.3 Parameters

This section describes the parameters to be filled during the calibration procedure.

#### **Parameter #1: Load Cell mV/V**

The mV/V output of your load cell: 1, 2 or 3 mV/V. In case the actual output is none of these values: choosing a value *higher* than the actual will result in loss of resolution; choosing a *lower* value may result in loss of range, that is, maximum smaller than the real one.

**Refer also to Appendix B.**

#### **Parameter #2: Unit**

The desired weight unit (g, kg, ton, oz or lb).

#### **Parameter #3: Load Cells Capacity**

The maximum capacity of all the load cells in the scale system, according to manufacturer's specifications. If the scale system has more than one load cell, e.g. four load cells of 10 kg then type in 40.

#### **Parameter #4: Maximum Load**

Fill in the actual full scale, i.e. the maximum load you plan to put on the scale.

#### **Note**

**Keep the following condition true:**

**Maximum load + Dead load  $\leq$  Load Cells Capacity.**

The known rated mV/V value of the load cell, the Maximum load and the Load Cells Capacity – in whatever measuring units – are required in order to calculate the optimal gain and zero thus drastically accelerating the calibration procedure.

#### **Parameter #5: Resolution**

Select the Resolution value of the displayed weight in the selected unit (1g, 0.05kg, etc.) that fits your application.

### **3.2.4 Calibration Efficiency (CE)**

The potential range of A/D points is between 0 and near  $\pm 8,400,000$ . The 'Calibration Efficiency' specifies what portion of this potential range is in use. The closer it is to 100%, the better accuracy / stability you have. However, in practice , 100% is a theoretical number and almost not reachable . The accuracy and stability will still be excellent even if 'Calibration Efficiency' is far lower than 100%.

## **3.3 The Settings Utility**

The LCIC-WIM-SETTINGS utility gives control to card's filters, analog output, fill mode parameters and more. The utility has three items:

- The Menu Bar
- Current Weight Display
- Parameters

The 'Current Weight Display' is rather obvious – it continuously shows the actual weight. The other two items are detailed below.

### **3.3.1 The Menu Bar**

The Menu Bar supplies some functions:

- Exit  
An alternative way to quit the utility.
- Tools / Analog Output  
Described below (section 3.3.1.1).
- Tools / Baud Rate for SCI port  
Described below (section 3.3.1.2).
- Tools / General Setpoints  
Described below (section 3.3.1.3).

### **3.3.1.1 Tools / Analog Output**

The 'Settings' utility gives access to the analog output mechanism:  
Click 'Tools' / 'Analog Output'.

For 'manual' mode uncheck the 'Activate Auto Mode' box.

For 'auto' mode check the 'Activate Auto Mode' box.

#### **Manual Mode**

On the top of the display, specify a desired voltage in the 'Manual Mode' square and click 'Send'.

#### **Auto Mode**

Specify the following parameters:

1. Voltage Max.
2. Weight Max.
3. Weight Min.
4. Update Frequency

The Auto Mode works like this:

- \* When the current weight = **Weight Min.** or less, the analog output is set to 0.
- \* When the current weight = **Weight Max.** or more, the analog output is set to **Voltage Max.**
- \* When the current weight is between **Weight Min.** and **Weight Max.**, the analog output is set between 0 and **Voltage Max.**, in the same ratio.
- \* The rate of updating the analog output is depends on the **Update Frequency** parameter. For example, when **Update Frequency** is 4, the analog output is updated 4 times a second.

### **3.3.1.2 Tools / Baud Rate for SCI port**

Click 'Tools' / 'Baud Rate for SCI port' to see the current baud rate for the RS232/RS485 serial port. It may be changed to some values between 19,200 and 115,200. (The baud rate for the USB need not be defined – usually it is 921,600.) **The change will take effect only after card reset.** The current b/r used by the board for the serial communication is shown for a while on the LED display upon card reset, prefixed by 'Sb' (=Serial baud rate). Do not mix the serial b/r with the b/r shown in the Communication box – they are not necessarily the same: The b/r in the Communication box specifies the *actual* b/r in which the Settings utility is communicating with the card. If the Settings utility communicates with the card through a serial port, the two b/r values will indeed be the same. However, in case the Settings utility communicates with the card through a USB port, the b/r displayed in the Communication box will be usually 921,600, regardless of the serial b/r.

Note: Normally, the b/r in the board side will be the same as the b/r in the PC side. However, it does happen that the communication is successful only when the PC sets the b/r to some other value. (This anomaly might occur only with the serial communication, not when using USB.) You may find out the required b/r in the PC side by watching any of the supplied utilities (Calibration, Settings or Monitor): upon the initialization they try to communicate with the board with various values of b/r.

### **3.3.1.3 Tools / General Setpoints**

The LCIC-WIM board has four digital outputs. Each of them may be defined – through the *Tools / General Setpoints* – either as a **manual** output, or as a **general setpoint** output:

- A **manual** output is controlled by a user's command sent from the PC (or another computer). That is, the user sends – either by his own application or by a general RS232 terminal (see section D.3) – commands to turn an output on or off. The user – and not the card – has the initiative to turn an output on or off.
- A **general setpoint** output is controlled by the card according to user pre-defined criterion. Say, initially the user defined the setpoint as 10 kg, then the card automatically turns the output off/on when the current weight is less/more than 10 kg. The user cannot explicitly turn an output on or off by a command.

*Note – these general setpoints are absolutely different from the setpoints in the Fill mode (section E.4). The general setpoints are inactive in the Fill mode.*

In order to configure an output:

- Select output's Mode as **Manual** or **Setpoint**.
- In case you selected '**Mode: Setpoint**', specify the setpoint value (e.g., 10).
- Click 'Save to Board' to validate your new configuration.
- You may turn each manual output on or off.



## **3.3.2 Parameters**

The following sections describe the various parameters. After changing parameter(s), click the ‘Save to Board’ button and wait a while until the new value(s) are accepted by the board.

### **3.3.2.1 Communication**

The Communication box refers to card’s communication port – either serial or USB. The current Port and Baud Rate are shown. They are ‘read only’, that is, not changeable. (About changing the Baud Rate for a serial port, refer to section 3.3.1.2.) A third item (‘Get results immediately’) is a parameter controlling card’s response in communication during the Fill mode.

#### **3.3.2.1.1 Port**

Shows port’s type and number, e.g., “USB port COM4” or “SCI port COM1” (SCI stands for ‘Serial Communications Interface’, that is, RS232 or RS485).

#### **3.3.2.1.2 Baud Rate**

Specifies the actual Baud Rate in which the Settings utility is communicating with the card. Refer also to section 3.3.1.2.

#### **3.3.2.1.3 RS485 Address**

Up to 64 LCIC-WIM boards may reside on one RS485 bus consuming only one PC port. In case you do not need this feature, specify RS485 Address = 0; this will simplify the coding of your application. If you do like to utilize this feature, specify the required RS485 Address – between 1 to 64.

**→ The address setting takes effect only upon board reset.**

#### **3.3.2.1.4 Get results immediately**

Controls card's response in communication during a special mode, such as the Fill mode:

When checked, the card assumes that the PC (or another remote computer) is continuously connected and listening to the communication port.

Therefore, the card takes the initiative and sends messages to the PC, reporting the process results immediately when they are available.

However, you might prefer the PC to poll the card from time to time drawing the results, so that the PC can handle other tasks too. In the latter case, uncheck the 'Get results immediately' option. For example, in the Fill mode, use the 'r' command as described in section E.6.

### **3.3.2.2 Auto Zero**

The 'Auto Zero' optional feature supplies an automatic correction to creeps in the zero level during a special mode (such as the fill-mode or the WIM-mode), caused by dust, temperature etc.

When this feature is activated and the card is inside the special mode, the board automatically clears the gross weight if some pre-defined condition is satisfied: All readings within some 'continuous duration' are inside the 'zero range'.

Say, all readings during 3 consecutive seconds are between  $-1\text{g}$  and  $+2\text{g}$ . These 'continuous duration' & 'zero range' are user-defined by the following parameters.

The auto Zero effect is temporary – it expires upon the exit from the special mode, or card reset.

#### **3.3.2.2.1 Activate**

Activates the 'Auto Zero' feature. When 'Activate' is deselected (unchecked), an "Auto Zero" operation will never take place.

#### **3.3.2.2.2 Max Zero**

The *upper* bound of the 'zero range' ( $+2\text{g}$  in the example).

Note: Max. Zero refers to the original zero level as was defined during the calibration procedure.

#### **3.3.2.2.3 Min Zero**

The *lower* bound of the 'zero range' ( $-1\text{g}$  in the example).

Note: Min. Zero refers to the original zero level as was defined during the calibration procedure.

#### **3.3.2.2.4 Time limit**

The size of the 'continuous duration' (3s in the example).

### **3.3.2.3 Start Fill-mode**

#### **3.3.2.3.1 Fill-mode starts automatically upon card reset**

When this option is activated (checked), the card starts automatically the Fill-mode upon reset. Otherwise (the option is unchecked), the cards 'awakes' in the upper level, referred to in this document as the 'general mode'.

#### **Notes**

1. In order to switch the card from Fill mode to General mode, use the 'x' command (small 'x').
2. In order to switch the card from General mode to Fill mode, use the 'F' command.
3. About the 'x', 'F' and other commands in Fill mode, refer to section E.6.

### **3.3.2.4 Filtering**

Board's **digital filtering** is used to "smooth" the read samplings by averaging a pre-set number of the internal readings. It's especially essential on a noisy environment, as this mechanism reduces system's susceptibility to short interferences. The 'noise' may be either mechanical (e.g., load cell vibrations), or electrical.

The digital filter averages the raw internal readings of the A/D, whose rate is 52,734 A/D readings per second. The behavior of the filtering mechanism is controlled by three parameters:

- **Filter1**
- **Filter2**
- **Decimator**

These three parameters, described below, are adjustable in order to make them suit best your application.

The filtering mechanism includes **two** levels:

Level 1: Each  $N_1$  successive A/D readings are averaged forming the **first level average**.

Level 2: Each  $N_2$  first level averages are re-averaged forming the **second level average**. Unlike the  $N_1$  readings of level 1, the  $N_2$  readings of level 2 are not successive: Only each  $N_3$ -th 'first level average' participates in the second level averaging procedure, the other  $N_3-1$  ones being ignored. As you might guess,  $N_1$  is Filter1,  $N_2$  is Filter2 and  $N_3$  is Decimator. The range of Filter1 & Filter2 is selectable from 2 to 256, where 2 is the **lowest** filtering and 256 is the **highest** filtering.

The range of Decimator is from 1 to 1000.

Note: As described in section 4.1, 4.2 & 4.3.2, user's application may select the filtering level it requests – either only Filter1, or Filter2 (=both Filter1 & Filter2). **However, the weight on board's numeric LED display is always after Filter2.**

### **3.3.2.5 Filling Definition**

The '**Filling Definition**' box includes the parameters used upon a filling operation in board's **Fill Mode**, as described in section E.4.

## 4. Programming your Application

The control of the board is by **commands** and **parameters**, described below. You may either use them directly (see also section D.3), or call an ActiveX (see section 4.3) that does the work.

### 4.1 Commands

\_ <c/r> signifies a carriage return.

#### Note about High Speed Commands:

Single character commands (where no <c/r> is required) are used for quick direct access/control of the card.

Command	Action
<b>a. Parameters: Read &amp; write:</b>	
<b>Rnn&lt;c/r&gt;</b>	Read parameter #nn. All values are returned in floating point scientific format, preceded by nn<c/r> echo and appended by a c/r. E.g., the answer to 'R23<c/r>' is '23<c/r>2.560000e+02<c/r>' indicating that the value of Filter1 is 256.
<b>Wnn&lt;c/r&gt;argument&lt;c/r&gt; (upper case W)</b>	Write the value <i>argument</i> to parameter #nn. Floating point values can either be normal or scientific representation. For example 0.003 could be sent as it is or as 3.0E-03. After the 'Wnn<c/r>' the board echoes 'nn<c/r>' and after the ' <i>argument</i> <c/r>' the board responds ' <i>argument</i> <c/r>OK<c/r>'. For example, PC sends 'W23<c/r>', the board answers '23<c/r>', the PC then sends '123<c/r>' and the board answers '123<c/r>OK<c/r>' changing Filter1 to 123.
<b>b. Fast Mode: Start &amp; stop (the Fast Mode is not available with RS485):</b>	
<b>u (lower case u)</b>	Start a 'Fast Mode' session <u>using Filter1</u> (no LED update). Terminated by the 'x' command. A timer stamp is appended. Its value is the time elapsed from start of transmission until end of transmission, in ms.
<b>U (upper case U)</b>	Start a 'Fast Mode' session <u>using Filter2</u> (no LED update). Terminated by the 'x' command. A timer stamp is appended. Its value is the time elapsed from start of transmission until end of transmission, in ms.
<b>x (lower case x)</b>	Exit Fast Mode and return to general mode. In order to exit the fast mode, the 'x' command should be <u>synchronised</u> , that is, issued upon receiving a block.

<b>c. Get a single reading of: weight, A/D or temperature:</b>	
.	Get weight (after Filter2, not rounded to resolution).
?	Get weight (after Filter2, rounded to resolution).
>	Get A/D reading after Filter1.
<	Get A/D reading after Filter2.
T	Get temperature
<b>d. Analog output: Read &amp; write voltage:</b>	
@	Gets the analog output voltage (in Volts).
vx (lower case v)	Sets the analog output voltage to x Volts. The analog output voltage is measured at pin 12 of CONN6 with respect to pin 13 which is ground.
<b>e. Digital Outputs: Write outputs:</b>	
a	Turn on Output1 if Manual
A	Turn off Output1 if Manual
b	Turn on Output2 if Manual
B	Turn off Output2 if Manual
c	Turn on Output3 if Manual
C	Turn off Output3 if Manual
d	Turn on Output4 if Manual
D	Turn off Output4 if Manual
<b>f. Digital Outputs: Read outputs:</b>	
O (upper case o)	Card returns a string of the form 'xxxx ', where x is either "1" or "0" representing the status of the 4 output opto relays OUT4,OUT3,OUT2,OUT1 respectively.
<b>g. Digital Inputs: Read inputs &amp; toggling counter:</b>	
I (upper case i)	Card returns a string of the form 'xxxx ', where x is either "1" or "0" representing the digital input status of IN4,IN3,IN2,IN1 respectively.
i	Gets a 16 bit 'toggling counter'. The 'toggling counter' increments each time when input #2 is toggled.

<b>h. RS485: Address selection, setting &amp; reading</b> <b>(for more details and examples refer to section D.5; except</b> <b>'Nx&lt;c/r&gt;', these commands are available also in the fill mode):</b>	
:x<c/r>	Activate the board addressed <b>x</b> . <b>x</b> is between 1 and 64. <u>Notes</u> 1. Wait 10 ms after sending the colon (':') before sending the rest of the command ('x<c/r>'). 2. Wait 30 ms after sending the ':x<c/r>' command before analyzing the response(s).
:999<c/r>	"Hello" command: prompt all existing boards to respond. <u>Notes</u> 1. Wait 10 ms after sending the colon (':') before sending the rest of the command ('999<c/r>'). 2. Wait 1000 ms after sending the ':999<c/r>' command before analyzing the response(s).
n	Read the RS485 address of the active board. Board's response: '#x<c/r>', where <b>x</b> is the address of the active board. <b>x=0</b> means that the board has been configured as a non-RS485 device. Other value (between 1 & 64) specifies the RS485 address of the active board. <u>Notes</u> 1. Wait 1 second after sending the 'n' command. 2. If no board is active, there will be no response. 3. In case of malfunction, more than one '#x<c/r>' will be responded in sequence. This is theoretically impossible but should be checked in order to be on the safe side.
Nx<c/r>	Set the required RS485 address. <b>x</b> is between 0 and 64. Board's response: 'OK<c/r>'. ( <b>x=0</b> sets the board as a non-RS485 device; this will facilitate your coding, as no ':x<c/r>' command will be needed.) <u>Notes</u> 1. <b>The address setting takes effect only upon board reset.</b> 2. When the RS485 address is not 0, its value is shown for a while on the LED display upon board reset.



i. Misc.:	
<b>z (lower case z)</b>	Manually zero the gross weight. The effect of this function is temporary — it expires upon card reset. Response (versions 1.12, 3.09, 6.01, 7.00 and up): 'z'<c/r>.
<b>Z (Upper case Z)</b> (versions 1.12, 3.09, 6.01, 7.00 and up)	Cancel the manual zero operation (the lower case 'z'). That is, return to the original calibration zero. Response: 'Z'<c/r>.
<b>S</b> (upper case S)	System reset (software reset to perform a restart as if turned off/on). Response (versions 1.12, 3.09, 6.01, 7.00 and up): 'S'<c/r> - possibly followed by additional binary character(s). <b>Wait 6 seconds before accessing the board again.</b>  Issue this command after changing any of the following parameters: * Auto transmit interval (parameter #20). * Filters & Decimators (parameters #23-25). * General Setpoints (parameters #101-104). * Output modes (parameters #111-114).
<b>F (upper case F)</b>	Start Fill-mode.
<b>V (upper case V)</b>	Get version no. of DSP software. Board's response: "LCIC-WIM/V#.#", where #.# = Version Number.

## Summary of Weight & A/D Reading Commands

		Reading Type			Filtering Level	
		A/D	Weight		Filter1	Filter2
			Rounded to Resolution	Not Rounded		
Single Reading	.			v		v
	?		v			v
	>	v			v	
	<	v				v
Fast Mode	u			v	v	
	U			v		v

## 4.2 Parameters

Parameter number	Parameter Description	Data Type
1	Load Cell mV/V (1, 2 or 3. 0 = unknown).	Float
2	Units: 0=g, 1=kg, 2=ton, 3=oz, 4=lb.	Float
3	Full Load Cell(s) Capacity	Float
4	Maximum Applied Capacity	Float
5	Resolution Index (0-17) Resolution Index is actually the index to an array of 18 defined values(0-17) like that:  0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50.  E.g., if Resolution Index=8, then system resolution = 0.05.	Float
8	Calibration Date formatted as "MMDDYY". So, 10107 <= Calibration Date <= 123199.	Float
11	Calibration Time formatted as "HHMM". So, 0 <= Calibration Time <=2359	Float
20	Auto transmit interval (3-52734, integer). <b>How many internal adc updates (52734 Hz) there are between auto transmissions.</b> This gives a theoretical reading rate from 17578 per sec to 1 per sec. (i.e. rate=52734/P20). Practically, the actual rate for low values of P20 is usually less than the theoretical rate. ("P20" stands for "Parameter #20".) Note that this rate relates to USB communication; upon using a serial port, the rate is far smaller. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float

23	Filter1 value: 2-256. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
24	Filter2 value: 2-256. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
25	Decimator: 1-1000. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
101	General Setpoint1. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
102	General Setpoint2. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
103	General Setpoint3. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
104	General Setpoint4. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
111	Output1 Mode 0=manual, 1=general setpoint. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
112	Output2 Mode 0=manual, 1=general setpoint. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
113	Output3 Mode 0=manual, 1=general setpoint. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
114	Output4 Mode 0=manual, 1=general setpoint. <b>Becomes effective only after a system reset (either power off/on or using the 'S' command).</b>	Float
115	Baud rate for the RS232/RS485 port. Possible values: 19200, 28800, 38400, 57600, 115200. <u>Note</u> The baud rate for the USB need not be defined – its upper limit is 921,600.	Float

1024-1034	Calibration Name (32 characters max.) (Organised 3 characters per location; in case the length is less than 32, the last character is followed by a binary zero byte.)	Float
1053	Analog Output Max Voltage	Float
1054	Analog Output Min Weight	Float
1055	Analog Output Max Weight	Float
1059	Analog Output Mode (0 = Manual, 1 = Auto)	Float
1066-1069	Card Serial Number (12 characters max.) (Organised 3 characters per location, in case the length is less than 12, the last character is followed by a binary zero byte.)	Float

## **4.3 LCIC-WIM ActiveX**

*Unless otherwise specified, a function returns a Boolean: True for success, or False for failure.*

### **4.3.1 Start/Stop Communication**

**Is\_LCIC\_WIM\_Port(*CommPortNumber*)**

Returns:

0 if the port does not respond.

1 if the port responds but not as an LCIC-WIM.

2 if the port responds as an LCIC-WIM.

***CommPortNumber*** (Integer): Number of communication port.

**OpenLCIC\_WIM(*CommPortNumber*, *Baud\_Rate*)**

Opens the specified port.

***CommPortNumber*** (Integer): Number of communication port.

***Baud\_Rate*** (Integer): Required Baud Rate (110, 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200, 230400, 460800 or 921600).

(Refer also to sections 3.3.1.2 & D.2.)

Note: Using USB, all these values of *Baud\_Rate* are relevant (even though usually 921600 is used). However, upon using serial communication, board's possible b/r values are limited to the range 19200-115200; please refer to section 3.3.1.2.

**CloseLCIC\_WIM()**

Closes the open port.

**Get\_First\_Free\_LCIC\_WIM\_PortNumber()**

Returns the number of the first port (from COM1 to COM16) that responds as an LCIC-WIM board.

If none is detected, 0 is returned.

## 4.3.2 Variables

The system has **variables** with which the user may adjust the system to his needs and communicate with the I/O. Actually these variables consist of parameters, inputs and outputs. A variable may be read and sometimes also may be written. The table below lists the variables, describes them and specifies which of them may be also written.

The methods to read and write a variable are:

### Read:

Get\_Variable(**r**<*Variable Name*>)

Returns a string with the value of the variable.

**r**<*Variable Name*> is the variable name as it appears in the table below, prefixed by 'r' for read, e.g., rOutput\_1\_Mode.

### Write:

Set\_Variable(**w**<*Variable Name*>, *Value*)

**w**<*Variable Name*> is the variable name as it appears in the table below, prefixed by 'w' for write, e.g., wOutput\_1\_Mode.

*Value* is a string with the value of the value to be written to the variable.

## Variables Table

Category	Variable Name	Description	Get	Set
Calibration Info	Calibration_Name	Name of calibration	V	
	Calibration_Date	Calibration date (MMDDYY).	V	
	Calibration_Time	Calibration time (HHMM, e.g., 1545).	V	
	Unit	Weighing unit: ton, kg, g, lb or oz.	V	
	Resolution	Weighing resolution: 0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20 or 50.	V	
	Full_Capacity	Full capacity of the load cell.	V	
	Maximum_Load	Maximum applied load on the load cell.	V	
	Load_Cell_Output	Output of the load cell: 0 = Unknown 1, 2 or 3 = 1, 2 or 3 mV/V.	V	
Filtering	Filter1	<p>There are <b>two</b> filters. The <b>first filter</b> is basically a 1st level moving average filter of size <b>Filter1</b> (2-256). Then depending on <b>Decimator</b> (1-1000), every Decimator-th result from the 1st level filter is put through another moving average filter of size <b>Filter2</b> (2-256), which is the <b>second filter</b>.</p> <p><b>* Setting</b> these three variables causes a <b>board reset</b>, so it is a <b>slow</b> function. Therefore, in order to set <b>all of them</b> there is one more option: the method <code>Set_Filtering</code>, which works faster than three individual activations of the <code>Set_Variable</code> method.</p>	V	*
	Filter2		V	*
	Decimator		V	*



### Variables Table (cont'd)

Category	Variable Name	Description	Get	Set
Analog Inputs	Weight_Native	Current weight after Filter2, not rounded.	V	
	Weight_Rounded	Current weight after Filter2, rounded to resolution.	V	
	A2D_F1	Current A/D after Filter1.	V	
	A2D_F2	Current A/D after Filter2.	V	
	Temperature	The temperature measured on the board (°C).	V	
Digital Outputs	Output_1_Mode	0 = Manual, 1 = General Setpoint.	V	V
	Output_2_Mode		V	V
	Output_3_Mode		V	V
	Output_4_Mode		V	V
	Output_1_Status	0 = off, 1 = on. * The 'Set' is relevant only if the corresponding Output_x_Mode is 'Manual'.	V	*
	Output_2_Status		V	*
	Output_3_Status		V	*
	Output_4_Status		V	*
Output_A_Status	Status of <b>all</b> outputs at once: # <sub>4</sub> # <sub>3</sub> # <sub>2</sub> # <sub>1</sub> # <sub>x</sub> = Status of output x: 0 = off, 1 = on (e.g., 0101).	V		
General Setpoints	Setpoint_1	The weight limit for output x. Relevant only if the corresponding Output_x_Mode is 'General Setpoint'. Irrelevant in Fill mode.	V	V
	Setpoint_2		V	V
	Setpoint_3		V	V
	Setpoint_4		V	V

### Variables Table (cont'd)

Category	Variable Name	Description	Get	Set
Analog Output  <i>(The analog output voltage is measured at pin 12 of CONN6 with respect to pin 13 which is ground.)</i>	Analog_Output_Mode	0 = Manual, 1 = Auto.	V	V
	Analog_Output_Level	Voltage in the analog output, in volts (0 – 2.5). * The ‘Set’ is relevant only if Analog_Output_Mode is ‘Manual’.	V	*
	Auto_Hi_Voltage	When Analog_Output_Mode = ‘Auto’: The voltage to be supplied for weight=Auto_Hi_Weight (in volts, up to 2.5).	V	V
	Auto_Lo_Weight	When Analog_Output_Mode = ‘Auto’: The weight for which 0V should be supplied.	V	V
	Auto_Hi_Weight	When Analog_Output_Mode = ‘Auto’: The weight for which maximal voltage (=Auto_Hi_Voltage) should be supplied.	V	V
Digital Inputs	Input_1_Status	0 = off, 1 = on.	V	
	Input_2_Status		V	
	Input_3_Status		V	
	Input_4_Status		V	
	Input_A_Status	Status of <b>all</b> inputs at once: # <sub>4</sub> # <sub>3</sub> # <sub>2</sub> # <sub>1</sub> # <sub>x</sub> = Status of input x: 0 = off, 1 = on (e.g., 0101).	V	
	Toggling_Counter	A 16 bit counter that increments when opto input #2 is toggled.	V	

### Variables Table (cont'd)

Category	Variable Name	Description	Get	Set
Fast Mode	FM_Updates	<p>FM_Updates = Each how many internal updates there will be a Fast Mode transmission (3 – 52,734, integer).            The frequency of the internal updates is 52,734 Hz.            So, the theoretical reading rate is from 17578 per sec to 1 per sec. (52734/FM_Updates).            Practically, the actual rate for low values of FM_Updates is usually less than the theoretical rate.</p> <p>Notes:            1. This rate relates to USB communication; upon using RS232, the rate is far smaller.            2. The Fast Mode is not available with RS485.</p>	V	V
Misc.	Version_ID	“ <u>LCIC-WIM/V#.#</u> ” #.# = Version Number.	V	
	Serial_Number	Card’s serial number.	V	

### 4.3.3 Filters

Set\_Filtering(*Filter1*, *Filter2*, *Decimator*)

*Filter1* (Integer): 2 – 256 or 0.

*Filter2* (Integer): 2 – 256 or 0.

*Decimator* (long): 1 – 1000 or 0.

(Refer to the ‘Filtering’ square in the variables table above.)

Set\_Filtering supplies a faster way to change the filtering parameters when more than one of them has to be changed, as the change operation causes **board reset** which is time consuming; individual activations of Set\_Variable would require this time more than once. **Specify ‘0’ for a parameter that needs no change.**

For example, in order to set Filter1, Filter2 & Decimator to 11, 22 & 33, respectively, apply Set\_Filtering(11, 22, 33). Now in order to change Filter1 & Filter2 to 10 & 20, respectively, and leave Decimator unchanged, apply Set\_Filtering(10, 20, 0). Finally, in order to leave both Filter1 & Decimator unchanged and set Filter2 to 2, apply either Set\_Filtering(0, 2, 0), or Set\_Variable(wFilter2, 2).

Get\_Filtering(*Filter1*, *Filter2*, *Decimator*)

*Filter1* (Integer): 2 – 256.

*Filter2* (Integer): 2 – 256.

*Decimator* (long): 1 – 1000.

(Refer to the ‘Filtering’ square in the variables table above.)

The Get\_Filtering is functionally equivalent to

Filter1 = Get\_Variable(rFilter1)

Filter2 = Get\_Variable(rFilter2)

Decimator = Get\_Variable(rDecimator)

and has no time advantage, as the get operation does *not* cause board reset; the Get\_Filtering function has been supplied just for symmetry with Set\_Filtering.

### 4.3.4 Fast Mode

(The Fast Mode is not available with RS485.)

During the Fast Mode there is auto high speed transmission of weight readings to the communication. About the transmission rate, refer to the 'Fast Mode' square in the variables table above.

At the end, a 'timer stamp' is appended. Its value is the time elapsed from start of transmission until end of transmission, in ms.

The readings that are returned in this mode are always integers and they equal the actual weight multiplied by 1, 10, 100, 1000 or 10000, depending upon the resolution setting. So, although the readings are integers, due to that multiplication the original precision is maintained. The readings are *not* rounded to the resolution. For example, if the resolution is 0.05, then the readings transmitted by the board will be multiplied by 100, so that '123' will represent '1.23'.

Start\_Fast\_Mode(*Filtering\_Level*)

Starts the Fast Mode.

Note: During the Fast Mode the LED Display is *not* updated.

*Filtering\_Level* (Integer):

1 = Supply readings after Filter1.

2 = Supply readings after Filter2.

Stop\_Fast\_Mode()

Stops the Fast Mode.

(A timer stamp is appended. Its value is the time elapsed from start of transmission until end of transmission, in ms.)

*The mechanism to receive the data uses events and methods as described below:*

The transmission sends blocks of information.

Stage 1

Except the last one, each block generates the event **DataArrivalInFastMode**. When the event occurs, run the method **Get\_CurrentBlock** to read the current block. The block consists of *integer weights* separated by a Carriage Return. At this stage, just store the blocks into a string array. This stage repeats until the last block arrives. That is, if there are 10 blocks, then 'stage 1' occurs 9 times.

Stage 2

The last block generates the event **DataArrivalLastInFastMode**. Run the method **Get\_LastBlock** in order to read the last portion of the *integer weights* and store them too in the string array used in stage 1. Run the method **Get\_Time\_ms** in order to get the time stamp.

Stage 3

After the last block was received and stored, the weights may be processed:

1. Recalling that the values are separated by Carriage Return, parse the string array and keep the individual values in a numeric array. One clean way to do that is write the array to file by Print and read back the file using Input. Note that a value may be split between two blocks, e.g., the value '123' may appear as '12' in the end of one block and '3' in the beginning of the next block. The above way using a file handles the parsing well.
2. As described in the beginning of this section, the values are integers that were accepted by multiplying the actual weight by 1, 10, 100, 1000 or 10000. You may find the actual weight by multiplying the integer weight by a "Resolution\_Factor" which is 1, 0.1, 0.01, 0.001 or 0.0001, respectively. You may find the Resolution\_Factor yourself, but for your convenience there is the method **Get\_Resolution\_Factor** which returns the proper value.

## How to work with the Fast Mode in VB using the ActiveX

During the Fast Mode process the board transmits mass data to the PC. Therefore, in order to avoid data loss, all the actions on your PC should be minimized.

1. Define string Array  
Dim Fast\_mode\_Data(1 to SizeOfArray) as String  
Dim fmCounter as long 'Current counter (index)
2. Select Filter:  
Filter=Filter2
3. To start the Fast Mode:  
Call LCICwim\_commands1.Start\_Fast\_Mode(Filter(
4. In the event DataArrivalInFastMode:  
' Get current block:  
fmCounter = fmCounter + 1  
Fast\_mode\_Data(fmCounter) =  
LCICwim\_commands1.Get\_CurrentBlock

So, all arrived data are stored in the array Fast\_mode\_Data

5. To terminate the Fast Mode:  
Call LCICwim\_commands1.Stop\_Fast\_Mode  
Except the last one, each block will still cause a  
DataArrivalInFastMode event, as described in para. 3.

In the DataArrivalLastInFastMode event that the last block will cause:

```
fmCounter = fmCounter + 1

Fast_mode_Data(fmCounter) =
    LCICwim_commands1.Get_LastBlock
TotalTimeInFastMode =
    LCICwim_commands1.Get_Time_ms
```

### **Interpreting the data in a block:**

Each block includes integer weight values separated by a c/r. In order to get the real weight values, the integer values should be multiplied by the current **Resolution Factor** (for details refer to the end of stage 3, above). You may get the current Resolution Factor using the method:  
LCICwim\_commands1.Get\_Resolution\_Factor

### **4.3.5 Misc.**

**Apply\_Temporary\_Zero()**  
Manually zero the gross weight. The effect of this function is temporary — it expires upon card reset.

**Reset\_Board()**  
Resets the board. Usually this function is not required.



## Appendix A: I/O & the LED Display

### A.1 General Notes about the I/O

\* **The digital I/O is available on CONN6 (15 pin Dsub).**

\* **Digital Outputs**

The outputs are opto-isolated 300mA 50V solid state relays. When activated (status LED is on), they switch the OUTPUT x (x=1,2,3, or 4) to I/O VOLTAGE OV. Hence the load would normally be connected between OUTPUT x and the I/O VOLTAGE+.

\* **Digital Inputs**

The digital inputs are designed to work with either npn or contact input devices. They are activated by an external device pulling INPUT x (x=1,2,3,or 4) down to I/O VOLTAGE OV. To work in this way an external IO VOLTAGE+ (in the range 10-30V) must be present.

\* **Analog Output**

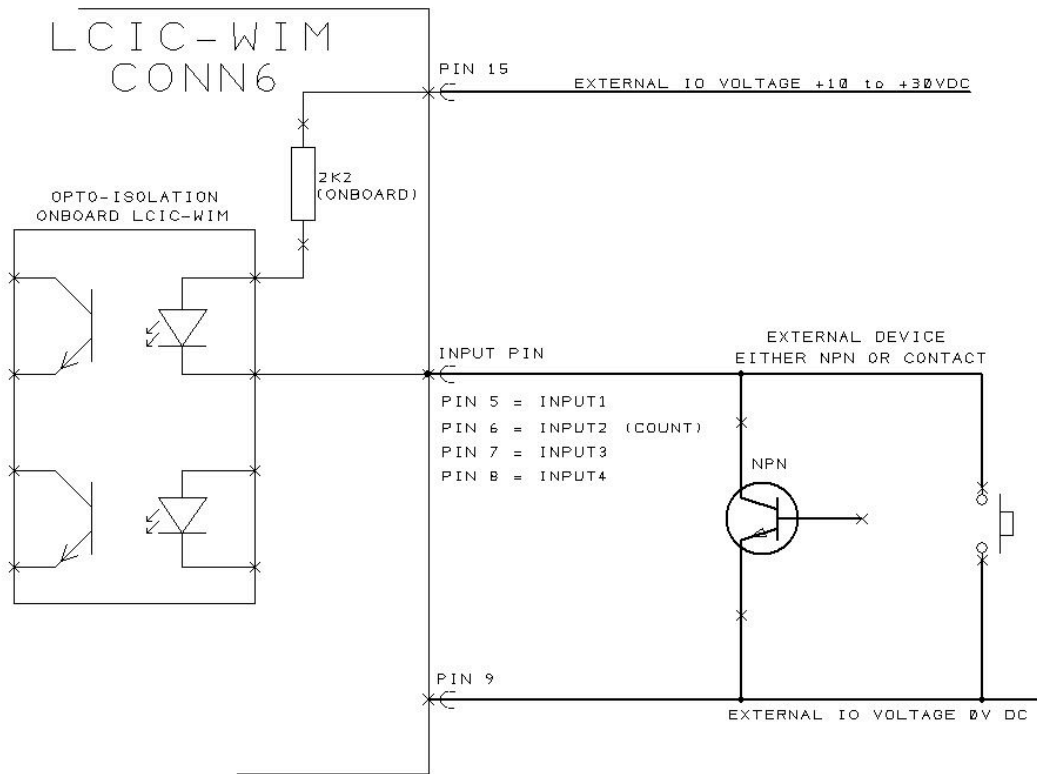
The analog output signal is set by a 16 bit DAC and appears on pin12 of CONN6. The output is with respect to the LCIC\_WIM board **ground** (not the I/O 0V). The board ground appears at pin13 of CONN6. The analog output control is application dependent.

**\* Connections**

The following table shows the I/O pinout:

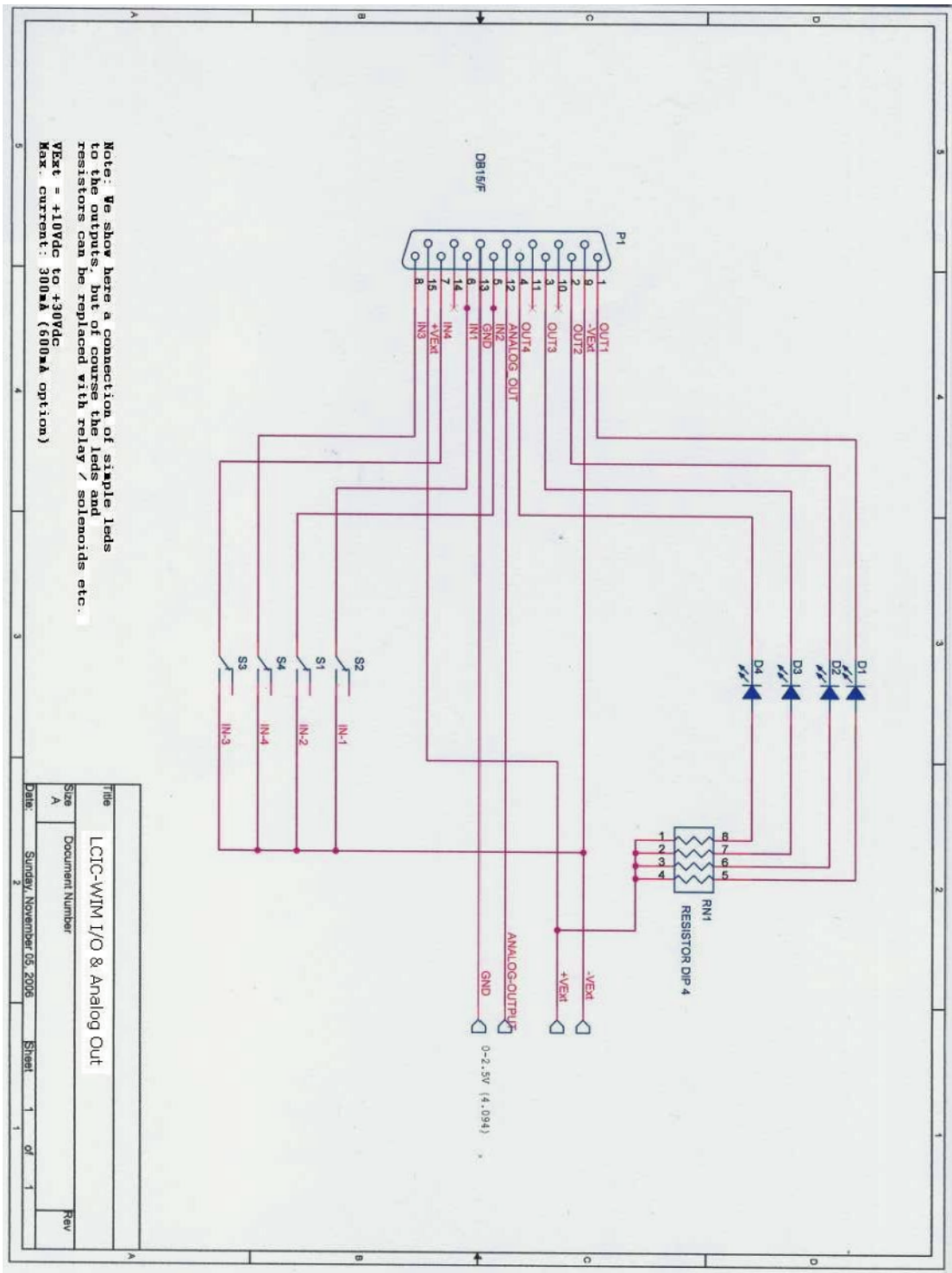
<b>Pin</b>	<b>Function</b>
<b>1</b>	Output 1
<b>2</b>	Output 2
<b>3</b>	Output 3
<b>4</b>	Output 4
<b>5</b>	Input 1
<b>6</b>	Input 2
<b>7</b>	Input 3
<b>8</b>	Input 4
<b>9</b>	I/O Voltage 0V
<b>10</b>	NC
<b>11</b>	NC
<b>12</b>	Analog Out Signal
<b>13</b>	Analog Out Gnd
<b>14</b>	NC
<b>15</b>	I/O Voltage+ (10 to 30V)

## A.2 Connecting External Devices to the Inputs



CONNECTING EXTERNAL DEVICES TO LCIC-WIM INPUTS (ONLY 1 SHOWN)

## A.3 LCIC-WIM I/O & Analog Out



## A.4 The LED Display

Upon **board restart**, the two following messages are shown on the LED display – each for a while:

LCIC **x.xx**            **x.xx** is board's DSP version.  
Sb **yyy**                **yyy** is current board's Serial baud-rate  
                              (refer to sections 3.3.1.2 & D.2).

Then the display shows the current data.

### Notes

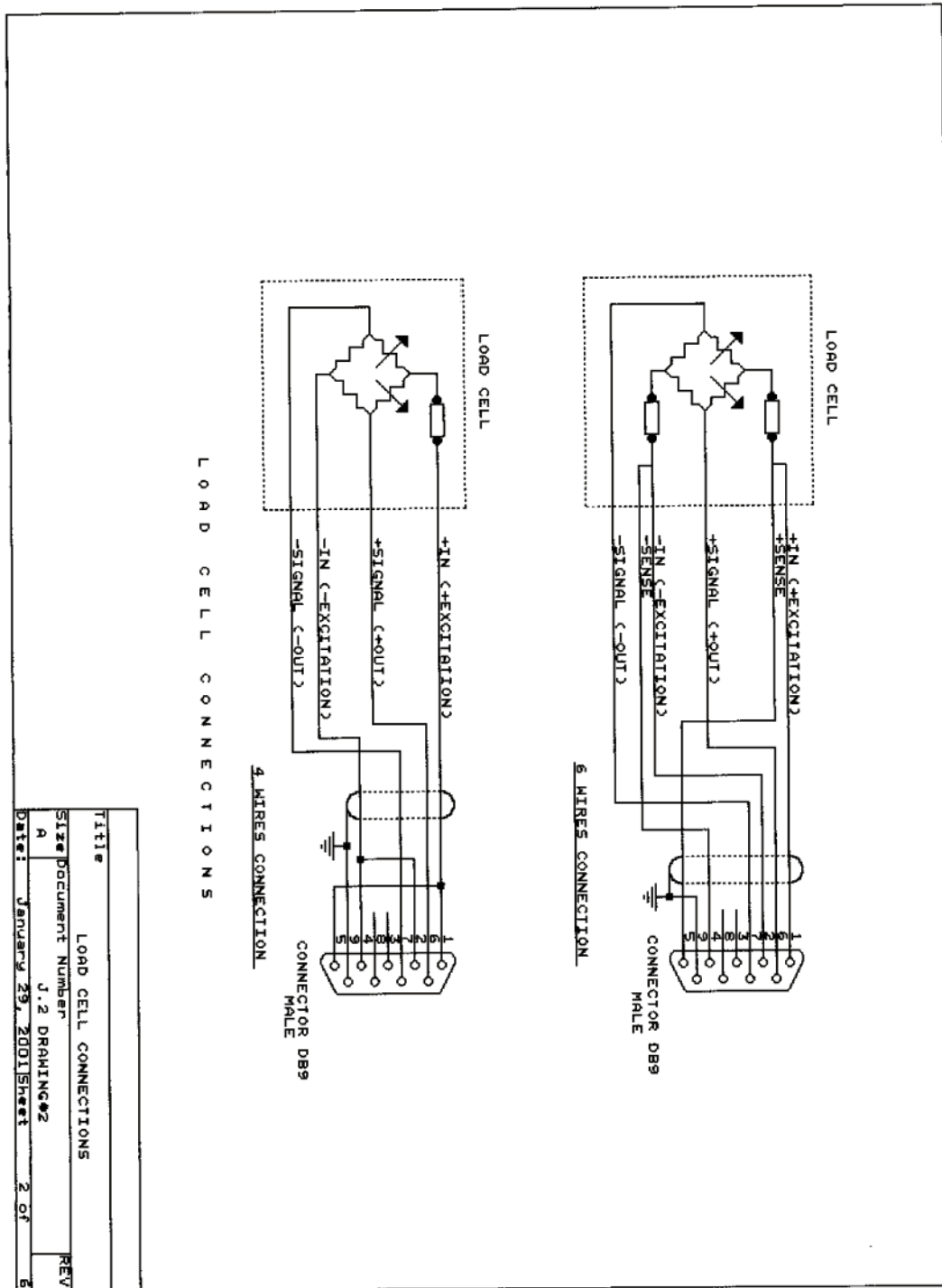
1. The weight on the LED display is always after Filter2 (refer to section 3.3.2.4).
2. In **Fill-mode**, the LED display shows additional information – refer to section E.5.
3. During the Fast Mode (section 4.3.4) the LED display is *not* updated.

## Appendix B: Scaling the Load Cell Input

The full scale of the input coming from the load cell may be adjusted by the LK4 jumper (which is next to load cell connector):

- Across the two **leftmost** pins (default):  
Load cell output is 1-2mV/V.
- Across the two **rightmost** pins:  
Load cell output is 3mV/V.

# Appendix C: Load Cell Connections



## Appendix D: USB, RS232 & RS485

In addition to USB, The LCIC has an option for both full-duplex RS232 and half-duplex RS485 interfaces. These are brought out on CONN3, a 9 way 'D' type connector. The pin-out is as follows:

CONN3 PIN	FUNCTION
1	RS485-
2	RS232 TX (out)
3	RS232 RX (in)
4	NC
5	SIGNAL GROUND
6	RS485+
7	NC
8	NC
9	NC

i.e., for RS232 use pins 2,3 & 5 and for RS485 use pins 1,5 & 6  
For RS232 connection, a standard direct (straight through) wired 9D M-F cable can be used for direct connection to a standard 9 way 'D' type PC COM port.

RS485 line termination – placing a jumper across the two rightmost pins of LK1 puts a 120 Ohm (ac coupled) impedance across the RS485 data lines. This should be used in single LCIC RS485 applications. In applications where multiple LCIC boards reside on one RS485 bus, the termination impedance should only be added on the last board on the bus.

### D.1 Communication Parameters

For both the USB, the full-duplex RS232 and half-duplex RS485 interfaces, the communications parameters are fixed as follows:

Data           8 bits  
Parity         None  
Stop Bits      1

\_ This is generally referred to as "8,N,1".



## **D.2 Baud Rate**

For the **USB**, the maximal baud rate is 921,600. The board responds well without a need to pre-define the used b/r.

For **serial communication**, the required b/r should be pre-defined by the user via the Settings utility (section 3.3). The available baud rates are between 19,200 and 115,200. The current b/r used by the board for the serial communication is shown for a while on the LED display upon card reset, prefixed by 'Sb' (=Serial baud rate). Refer also to sections 3.3.1.2 & 3.3.2.1.2.

## **D.3 General RS232 Terminal**

You may talk with the card either by your own application or by a general RS232 terminal. One simple one called Termite is available for free at [http://www.compuphase.com/software\\_termite.htm](http://www.compuphase.com/software_termite.htm)

## **D.4 Serial Communication & PC Power On/Off**

After PC power on or off the serial communication (RS232/RS485) is likely to drop. A card reset is needed in this case.

## **D.5 RS485**

Up to 64 LCIC-WIM boards may be connected to one PC port.

In the **PC side**, use a converter either from the RS232 port, or from the USB port (that is, RS232 to RS485 converter, or USB to RS485 converter). In the **board side**, use the serial port (CONN3) – refer to the table in the beginning of this appendix.

Using the LCIC-WIM-SETTINGS utility, each board is assigned a unique address between 1 and 64 – refer to section 3.3.2.1.3 (**never give the same address to more than one board**).

An RS485 board may be either ‘active’ or ‘inactive’. An ‘active’ board honours all commands, while an ‘inactive’ board honours only the RS485 address handling commands. Upon board reset it ‘wakes up’ inactive. A board **starts** being active when it receives **its** ‘activate board’ command and **stops** being active when an ‘activate board’ command to any other board is sent. Hence, only one board (at most) may be active at the same time.

About the handling of the RS485 address while coding your application please refer to the RS485 Commands and RS485 Responses sections below, which expand the summary given in section 4.1/h (‘RS485: Address selection, setting & reading’).

→ When a board is configured as an RS485 device, its address is shown for a while on the LED display upon board reset, for example “rS.485-18” for the board addressed 18.

## RS485 Commands

**Except 'Nx<c/r>' (paragraph d), these commands are available also in the fill mode.**

**a. Activate address x (x=1, 2, 3, ..., 62, 63, 64):**

**:x<c/r>**

Board(s) response:

1. If address x is already active:

**!x<c/r>**

(The board reports that it is already active and has nothing to do.)

2. If address x is not active:

2.1 If another address is currently active:

**^x<c/r>** (*I'm going to sleep*)

(The 'falling' board reports that it received an activation command to another board, then it makes itself inactive.)

2.2 If address x exists:

**Ax<c/r>** (*I'm becoming active*)

(The 'rising' board reports that it makes itself active.)

*(Please refer to the notes on the next page.)*

## Notes

1. Wait 10 ms after sending the colon (':') before sending the rest of the command ('**x**<c/r>').
2. Wait 30 ms after sending the ':**x**<c/r>' command before analyzing the response(s).
3. There might be four cases with the conditions of 2.1 & 2.2:
  - #1: Both conditions are true:  
Both responses will be transmitted – first '^**x**<c/r>' and then '**Ax**<c/r>'.  
(Old address retired and address **x** became active.)
  - #2: None of the conditions is true:  
There will be no response.  
(No address was active before, none is active now.)
  - #3: Only 2.1 is true:  
'^**x**<c/r>' will be transmitted.  
(No address is active, as old address retired and the new one does not exist.)
  - #4: Only 2.2 is true:  
'**Ax**<c/r>' will be transmitted.  
(Address **x** became active.)
4. Verify normal address switching by the 'n' command. In case of unexpected response to the 'n' command, repeat the ':**x**<c/r>' command. In case #1 of note 3 (which is the most frequent) this verification is not needed, simplifying the switching procedure.

**b. Hello:**

:999<c/r>

The 'hello' command is like a 'who is alive?' question. This is useful in order to detect which addresses exist in the system. Each existing board, whether active or not, responds '\*x<c/r>', where x is its address.

Notes

1. Wait 10 ms after sending the colon (':') before sending the rest of the command ('999<c/r>').
2. The '\*x<c/r>' responses will be transmitted in sequence. That is, if all 64 addresses exist, then first '\*1<c/r>' will be transmitted, then '\*2<c/r>' and so on, and finally '\*64<c/r>'.
3. The 'hello' command does not change the 'active' mode of the address – it will remain the same as before.
4. After sending the 'hello' command, wait 1 second to give chance to all 64 potential addresses to respond.

### c. Read the RS485 address of the active board:

n

Board's response:

'#x</r>', where **x** is the address of the active board.

**x**=0 means that the board has been configured as a non-RS485 device.

Other value (between 1 & 64) specifies the RS485 address of the active board.

#### Notes

1. After sending the 'n' command, wait 1 second to give chance to all 64 potential addresses to respond.
2. If no board is active, there will be no response.
3. In case of malfunction, more than one '#x</r>' will be responded, in sequence. This is theoretically impossible but should be checked in order to be on the safe side. In this case re-activate the required address by the 'activate address' command (paragraph a).

### d. Set the specified RS485 address:

N**x**</r>

This command sets the board's RS485 address. **x** is between 0 and 64. (**x**=0 sets the board as a non-RS485 device; this will facilitate your coding, as no ':**x**</r>' command will be needed.) Note: When the RS485 address is not 0, its value is shown for a while on the LED display upon board reset.

Board's response: 'OK</r>'.

Normally a user application will never use this command, as the address setting is carried out using the LCIC-WIM-SETTINGS utility.

## RS485 Responses

(Most responses are already described in the ‘Commands’ section.)

!x<c/r>	Address <b>x</b> is already active. See Commands/a/1.
^x<c/r>	Becoming inactive on behalf of address <b>x</b> . See Commands/a/2.1.
Ax<c/r>	Address <b>x</b> becomes active. See Commands/a/2.2.
*x<c/r>	Address <b>x</b> is alive. See Commands/b.
?x<c/r>	A ‘:x<c/r>’ command was received but <b>x</b> is illegal. That is, <b>x</b> is neither in the range (1,...,64), nor 999.
#x<c/r>	Address <b>x</b> is active. See Commands/c.

## Examples

Suppose there are 3 boards in the system, addressed 1, 2 & 3.  
(The **blue** text is the PC side, the **red** text is the response from the board(s), and the black text is our comments)

### Example #1: Everything goes fine

**:999**

**\*1**

**\*2**

**\*3**

**n**

(no response as no board is active)

**:1**

**A1** Board #1 becomes active

**n**

**#1**

**:2**

**^2** Responded by board #1. Means: I (board #1) am becoming inactive in favor of board #2, even though it's unknown for me whether board #2 exists or not.

**A2** Responded by board #2. Means: I (board #2) am becoming active, assuming that no other board is active.

(Note: Comments analogue to the above two ones (about ^2 and A2) are relevant also for the other ^x and Ax commands; however, for the sake of readability, the following comments are shorter.)

**n**

**#2**

**:3**

**^3** Board #2 becomes inactive in favor of board #3

**A3** Board #3 becomes active

**n**

**#3**

**:1**

**^1** Board #3 becomes inactive in favor of board #1

**A1** Board #1 becomes active

**n**

**#1**



*So far everything was smooth; however, the quality of the communication depends – besides the board and the PC – also on the environment. Hence, there might be irregular situations that the user should know to handle – this is what the following two examples (#2 and #3) explain.*

#### Example #2: No response from the **new** board

n

#1

:2

^2 Board #1 becomes inactive in favor of board #2

But the 'A2' response, telling that board #2 became active, did not arrive! Where is the disorder? Maybe board #2 did receive the command and is indeed active, just the 'A2' response was lost, and everything is OK (case #1); but maybe board #2 did not receive the command and no board is active (case #2). In both cases sending the ':2' command again will throw light on the situation: In case #1 '!2' will be responded telling that board #2 was already active; in case #2 'A2' will be responded telling that board #2 became now active. In both cases everything is ok and we may proceed. However, if there is no response at all on the ':2' command, repeat it say, 3 or 4 times and if there is still no response, then there is some severe problem requiring a human action, maybe board reset.

Note that theoretically the 'n' command could be used as well, but practically it is recommended to avoid using the 'n' command – where possible – for some reasons:

1. The 'n' command is time consuming – it requires waiting 1 second in order to let all potential boards respond. (This is unavoidable as – by definition – the mechanism of the 'n' command takes into consideration also a faulty situation in which two (or even more) boards are active at the same time. This mechanism ensures that the responses will arrive in sequence and not simultaneously, therefore it consumes so much time.)
2. The conclusion derived according to the response on the 'n' command is not always clear. That might require sending the 'n' command again, requiring another 1 second.
3. The 'n' command is only informative, it does not fix anything. If it's possible – as in our case – to both fix a problem and get information at the same time, then it is preferable.

### Example #3: No response from the **old** board

n

#1

:2

A2 Board #2 becomes active

But the '^2' response, telling that board #1 became inactive, did not arrive! Where is the disorder? Maybe board #1 did receive the command and is indeed inactive, just the '^2' response was lost, and everything is OK (case #1); but maybe board #1 did not receive the command and both boards are active (case #2). This is a dangerous situation and definitely should be avoided. This problem is more complicated than the previous one (in example #2) – sometimes re-sending the ':2' command won't help: In case #2 re-sending the ':2' command should be responded by '^2' which indicates that everything is OK. ('!2' will be responded as well, but this does not add us any new information.) However, if the '^2' response does not arrive, we have to use the 'n' command in order to make sure that we are not in the situation of two boards active at the same time, which should be avoided. Therefore, we should send the 'n' command and expect to get only '\*2' (and not '\*1'). Although it's time consuming, we have to repeat the 'n' command at least once in order to have no doubt that board #1 is really inactive.

## **Board selection by the supplied utilities**

The three supplied utilities – LCIC-WIM-CALIBRATION, LCIC-WIM-SETTINGS & LCIC-WIM-MONITOR – enable easy selection of the required board:

- Upon program start, all detected addresses are reported. Verify that the total number of boards detected (reported at the bottom of the display) corresponds the real number. It might occur that the automatic detection fails to detect a board. If you know that some board does exist although it was not detected, click its address manually. Finally, select the board that will be activated first by right-clicking its address (or leave the default selection) and click 'Continue' (or just wait).
- When you are accessing some board, usually you may switch to another one. (In the LCIC-WIM-CALIBRATION this option is available only in step 1.) In order to switch the active board, right click the mouse. You'll get a list of all existing boards, with the active one dimmed and checked. You may select another board to be activated by clicking its address.

## Appendix E: Fill Mode

### E.1 Introduction

The LCIC-WIM supplies a *Fill Mode* in which it may control a filling operation, using the *hardware inputs* (section E.2) and *hardware outputs* (section E.3). The character of the filling operation is determined by *parameters* (section E.4) set via the *Settings utility* (section 3.3). The filling results are shown on the *LED display* (section E.5). The Fill Mode supports also *commands* (section E.6) sent through the communication line (USB, RS232 or RS485). These commands may draw the filling results as well as send operational instructions, such as start a filling operation. Both the display and the communication line may indicate an *error code* (section E.7). During the filling cycles the board learns the system in order to improve the results of the following cycles by compensating system's unavoidable inaccuracy. The *first* cycle is split into two parts (start-stop-start-stop) enabling the board to learn the system, thus apply already the above compensation, trying to achieve a correct result even in the first cycle.

## E.2 Hardware Inputs

<b>Input #1</b>	<b>Input #2</b>	<b>Input #3</b>		<b>Input #4</b>
<b>OFF = Manual</b>	ON = Turn output #1 on	ON = Turn output #2 on		N/A
<b>ON = Auto</b>	ON = Start	When in error status:	ON = Terminate an Error Status	ON = Emergency Stop
		When <u>not</u> in error status <b>and</b> <u>not</u> during filling:	<u>Long</u> ON = Special Mode (*)	

### \* The Special Mode

In this mode:

- \* 'SP' flashes on the left side of the LED display.
- \* The three setpoints are shown, in turn, on the LED display.

In this state (while input #3 is still on) you have two options:

#### 1. Switch the setpoint

Turn input #3 off.

The setpoint currently shown on the LED display becomes effective.

The LED display will show 'SUCCESS' and then the board will return to regular auto-mode. You will be able to see the effective setpoint on the LED display (prefixed by 'SP'). If the 'get results immediately' option is selected, the board reports to the PC: 'Setpoint selected as...'.

#### 2. Switch to general (non-fill) mode

Turn input #2 on for ~2 seconds.

The LED display will show 'Gen.Mode' and then the board will switch to the general mode.

Note: When the board switches to the general mode, make sure to turn input #3 off, otherwise the manual zero operation will take place – refer to section H.1 / Manual Zero / Hardware input.

### E.3 Hardware Outputs

<b>Output #1</b>	<b>Output #2</b>	<b>Output #3</b>	<b>Output #4</b>
<b>Option #1</b>		Error	Filling Complete (Only with board firmware 1.11 or higher.)
Fast Valve	Slow Valve		
<b>Option #2</b>			
Fast Valve = Output #1 + Output #2 Slow Valve = Output #2 only			

(About [Options #1](#) and [Option #2](#) refer to 'Fast Speed Config' in section E.4.3)

## **E.4 Filling Parameters**

### **Notes**

- 1. There is a set of three setpoints. Once they are specified (using the Settings utility), the user may switch to another setpoint without needing a PC. This gives more flexibility when several setpoints are needed. For details about the switching procedure refer to section E.2.*
- 2. These setpoints are absolutely different from the four general setpoints (section 3.3.1.3). The general setpoints are inactive in the Fill mode.*

### **Filling By**

#### **Weight**

Means that **Setpoint** is defined in *weighing units*.

#### **Time**

Means that **Setpoint** is defined in *time units*.

### **E.4.1 Filling By = Weight**

#### **E.4.1.1 Filling by Weight Parameters**

##### **Setpoint #x**

The required total filling weight when the user selects setpoint #x (x = 1, 2 or 3).

##### **Slow Amount**

The required slow filling weight, in % of the current Setpoint (a tip shows the value of **Slow Amount** in weight units). Specify '0' when only one speed is required.

##### **Filling Timeout**

Time limitation for the filling process (in ms).

Note: The 'Filling Timeout' parameter is common for all the three setpoints. Therefore, specify a 'Filling Timeout' value large enough to cover all the setpoints used.

## **Auto Correction & Averaging *x* last fillings**

When 'Auto Correction' is checked, the board tries to correct the filling amount, based on the results of the last *x* fillings.

### **E.4.1.2 Auto Tare**

#### **Activate**

- When *not checked*, the Setpoint defines the requested **final gross weight**.  
That is, if the setpoint is 100 kg and the starting gross weight is 90 kg, the filling amount will be 10 kg.  
The next two parameters (Hi/Lo Limits) are irrelevant.
- When *checked*, the Setpoint defines the requested **filling amount**.  
That is, if the setpoint is 100 kg, the filling amount will also be 100 kg, regardless of the starting gross weight.  
However, the filling operation will take place only if the starting gross weight, as found in accordance with the Stabilization Criterion (section E.4.1.4), is within the range defined by the following two parameters (Hi/Lo Limits).  
Otherwise, the filling operation will be rejected and an error will be reported.

#### **Hi Limit**

When 'Auto Tare' is activated, specifies the *high* allowed tare limit.

#### **Lo Limit**

When 'Auto Tare' is activated, specifies the *low* allowed tare limit.



### **E.4.1.3 Valid Results Limits**

The resulting filling weight should normally be inside a user pre-defined 'valid range'. In case the weight exceeds that range, an error situation will occur. Specify 'Valid Limits ( $\pm$ )' = 0 if you don't need this check.

#### **Start from filling # ...**

Specifies the first filling to be checked, letting you disregard some filling until the board learns the system thus gives good results. For example, if 7 is specified, then no error situation will occur upon the first 6 fillings, whatever the results will be.

#### **Valid Limits ( $\pm$ ) ... %**

Specifies the acceptable deviation for a valid result, as a percent of the current setpoint (a tip shows the value of that acceptable deviation in weight units).

For example, if the current setpoint = 100 kg and 'Valid Limits ( $\pm$ )' = 1%, then the result will be considered as 'valid' when it is in the range [99,101] kg. Otherwise, error #101 or #102 will occur. Specify 'Valid Limits ( $\pm$ )' = 0 if you don't need the 'Valid Results Limits' check.

#### **E.4.1.4 Stabilization Criterion (Tare & Stop)**

At the beginning and at the end of a filling cycle the board waits for the scale to stabilize in order to read its weight. Hence, some *stability criterion* is required. The board requires that all readings within ‘Delta Time’ will be inside a range whose width is ‘Delta Weight’, both at the beginning (Tare) and at the end (Stop) of the filling cycle. The waiting for that stabilization is limited by ‘Timeout’, causing an error situation in case of failure.

##### **Delta Weight**

See the description above.

##### **Delta Time**

See the description above.

##### **Stabilization Timeout**

See the description above.

##### **Impacts**

The system copes with impacts and automatically resumes the filling in case the current weight – after stabilization – is less than the current setpoint.

#### **E.4.1.5 Lazy Filling**

##### **Activate**

When checked, the board will identify too slow filling, thus reporting an error earlier than the ‘Timeout’ check would do.

## **E.4.2 Filling By = Time**

### **E.4.2.1 Filling by Time Parameters**

#### **Setpoint #x**

The required total filling time when the user selects setpoint #x (x = 1, 2 or 3).

#### **Slow Amount**

The required slow filling time, in % of the current Setpoint (a tip shows the value of **Slow Amount** in ms). Specify '0' when only one speed is required.

### **E.4.2.2 Stabilization Criterion (Stop)**

At the end of a filling cycle the board waits for the scale to stabilize before proceeding to the next cycle. The size of this waiting delay is specified by the “**Delta Time**” parameter.

#### **Delta Time**

See the description above.

## **E.4.3 Fast Speed Config**

#### **Option1**

Fast Speed = Output #1

Slow Speed = Output #2

#### **Option2**

Fast Speed = Output #1 + Output #2

Slow Speed = Output #2

## E.4.4 The Filling Configurations Library

There is a 'Filling Configurations Library' in which you may save sets of filling configuration parameters. This is useful in case you have more than one type of filling, letting you switch easily and reliably from one configuration to another.

(Note: This library has nothing to do with the *calibration* library mentioned in section 3.2.2.)

Refer to the 'Library' box.

### Save

If you like your configuration to be saved in the library, specify a Library Name and check the option box. (It is recommended to specify a meaningful Library Name so that later you'll recognize the various files you created in the library.) Your filling configuration will be saved in the library only upon clicking the 'Save to Board' button. In case a library file having the name you specified already exists, you'll have to select either to overwrite it, or to use another name.

### Recall or Delete a File in the Library

Click the Select button in order to select a file to be recalled or deleted.

You may watch the various files and their contents. In order to **delete** a file, click the Delete button. The procedure to **recall** a file has two or three steps:

1. Click the Confirm button. This will insert file's parameters to the 'Filling Definition' box.
2. Optionally, you may modify some of the parameters, thus using the library file as a draft to make your changes easier. In case you want the new values to be saved in a file (either the original one or another), make sure to check the option in the 'Library' box.
3. Click the 'Save to Board' button.

## E.5 LED Display Notations

In **Fill-mode**, the LED display shows the current sub-mode:

m Fill      Manual Fill-mode (input #1 is off)  
AutoFill    Auto Fill-mode (input #1 is on)

Likewise, the value shown on the LED display is prefixed by one or two letters:

C      Current weight  
F      Weight during a filling cycle  
A      Actual weight (the resulting final filling weight)  
SP     SetPoint

### The scale of the displayed weight

The displayed weight is sometimes **gross** and sometimes **net**:

\* During a Filling Cycle, **and** the user selected Auto Tare:

    The displayed weight is **net**.

\* In all other cases (that is, not during a Filling Cycle, or even during a Filling Cycle but the user did **NOT** select Auto Tare):

    The displayed weight is **gross**.

## E.6 Commands

### Enter & Exit Fill Mode

- F Enter Fill Mode (from General Mode)
- x Exit Fill Mode (to General Mode) (small 'x')

### Inside the Fill Mode

- g Start filling ('g' stands for 'go') (like input #2 does).
- t Terminate an error status (like input #3 does).
- e Emergency stop (like input #4 does).
- r Get filling(s) report  
Response Example:  
**# 1 A= 40.99 Tr= 6.65 Cv= 0.00 Ft= 7527 ms St= 2554 ms Cc= 0**  
Legend: Filling #1, Actual (=final) weight=40.99,  
Tare=6.65, Correction Value=0.00,  
Fast time=7527 ms, Slow time=2554 ms,  
Completion Code=0 (0 is normal, otherwise it's an error code).  
Notes:
  1. The actual (final) weight (40.99 in the example) is:
    - \* *net weight* if Auto Tare is selected
    - \* *gross weight* if Auto Tare is **not** selected.
  2. The reports are accumulated in a FIFO whose size is 30 fillings. That is, only up to last 30 reports are available.  
Note: The FIFO size (30) is subject to change in the various versions.
  3. Please refer also to section 3.3.2.1.4.
- F Get fillings summary:  
Number of normal fillings.  
Number of each error type cases for each error type that occurred at least once.

- s Get current status (small 's')  
Response Example:  
**Current\_Status: W= 17.14 Tr= 6.65 Cv= 0.00 M=F A=I S= 0**  
Legend: Current (gross) weight=17.14, Last Tare=6.65,  
Next Correction Value=0.00, Mode=Fill mode,  
Activity=Idle (or: Tare, Fast, Slow)  
System status=0 (0 is normal, otherwise it's an error code).
- p Get parameters list (small 'p').
- i Turn **off** the "Get results immediately" feature.
- I Turn **on** the "Get results immediately" feature.
- V Get current mode (upper case V).  
Board's response: 'Fill-mode'.  
Note: This command is useful in order to find out board's mode. In the general mode the response to the 'V' command is '**LCIC-WIM/V#.#**' (refer to section 4.1/i), as opposed to the 'Fill-mode' response in the fill mode.

## **E.7 Error Codes**

- 101 Actual Filling Weight < Low Limit of 'Valid Result Limits'.
- 102 Actual Filling Weight > High Limit of 'Valid Result Limits'.
- 103 SetPoint < Low Limit of 'Valid Result Limits'.
- 104 SetPoint > High Limit of 'Valid Result Limits'.
- 105 High Tare Limit < Low Tare Limit
- 106 Actual Tare < Low Tare Limit
- 107 Actual Tare > High Tare Limit
- 108 Valid High Limit < Valid Low Limit
- 109 Lazy Filling
- 111 Filling Timeout
- 112 Low Tare Limit > High Tare Limit
- 113 SetPoint < Stabilization Delta Weight
- 115 Tare Stabilization Timeout
- 116 Initial weight is too large
- 117 Slow amount is too large
- 118 End Stabilization Timeout
- 119 Stabilization Delta Weight < Resolution
- 120 Memory Failure
- 121 User Emergency Stop
- 122 Stabilization Delta Time > Stabilization Timeout
- 123 Filling Timeout > Stabilization Timeout



## **Appendix F: Specifications**

### **F.1 Load Cell Input**

- 5 Volt excitation for upto 10 load cells (350 Ohm)
- Compatible with 1, 2 & 3 mV/V load cells
- Low noise wide bandwidth amplifier & 24 bit ADC

### **F.2 A/D**

- Very high speed A/D: upto 52,000 samples per second
- 24 Bit A/D with  $\pm 8$  million counts for tension and compression applications

### **F.3 Digital Inputs**

- 4 opto-isolated inputs with 10-30 VDC range, each with status LED
- Input #2 configurable as high speed counter

### **F.4 Digital & Analog Outputs**

- 4 opto-isolated solid state relays rated at 50V, 300mA, each with status LED. Configurable as setpoints or manual outputs.
- Analog output of 0 to 2.5V with 16 bit resolution.

### **F.5 Standard Interfaces**

- USB 2.0 Full Speed compatible
- Combined RS232/RS485
- Multiple boards may be connected via USB or RS232/RS485
- Ideal for PLC based applications

### **F.6 Software**

- LCIC-WIM Calibration Wizard software is included.
- Included is the LCIC-WIM Monitor utility which is a vital tool for analyzing dynamic load/force systems. It takes full advantage of the board's speed.
- Also included the LCIC-WIM-SETTINGS utility which gives control to card's filters, analog output, filling parameters and other settings.
- An ActiveX interface is supplied for easier programming of user's application; however, direct conversation with the board is available either.

### **F.7 Dimensions (mm)**

- Standard OEM model 160 x 100pcb (Eurocard)
- ABS cased option

### **F.8 Misc.**

- Powerful 32 bit / 135 MIPS DSP for high speed onboard processing.
- 8 digit LED display
- On board temperature sensor
- Card includes an integral Fill Mode supplying an independent filling control.

## **Appendix G: Trouble-shooting**

### **G.1 Card does not respond after PC power-on**

**Q.** Everything was OK, but after PC restart the card suddenly stopped responding.

**A.** As specified in section D.4, after PC power on or off the serial communication (RS232/RS485) is likely to drop. A card reset is needed in this case.

## Appendix H: Zero & Tare

There are two functions which are similar, yet actually different: Zero & Tare:

- The *Zero* function supplies both manual and automatic ways to clear the *gross weight*.
- The *Auto-Tare* function supplies a way to define the meaning of the *setpoint parameter*.

### H.1 The Zero function

This function supplies both manual and automatic ways to clear the *gross weight*:

#### Manual Zero

The manual zero is available only in the general mode, **not** in the fill-mode. Its effect is temporary – it expires upon card reset.

The manual zero is accessible via two ways:

- \* Hardware input: Turn on digital input #3 for two seconds.
- \* Communication:
  - \* When accessing the communication directly:  
Send the ‘z’ command (section 4.1/i).
  - \* When using the LCIC-WIM ActiveX:  
Call the Apply\_Temporary\_Zero function (section 4.3.5).

#### Auto Zero

The auto zero is available only in the fill-mode, **not** in the general mode. Its effect is temporary – it expires upon card reset or upon exiting the fill-mode.

The auto zero operation occurs when some user’s pre-defined condition is satisfied. For more details refer to section 3.3.2.2.

## **H.2 The *Auto-Tare* function**

This function supplies a way to define the meaning of the *setpoint parameter*:

- \* When the 'AutoTare' option is **not** activated, the setpoint defines the requested **final gross weight**.  
That is, if the setpoint is 100 kg and the starting gross weight is 90 kg, the filling amount will be 10 kg.
- \* When the 'AutoTare' option is activated, the setpoint defines the requested **filling amount**.  
That is, if the setpoint is 100 kg, the filling amount will also be 100 kg, regardless of the starting gross weight.  
However, the filling operation will take place only if the starting gross weight is within a user pre-defined range; otherwise, the filling operation will be rejected and an error will be indicated.

The activation of the 'Auto-Tare' option as well as its parameters are described in section E.4.1.2.

## WARRANTY/DISCLAIMER

OMEGA ENGINEERING, INC. warrants this unit to be free of defects in materials and workmanship for a period of **13 months** from date of purchase. OMEGA's WARRANTY adds an additional one (1) month grace period to the normal **one (1) year product warranty** to cover handling and shipping time. This ensures that OMEGA's customers receive maximum coverage on each product.

If the unit malfunctions, it must be returned to the factory for evaluation. OMEGA's Customer Service Department will issue an Authorized Return (AR) number immediately upon phone or written request. Upon examination by OMEGA, if the unit is found to be defective, it will be repaired or replaced at no charge. OMEGA's WARRANTY does not apply to defects resulting from any action of the purchaser, including but not limited to mishandling, improper interfacing, operation outside of design limits, improper repair, or unauthorized modification. This WARRANTY is VOID if the unit shows evidence of having been tampered with or shows evidence of having been damaged as a result of excessive corrosion; or current, heat, moisture or vibration; improper specification; misapplication; misuse or other operating conditions outside of OMEGA's control. Components in which wear is not warranted, include but are not limited to contact points, fuses, and triacs.

**OMEGA is pleased to offer suggestions on the use of its various products. However, OMEGA neither assumes responsibility for any omissions or errors nor assumes liability for any damages that result from the use of its products in accordance with information provided by OMEGA, either verbal or written. OMEGA warrants only that the parts manufactured by it will be as specified and free of defects. OMEGA MAKES NO OTHER WARRANTIES OR REPRESENTATIONS OF ANY KIND WHATSOEVER, EXPRESS OR IMPLIED, EXCEPT THAT OF TITLE, AND ALL IMPLIED WARRANTIES INCLUDING ANY WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE HEREBY DISCLAIMED. LIMITATION OF LIABILITY: The remedies of purchaser set forth herein are exclusive, and the total liability of OMEGA with respect to this order, whether based on contract, warranty, negligence, indemnification, strict liability or otherwise, shall not exceed the purchase price of the component upon which liability is based. In no event shall OMEGA be liable for consequential, incidental or special damages.**

CONDITIONS: Equipment sold by OMEGA is not intended to be used, nor shall it be used: (1) as a "Basic Component" under 10 CFR 21 (NRC), used in or with any nuclear installation or activity; or (2) in medical applications or used on humans. Should any Product(s) be used in or with any nuclear installation or activity, medical application, used on humans, or misused in any way, OMEGA assumes no responsibility as set forth in our basic WARRANTY/DISCLAIMER language, and, additionally, purchaser will indemnify OMEGA and hold OMEGA harmless from any liability or damage whatsoever arising out of the use of the Product(s) in such a manner.

## RETURN REQUESTS/INQUIRIES

Direct all warranty and repair requests/inquiries to the OMEGA Customer Service Department. BEFORE RETURNING ANY PRODUCT(S) TO OMEGA, PURCHASER MUST OBTAIN AN AUTHORIZED RETURN (AR) NUMBER FROM OMEGA'S CUSTOMER SERVICE DEPARTMENT (IN ORDER TO AVOID PROCESSING DELAYS). The assigned AR number should then be marked on the outside of the return package and on any correspondence.

The purchaser is responsible for shipping charges, freight, insurance and proper packaging to prevent breakage in transit.

FOR **WARRANTY** RETURNS, please have the following information available BEFORE contacting OMEGA:

1. Purchase Order number under which the product was PURCHASED,
2. Model and serial number of the product under warranty, and
3. Repair instructions and/or specific problems relative to the product.

FOR **NON-WARRANTY** REPAIRS, consult OMEGA for current repair charges. Have the following information available BEFORE contacting OMEGA:

1. Purchase Order number to cover the COST of the repair,
2. Model and serial number of the product, and
3. Repair instructions and/or specific problems relative to the product.

OMEGA's policy is to make running changes, not model changes, whenever an improvement is possible. This affords our customers the latest in technology and engineering.

OMEGA is a registered trademark of OMEGA ENGINEERING, INC.

© Copyright 2008 OMEGA ENGINEERING, INC. All rights reserved. This document may not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of OMEGA ENGINEERING, INC.

# Where Do I Find Everything I Need for Process Measurement and Control? **OMEGA...Of Course!** *Shop online at [omega.com](http://omega.com)<sup>sm</sup>*

## **TEMPERATURE**

- ☑ Thermocouple, RTD & Thermistor Probes, Connectors, Panels & Assemblies
- ☑ Wire: Thermocouple, RTD & Thermistor
- ☑ Calibrators & Ice Point References
- ☑ Recorders, Controllers & Process Monitors
- ☑ Infrared Pyrometers

## **PRESSURE, STRAIN AND FORCE**

- ☑ Transducers & Strain Gages
- ☑ Load Cells & Pressure Gages
- ☑ Displacement Transducers
- ☑ Instrumentation & Accessories

## **FLOW/LEVEL**

- ☑ Rotameters, Gas Mass Flowmeters & Flow Computers
- ☑ Air Velocity Indicators
- ☑ Turbine/Paddlewheel Systems
- ☑ Totalizers & Batch Controllers

## **pH/CONDUCTIVITY**

- ☑ pH Electrodes, Testers & Accessories
- ☑ Benchtop/Laboratory Meters
- ☑ Controllers, Calibrators, Simulators & Pumps
- ☑ Industrial pH & Conductivity Equipment

## **DATA ACQUISITION**

- ☑ Data Acquisition & Engineering Software
- ☑ Communications-Based Acquisition Systems
- ☑ Plug-in Cards for Apple, IBM & Compatibles
- ☑ Datalogging Systems
- ☑ Recorders, Printers & Plotters

## **HEATERS**

- ☑ Heating Cable
- ☑ Cartridge & Strip Heaters
- ☑ Immersion & Band Heaters
- ☑ Flexible Heaters
- ☑ Laboratory Heaters

## **ENVIRONMENTAL MONITORING AND CONTROL**

- ☑ Metering & Control Instrumentation
- ☑ Refractometers
- ☑ Pumps & Tubing
- ☑ Air, Soil & Water Monitors
- ☑ Industrial Water & Wastewater Treatment
- ☑ pH, Conductivity & Dissolved Oxygen Instruments