

# COMPOSER USER MANUAL ADDENDUM 1

*Preliminary*

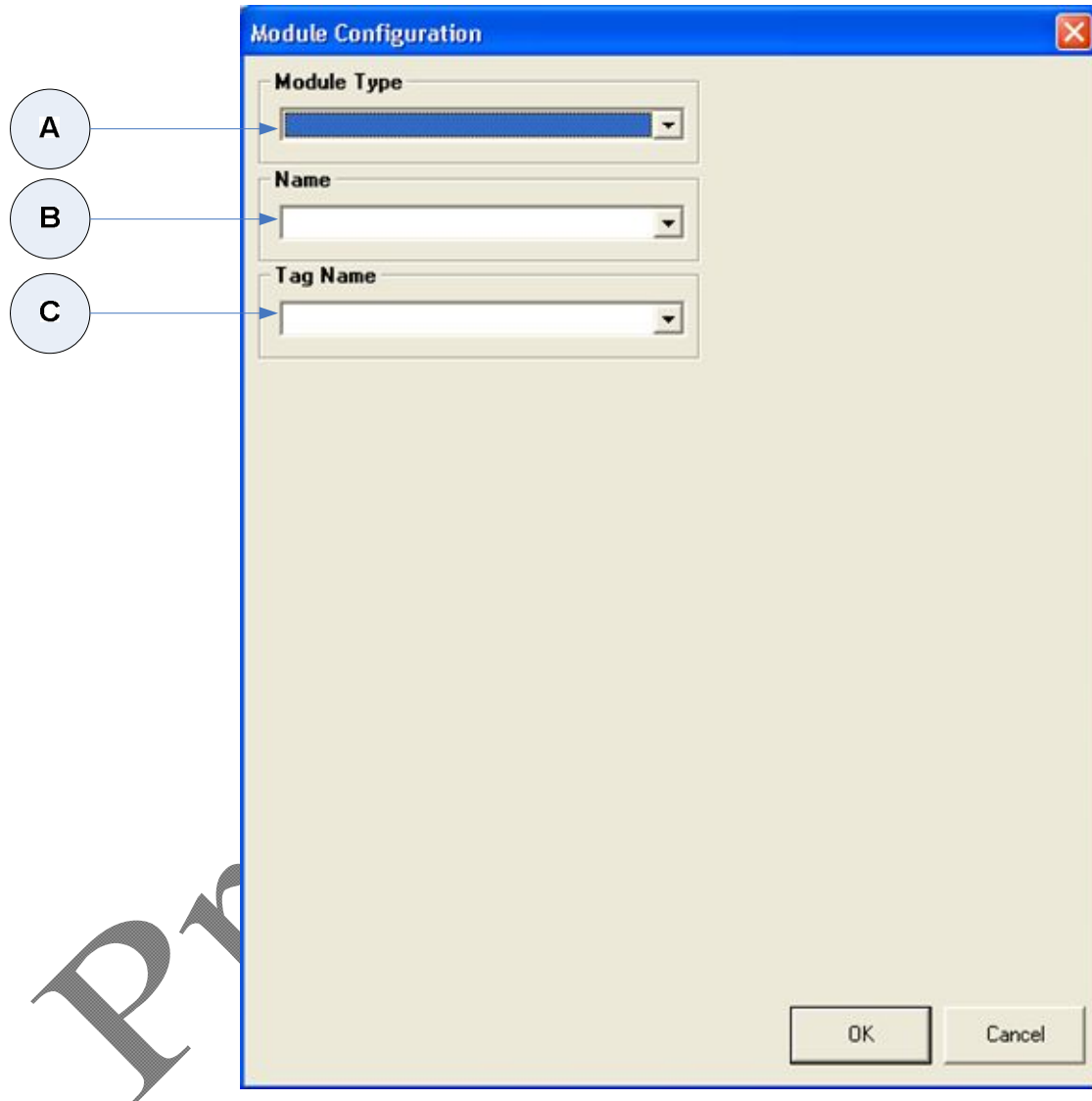
# MODULE CONFIGURATION WINDOW

## Description

Module Configuration

Module View

ADD



## Controls

(A) Module Select Dropdown Menu

ADD

Module View

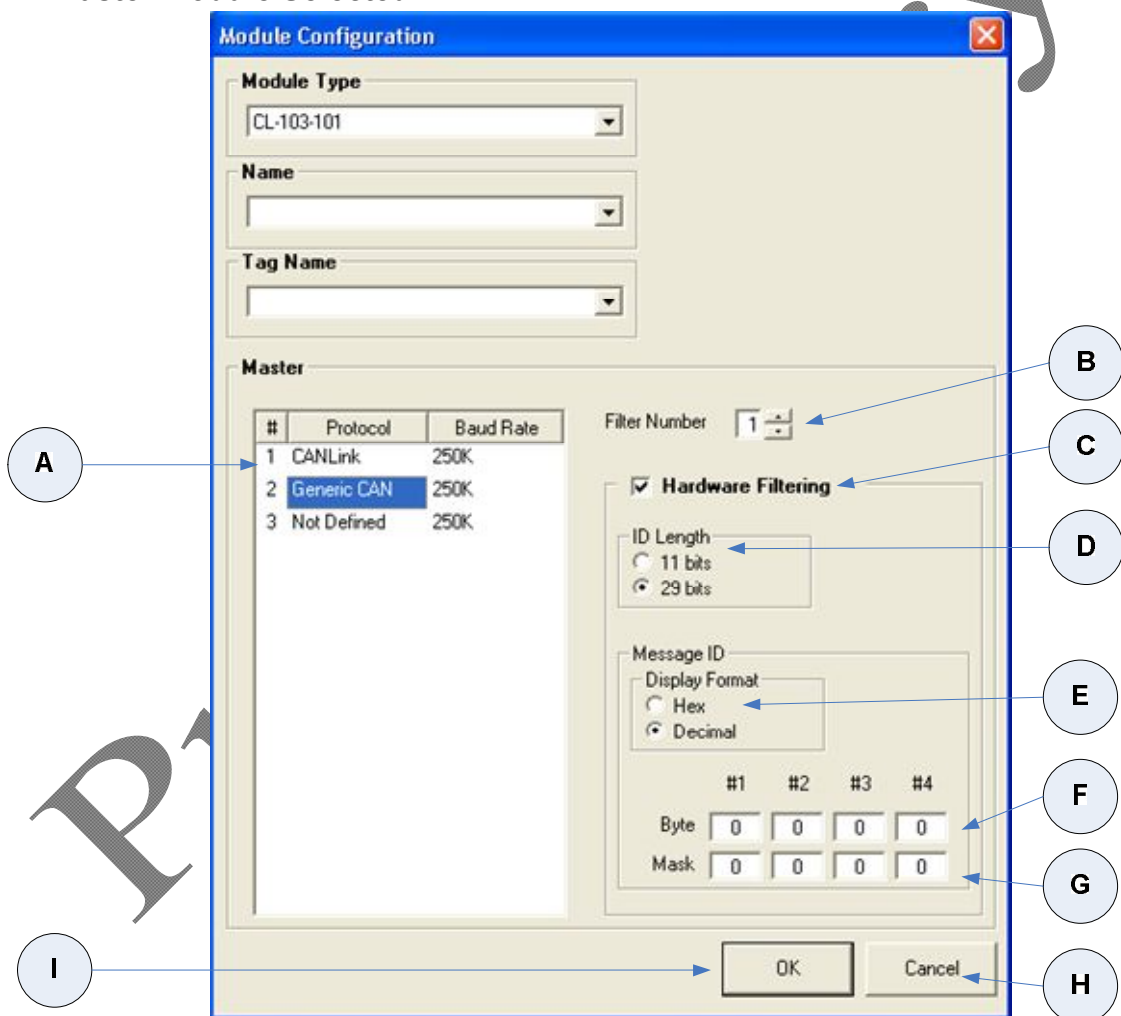
Module View List Box

(B) Module Name Text Box

(C) Module Tag Text Box

Comparison    Operator Blocks    Rung View  
Name

Master Module Selected



**(A) CAN Line Configuration**

This box will list all the CAN lines that are available on the module. They may be configured as 3 different types

1) Not Defined–The CAN line may not be used for modules, CAN receives, CAN transmits, or comm. bridges. This is the default value.

2) CANLink – Define a CAN line as this so it may be used to control modules. CAN receives, CAN transmits and comm. bridges may be defined on these lines however the message must not interfere with the standard CANLink messages. The hardware filters are not available for configuration in this mode.

3) Generic CAN – Define a CAN line as this so it may be used to control modules. CAN receives, CAN transmits and comm. bridges may be defined on these lines however Modules may not. Hardware filters are available to be configured to minimize the number of message the Module will receive.

**(B) Filter Number**

Each CAN line may have multiple hardware filters available and this selects which one is going to be configured. This is a hardware specific number which will vary from module to module.

**(C) Hardware Filtering Enable**

This check box enables and disables the hardware filter. If the filter is disabled it will block all messages. At least one filter should be enabled otherwise all messages will be blocked. Please note that the filters are used in parallel not series.

**(D) ID Length**

This allows the user to select which type of identifier to filter.

**(E) Hex or Dec**

This changes how the id and mask are displayed.

**(F) Byte 1-4**

These 4 bytes build up the identifier the module should accept. The max value that may be entered depends on which id is selected.

**(G) Mask 1-4**

This defines which bits of the incoming message need to match the entered Rx Byte. Mask bits that are high (1) mean it will check the bit and bits that are low (0) will be ignored.

**(H) Cancel Button**

Returns control to the Module View window without making changes.

**OK Button**

Updates and returns control back to the **Module View** window. If there are any errors with the values entered the program will prompt that changes need to be made. At any point cancel may be selected to leave without making any changes.

## Slave Module Selected

The screenshot shows the 'Module Configuration' dialog box. At the top, the 'Module Type' is set to 'CL-201-101'. Below it are fields for 'Name' and 'Tag Name'. The 'Slave' section contains a list of 'Harness No.' options from 0 to 15, each with a checkbox. To the right of this list is a 'Dual CAN' checkbox. Below that is a table for 'Slave CAN Line' and 'Master CAN Line' configuration. The 'Slave CAN Line' is set to 'Primary' and '1'. The 'Master CAN Line' is set to a dropdown menu. At the bottom right are 'OK' and 'Cancel' buttons. Callouts A through E point to the Harness No. list, Dual CAN checkbox, Master CAN Line dropdown, Cancel button, and OK button respectively.

Slave CAN Line	Master CAN Line
Primary	1

### (A) Harness No.

Select the check box for the harness code for the module. Harness codes listed in red are already selected for another module of the same module type.

### (B) Dual CAN

This check box enables dual CAN for the modules. The check box is only available if the module supports dual CAN and in the System Setup the system is configured for dual CAN

### (C) Slave CAN Configuration

This is where the slave CAN line is mapped to a Master CAN line. The pull down will only display Master CAN Lines that have been configured as CANLink. A pull down for the Secondary CAN line is displayed if Dual CAN is enabled and selected.

## COMMUNICATION BRIDGE

### Definition

A communication bridge takes a received CAN message and automatically transmits it. The transmit can be done on every message received, on report rate (set period of time) or disabled. The number of communication bridges varies from master module to master module. The compiler will give an error if the max number has been exceeded.

Each defined Communication Bridge generates 11 read/write variables in the IOMap all of which can be accessed in *Comparison* and *Operator Blocks*:

- Rx Status – This gives the application software feedback that a message has been received. The two options are CLEAR (no message received) and RECEIVED (message has been received). The application code can change the Rx Status.
- Tx Status – This gives the application software the ability to change how the message is being transmitted. The three options are DISABLED (message will not be sent), PASS THROUGH (every received message will be sent) and REPORT RATE (the message will be sent at whatever the Tx Report Rate is set to)
- Tx Report Rate – If the Tx Status is set to REPORT RATE, this is how fast the message will be sent. The units are multiples of the master's loop time.
- Data Items 1-8 – These are filled with the data bytes of the received message every time a message is received. This values can be changed in the rungs however it will only effect messages sent on report rate.

# CREATE / EDIT COMMUNICATION BRIDGE WINDOW

## CREATE / EDIT COMMUNICATION BRIDGE

Multi-View

ADD

The screenshot shows the 'Create / Edit Communication Bridge' dialog box. It is divided into several sections:

- General Information:** Name, Tag Name, Conductor Security Level (Level 2), Group Names (optional) for Group 1 and Group 2.
- Default Tx Status:** Radio buttons for Pass Through (selected), On Report Rate, and Disabled.
- Default Transmit Rate:** Text box containing '100' followed by 'x loop time (in Decimal)'. Callout B points to this field.
- Display Format:** Radio buttons for Hex (selected) and Decimal. Callout C points to this section.
- RECEIVE Section:**
  - Module: None Selected (Callout D)
  - CAN Line: (Callout E)
  - ID Length: Radio buttons for 11 bits (selected) and 29 bits (Callout F)
  - Byte: Four input boxes with values .., .., 00, 00 (Callout G)
  - Mask: Four input boxes with values .., .., 00, 00 (Callout H)
  - Data Byte Filtering: Unchecked checkbox (Callout I)
  - Data Bytes: Eight input boxes with values 00, 00, 00, 00, 00, 00, 00, 00 (Callout J)
- TRANSMIT Section:**
  - Module: None Selected (Callout L)
  - CAN Line: (Callout M)
  - ID Value Adjustment: Unchecked checkbox (Callout N)
  - ID Length: Radio buttons for 11 bits (selected) and 29 bits (Callout O)
  - Byte: Four input boxes with values .., .., 00, 00 (Callout P)
  - Data Length Adjustment: Unchecked checkbox (Callout Q)
  - Data Length = Length of Receive Message
  - Data Bytes: Eight input boxes with values 1, 2, 3, 4, 5, 6, 7, 8 (Callout R)
  - Data Byte Order(1-8) / Set

Buttons for Cancel and OK are at the bottom.

### (A) Default Transmit Status

Select the radial button to select the default transmit status. The Communication Bridge Tx Status will start with this status on startup. The Tx status may be changed at run time in the rungs.

### (B) Default Transmit Report Rate

Enter in the text box the default transmit report rate. The range is 0 to 65535. The comm. bridge tx report rate will start with this value on startup. The Tx report rate may be changed at run time in the rungs.

**(C) Display Format**

This changes how the id, data bytes and masks are displayed. Note: Transmit Rate is only displayed in decimal.

**(D) Rx Module**

This defines the module the message will be received on. The list is only populated with modules that have generic CAN lines.

**(E) Rx Module CAN Line**

This defines the module's CAN line the message will be received on. The list is only populated with generic CAN lines define on the selected module.

**(F) Rx ID Length**

This defines the type of ID the received message should be.

**(G) Rx Byte**

This defines the data in the ID the received message should match.

**(H) Rx Mask**

This defines which bits of the incoming message need to match the entered Rx Byte. Mask bits that are high (1) mean it will check the bit and bits that are low (0) will be ignored.

**(I) Data Byte Filtering check box**

Clicking this check box enables the data byte filtering. If it is not checked the module will ignore all the data bytes and if it is checked then it will compare all the data bytes.

**(J) Data Bytes**

This defines the data in the data bytes the received message should match.

Note: If the data byte filtering check box is not checked this section is grayed out and ignored by the firmware.

**(K) Data Bytes Mask**

This defines which bits of the incoming message need to match the entered Rx Byte. Mask bits that are high (1) mean it will check the bit and bits that are low (0) will be ignored.

Note: If the data byte filtering check box is not checked this section is grayed out and ignored by the firmware.

**(L) Tx Module**

This defines the module the message will be transmitted on. The list is only populated with modules that have generic CAN lines.

**(M) Tx Module CAN Line**

This defines the module's CAN line the message will be transmitted on. The list is only populated with generic CAN lines define on the selected module.

**(N) Tx ID Value Adjustment check box**

Determines what will be sent as the identifier in the transmitted message. If this box is checked, the transmitted message will use the identifier entered in the ID values. If the box is not checked, it will use whatever the identifier was on the received message.

**(O) Tx ID Length**

This defines the type of ID the transmitted message will have. Note this section is grayed out and ignored by the firmware if the ID value adjustment check box is not checked.

**(P) ID Value**

This defines the identifier value the transmitted message will have. Note This section is grayed out and ignored by the firmware if the ID value adjustment check box is not checked.

**(Q) Tx Data Length Adjustment check box**

This check box determines what will be sent for the message length in the transmitted message. If the box is checked, a text box is enabled to allow the transmitted message's data length to be entered. If the box is not checked it will use whatever the data length was on the received message.

**(R) Tx Data Bytes**

This section sets up what the data bytes will be filled with. Each data byte has its own pull down box with 1-8 and S. Selecting 1-8 will fill the tx message data byte with the corresponding received message's data byte. Selecting S enables the text box above it and it can be filled with a number 0-255. The transmitted message will use the set value in this case. The default setting is setup with each byte getting the respective rx byte so the data bytes will be unchanged.

Preliminary