HED®
Hydro Electronic Devices, Inc.

composer
CANLINK

CANLink® Composer™ USER'S MANUAL

V03

Information contained within this document is accurate at the time of publication. HED®, Inc. reserves the right to change specifications without notice.

IMPORTANT NOTICE - PLEASE READ FIRST

THIS SOFTWARE IS USED AT YOUR OWN RISK. IMPROPER USE OF THIS SOFTWARE MAY RESULT IN IMPROPER FUNCTIONING OF A MODULE. THE USER IS SOLELY RESPONSIBLE FOR THE RESULTS OF ANY USE OF THE SOFTWARE. HED MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THE SOFTWARE, ITS USE, OR ANY RESULT THAT MAY BE OBTAINED THROUGH ITS USE. USE OF THE SOFTWARE REQUIRES PROFESSIONAL JUDGMENT AND IT IS SOLELY THE USER'S RESPONSIBILITY TO ASSESS THE APPROPRIATENESS OF ANY APPLICATION OF THE SOFTWARE. HED WILL NOT BE LIABLE TO THE USER OR ANY THIRD PARTY FOR ANY DAMAGES OF ANY KIND, DIRECT, CONSEQUENTIAL OR OTHERWISE, REGARDLESS OF THE LEGAL THEORY, ARISING FROM OR ASSOCIATED WITH THE SOFTWARE OR ITS USE.

DISCLAIMER OF WARRANTY. The Software is provided on an "AS IS" basis, without warranty of any kind, including, without limitation, the warranties of merchantability, fitness for a particular purpose and non infringement. The entire risk as to the quality and performance of the Software is borne by You. Should the Software prove defective, You, not Vendor or its licensors, assume the entire cost of any service and repair. If the Software is intended to link to, extract content from or otherwise integrate with a third party service, Vendor makes no representation or warranty that Your particular use of the Software is or will continue to be authorized by law in Your jurisdiction or that the third party service will continue to be available to You. This disclaimer of warranty constitutes an essential part of the agreement. VENDOR HEREBY EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL VENDOR OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PERSON FOR (AND VENDOR HEREBY EXPRESSLY DISCLAIMS ANY AND ALL LIABILITY FOR) ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR WORK STOPPAGE, COMPUTER FAILURE OR LOSS OF REVENUES, PROFITS, GOODWILL, USE, DATA OR OTHER INTANGIBLE OR ECONOMIC LOSSES. IN NO EVENT WILL VENDOR OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES IN EXCESS OF THE AMOUNT PAID TO LICENSE THE SOFTWARE, EVEN IF YOU OR ANY OTHER PARTY SHALL HAVE INFORMED VENDOR OR ITS LICENSORS OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM. NO CLAIM, REGARDLESS OF FORM, MAY BE MADE OR ACTION BROUGHT BY YOU MORE THAN ONE (1) YEAR AFTER THE BASIS FOR THE CLAIM BECOMES KNOWN TO THE PARTY ASSERTING IT.

INDEMNIFICATION. You and Your agents will indemnify and hold harmless Vendor and its officers, directors, shareholders, employees, agents, successors, assigns and affiliates from and against any award of costs or damages brought against Vendor to the extent that such award is (i) based on a claim regarding the use, installation or configuration of Software by You or Your agents; or (ii) based on a claim regarding modification, translation, customization or localization to the Software by You or Your agents. Vendor shall indemnify and hold You harmless against liability to third parties (only for liability solely the fault of Vendor) to the extent arising from the violation of any third party's copyright rights in connection with the use by You of the Software (a) as delivered by Vendor to You and (b) in accordance with this License Agreement, provided that (i) Vendor shall have the right to conduct any defense and/or settlement in any such third party action arising as described herein, (ii) You shall fully cooperate with such defense, and (iii) Vendor receives prompt written notice from You. In no event shall You settle any such claim, lawsuit, or proceeding without Vendor's prior approval, and Vendor shall have no liability for any such unapproved settlement so made. This indemnification is limited to the Software delivered to You or as modified by Vendor and does not cover third party claims arising from modifications to or uses of the Software in a manner not authorized by Vendor.

FULL TEXT OF THE CANLink® Composer™ LICENSE IS INCLUDED WITH THE SOFTWARE

Table of Contents

— General Information —

— Quick Start —

— Modules and Data Items —

— Function Blocks —

— Operator Blocks —

— Logic Blocks —

— Compiling Database —

— Download Database —

— Printing —

— Supplemental Information —

— Licensing —

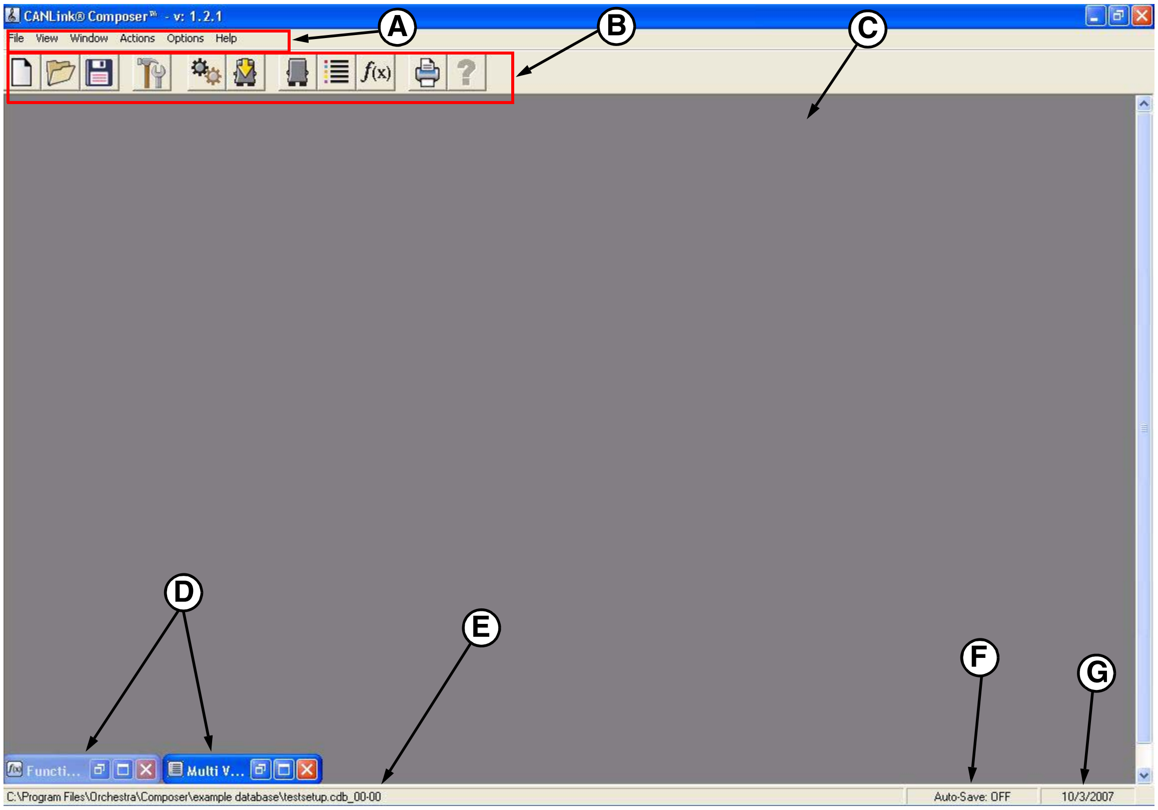
— Company Information —

General Information

SOFTWARE INSTALLATION

Refer to the CANLink® Orchestra™ Software Installation Manual for system requirements and for procedures on how to install the CANLink Composer software.

MAIN WINDOW



C:\Program Files\Orchestra\Composer\example database\testsetup.cdb_00-00

Auto-Save: OFF

10/3/2007

Figure 1

A. Program Menu
B. Tool Bar Buttons
C. Main Window
D. Minimized Windows
E. Opened Database Location
F. Auto-Save Status
G. Current Date

Menus

File Menu

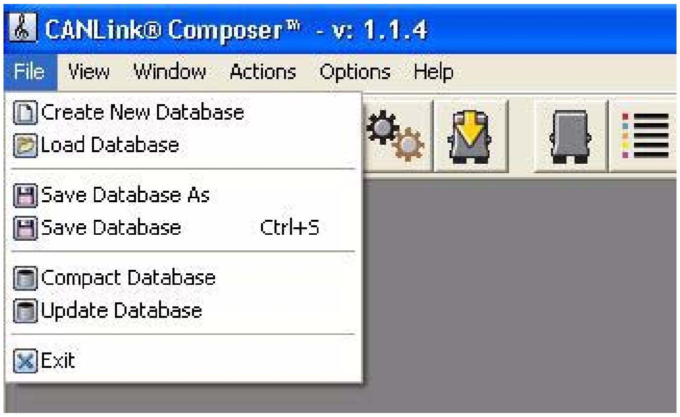


Figure 2

Use this function to create new, load, save, compact and update a database, or exit Composer.

View Menu

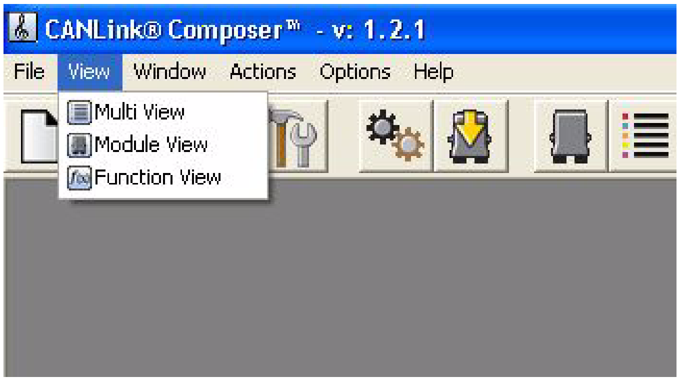


Figure 3

Use this function to open **Multi View**, **Module View** or **Function View** windows.

Window Menu

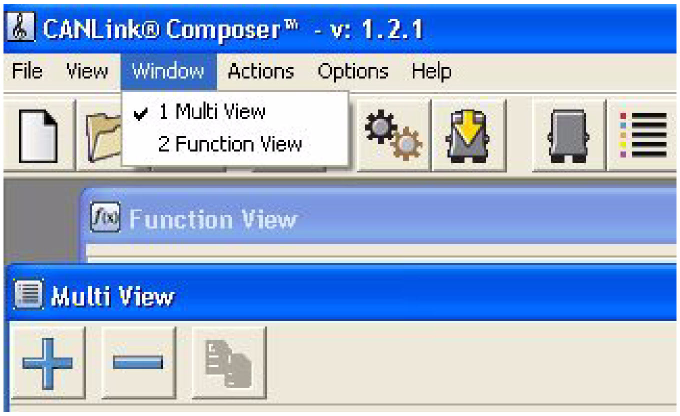


Figure 4

Use this function to pull the currently opened windows to the foreground.

Actions Menu

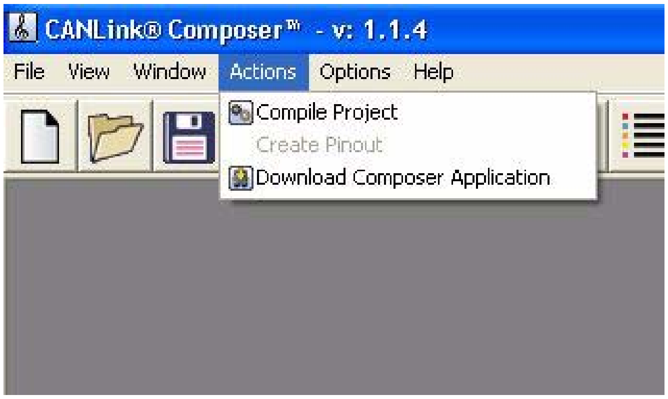


Figure 5

Use this function to compile the project database, create the project pinout, or download the Composer application to the Master Module.

Options Menu

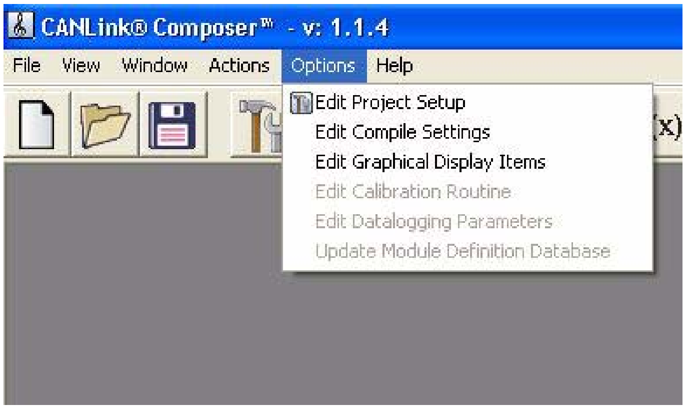

CANLink® Composer™ - v: 1.1.4
File
View
Window
Actions
Options
Help
Edit Project Setup
Edit Compile Settings
Edit Graphical Display Items
Edit Calibration Routine
Edit Datalogging Parameters
Update Module Definition Database
(x)

Figure 6

Use this function to edit the project setup (edit application version, add a project description, turn Auto-Save on or off, etc.), edit the compile settings, edit graphical display items, edit calibration routine, edit datalogging parameters, or update the module definition database.

Help Menu

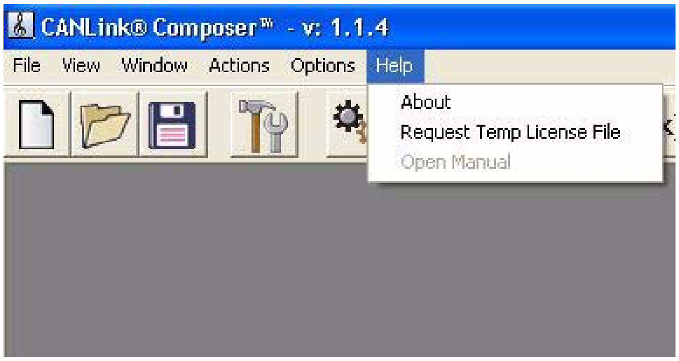


Figure 7

Use this feature to find out about Composer, request a temporary license file, or open a user's manual.

Tool Bar Buttons

Create New Database

This will create a new database. If there is a database currently open, you will be prompted to save the current file. *Yes* will save and close the open database. *No* will close the open database and proceed to creating a new database.

Open Existing Database

This will open an existing database. If there is a database currently open, you will be prompted to save the current database, then asked if you are sure you want to open another database. *Yes* will open a window to choose the new database. *No* will keep the current database open.

Save Database

(Key Command Shortcut - Ctrl+s)
This will save the changes to the current database.

Project Setup

This will open the **Project Setup** window. Here you can edit the version, project description, turn on or off the “Safe Mode Below Voltage” option and define the voltage, choose single or dual CAN system, or turn Auto-Save on or off.

Compile Database

This will compile the system and create the necessary files for downloading.

Download Composer Application

This will download the Composer application to the Master Module.

Open Module View

This will display all the modules currently added to the project. From here you can add, delete, and edit modules.

Open Multi View

This will display all the *Inputs*, *Outputs*, and virtual *Inputs* and *Outputs*. From here you can open the *Input* or *Output* **Assignment** window and **Properties** window.

Open Function View

This will display the *Rungs* in the *Function Blocks* and allow you to create and edit the *Function Blocks*.

Print

This will create a text file that can be opened and printed using a text editor such as Notepad or Microsoft® Word.

Help

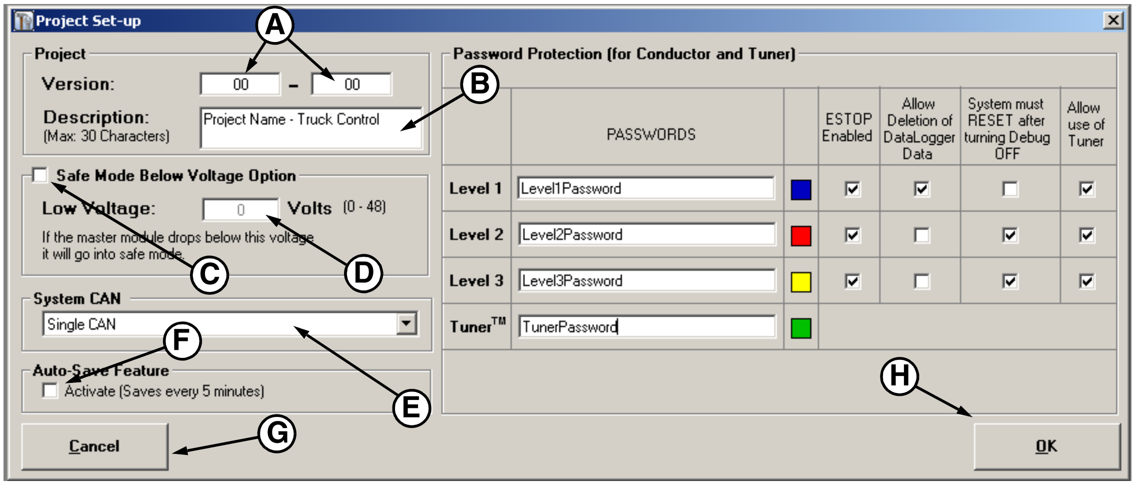
This opens a dialog box that asks if you wish to open this user’s manual.

PROJECT SETUP WINDOW

Overview

From this window you can edit the version, project description, turn on or off the “Safe Mode Below Voltage” option and define the voltage, choose single or dual CAN system, or turn Auto-Save on or off.

This window is opened by clicking the *Project Setup* button in the **Main** window, or by selecting *Options>Edit Project Setup* from the **File** menu.



Project

Version: 00 - 00

Description: Project Name - Truck Control
(Max: 30 Characters)

□ Safe Mode Below Voltage Option

Low Voltage: 0 Volts (0 - 48)

If the master module drops below this voltage it will go into safe mode.

System CAN

Single CAN

Auto-Save Feature

□ Activate (Saves every 5 minutes)

Password Protection (for Conductor and Tuner)

	PASSWORDS		ESTOP Enabled	Allow Deletion of DataLogger Data	System must RESET after turning Debug OFF	Allow use of Tuner
Level 1	Level1Password	🟦	☑	☑	□	☑
Level 2	Level2Password	🟥	☑	□	☑	☑
Level 3	Level3Password	🟨	☑	□	☑	☑
Tuner™	TunerPassword	🟩				

Figure 8

Controls

(A) Application Version Text Boxes

Enter the revision data here. This information will be appended to the filename extension of all files associated with database in the form of *Filename + .Extension + Application Version.* Example: *filename.cdb_00-00.*

(B) Project Description Text Box

Enter a short description of the database. Maximum 30 characters.

(C) Safe Mode Below Voltage Option Check Box

Check this to force the Master Module to go into Safe Mode if the voltage drops below the threshold entered in (D).

(D) Low Voltage Text Box

Enter the threshold voltage at which you want the Master Module to enter Safe Mode.

(E) System CAN Dropdown Menu

Select single or dual CAN system.

Single CAN means there is only one CAN line connected to each of the CANLink modules (Master Module may still have more than one in a single CAN system).

Dual CAN means there are at least two CAN lines connected to the Master Module and allows it to use a redundant CAN system setup. This allows the Master Module to switch to CAN line 2 if CAN line 1 fails (both CAN lines should be run to other CANLink modules for redundancy).

(F) Auto-Save Check Box

When this box is checked, Composer will automatically save the project every five (5) minutes.

(G) Cancel Button

This will close the **Project Setup** window without making changes.

(H) OK Button

The **Project Setup** window will close and apply the changes you have made.

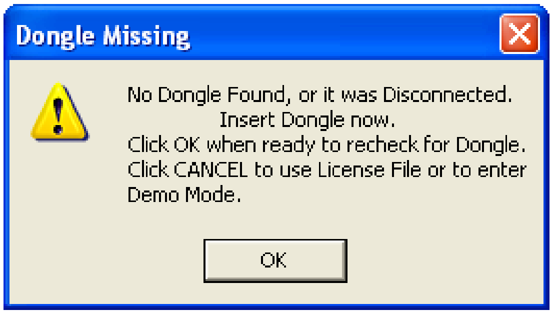
Quick Start

GENERAL INFORMATION

This guide will outline the steps needed to add one *Input*, one *Output* and two modules, and also how to create a *Rung* where the *Input* controls the *Output*. This assumes that the Master Module has been programmed with the appropriate firmware and that all connections to the module have been made.

GETTING STARTED

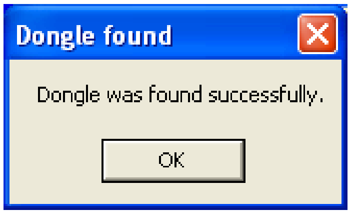
Note: If the dongle is removed while Composer is running, Composer will close automatically after one minute but will allow you to save your work before it does.



No Dongle Found, or it was Disconnected.
Insert Dongle now.
Click OK when ready to recheck for Dongle.
Click CANCEL to use License File or to enter Demo Mode.

Figure 1

1. Connect the dongle supplied with your program into an available USB port. If you attempt to start Composer without the dongle connected, an *Error* message will be displayed **(Figure 1)**. When the dongle is connected, a “Dongle Found” message will appear **(Figure 2)**. Click the *OK* button and proceed as directed by the software.



Dongle was found successfully.

Figure 2

2. Open Composer.

Click *Start>Programs>Orchestra>Composer*.

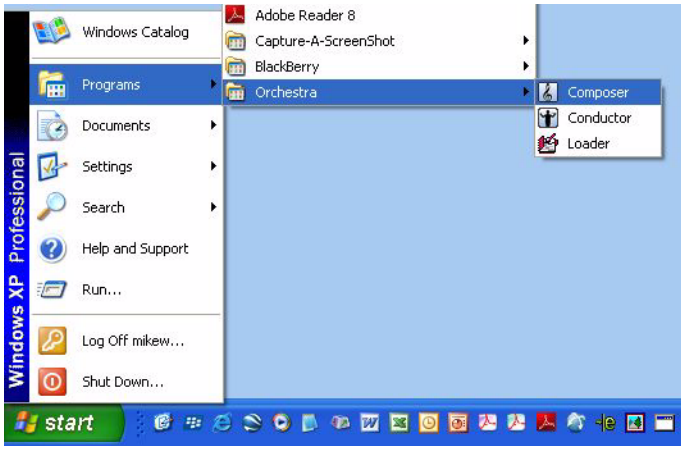
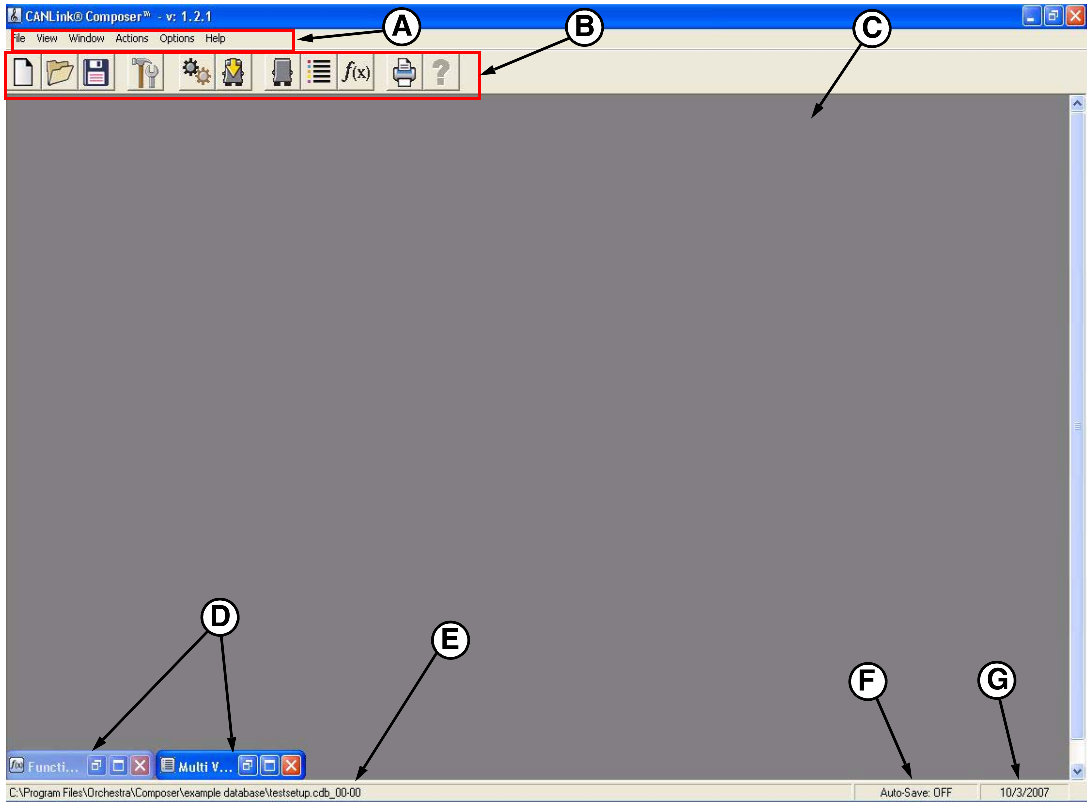


Figure 3

The **Main** window will open.


CANLink® Composer™ - v: 1.2.1
File View Window Actions Options Help
A
B
C
D
E
F
G
Functi...
Multi V...

C:\Program Files\Orchestra\Composer\example database\testsetup.cdb_00-00 Auto-Save: OFF 10/3/2007

Figure 4

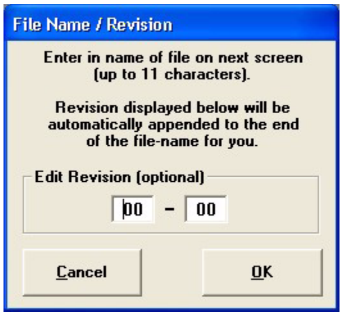
A. **Program Menu**
B. **Tool Bar Buttons**
C. **Main Window**
D. **Minimized Windows**
E. **Open Database Location**
F. **Auto-Save Status**
G. **Date**

CREATE A DATABASE

Start Here

1. Click the *Create New Database* button on the *Main Window* tool bar.

The **File Name/Revision** window will open **(Figure 5)**. Enter a revision number if desired. Click the *OK* button to continue.



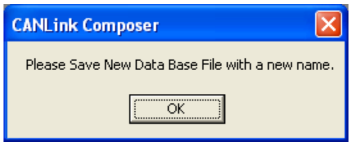
Enter in name of file on next screen (up to 11 characters).

Revision displayed below will be automatically appended to the end of the file-name for you.

Edit Revision (optional)
00 – 00

Figure 5

2. Click the *OK* button to save the database **(Figure 6)**.

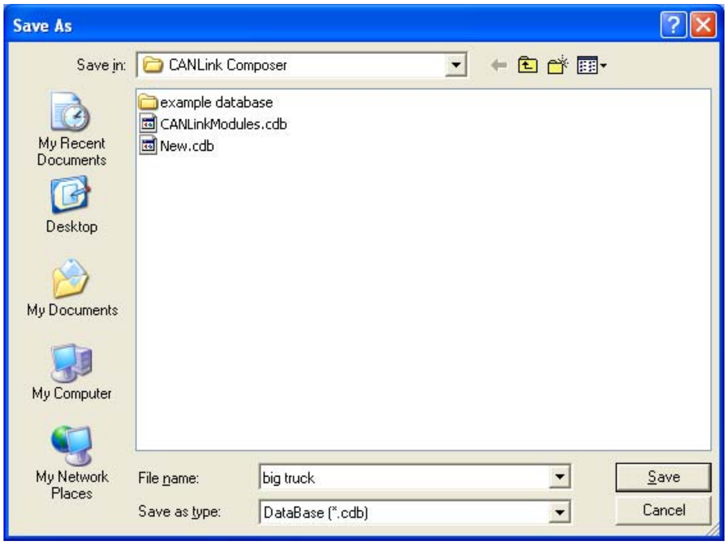


Please Save New Data Base File with a new name.

Figure 6

*Note: Do not save the database in the Composer install directory. It is recommended each database project be saved in a separate folder. New folders can be created from within the **Save As** window.*

3. Enter a name (11 characters maximum), choose a location (create a new folder if necessary), and click the *Save* button **(Figure 7)**.



Save in: CANLink Composer

example database
CANLinkModules.cdb
New.cdb

File name: big truck
Save as type: DataBase (*.cdb)

Figure 7

Add an Input

4. The **Multi View** window opens automatically **(Figure 8)**.
5. Click the + *ADD* button and select the type of *Data Item* desired from the dropdown menu. In this case choose “Input” as the *Data Item* type.

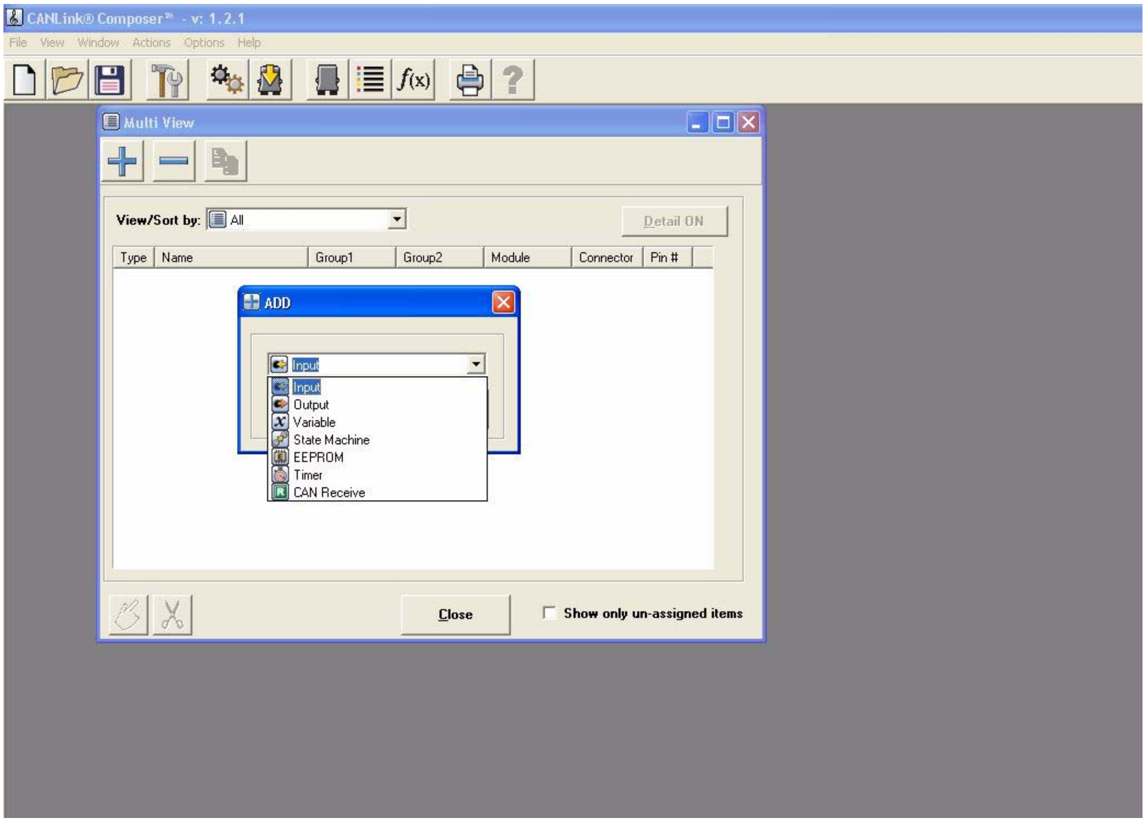
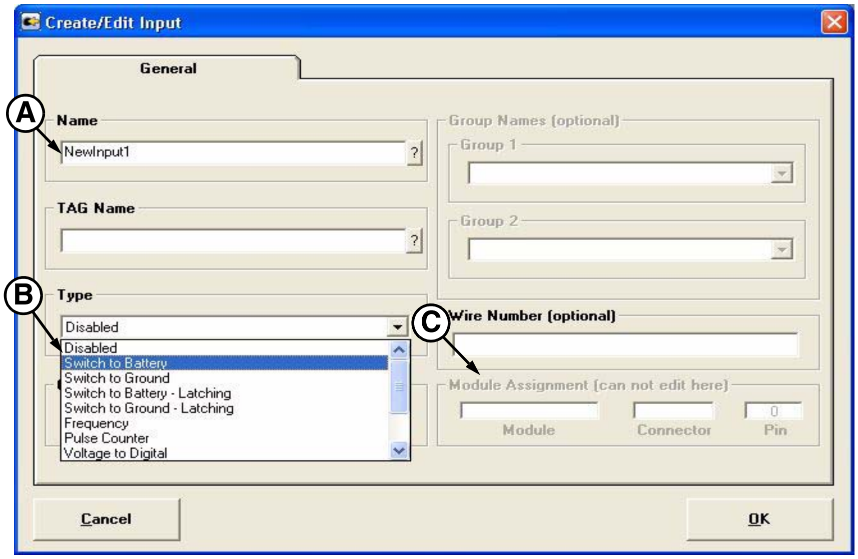

CANLink® Composer™ - v: 1.2.1
File View Window Actions Options Help
Multi View
View/Sort by: All
Detail ON
Type Name Group1 Group2 Module Connector Pin #
ADD
Input
Input
Output
Variable
State Machine
EEPROM
Timer
CAN Receive
Close
Show only un-assigned items

Figure 8

6. The **Create/Edit Input** window will open. Enter data in the appropriate boxes. In this case enter a *Name* of “NewInput1” **(Figure 9, A)** and select “Switch to Battery” in the *Type* dropdown menu **(Figure 9, B)**. The *TAG Name* will generate automatically after clicking *OK* if nothing is entered in the text box. The *Module Assignment* box **(Figure 9, C)** is read-only data and will contain data after the *Input* has been assigned to a *Module*. *See Assign Inputs and Outputs on page 15*.



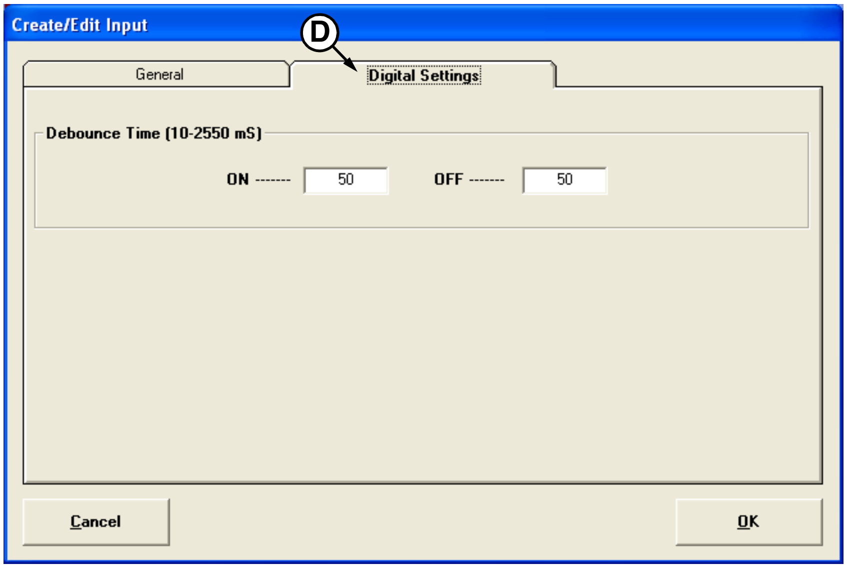
Name: NewInput1

Type: Disabled

Pin: 0

Figure 9

Depending on the *Input Type* chosen, one or more tabs **(Figure 10, D)** will appear which may require data *Input*. Default values will be inserted automatically and can be changed if needed.



Debounce Time (10-2550 mS)

ON ------- 50

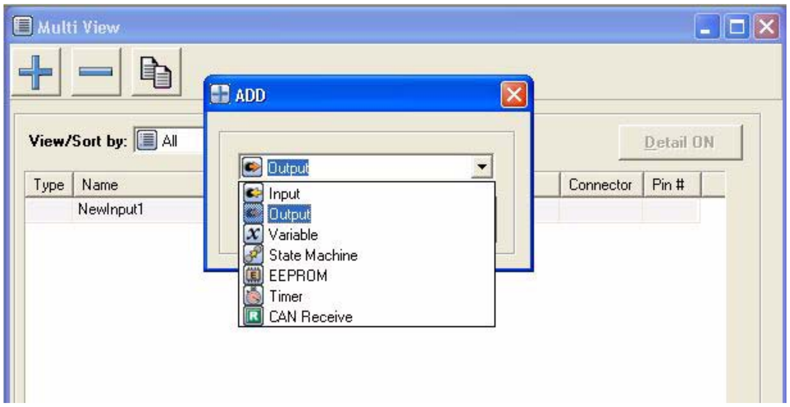
OFF ------- 50

Figure 10

Click the *OK* button when all data is entered. If a duplicate name has been entered, an *Error* message will appear requiring that a different name be used.

Add an Output

7. Click the + *ADD* button in the **Multi View** window and select “Output” to create an *Output Data Item* type.



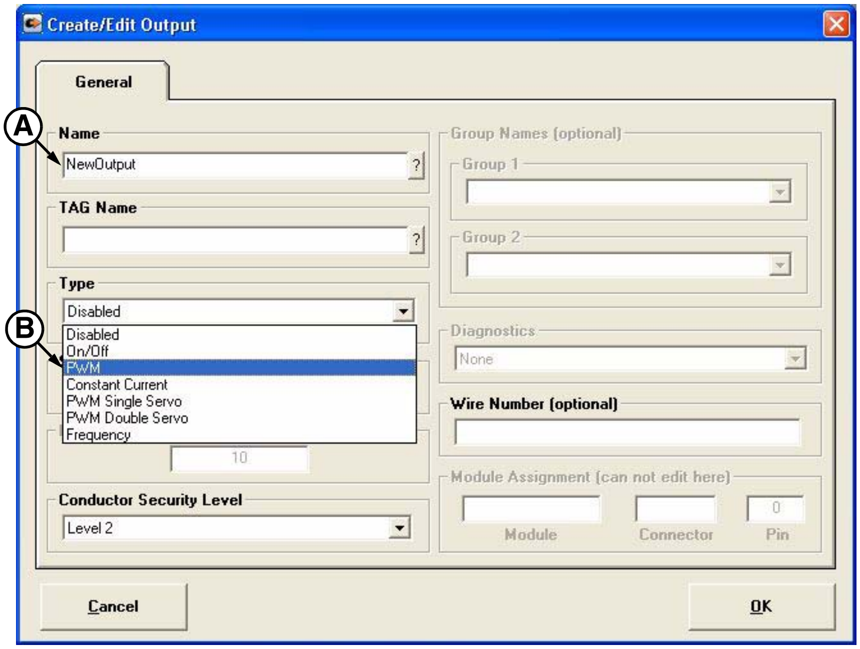
Multi View

View/Sort by: All

ADD: Output

Figure 11

8. Enter a *Name* of 20 characters or less **(Figure 12, A)** and choose “PWM” (pulse width modulated) as the *Type* **(Figure 12, B)**. The *TAG Name* will generate automatically after clicking *OK* if nothing is entered in the text box.



Create/Edit Output

Name: NewOutput

TAG Name:

Type: Disabled

Conductor Security Level: Level 2

Diagnostics: None

Wire Number (optional):

Module Assignment (can not edit here): Module, Connector, Pin 0

Figure 12

Add Modules

9. Click the *Open Module View* button in the **Main** window to open **Module View** window.

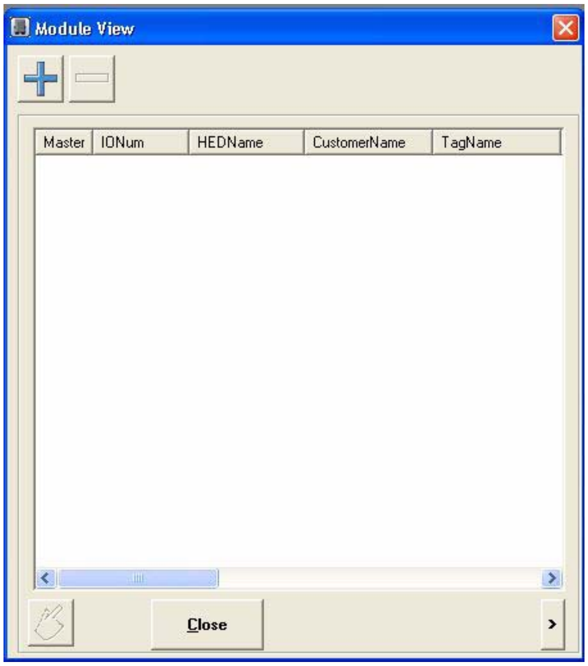
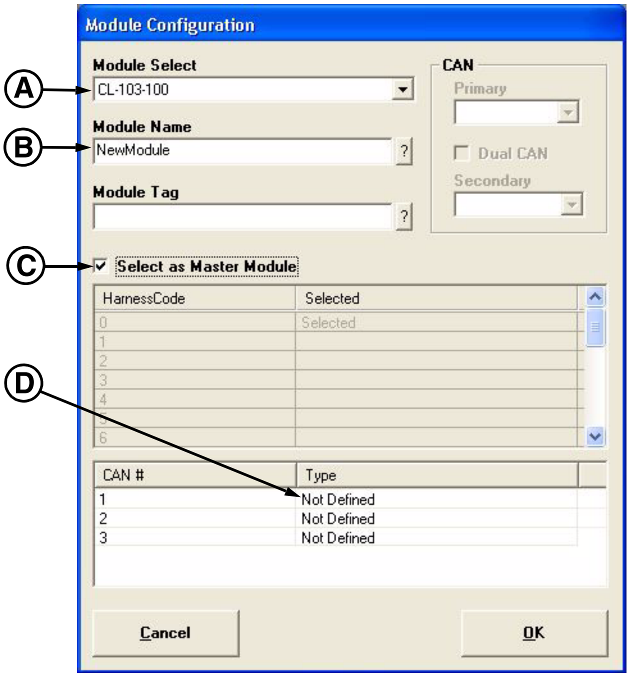


Figure 13

10. Click the *ADD* button in **Module View**. The **Module Configuration** window will open **(Figure 14)**.
11. Choose “CL-103-100” in the *Module Select* dropdown menu **(Figure 14, A)**.

12. Enter a *Name* of 20 characters or less **(Figure 14, B)** and check the *Select as Master Module* check box **(Figure 14, C)**. The *Module Tag* name will generate automatically after clicking *OK* if nothing is entered in the text box.
13. At least one *CAN # Type* must be set to "Module." Change each *CAN # Type* to "Module" **(Figure 14, D)** by double-clicking on the line item you want to change. The *CAN # Type* will cycle through several criteria.



Module Configuration

Module Select: CL-103-100

Module Name: NewModule

Module Tag:

CAN
Primary
□ Dual CAN
Secondary

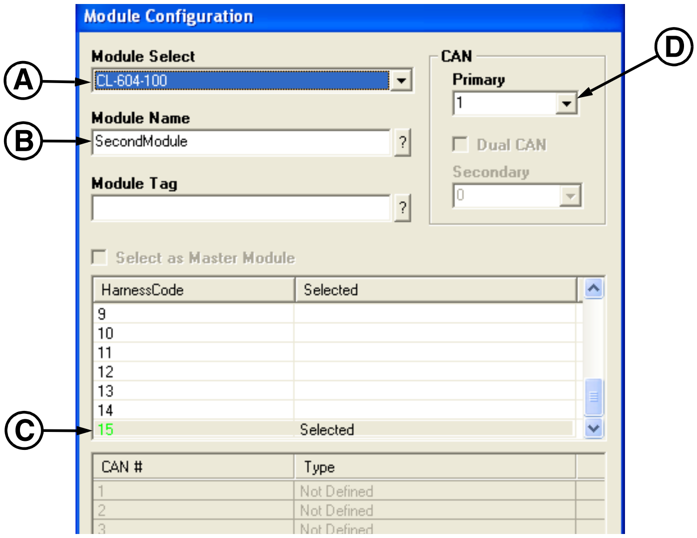
☑ Select as Master Module

HarnessCode	Selected
0	Selected
1	
2	
3	
4	
5	
6	

CAN #	Type
1	Not Defined
2	Not Defined
3	Not Defined

Figure 14

14. Click *OK*. You will be returned to the **Module View** window.
15. Click the + *ADD* button to add a second module. Choose "CL-604-100" in the *Module Select* dropdown menu **(Figure 15, A)**.
16. Enter a *Name* **(Figure 15, B)**. The *Module Tag* will be generated automatically.
17. Select *Harness Code* "15" by double-clicking the appropriate harness code **(Figure 15, C)** and set *CAN Primary* to "1" **(Figure 15, D)**.
18. Click *OK.*



Module Configuration

Module Select: CL-604-100

Module Name: SecondModule

Module Tag:

CAN
Primary: 1
□ Dual CAN
Secondary: 0

□ Select as Master Module

HarnessCode	Selected
9	
10	
11	
12	
13	
14	
15	Selected

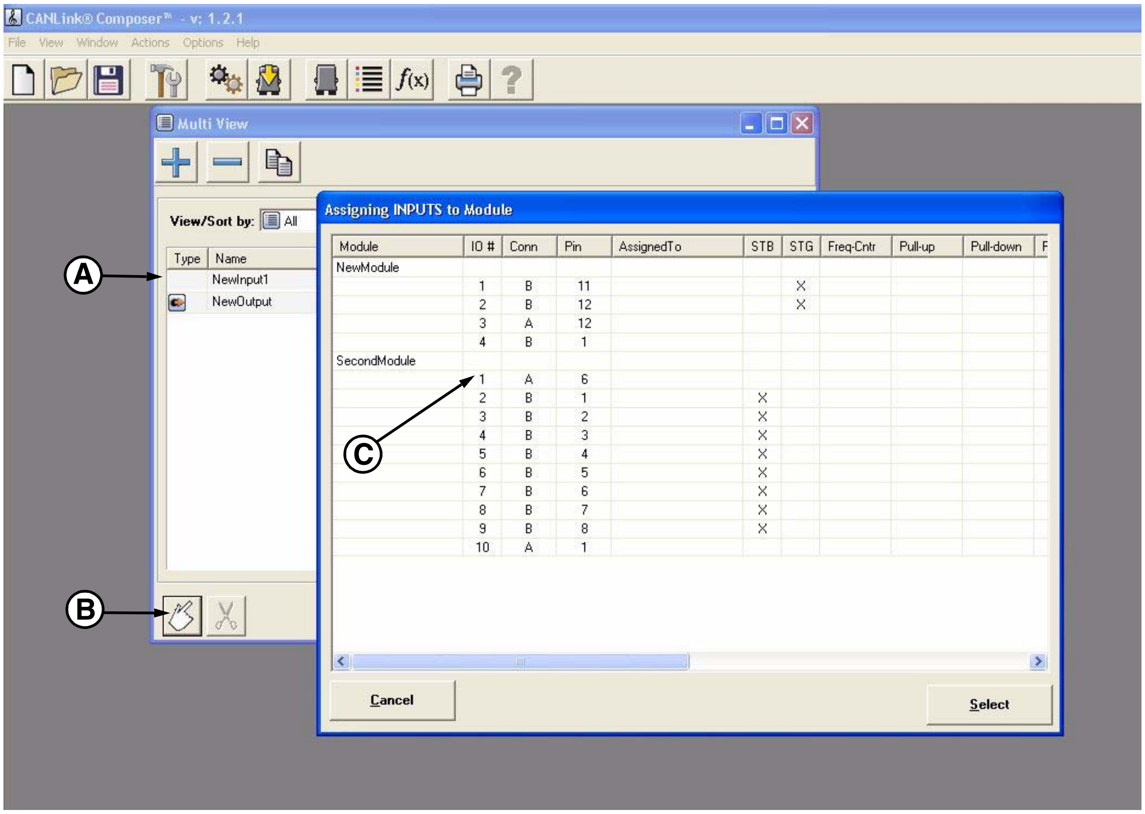
CAN #	Type
1	Not Defined
2	Not Defined
3	Not Defined

Figure 15

Assign Inputs and Outputs

19. Click the *Open Multi View* button in the **Main** window. The **Multi View** window will open.

20. Single-click the *Input* called "NewInput1" **(Figure 16, A)**. Do not double-click. If the **Create/Edit Input** window opens, click *Cancel* and continue.


CANLink® Composer™ - v: 1.2.1
File View Window Actions Options Help
f(x)
Multi View
View/Sort by: All
Assigning INPUTS to Module
A
B
C
Cancel
Select

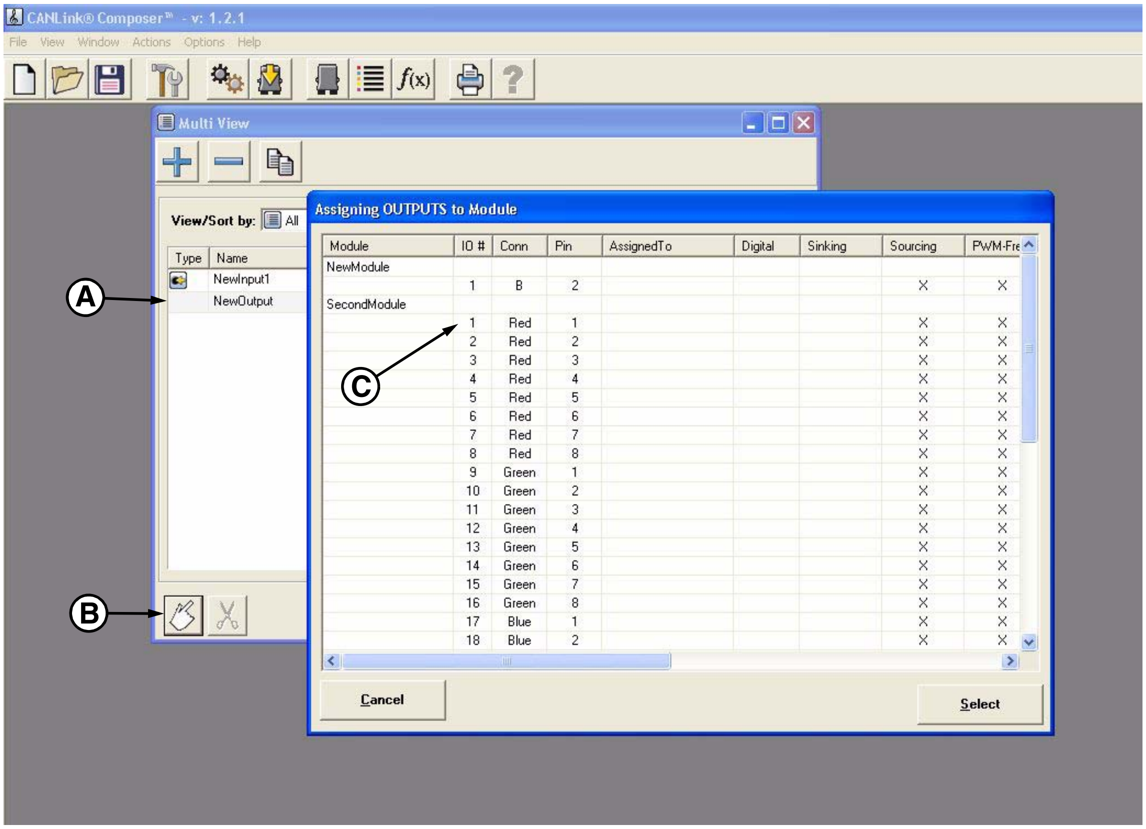
Type	Name
	NewInput1
	NewOutput

Module	IO #	Conn	Pin	AssignedTo	STB	STG	Freq-Cntr	Pull-up	Pull-down	F
NewModule										
	1	B	11			X				
	2	B	12			X				
	3	A	12							
	4	B	1							
SecondModule										
	1	A	6							
	2	B	1		X					
	3	B	2		X					
	4	B	3		X					
	5	B	4		X					
	6	B	5		X					
	7	B	6		X					
	8	B	7		X					
	9	B	8		X					
	10	A	1							

Figure 16

21. Click the *Assign* button **(Figure 16, B)** to open the **Assigning INPUTS to Module** window. Double-click *IO #* "1" under *SecondModule* **(Figure 16, C)** to assign the *Input* to the first *Input* on the *SecondModule* (CL-604 module). The **Assigning INPUTS to Module** window will automatically close.

22. Single-click the *Output* called "NewOutput" **(Figure 17, A)** and click the *Assign* button **(Figure 17, B)**.



CANLink® Composer™ - v: 1.2.1
File View Window Actions Options Help
Multi View
View/Sort by: All
Type
Name
NewInput1
NewOutput
Assigning OUTPUTS to Module
Cancel
Select
A
B
C

Module	IO #	Conn	Pin	AssignedTo	Digital	Sinking	Sourcing	PWM-Fre
NewModule								
	1	B	2				X	X
SecondModule								
	1	Red	1				X	X
	2	Red	2				X	X
	3	Red	3				X	X
	4	Red	4				X	X
	5	Red	5				X	X
	6	Red	6				X	X
	7	Red	7				X	X
	8	Red	8				X	X
	9	Green	1				X	X
	10	Green	2				X	X
	11	Green	3				X	X
	12	Green	4				X	X
	13	Green	5				X	X
	14	Green	6				X	X
	15	Green	7				X	X
	16	Green	8				X	X
	17	Blue	1				X	X
	18	Blue	2				X	X

Figure 17

23. Double-click *IO #* "1" under *SecondModule* **(Figure 17, C)** to assign the *Output* to the first *Output* on *SecondModule* (CL-604 module). The window will close automatically.

Add Rungs to the Function View

24. Click *f(x)* *Open Function View* button to open the **Function View** window. Click the *ADD* button **(Figure 18, A)** to open the **Rung Editor** window.

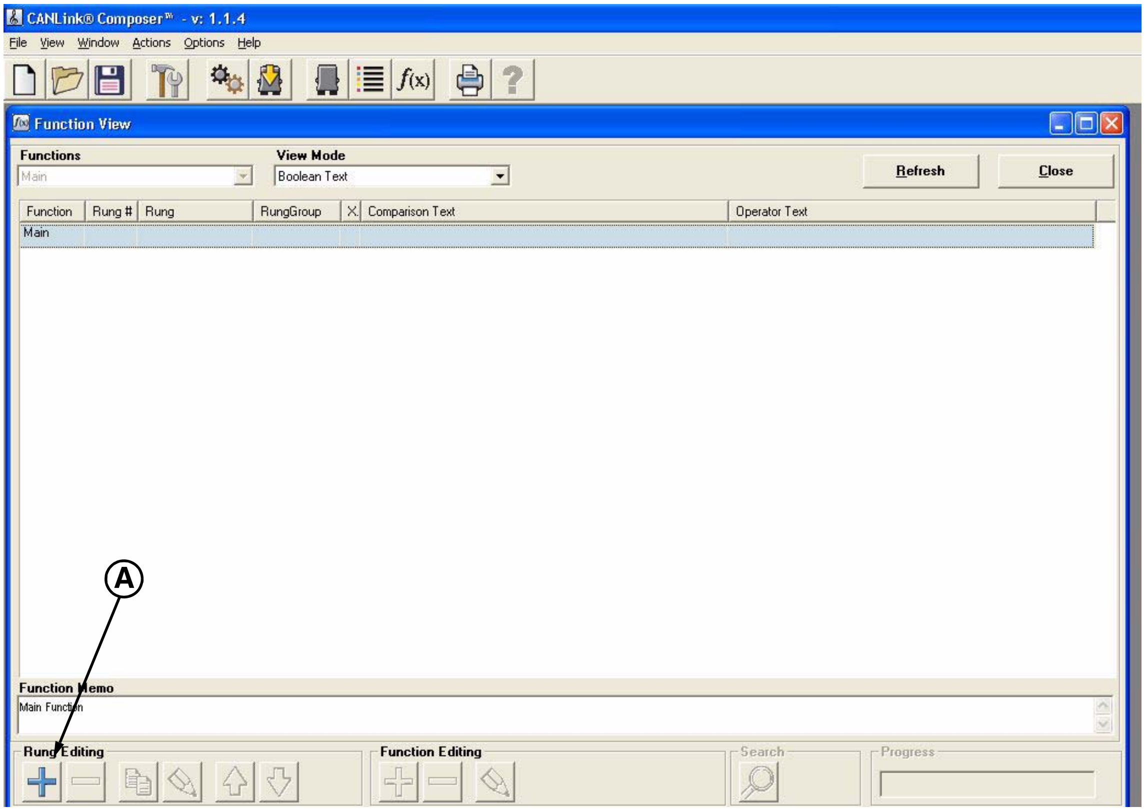


Figure 18

25. Single-click the line under "Block 1" **(Figure 19, A)**. The entire block will be highlighted gray.

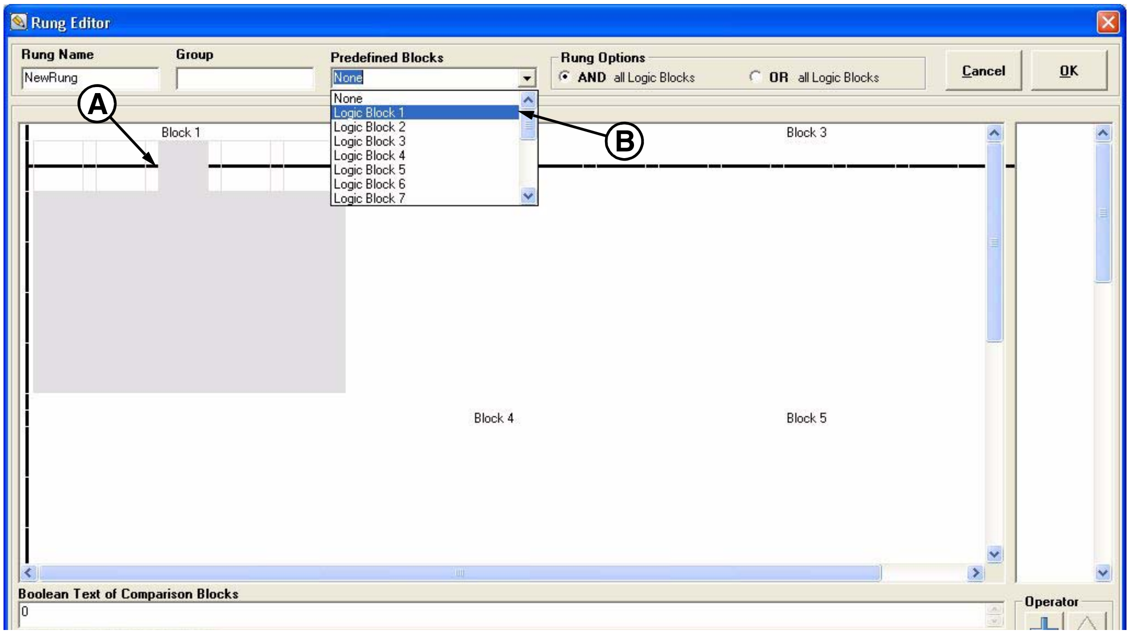
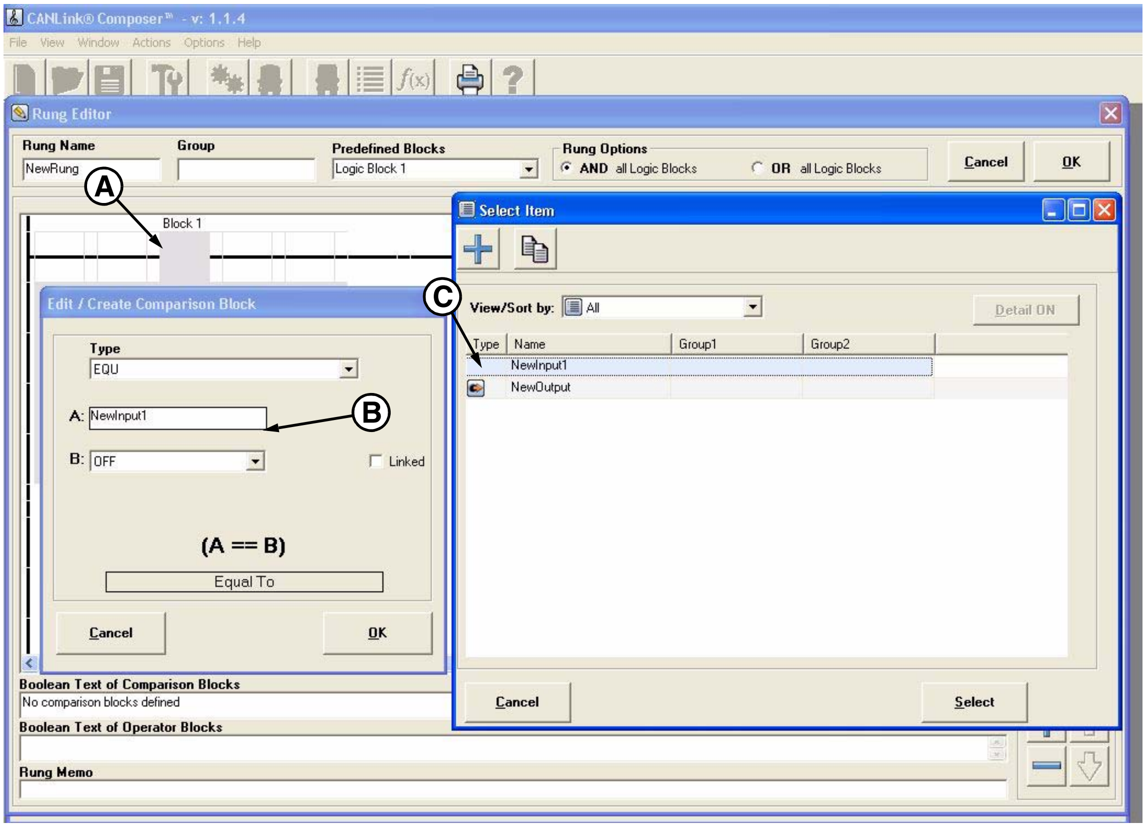


Figure 19

26. Select "Logic Block 1" in the *Predefined Blocks* dropdown menu **(Figure 19, B)**.

27. Double-click Block 1 **(Figure 20, A)** to open the **Edit/Create Comparison Block** window **(Figure 21)**.
28. Double-click text box *A:* to open the **Select Item** window **(Figure 20, B)**.
29. Double-click “NewInput1” **(Figure 20, C)** to select it.


CANLink® Composer™ - v: 1.1.4
File View Window Actions Options Help
Rung Editor
Cancel
OK
A
B
C
Block 1
Select Item
Detail ON
Cancel
Select
Boolean Text of Operator Blocks
Rung Memo

Rung Name: NewRung
Group:
Predefined Blocks: Logic Block 1
Rung Options: ● AND all Logic Blocks ○ OR all Logic Blocks

Edit / Create Comparison Block
Type: EQU
A: NewInput1
B: OFF
□ Linked
(A == B)
Equal To
Cancel OK

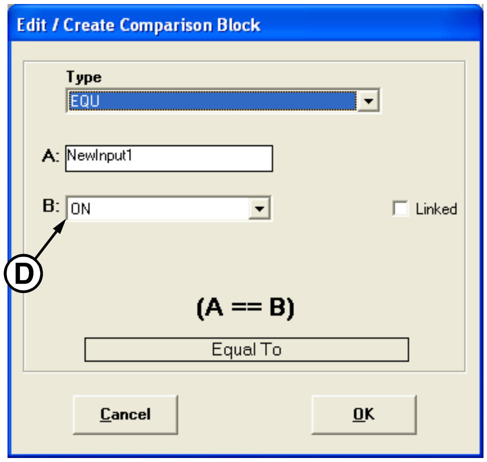
Boolean Text of Comparison Blocks: No comparison blocks defined

View/Sort by: All

Type	Name	Group1	Group2
	NewInput1		
	NewOutput		

Figure 20

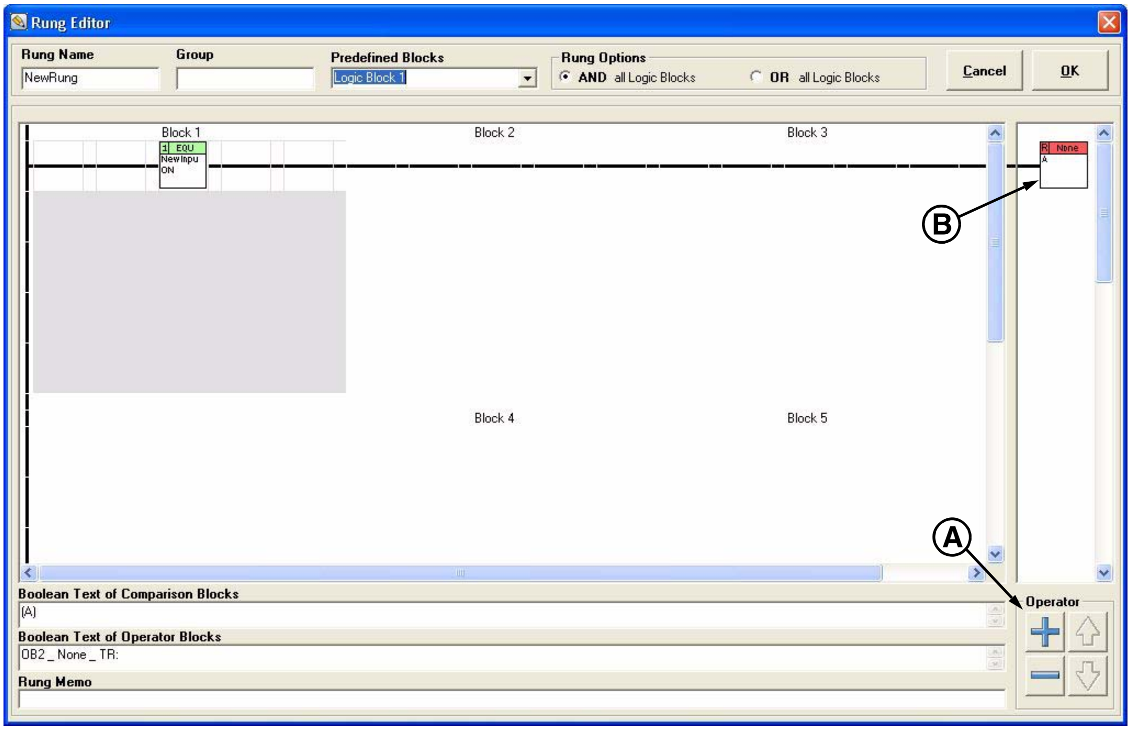
30. In the **Edit/Create Comparison Block** window, set *B:* to “ON” **(Figure 21, D)**. Leave *Type* set to “EQU.” Click *OK*.



Edit / Create Comparison Block
Type: EQU
A: NewInput1
B: ON
□ Linked
(A == B)
Equal To

Figure 21

31. Click the + *ADD* button in the *Operator* frame **(Figure 22, A)**.
32. Double-click the newly added block **(Figure 22, B)** to open the **Edit/Create Operator Block** window.



Rung Name: NewRung

Group:

Predefined Blocks: Logic Block 1

Rung Options: ● AND all Logic Blocks ○ OR all Logic Blocks

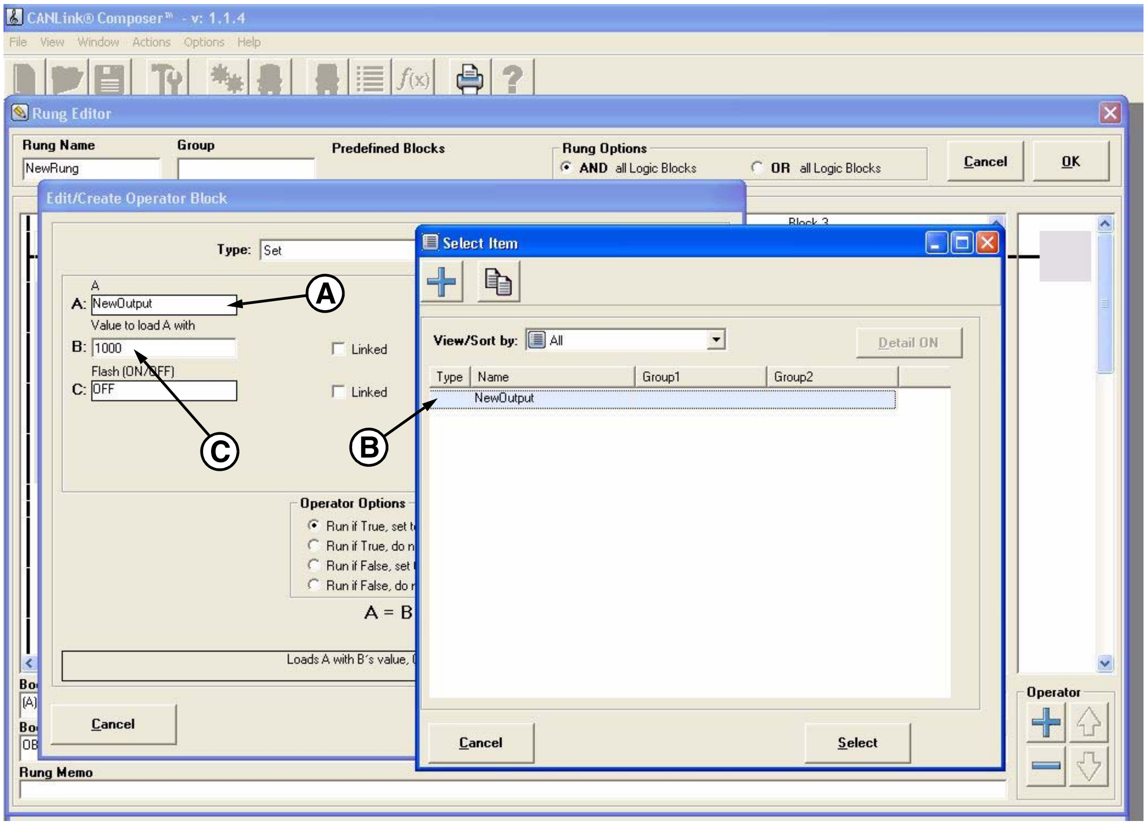
Boolean Text of Comparison Blocks: (A)

Boolean Text of Operator Blocks: OB2 _ None _ TR:

Rung Memo:

Figure 22

33. Double-click text box *A:* **(Figure 23, A)** to open the **Select Item** window. Select "NewOutput" **(Figure 23, B)** and click the *Select* button. "NewOutput" will be added to text box *A:* **(Figure 23, A)** and a third text box will appear named *Flash (ON/OFF).*
34. Enter "1000" in text box *B:* **(Figure 23, C)**. Entering "1000" will turn the PWM *Output* on, 100.0% duty cycle.
35. Click *OK*.



Rung Name: NewRung
Group:
Rung Options: ● AND all Logic Blocks ○ OR all Logic Blocks

Type: Set

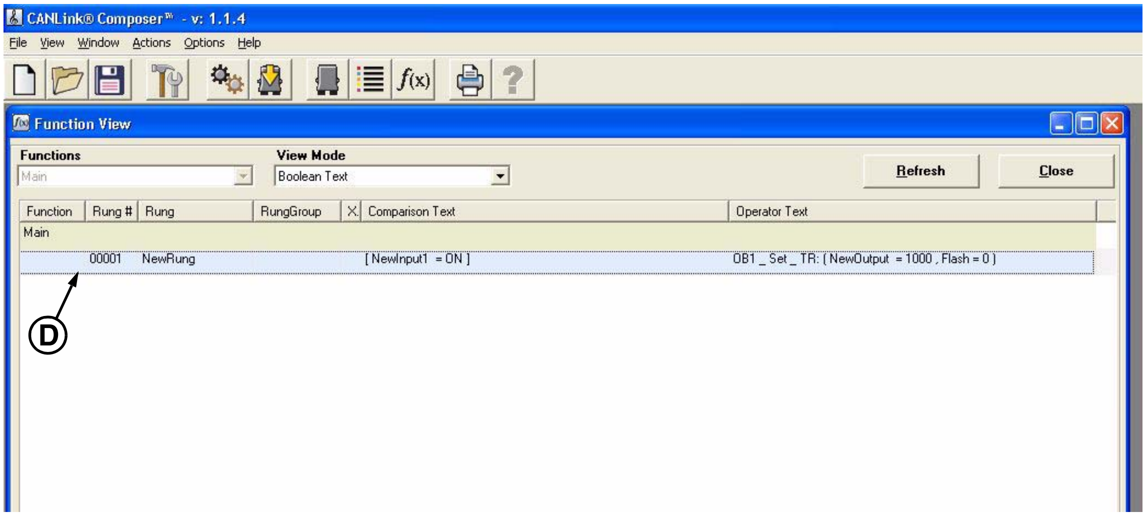
A: NewOutput
Value to load A with
B: 1000
Flash (ON/OFF)
C: OFF

View/Sort by: All

Type	Name	Group1	Group2
	NewOutput		

Figure 23

36. Click the *OK* button in the **Rung Editor** window. One *Rung* is now added. The newly added *Rung* **(Figure 24, D)** will appear in the **Function View** window.
37. Click the *Close* button in the **Function View** window.


CANLink® Composer™ - v: 1.1.4
File View Window Actions Options Help
Function View
Refresh
Close
D

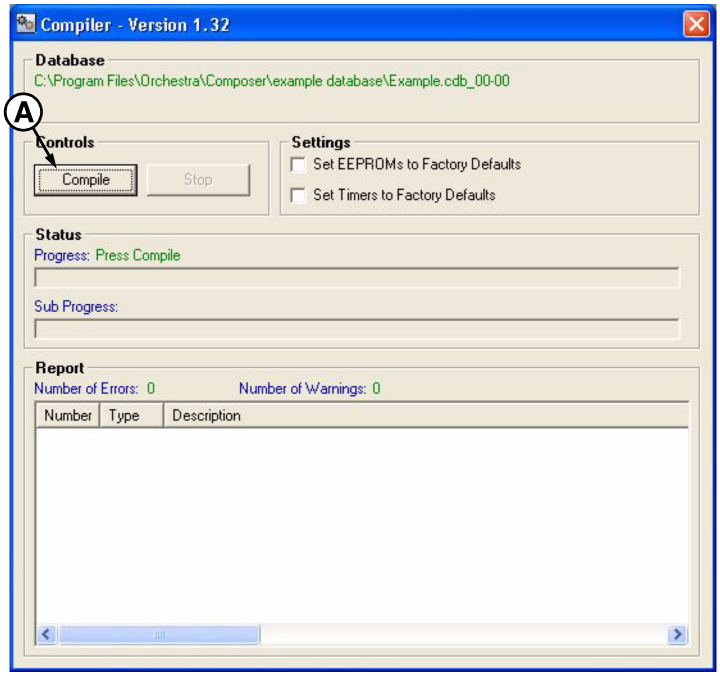
Functions: Main
View Mode: Boolean Text

Function	Rung #	Rung	RungGroup	X	Comparison Text	Operator Text
Main						
	00001	NewRung			[NewInput1 = ON]	OB1 _ Set _ TR: (NewOutput = 1000 , Flash = 0)

Figure 24

COMPILE DATABASE

38. Click the *Compile Database* button in the **Main** window to compile the database. You will be prompted to save your database.
    - *Yes* button will save changes and proceed.
    - *No* button will not save changes and proceed.
39. You will be prompted to overwrite the existing Compiler files.
    - *Yes* button will open the Compiler window.
    - *No* button will cause the compile to stop and return you to the **Main** window.
40. Click the *Compile* button **(Figure 25, A)** to begin compiling the database.



Database
C:\Program Files\Orchestra\Composer\example database\Example.cdb_00-00

Settings
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

Status
Progress: Press Compile
Sub Progress:

Report
Number of Errors: 0 Number of Warnings: 0

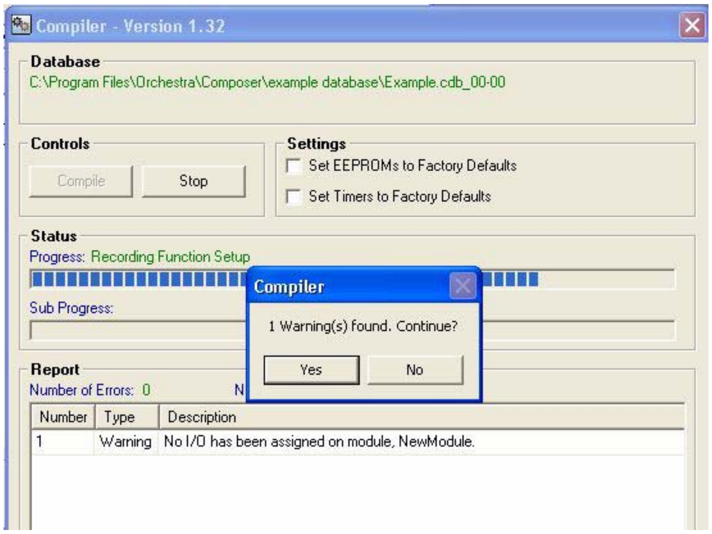
Number	Type	Description

Figure 25

41. Address problems presented in the *Report* box of the Compiler window as follows:

- **(Figure 26)** *Warnings* are non-fatal and will not cause the compile to fail. Correction is optional.
- **(Figure 27)** *Errors* are fatal and must be corrected before the database will compile successfully. Exit Compiler. Make any corrections and compile again.

Errors and *Warnings* are logged to a file. The file is located in the same path and same name as the database with the extension .err. This file can be used as an aid while correcting *Errors*.



Compiler - Version 1.32

Database
C:\Program Files\Orchestra\Composer\example database\Example.cdb_00-00

Settings
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

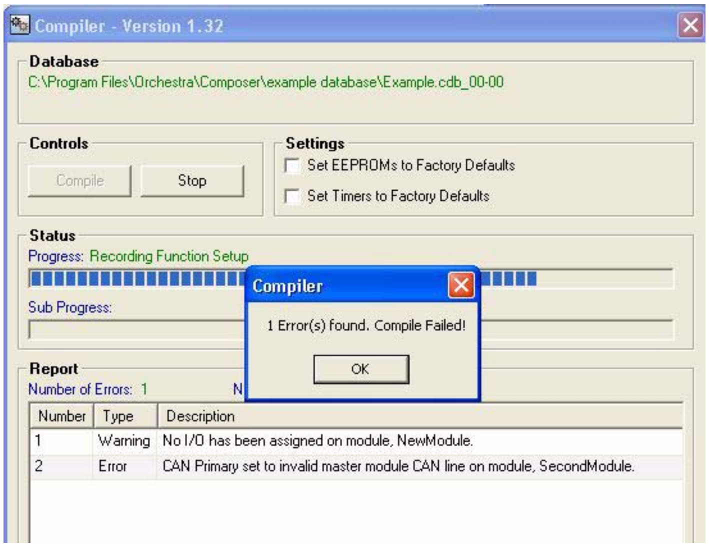
Status
Progress: Recording Function Setup
Sub Progress:

Compiler
1 Warning(s) found. Continue?

Report
Number of Errors: 0

Number	Type	Description
1	Warning	No I/O has been assigned on module, NewModule.

Figure 26



Compiler - Version 1.32

Database
C:\Program Files\Orchestra\Composer\example database\Example.cdb_00-00

Settings
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

Status
Progress: Recording Function Setup
Sub Progress:

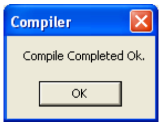
Compiler
1 Error(s) found. Compile Failed!

Report
Number of Errors: 1

Number	Type	Description
1	Warning	No I/O has been assigned on module, NewModule.
2	Error	CAN Primary set to invalid master module CAN line on module, SecondModule.

Figure 27

When a successful compile is completed, a “Compile Completed OK” message will appear. Click *OK* to continue **(Figure 28)**.



Compiler
Compile Completed Ok.

Figure 28

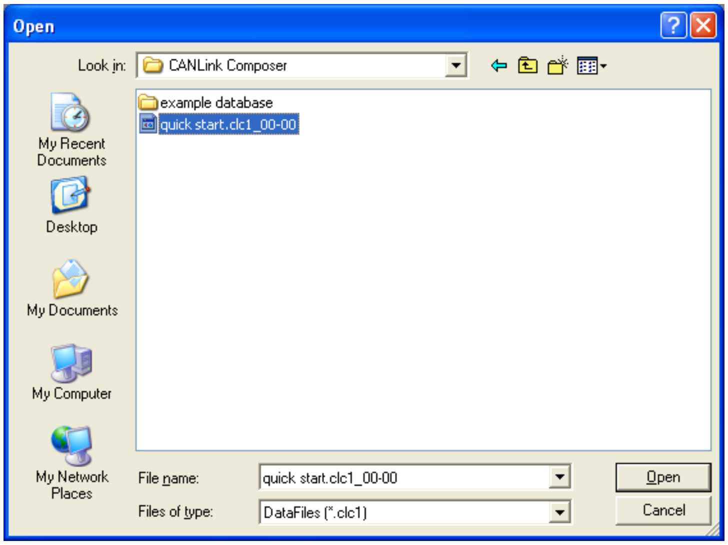
DOWNLOAD COMPOSER APPLICATION

Note: The following steps require that the computer is connected to a CANLink module and that the module is powered up.

42. Click the *Download Composer Application* button in the **Main** window to download the database to module "CL-103-100."

43. Choose the database to download and click *Open*. Database files will have the .clc1_xx-xx extension where the x represents the revision number.

Note: If no file exists with this extension, the database has not been successfully compiled.

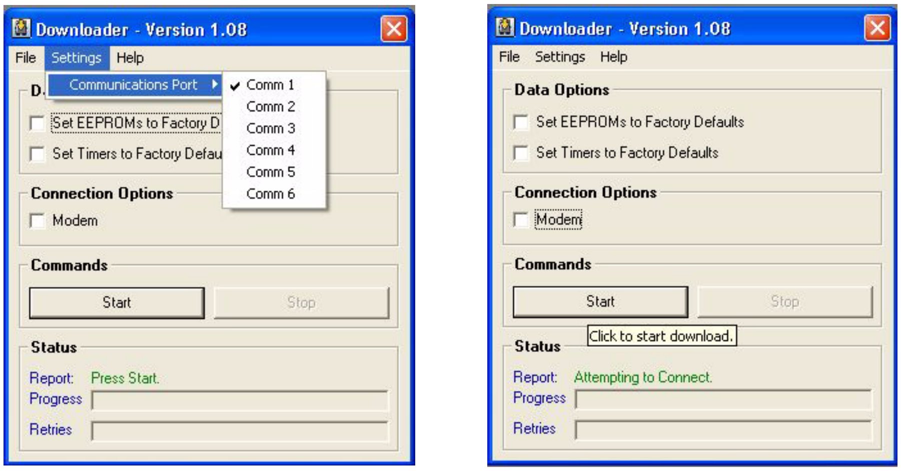


Look in: CANLink Composer

File name: quick start.clc1_00-00

Files of type: DataFiles (*.clc1)

44. Select the appropriate communications port.



☐ Set EEPROMs to Factory Defaults

☐ Set Timers to Factory Defaults

☐ Modem

Report: Press Start.

Click to start download.

Report: Attempting to Connect.

Figure 29

45. Click the *Start* button to begin the download. Follow any on-screen prompts.

Note: If the download fails, verify that the cable is securely connected and that the correct communications port is selected.

VERIFY OPERATION

46. Verify that when button 1 on the CL-604 switch panel is pressed, the red light on button 1 turns on.

Modules and Data Items

DEFINITIONS

Data Item Definitions

Input

Input on a module. These are read-only values.

Output

Output on a module. These are read/write values. Some module *Outputs* can generate *Status* or *Current Data Item Sub Types* that can be used in the *Comparison Blocks*. All *Outputs* generate a *Flashing Data Item Sub Type* that can be used in *Comparison* and *Operator Blocks*. However, *Status*, *Current* and *Flashing Data Item Sub Types* are not shown in the **Multi View** window.

Variable

These can be used to save temporary values, calculated values, etc. These are read/write values.

State Machine

This is a special type of *Data Item*. These are read/write values that are updated only after the last *Rung* of the Main Function has finished executing. You are able to enumerate the values in the Composer. The limit of the number of *State Machines* varies depending on the Master Module selected.

Timer

These are read/write values that are incremented or decremented by a defined amount of time as chosen from a menu. The options are 10 msec, 1 second, 10 seconds, 1 minute, and 10 minutes. The default is 10 msec. The limit of the number of *Timers* varies depending on the Master Module selected. *Timer* values can be saved into *Non-Volatile Memory* if the *Save on Shut-Down* option is checked and the module is capable with that feature. All *Timers* generate a *Status Data Item Sub Type* that can be used in *Comparison* and *Operator Blocks*. However, *Status Data Item Sub Type* is not shown in the **Multi View** window. The *Status* of a *Timer* can be either RUN or PAUSE.

EEPROM

EEPROM values get read from *Non-Volatile Memory* and are put into temporary *Variable* locations before any *Rungs* are processed. When you write to an *EEPROM Data Item*, the program actually writes to a temporary *Variable* location, not the actual *EEPROM* (*Non-Volatile Memory*) location. In order to write and save the *EEPROM Data Item* values, the Master Module must have an auxiliary power line and the *EEPROM Data Item* must have the *Save* option checked. The limit of the number of *EEPROMs* varies depending on the Master Module selected.

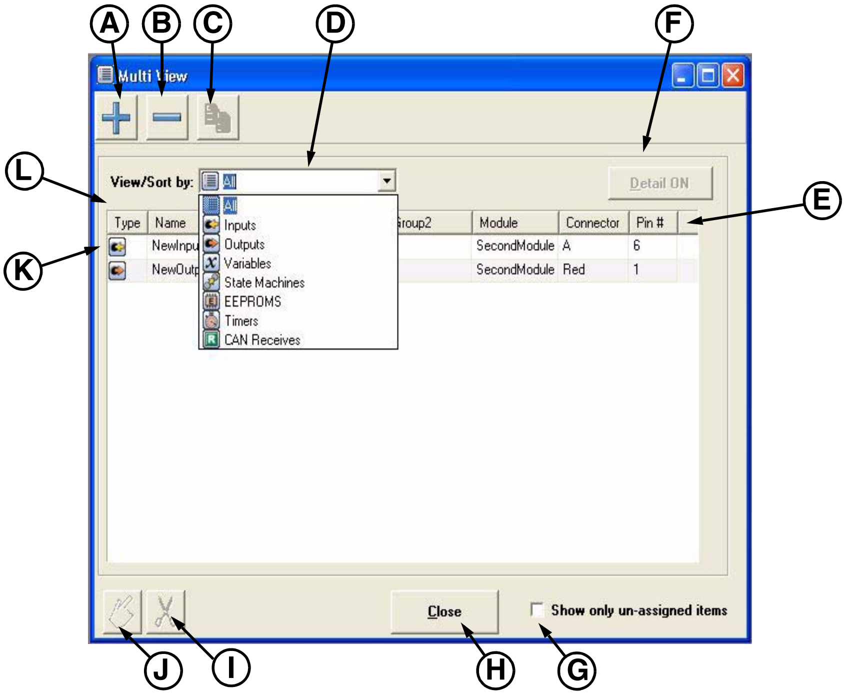
CAN Receive

This takes part of a CAN message and saves it to an IO Map location. You can define your own message or pick from a list of standard J1939 messages. You will need to select a CAN line. The limit of the number of *CAN Receives* varies depending on the Master Module selected. These are read-only values. All *CAN Receives* generate a *Status Data Item Sub Type* that can be used in *Comparison* and *Operator Blocks*. However, *Status Data Item Sub Type* is not shown in the **Multi View** window. The *Status* of a *CAN Receive* can be either RECEIVED or CLEAR. Every time a message is received for this *CAN Receive*, the *Status* is set to RECEIVED. This is a read/write *Variable*.

MULTI VIEW WINDOW

Description

The **Multi View** window displays all the *Inputs*, *Outputs*, and other *Data Items*. It can open the IO **Assignment** window and the **Create/Edit** window.



Type	Name	roup2	Module	Connector	Pin #
	NewInpu		SecondModule	A	6
	NewOutp		SecondModule	Red	1

Figure 1

Controls

(A) ADD Button

This will open the **ADD** window. Choose the *Type* of *Data Item* you wish to add (Input, Output, etc., as described in *Data Item Definitions on page 25*). Click the *OK* button and the **Create/Edit** window opens.

(B) DELETE Button

This will delete a *Data Item* from the list. A window will appear to verify the action. There is no "undo," so be certain before you click *Yes*.

(C) Copy Button

This is similar to the *ADD* button except that it will open the **Create/Edit** window and copy all the same attributes as the currently selected *Data Item*.

(D) View/Sort by: Dropdown Menu

Allows the *Data Items* to be sorted and displayed in different ways. "All" will display every *Data Item* type. Any other selection will display only the *Data Item* type chosen.

(E) Data Properties (Name, Group1, Group2, Module, Connector, Pin #)

Displays the *Data Items* that are currently selected in the *View/Sort by:* dropdown menu. Double-clicking a *Data Item* will open a **Create/Edit** window for that item.

(F) Detail ON Button

Available if only one *Data Item* type has been selected in the *View/Sort by:* dropdown menu (D). This will toggle between displaying all properties and standard properties.

(G) Show Only Un-assigned Items Check Box

Only available if *Inputs* or *Outputs* has been selected in the *View/Sort by:* dropdown menu (D). This will display only the *Data Items* that are currently not assigned to a module. Only *Inputs* and *Outputs* need to be assigned.

(H) Close Button

This will close the window.

(I) Un-Assign Button

This will remove the module, connector, and pin number assignments from the selected *Data Item*. No verification notice appears, so be certain you want to make the change before you click this button.

(J) Assign Button

This will open the **Assigning I/O** window where you can assign the currently selected *Input* or *Output* to a module. This button is available only for *Input* and *Output Data Items* and will be grayed out for all other *Data Items*.

(K) Data Item Type Icon

This is an icon that gives quick visual reference to the type of *Data Item* in that line, e.g., *Inputs*, *Outputs*, *Variables*, *State Machines*, *EEPROMs*, *Timers* and *CAN Receives*.

(L) Column Headers

These describe the type of *Data Item* shown in the column below each one. Sorting of the *Data Item* can be accomplished by clicking the column header to sort *Data Items* based on that column. A single click will sort ascending and a second click will sort descending.

Create/Edit Window

The **Create/Edit** window is accessed by clicking the *ADD* button, *Copy* button, or by double-clicking an existing *Data Item* in the **Multi View** window. If you click the *ADD* button, you will first have to choose the *Type* of *Data Item* you want to add from a dropdown list before the **Create/Edit** window opens.

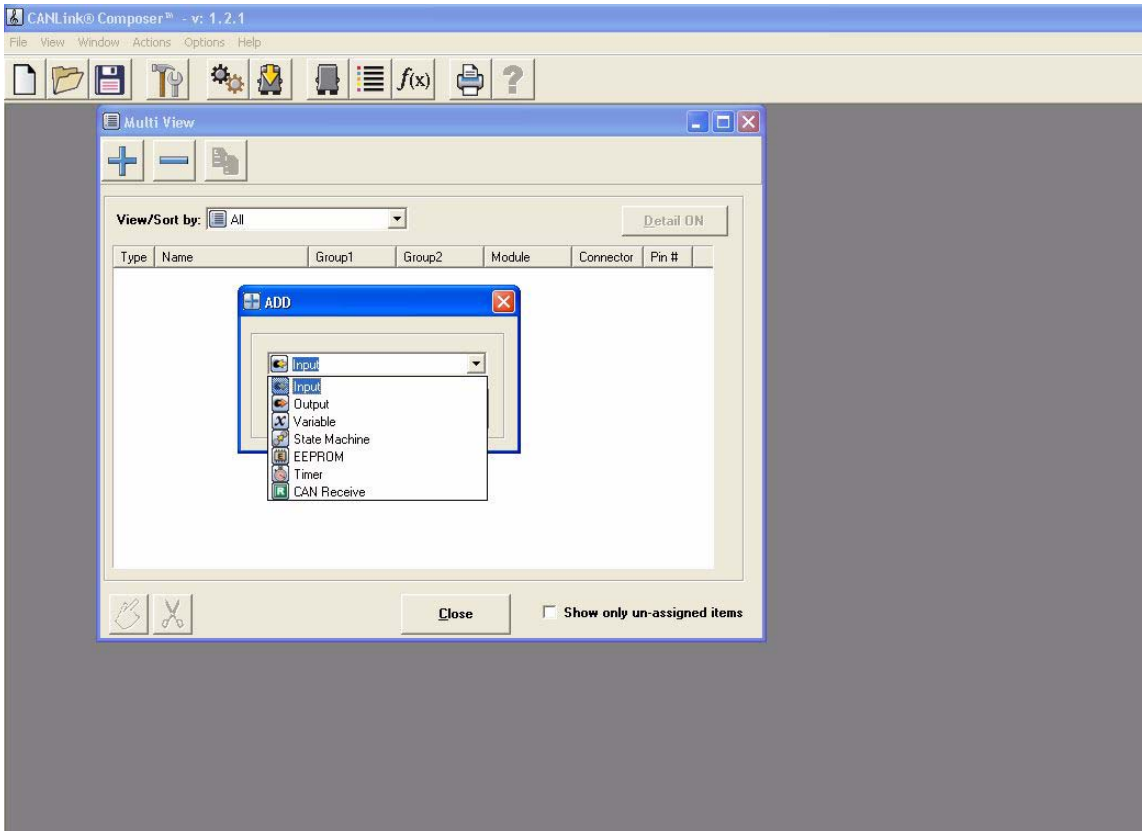
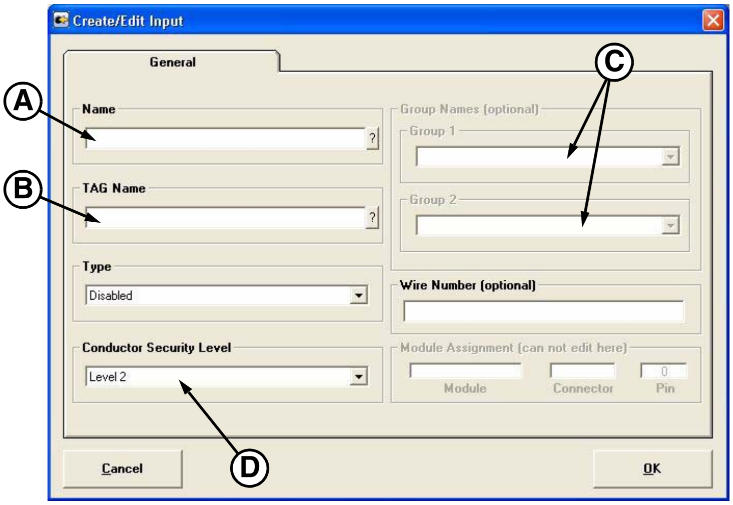


Figure 2

Once the *Data Item* type has been chosen, click *OK* to proceed or *Cancel* to abort.

Common Fields

When creating *Inputs*, *Outputs*, or other types of *Data Items* in the **Multi View** window, some fields are common to all types of *Data Items*.



Type: Disabled

Conductor Security Level: Level 2

Figure 3

Common to all *Data Item* types are:

(A) Name

This is a user-defined name of the *Data Item*. It must be 20 characters or less, and must be a unique name.

(B) TAG Name

This is a short name of the *Data Item*. This will appear in the block icon displayed in *Comparison* and *Operator Blocks* within the **Rung Editor** window. This does not need to be a unique name. If no name is entered, Composer will automatically use the first seven letters of the *Name*.

Use up to six characters for:

- *Timers*
- *Outputs*
- *CAN Receives*
- *Modules*

Note: A seventh character will be automatically tagged on to the end of the six characters:
Timers extra character: V = value, S = status
Outputs extra character: V = value, S = status, C = current, F = flashing
CAN Receives extra character: V = value, S = status
Modules extra character: M = Module status, C = CAN status

Use up to seven characters for:

- *Inputs*
- *EEPROMs*
- *Variables*
- *State Machines*

(C) Group Names (optional)

A *Data Item* can be grouped with other *Data Items*. *Groups* are used for sorting in the CANLink® Conductor™ and **Multi View** window. All **Create/Edit** windows provide two *Groups* to use. The *Group Name* must be 20 characters or less and can be either typed in or chosen from a previously defined *Group Name* dropdown menu.

(D) Conductor Security Level

This is the minimum level of Password/Dongle Security required to allow the user of Conductor to alter this *Data Item*.

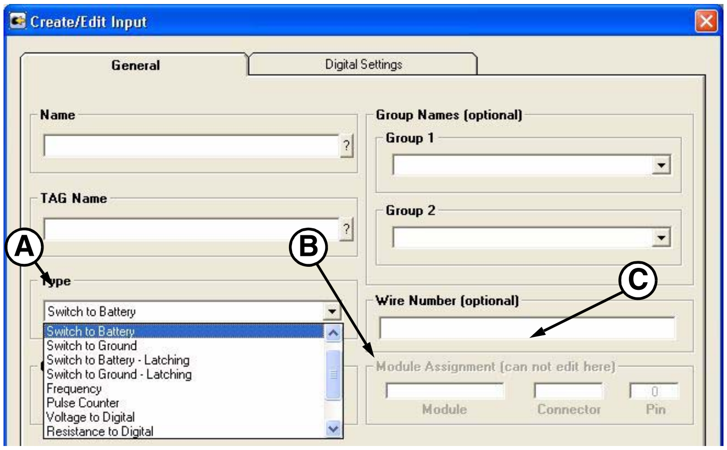
Example 1) Selecting “Level 3” allows Level 1, Level 2 and Level 3 Conductor users to alter this particular *Data Item*.

Example 2) Selecting “Level 1” allows only Level 1 Conductor users to alter this particular *Data Item*.

Create/Edit Input Window - Unique Fields

Depending on the *Input Type* chosen, additional tabs will be added to the window. Following is a description of each field in the various tabs.

General Tab



Type: Switch to Battery

Pin: 0

Figure 4

(A) Type

This defines the *Input Type* you want to work with.

The options are:

- STB "Switch to Battery" - Input turns ON when battery voltage is applied, and OFF when battery voltage is removed.
- STG "Switch to Ground" - Input turns ON when it is connected to ground, and OFF when ground connection is removed.
- STB "Switch to Battery - Latching" - Input turns ON when there is a battery connected, and turns OFF when the battery is disconnected and then reconnected.
- STG "Switch to Ground - Latching" - Input turns ON when ground is connected, and turns OFF when ground is disconnected and then reconnected.
- FRQ "Frequency" - Reports the frequency in Hz between pulses.
- PCT "Pulse Counter" - Counts the number of pulses.
- VTD "Voltage to Digital" - Reports the voltage in mV x 1000.
- RTD "Resistance to Digital" - Reports the resistance in ohms.
- FRQ "Frequency - AVG" - Similar to Frequency except it reports the average between the minimum value and the maximum value.
- VTD "Voltage to Digital - AVG" - Similar to Voltage to Digital except it reports the average between the minimum value and the maximum value.
- RTD "Resistance to Digital - AVG" - Similar to Resistance to Digital except it reports the average between the minimum value and the maximum value.

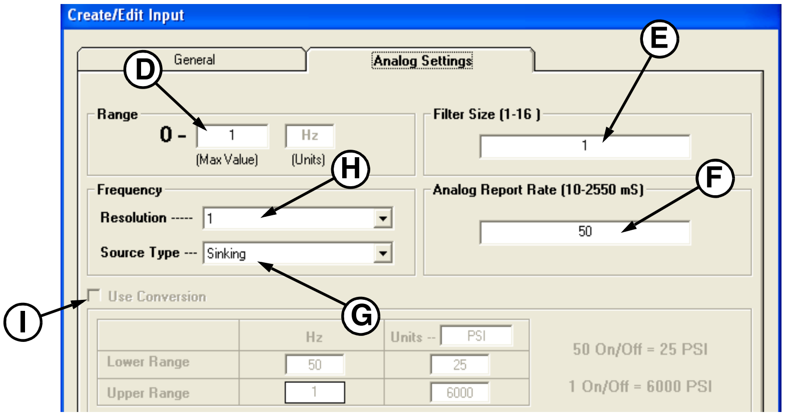
(B) Module Assignment

This displays the module name, connector, and pin that the *Input* is assigned to. It cannot be edited from this window. It will be blank if it has not been assigned.

(C) Wire Number (optional)

This is an optional description for the *Input/Output* to assist the Conductor user in sorting and troubleshooting a system.

Analog Tab


Create/Edit Input

General | Analog Settings

Range
0 - 1 (Max Value) Hz (Units)

Filter Size (1-16)
1

Frequency
Resolution ----- 1
Source Type --- Sinking

Analog Report Rate (10-2550 mS)
50

□ Use Conversion

	Hz	Units -- PSI
Lower Range	50	25
Upper Range	1	6000

50 On/Off = 25 PSI
1 On/Off = 6000 PSI

Figure 5

(D) Range

This defines the maximum value for the *Input*.

Applicable to the following *Input Types*:

Frequency

- Default: 1 Hz
- Acceptable values: 1 - 10000

Voltage to Digital

- Default: 1 mV
- Acceptable values: 1 - 65535

Resistance to Digital

- Default: 1 ohm
- Acceptable values: 1 - 65535

Frequency - AVG

- Default: 1 Hz
- Acceptable value: 1 - 10000

Voltage to Digital - AVG

- Default: 1 Hz
- Acceptable values: 1 - 65535

Resistance to Digital - AVG

- Default: 1 ohm
- Acceptable values: 1 - 65535

(E) Filter Size

This defines the maximum running average. The range is 1-16 samples, each sample being taken once a loop. The default is 4.

Applicable to the following *Input Types*:

1. Frequency
2. Voltage to Digital
3. Resistance to Digital

(F) Analog Report Rate

This defines how often the value is reported from the module back to Master Module and is defined in 10 msec increments. The default is 50 msec. This does not apply to the Master Module since analog *Inputs* are automatically updated once a loop.

Applicable to the following *Input Types*:

1. Frequency
2. Pulse Counter
3. Voltage to Digital
4. Resistance to Digital
5. Frequency - AVG
6. Voltage to Digital - AVG
7. Resistance to Digital - AVG

(G) Frequency - Source Type

This defines the type of *Output* the sensor will provide to the *Input*. Options are “Sourcing” (requires sensor to supply battery when active) and “Sinking” (requires sensor to supply ground when active). The default is “Sinking.”

(H) Frequency - Resolution

This defines the resolution (per bit) that the slave module reports back to the Master Module. Options are 0.01, 0.05, 0.1, 0.5, 1.0 and 2.0 Hz per bit. The default is 1.0 Hz.

Example:

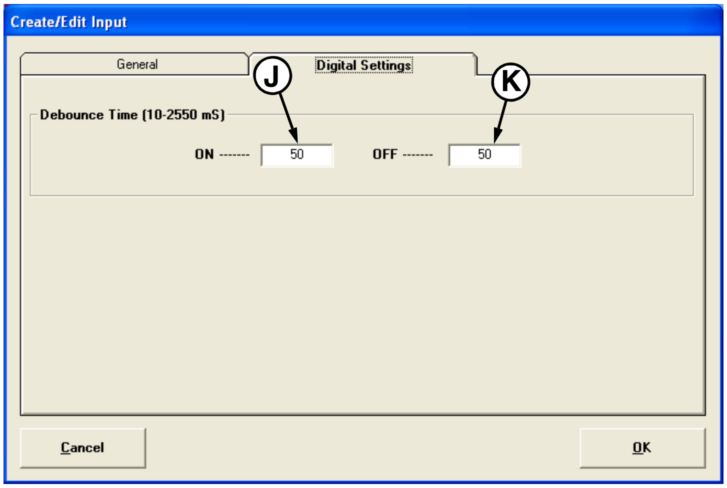
A 16 bit number is used equaling 0 - 65535. If the *Frequency Resolution* is set to 0.01 per bit, the resulting frequency range would be 0 - 655.35 Hz in 0.01 Hz steps. If the *Frequency Resolution* is set to 0.05 Hz, the resulting frequency range would be 0 - 3276.75 Hz in 0.05 Hz steps.

(I) Conversion

This converts *Range (Units)* to a different type of unit (PSI, temperature, etc.). Conversion is accomplished by selecting two points on a line. The “Upper Range” value is automatically inserted from the *Max Value* in *Range*. This is only applicable for the values in Conductor.

- Slope = ((Units Upper - Units Lower) / (Input Upper - Input Lower))
- Offset = Units Upper - (Slope * Input Upper)

Digital Tab



Debounce Time (10-2550 mS)

ON ------- 50 OFF ------- 50

Figure 6

(J) Debounce Time - ON

This defines the amount of time the *Input* must be active before it will switch from the inactive state to the active state. The range is 10 - 2550 in 10 msec increments. The default is 50 msec.

This is applicable for the following *Input Types*:

1. Switch to Battery
2. Switch to Ground
3. Switch to Battery - Latching
4. Switch to Ground - Latching

(K) Debounce Time - OFF

This defines the amount of time the *Input* must be inactive before it will switch from the active state to the inactive state. The range is 10 - 2550 in 10 msec increments. The default is 50 msec.

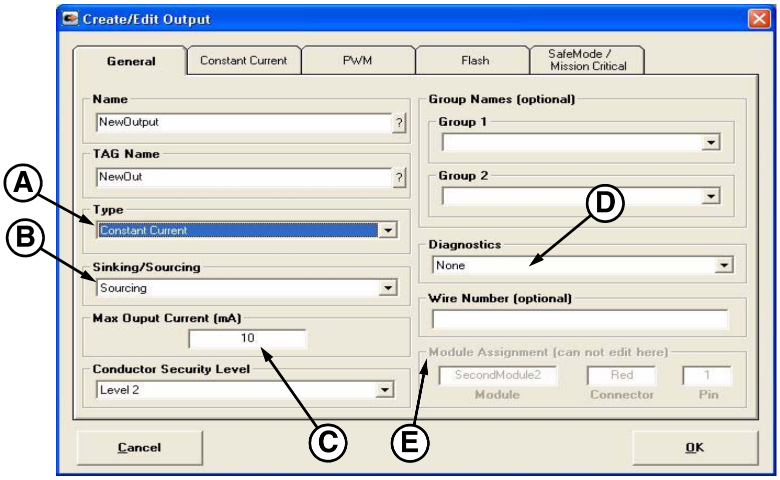
This is applicable for the following *Input Types*:

1. Switch to Battery
2. Switch to Ground
3. Switch to Battery - Latching
4. Switch to Ground - Latching

Create/Edit Output Window - Unique Fields

Depending on the *Output Type* chosen, additional tabs will be added to the window. Below is a description of each field in the various tabs.

General Tab



Name: NewOutput

TAG Name: NewOut

Type: Constant Current

Sinking/Sourcing: Sourcing

Max Ouput Current (mA): 10

Conductor Security Level: Level 2

Group Names (optional)

Group 1:

Group 2:

Diagnostics: None

Wire Number (optional):

Module Assignment (can not edit here)

Module: SecondModule2

Connector: Red

Pin: 1

Figure 7

(A) Type

This defines the *Output Type* you want to work with.

The options are:

- DIG "On/Off" - Turns *Output* on or off.
- PWM "PWM" - Can turn on at a percentage of 0-100%, in 0.1% increments.
- CC1 "Constant Current" - Controls the current of 0 - 4.095 Amps, in 0.001 Amp increments.
- SV1 "PWM - Single Servo" - Can turn on at a percentage of 0-100% in 0.1 increments. It also has a direction. Switch two pins between sourcing and sinking *Outputs*; one sinks and one sources.
- SV2 "PWM - Double Servo" - Can turn on at a percentage of 0-100% in 0.1 increments. It also has a direction. Switch two pins between sourcing and sinking *Outputs*; both sink or both source.
- FRQ "Frequency" - Can control the frequency of the PWM *Output* of 0 - 10,000 Hz in 1 Hz increments.

(B) Sinking/Sourcing

This defines the polarity of the *Output* when it is on. Options are "Sinking" (*Output* supplies ground when active) and "Sourcing" (*Output* supplies battery when active). The default is "Sourcing."

Applicable to the following *Output Types*:

1. On/Off
2. PWM
3. Frequency

(C) Max Output Current

This defines the maximum current that the load will draw. This is entered in mA. The range is 10 - 65535 mA. The default value is 10 mA.

(D) Diagnostics

This defines the type of diagnostics the *Output* has. Default value is "None."

The options are:

- "None"
- "Status"
- "Current" *
- "Status/Current" *

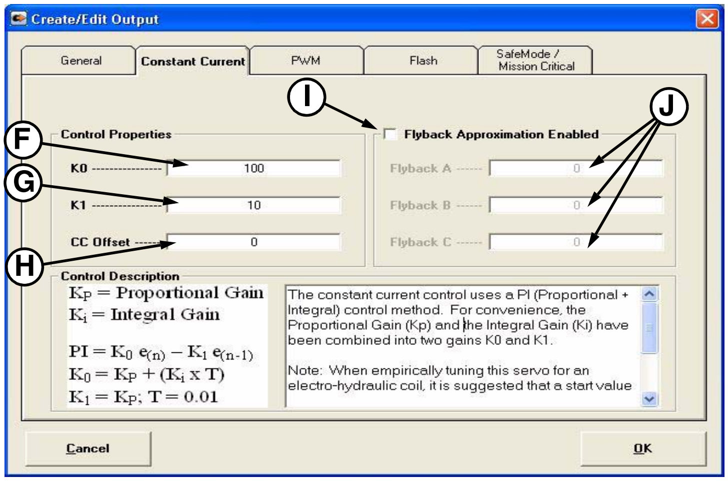
** Selecting either of these will cause a Digital Fuse/Report Rate tab to appear in place of the Constant Current tab.*

Note: Not all Outputs have Diagnostics, so if you select it here, you need to make sure that the Output you assign it to also has the necessary diagnostics. Also, if you select "None" but assign it to an Output that has diagnostics, the diagnostics will still be active.

(E) Module Assignment

This displays the module name, connector, and pin that the *Output* is assigned to. It cannot be edited from this window. This will be blank if the *Output* has not been assigned.

Constant Current Tab



Create/Edit Output

General | Constant Current | PWM | Flash | SafeMode / Mission Critical

Control Properties

K0 ---------------- 100

K1 ---------------- 10

CC Offset ------ 0

□ Flyback Approximation Enabled

Flyback A ------ 0

Flyback B ------ 0

Flyback C ------ 0

Control Description

K_P = Proportional Gain
K_i = Integral Gain

$PI = K_0\, e_{(n)} - K_1\, e_{(n-1)}$
$K_0 = K_P + (K_i \times T)$
$K_1 = K_P;\ T = 0.01$

The constant current control uses a PI (Proportional + Integral) control method. For convenience, the Proportional Gain (Kp) and the Integral Gain (Ki) have been combined into two gains K0 and K1.

Note: When empirically tuning this servo for an electro-hydraulic coil, it is suggested that a start value

Figure 8

(F) K_0

This combines with K_1 to control the speed that the *Output* will adjust its duty cycle to get the commanded current. Range is 1-255. The higher the number the faster it will try to compensate.

(G) K_1

This combines with K_0 to control the speed that the *Output* will adjust its duty cycle to get the commanded current. Range is 1-255. The higher the number the slower it will try to compensate.

(H) Constant Current Properties - CC Offset

This is the duty cycle that the current control will start at. This should be 2-3% below the threshold value. This will make the *Output* turn on quicker. Range is 0-100% in 1% increments. The default is 0%. This offset is only applicable for the “Constant Current” *Output Type*.

(I) Constant Current Properties - Flyback Check Box

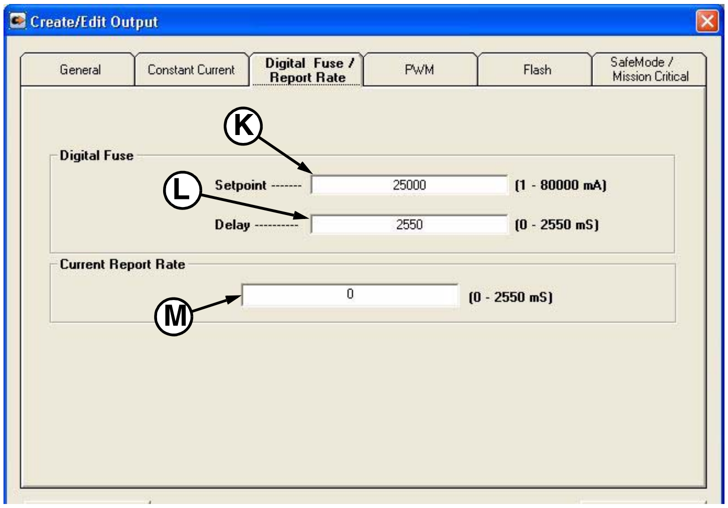
If this box is not checked, the flyback factors are disabled and will default to 0. If this box is checked, the flyback factors are enabled.

(J) Constant Current Properties - Flyback Factor A, B, C

The flyback factors are used to approximate the flyback current and are only necessary if using single wire constant current (Follow the *Flyback Calculation Procedure on page 94* if you need help calculating these factors). These factors are only applicable for the “Constant Current” *Output Type*. Ranges are:

- A - 1 - 65535
- B - 1 - 65535
- C - 1 - 65535

Digital Fuse/Report Rate Tab



Create/Edit Output

Digital Fuse

Setpoint ------- 25000 [1 - 80000 mA]

Delay --------- 2550 [0 - 2550 mS]

Current Report Rate

0 [0 - 2550 mS]

Figure 9

(K) Digital Fuse Setpoint

If the current goes above this value for the time defined in the *Digital Fuse Delay Time (L)*, it will shut the *Output* OFF and be marked as over-current. The range is 1 - 80000 mA. Default value is 25000 mA. This is only applicable for *Outputs* with current feedback.

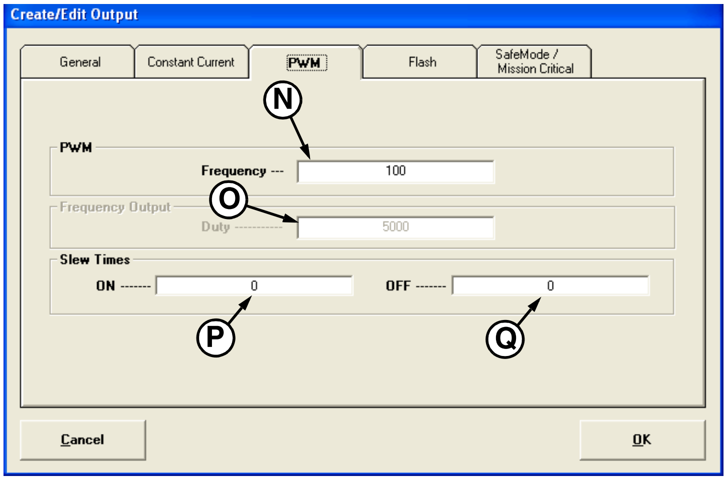
(L) Digital Fuse Delay

This defines the amount of time which the current can be above the value set in the *Digital Fuse Setpoint* before the *Output* will shut OFF and be marked as over-current. The range is 0 - 2550 mS. Default value is 2550 mS. This is only applicable for *Outputs* with current feedback.

(M) Current Report Rate

This defines how fast the current will be reported back. This only needs to be less than 2.5 seconds if it is being used in a *Comparison Block*. The range is 0 - 2550 mS. Default value is 0. This is only applicable for *Outputs* with current feedback.

PWM Tab



PWM

Frequency --- 100

Frequency Output

Duty ---------- 5000

Slew Times

ON ------- 0

OFF ------- 0

Figure 10

(N) PWM - Frequency

This defines the frequency of the *Output* in Hz. Range is 1 - 10,000 Hz. The default is 100 Hz.

Applicable to the following *Output Types*:

1. PWM
2. Constant Current
3. PWM - Single Servo
4. PWM - Double Servo

(O) Frequency Output - Duty

This defines the duty cycle of the *Output*. Range is 10 - 990. The default is 500. This is only applicable for *Frequency Outputs*.

(P) Slew Times - ON

This defines the amount of time used by the *Output* to go from off to maximum value. The range is from 0 - 2.5 seconds in 10 msec increments. The default is 0.

This is applicable to the following *Output Types*:

1. PWM
2. Constant Current
3. PWM - Single Servo
4. PWM - Double Servo
5. Frequency

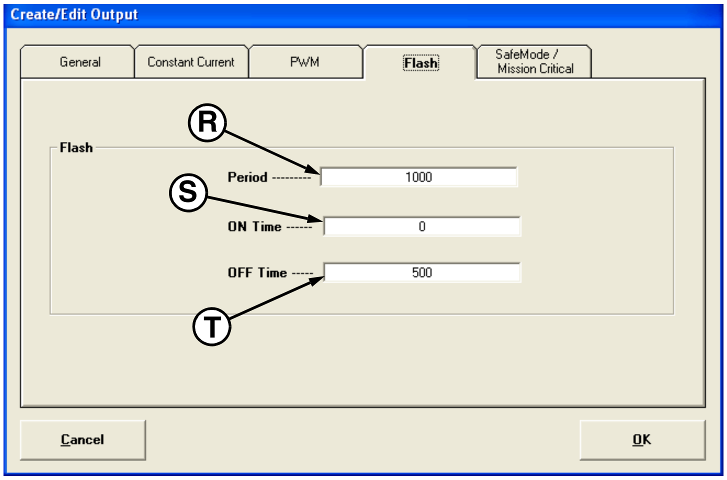
(Q) Slew Times - OFF

This defines the amount of time used by the *Output* to go from maximum value to off. The range is from 0 - 2.5 seconds in 10 msec increments. The default is 0.

This is applicable to the following *Output Types*:

1. PWM
2. Constant Current
3. PWM - Single Servo
4. PWM - Double Servo
5. Frequency

Flash Tab



Flash

Period --------- 1000

ON Time ------ 0

OFF Time ----- 500

Figure 11

(R) Flash - Period

This defines how long a flash cycle takes to complete. The range is 1 - 65535 in 10 msec increments. The default is 1000 msec.

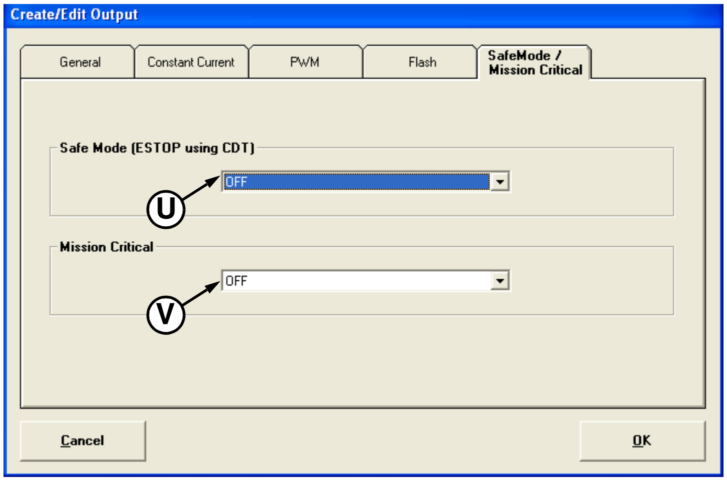
(S) Flash - ON Time

This defines the point in *Flash - Period* at which the *Output* will turn off. *ON Time* must be less than *OFF Time* and less than the *Period*. The range is 0 - 65535 in 10 msec increments. The default is 0 seconds.

(T) Flash - OFF Time

This defines the point in *Flash - Period* at which the *Output* will turn off. *OFF Time* must be greater than *ON Time* and less than the *Period*. The range is 0 - 65535 in 10 msec increments. The default is 500 msec.

SafeMode/Mission Critical Tab



Safe Mode (ESTOP using CDT): OFF

Mission Critical: OFF

Figure 12

(U) Safe Mode

This defines what the *Output* will do when it goes into *Safe Mode*. The modules can be forced into *Safe Mode* by pressing the space bar while using Conductor. The module will also go into *Safe Mode* when it drops below the low voltage threshold if it is enabled. The default value is “OFF.”

Options are:

- “OFF” - The *Output* turns off regardless of the last state.
- “ON” - The *Output* turns on regardless of the last state.
- “Maintain State” - The *Output* will maintain the state it was in at the time of entering *Safe Mode*.
- “ON with FLASH” - The *Output* will turn on and flash when entering *Safe Mode*.

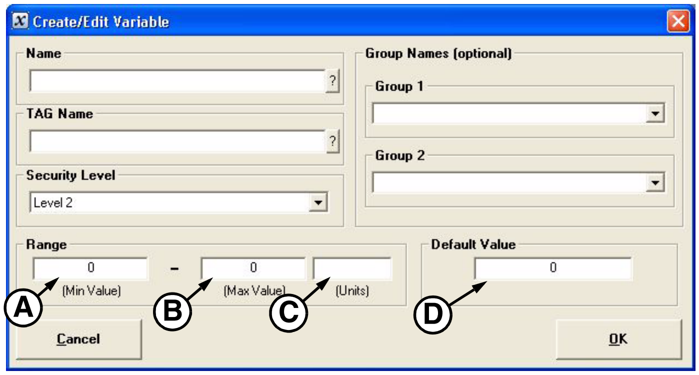
(V) Mission Critical

This defines what the *Output* will do when it goes into *Comm Fail* mode. This is applicable only to Client Modules. For a module to enter *Mission Critical* mode the module must lose communications with the Master Module. The default value is “OFF.”

Options are:

- “OFF” - The *Output* turns off regardless of the last state.
- “ON” - The *Output* turns on regardless of the last state.
- “Maintain State” - The *Output* will maintain the state it was in at the time of entering *Mission Critical*.
- “ON with FLASH” - The *Output* will turn on and flash when entering *Mission Critical*.

Create/Edit Variable Window



Security Level: Level 2

Range: 0 - 0

Default Value: 0

Figure 13

(A) Range - Min Value

Range value must be between 0 - 4294967294.

The *Min Value* must be smaller than the *Max Value*.

(B) Range - Max Value

Range value must be between 1 - 4294967295.

The *Max Value* must be greater than the *Min Value*. The *Max Value* determines the size of the *Variable*. *Max Value* in the range of 0 - 65535 will be treated as 2 bytes of information. *Max Value* from 65536 - 4294967295 will be treated as 4 bytes.

(C) Range Units

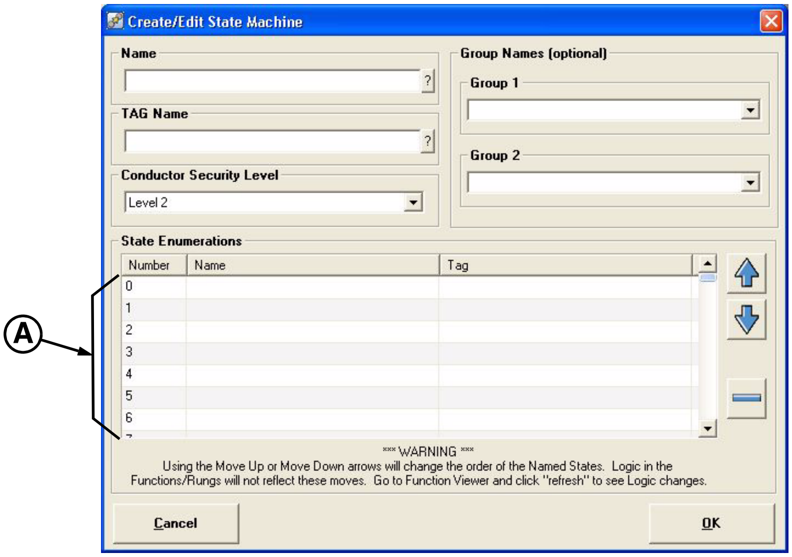
The user can enter whatever they want for units. This is used for displaying data in Conductor.

(D) Default Value

This defines the value at startup.

Create/Edit State Machine Window

Double-click on the line item **(Figure 14, A)** you want to edit or create. The **Create/Edit State Names** window will open **(Figure 15)**.



Conductor Security Level: Level 2

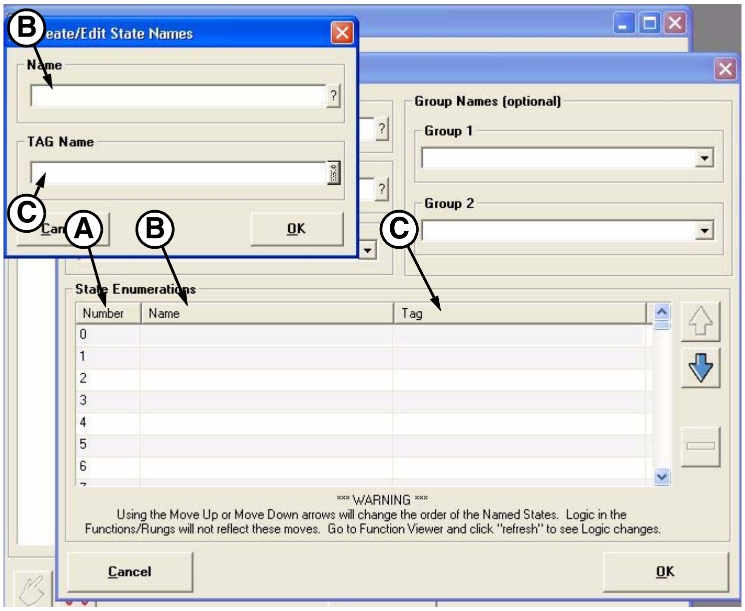
Number	Name	Tag
0		
1		
2		
3		
4		
5		
6		

*** WARNING ***
Using the Move Up or Move Down arrows will change the order of the Named States. Logic in the Functions/Rungs will not reflect these moves. Go to Function Viewer and click "refresh" to see Logic changes.

Figure 14

Controls

- The ⇧ UP and ⇩ DOWN arrows will move the highlighted line item to change the order of the Named States. Logic in the **Rung Editor** and **Function View** windows will not reflect these changes. Go to the **Function View** window and click *Refresh* to see the logic changes.
- The ▬ *DELETE* button will delete the highlighted line item.



*** WARNING ***
Using the Move Up or Move Down arrows will change the order of the Named States. Logic in the Functions/Rungs will not reflect these changes. Go to Function Viewer and click "refresh" to see Logic changes.

Figure 15

(A) State Enumerations - Number

Each *State Machine* can have up to 255 states.

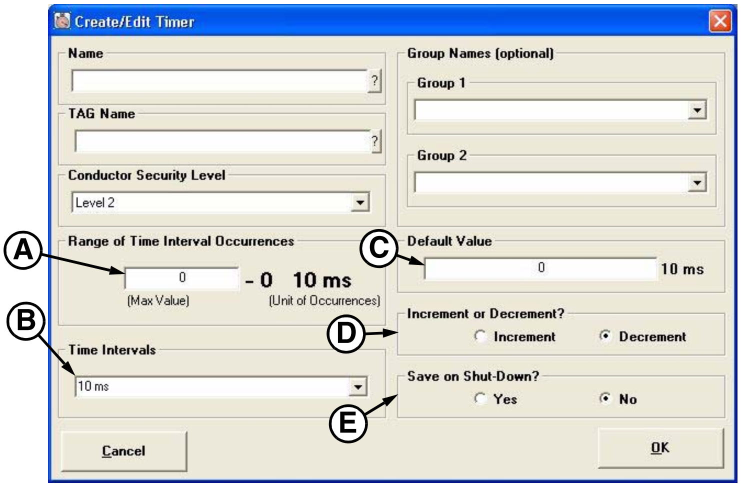
(B) State Enumerations - Name

This is a 1-20 character description of the enumerated *State* that will be shown in the **Rung Editor** and **Function View** windows. The enumeration name must be unique to the specific *State Machine*.

(C) State Enumerations - Tag

This is a 1-7 character description of the enumerated name. This is used to display the name inside the *Operator* and *Comparison Blocks* of the *Rung* in the **Rung Editor** window. This does not need to be a unique name. If nothing is entered, it will default to the first seven characters of the enumeration name.

Create/Edit Timer Window



Create/Edit Timer

Name:

TAG Name:

Conductor Security Level: Level 2

Range of Time Interval Occurrences: 0 - 0 10 ms (Max Value) (Unit of Occurrences)

Time Intervals: 10 ms

Group Names (optional)

Group 1:

Group 2:

Default Value: 0 10 ms

Increment or Decrement? ○ Increment ● Decrement

Save on Shut-Down? ○ Yes ● No

Figure 16

(A) Range of Time Interval Occurrences

This defines the range of the *Timer*. Range value must be between 0 - 4294967294.

The *Min Value* must be smaller than the *Max Value*.

(B) Time Intervals

This defines the unit of time the *Timer* is in. The options are 10 msec, 1 second, 10 seconds, 1 minute, and 10 minutes. The default is 10 msec.

(C) Default Value

This is used two different ways:

- If it is saved on shut-down, this will be the default value the first time the unit runs or if *Restore Factory Defaults* is selected at programming time.
- If it is not saved on shut-down, this will be the value on startup.

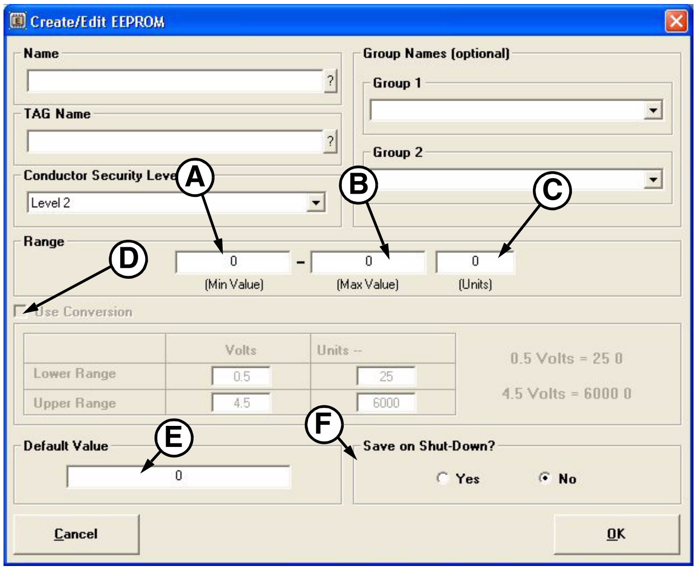
(D) Increment or Decrement?

This defines if the *Timer* will count up (“Increment”) or count down (“Decrement”). The default is “Decrement.” For Decrements the *Timer* counts down until it reaches 0, then stops. For Increments the *Timer* counts up to a value of 65535 or 4294967295, depending on the range, then stops.

(E) Save on Shut-Down?

This defines whether or not the latest *Timer* value is saved on shut-down. The saved value will be loaded on the next module startup. The default is “No.”

Create/Edit EEPROM Window



Name

TAG Name

Conductor Security Leve

Level 2

Group Names (optional)

Group 1

Group 2

Range

0 - 0 0
(Min Value) (Max Value) (Units)

□ Use Conversion

	Volts	Units --
Lower Range	0.5	25
Upper Range	4.5	6000

0.5 Volts = 25 0

4.5 Volts = 6000 0

Default Value

0

Save on Shut-Down? ○ Yes ● No

Figure 17

(A) Range - Min Value

Range value must be between 0 - 4294967294.

The *Min Value* must be smaller than the *Max Value*.

(B) Range - Max Value

Range value must be between 1 - 4294967295.

The *Max Value* must be greater than the *Min Value*. The *Max Value* determines the size of the *Variable*. *Max Value* in the range of 0 - 65535 will be treated as 2 bytes of information. *Max Value* from 65536 - 4294967295 will be treated as 4 bytes.

(C) Range - Units

The user can enter whatever they want for units. This is used for displaying data in Conductor.

(D) Use Conversion

This converts *Range (Units)* to a different type of unit (PSI, temperature, etc.). Conversion is accomplished by selecting two points on a line. The “Upper Range” value is automatically inserted from the *Max Value* in *Range*. This is only applicable for the values in Conductor.

- Slope = ((Units Upper - Units Lower) / (Input Upper - Input Lower))
- Offset = Units Upper - (Slope * Input Upper)

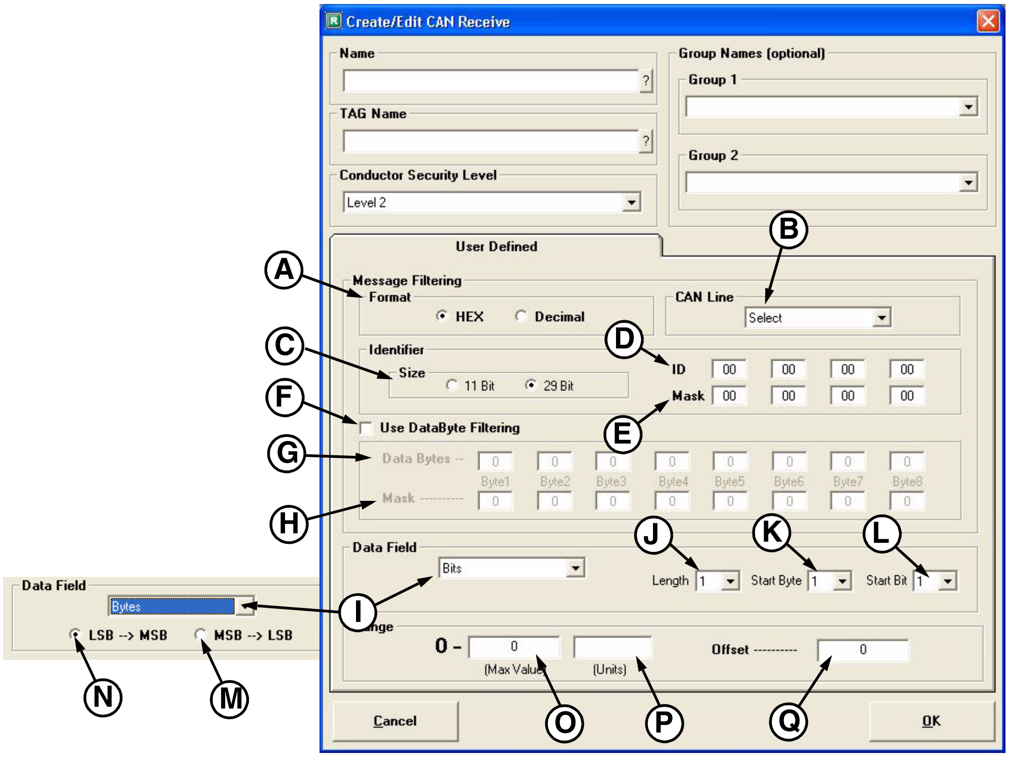
(E) Default Value

This is what the value will be if the *EEPROM* has never been written to or, if in Conductor, the *Restore Factory Defaults* button is pressed.

(F) Save on Shut-Down

This defines if the *EEPROM value* is saved on shut-down. The default is “No.”

Create/Edit CAN Receive Window



Create/Edit CAN Receive

Name

TAG Name

Conductor Security Level: Level 2

Group Names (optional)

Group 1

Group 2

User Defined

Message Filtering

Format: ● HEX ○ Decimal

CAN Line: Select

Identifier

Size: ○ 11 Bit ● 29 Bit

ID: 00 00 00 00

Mask: 00 00 00 00

□ Use DataByte Filtering

	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Data Bytes --	0	0	0	0	0	0	0	0
Mask ---------	0	0	0	0	0	0	0	0

Data Field: Bits

Length 1 Start Byte 1 Start Bit 1

Range: 0 - 0 (Max Value) (Units)

Offset --------- 0

Data Field: Bytes

● LSB --> MSB ○ MSB --> LSB

Figure 18

(A) Message Filtering - Format

This changes the type of data entered into fields (D, E, G, and H). Choices are hexadecimal or decimal.

(B) Message Filtering - CAN Line

Selects which CAN line the message is to be received on.

(C-E) Message Filtering - Identifier

- *(C) Size - There are 2 different types of CAN Identifiers: 11 bit and 29 bit. Changing this affects the size of (D) and (E).*
- *(D) ID - Enter the 11 or 29 bit CAN Identifier you want the module to accept.*
- *(E) Mask - Setting a bit to 1 in the Mask (E) will make the code check the same bit in ID (D). Setting a bit to 0 will make the code not check the bit in (D).*

Example 1:

ID = 00 00 01 00
Mask = 00 00 FF 00

Module will accept any message as long as the third byte is 01. Only the third byte in the mask is set to match.

Example 2:

ID = 00 00 01 00
Mask = FF FF FF FF

Module will accept only message 00 00 01 00. The mask is set to match all bits.

(F-H) Message Filtering - Use DataByte Filtering

- *(F) Use DataByte Filtering Check Box - If this is checked, the data bytes can be used for additional message acceptance filtering.*
- *(G) Data Bytes - Enter the data byte values you want the module to accept.*
- *(H) Mask - Setting a bit to 1 in the Mask (H) will make the code check the same bit in Data Bytes (G). Setting a bit to 0 will make the code not check the bit in (G).*

(I-N) Data Field

- *(I) Type - Set the kind of data you want to get: Bytes or Bits. Selecting Bytes adds properties (M) and (N).*
- *(J) Length - This changes based on (I). States the number of bytes or bits you wish to look at. You can look for 4 bytes or 32 bits.*
- *(K) Start Byte - Determines which byte of data to start looking at. There are 8 data bytes per CAN message. So if your data is in byte 2, set this to 2.*
- *(L) Start Bit - Used only when (I) is set to Bits. Determines which bit to start looking at in a data byte. There are 8 bits per byte.*
- *(M) MSB-LSB - Order of data fields is Most Significant Bit to Least Significant Bit. Used when (I) is set to bytes and the length is set to more than 2 bytes.*
- *(N) LSB-MSB - Order of data fields is Least Significant Bit to Most Significant Bit. Used when (I) is set to bytes and the length is set to more than 2 bytes.*

(O-Q) Range

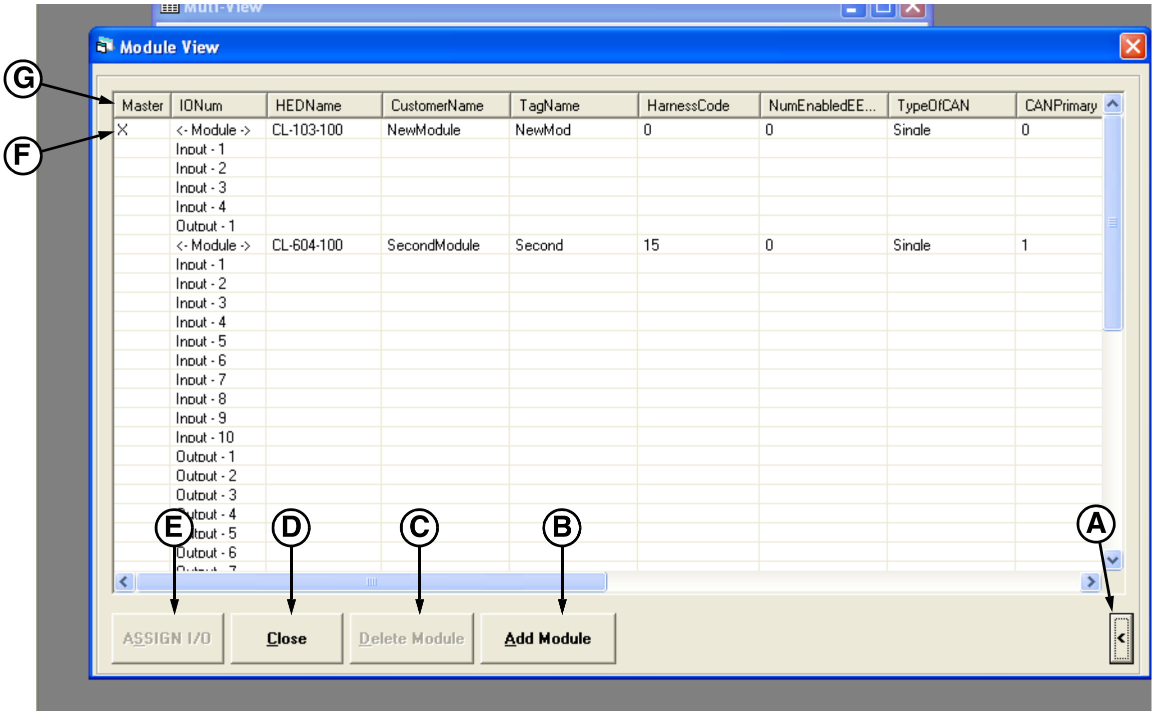
- *(O) Max Value - Range value must be between 1 - 4294967295.
The Max Value must be greater than the Min Value. The Max Value determines the size of the CAN Receive Data Item. Max Value in the range of 0 - 65535 will be treated as 2 bytes of information. Max Value from 65536 - 4294967295 will be treated as 4 bytes.*
- *(P) Units - The user can enter whatever they want for units. This is used for displaying data in Conductor.*
- *(Q) Offset - The offset can be a positive or negative number and will be subtracted from or added to the received value.*

MODULE VIEW WINDOW

Description

This window is accessed from the **Main** window by either clicking the *Open Module View* button or choosing *View>Module View* from the file menu.

The **Module View** window shows the modules and the IO associated with each. It can open the **Module Configuration** window.



Master	IONum	HEDName	CustomerName	TagName	HarnessCode	NumEnabledEE...	TypeOfCAN	CANPrimary
X	<- Module ->	CL-103-100	NewModule	NewMod	0	0	Single	0
	Input - 1							
	Input - 2							
	Input - 3							
	Input - 4							
	Output - 1							
	<- Module ->	CL-604-100	SecondModule	Second	15	0	Single	1
	Input - 1							
	Input - 2							
	Input - 3							
	Input - 4							
	Input - 5							
	Input - 6							
	Input - 7							
	Input - 8							
	Input - 9							
	Input - 10							
	Output - 1							
	Output - 2							
	Output - 3							
	Output - 4							
	Output - 5							
	Output - 6							
	Output - 7							

Figure 19

Controls

(A) Expand Button

This will expand the size of the window.

(B) Add Module Button

Opens the **Module Configuration** window allowing you to create a module.

(C) Delete Module Button

Used to delete the module that is highlighted in the *List Box*. A window will appear to verify the action. There is no “undo,” so be certain before you click *Yes*.

(D) Close Button

This will close the **Module View** window.

(E) Assign I/O Button

Opens the **Assignment** window.

(F) List Box

Displays the module and its various properties. Double-clicking on a module will open the **Module Configuration** window where the module can be edited.

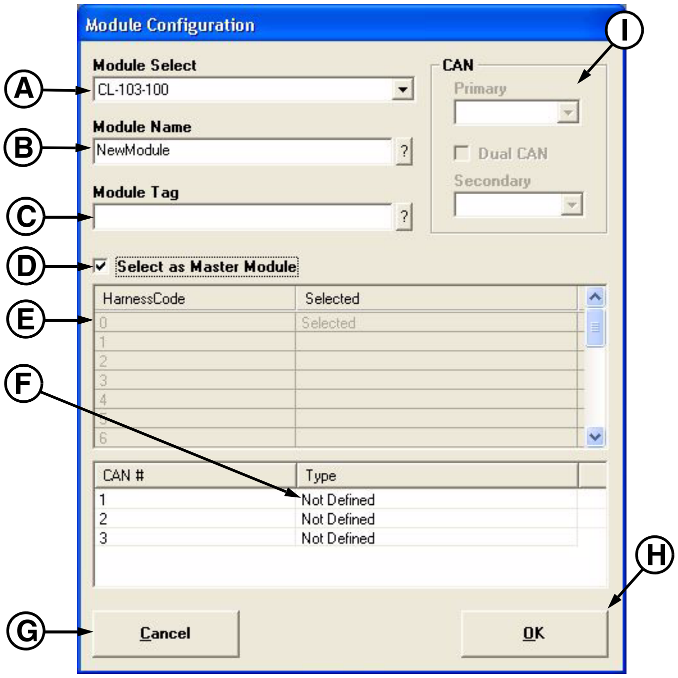
(G) Column Headers

These describe the type of *Data Item* shown in the column below each one. Clicking on any of these will not cause the *Data Item* to sort.

MODULE CONFIGURATION WINDOW

Description

The **Module Configuration** window will be used to add or edit a module. This window can only be opened from the **Module View** window by double-clicking a module *Name* or clicking the + *ADD* button.



Module Configuration

Module Select: CL-103-100

Module Name: NewModule

Module Tag:

☑ Select as Master Module

CAN

Primary:

☐ Dual CAN

Secondary:

HarnessCode	Selected
0	Selected
1	
2	
3	
4	
5	
6	

CAN #	Type
1	Not Defined
2	Not Defined
3	Not Defined

Figure 20

Controls

(A) Module Select Dropdown Menu

This contains a list of modules from the Module Database. If this window is opened by clicking the + *ADD* button in **Module View**, this box will be active. If this window is opened by double-clicking a module in the **Module View** *List Box*, this box will be grayed out and contain the name of the module type.

(B) Module Name Text Box

Enter a unique name in this box for the module. The name can be 1-20 characters.

(C) Module Tag Text Box

This is a description of the module and must be six characters or less. This name will be displayed inside *Comparison* and *Operator Blocks* within the **Rung View** window. This does not need to be a unique name. If no name is entered, Composer will automatically use the first six letters of the *Name*.

(D) Select as Master Module

Assigns the module as the Master Module. Only one module can be used as a Master Module. This is active only if the module is capable of being a master, as defined by (A), and no other module has been selected as the Master Module.

(E) Harness Information Text Box

Shows the available harness number for the module. If a harness number is in red, that means it is already being used by a module of the same type and cannot be selected for this module. This is not active if the module has been selected as a Master Module.

Note: To calculate the module's harness code, start with 15 and then:
subtract 1 if Harness 1 is grounded.
subtract 2 if Harness 2 is grounded.
subtract 4 if Harness 3 is grounded.
subtract 8 if Harness 4 is grounded.

(F) CAN Information Text Box

This displays the number of CAN lines on the Master Module.

Double-clicking on a line will cause the *Type* to cycle through “Not Defined,” “Module,” and “J1939” options. This box is active only if the module is selected as Master Module.

- “Not Defined” means Disable CAN line.
- “Module” means other HED CANLink modules.
- “J1939” means non-HED CANLink modules.

(G) Cancel Button

Returns control to the **Module View** window without making changes.

(H) OK Button

Updates and returns control back to the **Module View** window. If module was selected as a Master Module and a Master Module has already been selected, you will be prompted to make the change. Click *OK* to change or *Cancel* to exit without making changes.

(I) CAN

This is active only for slave modules.

Primary Dropdown Menu

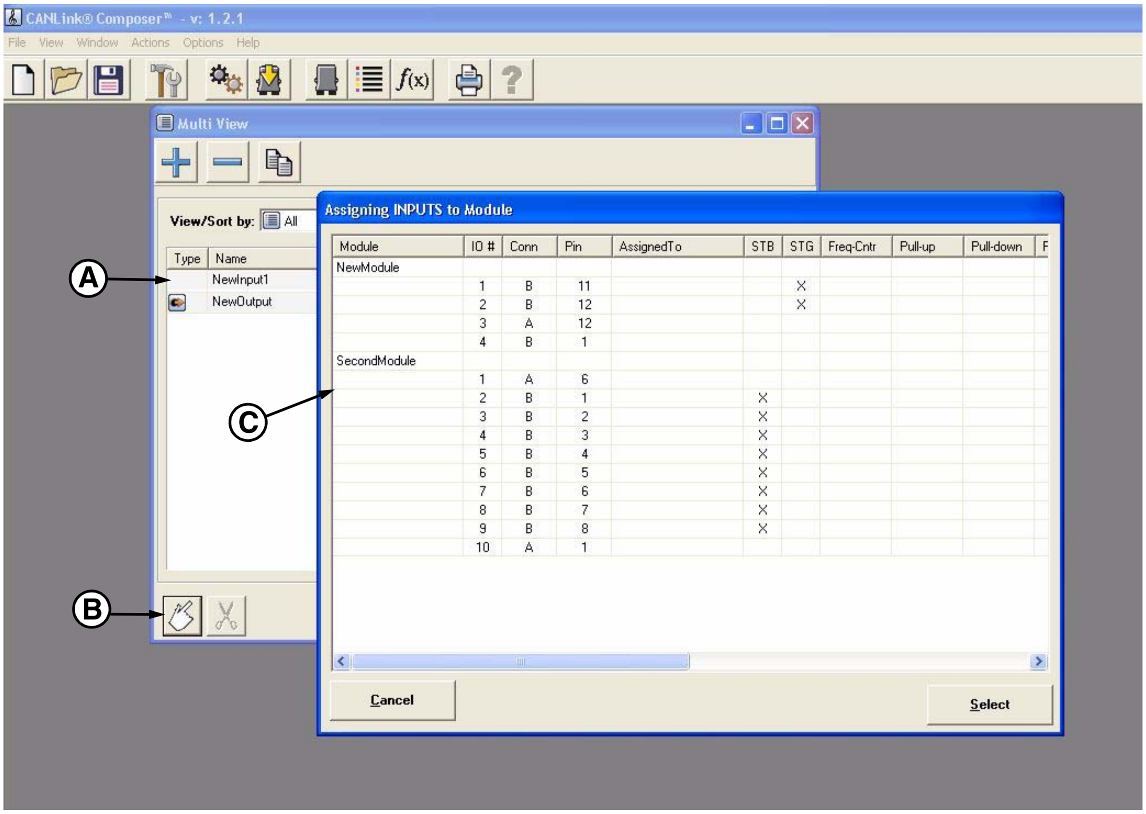
Dual CAN Check Box

Secondary Dropdown Menu

ASSIGNMENT WINDOW

Description

This matches the *Inputs* or *Outputs* of a module to a **Multi View** *Input* or *Output Data Item*. This window is accessed from the **Multi View** window by first highlighting the *Data Item* **(Figure 21, A)** to be assigned, then clicking the *Assign* button **(Figure 21, B)**. Double-click an *IO #* line item **(Figure 21, C)** to assign an *IO #* to the selected *Data Item*. The **Assignment** window will automatically close.


CANLink® Composer™ - v: 1.2.1
File View Window Actions Options Help
f(x)
Multi View
View/Sort by: All
Type Name
NewInput1
NewOutput
Assigning INPUTS to Module
Cancel
Select
A
B
C

Module	IO #	Conn	Pin	AssignedTo	STB	STG	Freq-Cntr	Pull-up	Pull-down	F
NewModule										
	1	B	11			X				
	2	B	12			X				
	3	A	12							
	4	B	1							
SecondModule										
	1	A	6							
	2	B	1		X					
	3	B	2		X					
	4	B	3		X					
	5	B	4		X					
	6	B	5		X					
	7	B	6		X					
	8	B	7		X					
	9	B	8		X					
	10	A	1							

Figure 21

Note: The example above shows Input assignment; however, the assignment window for Outputs is basically the same.

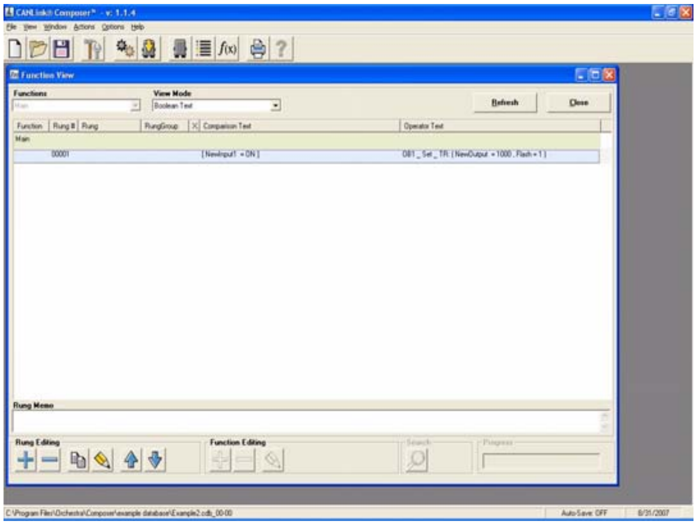
Function Blocks

OVERVIEW

Function Blocks determine how the application works, using a type of ladder logic. Each *Function Block* is made up of *Rungs*. Each *Rung* can have up to 5 *Logic Blocks* and 25 *Operator Blocks*.

The *Rungs* can be viewed as either text **(Figure 1: Function View)** or in a graphical form **(Figure 2: Rung Editor)**. Both views list all the information about the *Comparison Blocks* and the *Operator Blocks* in the *Rung*.

By default, there will be a main *Function Block* where the flow will start from. Additional functions can be called from the main or from any other function.



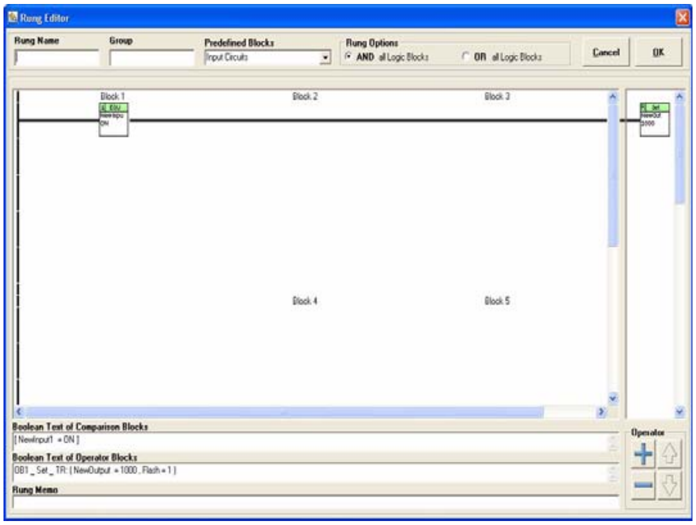
Functions: Main

View Mode: Boolean Text

Function	Rung #	Rung	RungGroup	X	Comparison Text	Operator Text
Main						
	00001				[NewInput1 = ON]	OB1 _ Set _ TR: [NewOutput = 1000 , Flash = 1]

C:\Program Files\Orchestra\Composer\example database\Example2.cdb_00-00 Auto-Save: OFF 8/31/2007

Figure 1: Function View



Rung Name:

Group:

Predefined Blocks: Input Circuits

Rung Options: ● AND all Logic Blocks ○ OR all Logic Blocks

Boolean Text of Comparison Blocks: [NewInput1 = ON]

Boolean Text of Operator Blocks: OB1 _ Set _ TR: [NewOutput = 1000 , Flash = 1]

Rung Memo:

Figure 2: Rung Editor

DEFINITIONS

Function Block

A *Function Block* is a collection of *Rungs*.

Rung

A *Rung* is a collection of 0-5 *Logic Blocks* and 1-25 *Operator Blocks*.

Logic Block

A *Logic Block* is a collection of 0-5 *Comparison Blocks*.

Comparison Block

A *Comparison Block* is one specific block inside a *Logic Block*. A *Comparison Block* maps the *Data Items* to specific decision-making controls. Each *Comparison Block* equates to a True/False result.

- A green block header means it is OK.
- A red block header means something is entered incorrectly in the block.
- A tan block header means it was just added.

Operator Block

An *Operator Block* maps the user-desired action into the *Data Items* via many controls.

- A green or blue block header means it is OK.
  - Green designates an action happens when the *Logic Blocks* equate true.
  - Blue designates an action happens when the *Logic Blocks* equate false.
- A red block header means something is entered incorrectly in the block.
- A tan block header means it was just added.

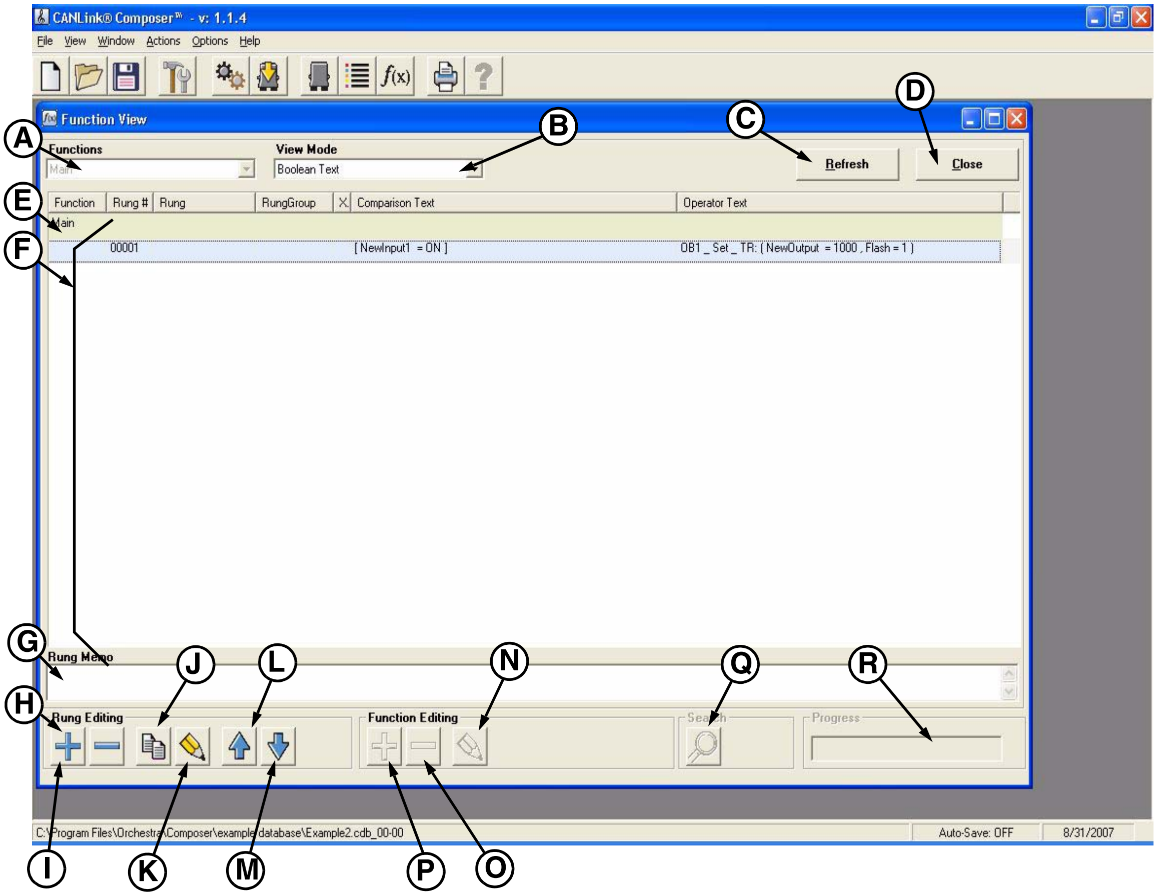
This Page Intentionally Left Blank

FUNCTION VIEW WINDOW

Description

The **Function View** window displays the *Rungs* in the *Function Blocks* and allows you to create, delete, and edit the *Function Blocks*.

This window is opened by clicking the f(x) *Function View* button in the **Main** window, or by selecting *View>Function View* from the **Program** Menu.



Functions: Main

View Mode: Boolean Text

Function	Rung #	Rung	RungGroup	X	Comparison Text	Operator Text
Main						
	00001				[NewInput1 = ON]	OB1 _ Set _ TR: (NewOutput = 1000 , Flash = 1)

C:\Program Files\Orchestra\Composer\example database\Example2.cdb_00-00 Auto-Save: OFF 8/31/2007

Figure 3

Controls

(A) Functions Dropdown Menu

This selects which *Function Block* is currently displayed in the *Rung List Box (F)*.

The options are:

- "Main" - This displays the main *Function Block Rungs*.
- "User-Created Names" - This displays the *Rungs* for the selected user-created *Function Blocks*.
- "All" - This displays all of the *Rungs* for all of the *Function Blocks*, showing a separation between blocks.

(B) View Mode Dropdown Menu

This selects how to view the *Rungs* in the *Rung List Box (F)*. After you have selected the type, it will also continue to show the currently selected type.

The options are:

- "Boolean Text" - This displays the *Rungs* in a text view using boolean text/logic.
- "Rung Memo" - This displays the descriptions that were entered for each *Rung*.

(C) Refresh Button

This will redraw the window to display updated information.

(D) Close Button

This will close the window and include the changes that were made.

(E) Function Description

This line displays the name of the current *Function Block*.

(F) List Box

This shows all the *Rungs* currently selected by the *Functions*. It will display them as selected by the *View Mode* dropdown menu (B). Double-clicking a *Rung* will open up the **Rung Editor** window.

Rungs can be sorted by clicking on any of the column headers. If the *Rungs* are sorted by any column other than *Rung #*, the *Rungs* can only be edited and no new *Rungs* can be created. Clicking the *Rung #* column will put them back in order, and allow *Rungs* to be deleted, moved, copied and new *Rungs* to be created.

Note: The Rung Editing right-click menu and buttons, except Rung Edit, are also disabled unless the Rungs are sorted by the Rung # column.

(G) Memo Text Box

This will display the description of the *Rung* when *View Mode>"Boolean Text"* is selected. This will display the *Boolean Text* information of the *Rung* when *View Mode >"Rung Memo"* is selected. This text is entered from the **Rung Editor** window. If a Function is selected, the *Memo* text box will display the description for that Function. This text is entered from the **Create/Edit Function** window.

(H-M) Rung Editing Box

(H) ➕ ADD Button

This will open the **Rung Editor** window in order to create a new *Rung* and add it to the end of the *Function Block*.

(I) ➖ DELETE Button

This button will delete the highlighted *Rung* and prompt you to verify the action. This has no "undo" function so be sure before clicking *Yes*.

(J) ⧉ COPY Button

This will create a copy of the currently highlighted *Rung*. After the new *Rung* is created, double-click the new *Rung* or click the *Edit* button to edit the data in the **Rung Editor** window.

(K) EDIT Button

This will open the currently selected *Rung* in the **Rung Editor** window and allow you to edit the current *Function Block*.

(L) MOVE UP Button

This will move the selected *Rung* up in the *List Box* and update the order in which *Rungs* are executed and saved.

(M) MOVE DOWN Button

This will move the selected *Rung* down in the *List Box* and update the order in which *Rungs* are executed and saved.

(N-P) Function Editing Box

(N) EDIT Button

This will open a **Create/Edit Function** window.

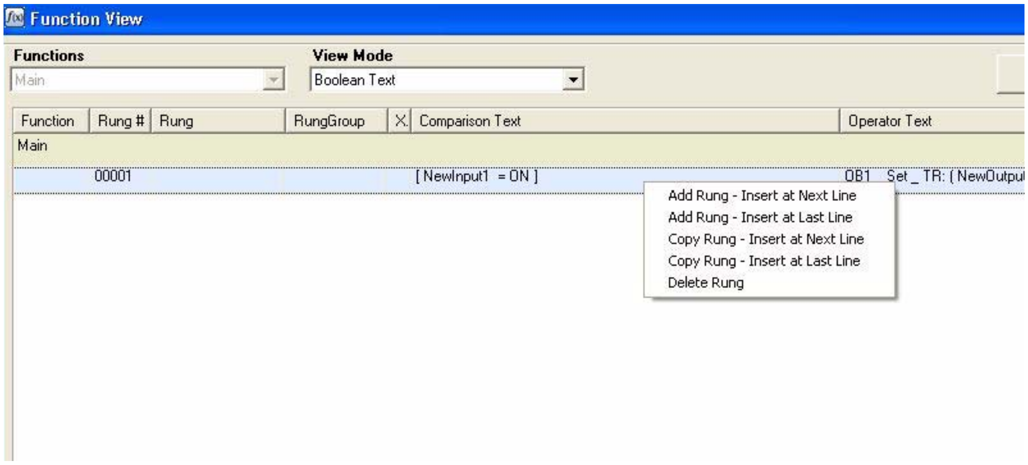
(O) DELETE Button

This will delete an entire function, including all its *Rungs*.

(P) ADD Button

This will open the **Create/Edit Function** window where a *Function* can be created.

Note: Right-clicking in the Function View window **(Figure 4)** *brings up a selectable menu for Adding, Copying and Deleting Rungs.*


Function View
Functions
Main
View Mode
Boolean Text
Add Rung - Insert at Next Line
Add Rung - Insert at Last Line
Copy Rung - Insert at Next Line
Copy Rung - Insert at Last Line
Delete Rung

Function	Rung #	Rung	RungGroup	X	Comparison Text	Operator Text
Main						
	00001				[NewInput1 = ON]	OB1 Set _ TR: (NewOutpu

Figure 4

Search Box

(Q) START Button

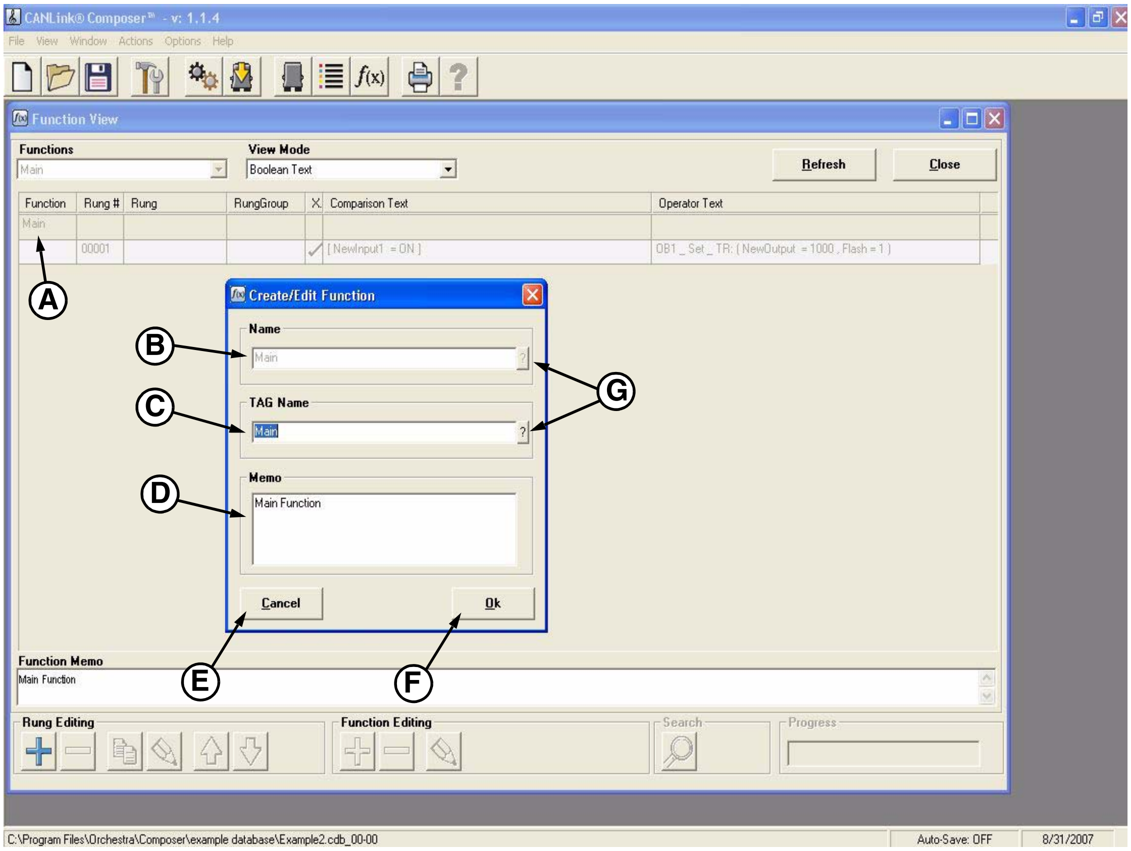
This will open the **Search** window and search the *Function Rungs* for the specified text.

(R) Progress Bar

This shows the progress of the operation being completed.

Create/Edit Function Window

This window is accessed by double-clicking on the *Function Description* **(Figure 5, A)** that you want to edit from **Function View**. It allows the *Name, Tag Name*, and *Memo* (description) of the function to be edited.



CANLink® Composer™ - v: 1.1.4

Function View

Functions: Main

View Mode: Boolean Text

Function	Rung #	Rung	RungGroup	X	Comparison Text	Operator Text
Main						
	00001			✓	[NewInput1 = ON]	OB1 _ Set _ TR: (NewOutput = 1000 , Flash = 1)

Create/Edit Function

Name: Main

TAG Name: Main

Memo: Main Function

Function Memo: Main Function

C:\Program Files\Orchestra\Composer\example database\Example2.cdb_00-00

Auto-Save: OFF

8/31/2007

Figure 5

Controls

(B) Name Text Box

Enter the *Name* that will be used to identify the *Function Block* in **Rung Editor, Function View** and in Conductor. It must be 1-20 characters in length and unique to all other *Function Block* names.

(C) TAG Name Text Box

This is a 1-7 character description that will be seen in the *Comparison* and *Operator Blocks* of the *Rungs*. It does not need to be unique. If nothing is entered here, Composer will automatically assign the first seven letters of the *Name*.

(D) Memo Text Box

Enter a short description of the *Function Block*. There is no limit to the number of characters used here.

(E) Cancel Button

This will cancel all changes and close the **Create/Edit Function** window.

(F) OK Button

When this button is clicked, the software will verify the *Name* is unique, make the changes, and close the window.

Warnings are non-fatal and will not cause the action to fail. Correction is optional. Click *OK* to continue or *Cancel* to change.

Errors are fatal and must be corrected. If the name is not unique, an **Invalid Entry** window will appear. Click *OK* and revise the *Name*.

To display a list of names or tags that are already in use, click the "?" **(Figure 5, G)** at the end of the text box.

SEARCH WINDOW

Description

This will search through the *Function Blocks* to find *Data Items*. This window can be open at the same time as the **Function View** window.

Controls

Search for Text Box

This will be either the data name that was selected after the link to check box was selected or the text that was entered by you.

Search for Link Check Box

Clicking this box will open the **Multi View** window to allow you to pick a name from the list.

Check all Functions Check Box

If this is checked, it will search through all of the functions. Otherwise it will only search the one that is currently being viewed.

Check Description Check Box

If this is checked, it will search through all *Rung* descriptions to find the search text.

Close Button

Returns control back to the **Function View** window. No action is taken.

Stop Search Button

This will stop the current search.

Find All Button

This will bring up a list of all the *Rungs* in the search results *List Box* that the *Data Item* was used in.

Find Next Button

This will bring up the next occurrence of a *Data Item*. This also updates the *Rung List Box* in the **Function View** window.

Find Previous Button

This will bring up the previous occurrence of the *Data Item*. This also updates the *Rung List Box* in the **Function View** window.

Search Results Window

Only displayed when the *Find All* button is pressed. This will display all of the *Rungs* that use the search criteria. It will display function *Name*, function *Rung* number, and the function description. If you double-click on a line item, it will bring it up in the **Function View** window.

Progress Bar

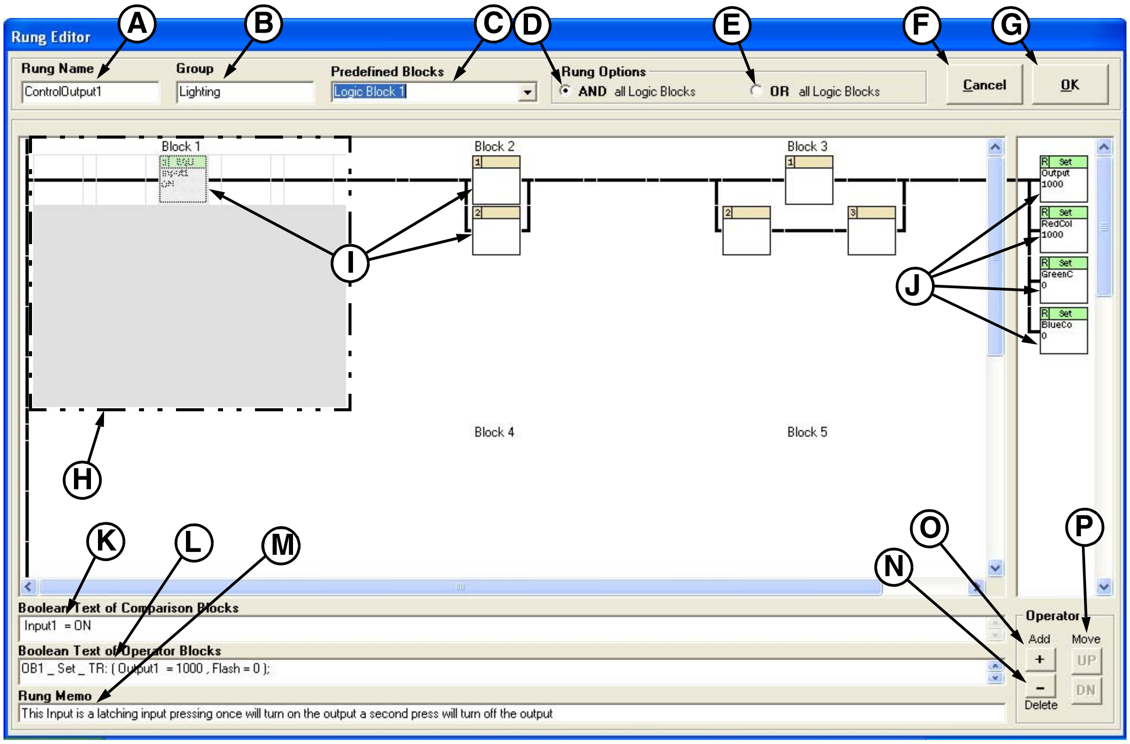
This will show the progress of the operation being completed.

RUNG EDITOR WINDOW

Description

This is the window where you define what the *Rung* will do. Available are five *Logic Blocks* (displayed in the left window), each containing up to five *Comparison Blocks* in various orders based on the *Predefined Blocks* chosen (*see Logic Blocks on page 79 for more information*), and 25 *Operator Blocks* (displayed in the right window).

This window is accessed by double-clicking a *Rung* from within **Function View** window, or clicking *+ ADD* or *EDIT* from the **Function View** window.



Rung Editor

Rung Name: ControlOutput1

Group: Lighting

Predefined Blocks: Logic Block 1

Rung Options: ● AND all Logic Blocks ○ OR all Logic Blocks

Boolean Text of Comparison Blocks: Input1 = ON

Boolean Text of Operator Blocks: OB1 _ Set _ TR: (Output1 = 1000 , Flash = 0);

Rung Memo: This Input is a latching input pressing once will turn on the output a second press will turn off the output

Figure 6

Controls

(A) Rung Name Text Box

This is a description of the *Rung* and will be used in the **Function View** window and Conductor.

(B) Group Text Box

Rungs can be grouped together and are used for sorting in Conductor or **Function View** window. *Rungs* can be grouped in one *Group*. The *Group* name must be 20 characters or less.

(C) Predefined Blocks Text Box

This is a list of block *Types* that can be used in the currently selected *Logic Block (H)*. When you select a block *Type*, it will update the currently selected *Logic Block* box in the **Rung Editor** window. *See Logic Blocks on page 79 for more information*.

(D and E) Rung Options Box

This determines how the *Logic Blocks* are used.

(D) AND Radio Button

When this option is chosen, the *Logic Blocks* will be displayed across the screen indicating that each *Logic Block* must be true for the entire *Rung* to be true.

(E) OR Radio Button

When this option is chosen, the *Logic Blocks* will be displayed down the screen indicating that if any of the *Logic Blocks* is true, the entire *Rung* is true.

(F) Cancel Button

This will cancel all changes and close the window.

(G) OK Button

When this button is clicked, the software will make the changes and close the window.

Warnings are non-fatal and will not cause the action to fail. Correction is optional. Click *OK* to continue or *Cancel* to change.

Errors are fatal and must be corrected.

(H) Logic Block

A *Logic Block* contains up to five *Comparison Blocks* that define how the *Logic Block* carries out its task. *Logic Blocks* should be defined from left to right and top to bottom.

Shown is the location for the first *Logic Block*. If the *AND* logic option is chosen, the remaining blocks will be to the right (as shown in ***Figure 6***). If the *OR* logic option is chosen, the remaining blocks will be shown below the first one.

(I) Comparison Block

A *Comparison Block* is a collection of properties that will compare one *Data Item* to a value or another *Data Item*. The compare must result in a True/False answer.

A new *Comparison Block* will have a tan header. A correctly defined *Comparison Block* will have a green header. If a *Comparison Block* is not defined, or incorrectly defined, its header will be red.

Comparison Blocks contain the formulas that will return a True (1) or False (0) value, determining in conjunction with the other *Comparison Block*s in the *Rung* if the *Rung* is true or false.

Single-clicking on a *Comparison Block* will highlight it and show the properties in the *Text Boxes (K, L, M)* at the bottom of the window. Double-clicking on a *Comparison Block* box will open the **Edit/Create Comparison Block** window allowing the logic criteria to be entered or edited. *See Edit/Create Comparison Block Window on page 62 for more information.*

(J) Operator Blocks

An *Operator Block* is a collection of properties that will perform an operation on a *Data Item*.

A new *Operator Block* will have a tan header. A correctly defined *Operator Block* that runs when the *Rung* is true will have a green block header. A correctly defined *Operator Block* that runs when the *Rung* is false will have a blue block header. An *Operator Block* that is not defined, or incorrectly defined, will have a red header.

This window defines what should happen based on the result acquired from the *Comparison Block*.

Single-clicking an *Operator Block* will highlight it and show the properties in the *Text Boxes (K, L, M)* at the bottom of the **Rung Editor** window. Double-clicking an *Operator Block* will open the **Edit/Create Operator Block** window allowing the logic criteria to be entered or edited. *See Edit/Create Operator Block Window on page 64 for more information.*

(K) Boolean Text of Comparison Blocks Text Box

This displays the values defined in the selected *Comparison Block*.

(L) Boolean Text of Operator Blocks Text Box

This displays the values defined in the selected *Operator Block*.

(M) Rung Memo Text Box

Enter a description of the entire *Rung*. This will also be displayed in the **Function View** window.

(N-P) Operator Box

(N) Delete

This will delete the selected *Operator Block*.

(O) Add

This will add a new *Operator Block* to the bottom of the list of *Operator Blocks*.

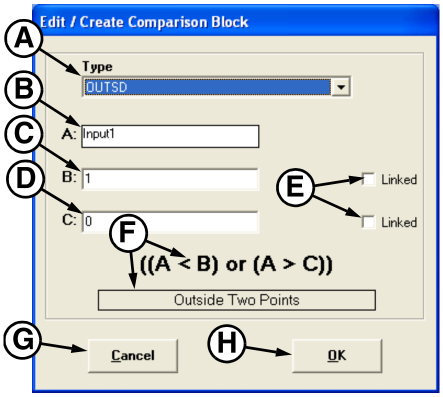
(P) Move

This will move the selected *Operator Block* up or down to adjust the order of execution.

Edit/Create Comparison Block Window

Description

This window is accessed by double-clicking on a *Comparison Block* in the **Rung Editor** window. It is used to define the *Comparison Block* properties.



Edit / Create Comparison Block

Type: OUTSD

A: Input1

B: 1 □ Linked

C: 0 □ Linked

((A < B) or (A > C))

Outside Two Points

Figure 7

Controls

(A) Type Dropdown Menu

Choose the *Type* of comparison formula from this menu. The Boolean equation of the chosen *Type* will be displayed in area (F) above the *Cancel* and *OK* buttons.

The choices are:

Type	Boolean	Description
EQU	(A == B)	A equals B
NEQU	(A <> B)	A does not equal B
LT	(A < B)	A is less than B
LTE	(A <= B)	A is less than or equal to B
GT	(A > B)	A is greater than B
GTE	(A >= B)	A is greater than or equal to B
BTWN	(B < A < C)	A is between two points
BTEQU	(B <= A <= C)	A is between or equal to two points
OUTSD	((A < B) or (A > C))	A is outside two points
OSEQU	((A <= B) or (A >= C))	A is outside or equal to two points
BITCmp	((A & B) = C))	Bit compare

(B) Text Box A:

This box displays the name of the *Data Item* assigned to the function of the selected *Comparison Block*. To enter or edit this function, double-click the text box.

(C) Text Box B:

This is a user-defined value that is either entered or chosen from a dropdown menu, depending on the *Type* of comparison formula chosen. This can also be linked to another *Data Item* by checking box (E).

(D) Text Box C:

This is a user-defined value that is either entered or chosen from a dropdown menu, depending on the *Type* of comparison formula chosen. This may also be linked to another *Data Item* by checking box (E).

(E) Linked Check Box

This enables the values to be assigned to a *Data Item*. By default it is unchecked. When the box is checked it will clear out the value in the text box or show a previously linked *Data Item*. Double-click the text box to open the **Multi View** window. The *Data Item* name selected will show up in the text box.

(F) Formula Description

This displays the Boolean equation of the comparison formula *Type*.

(G) Cancel Button

This will cancel all changes and close the window.

(H) OK Button

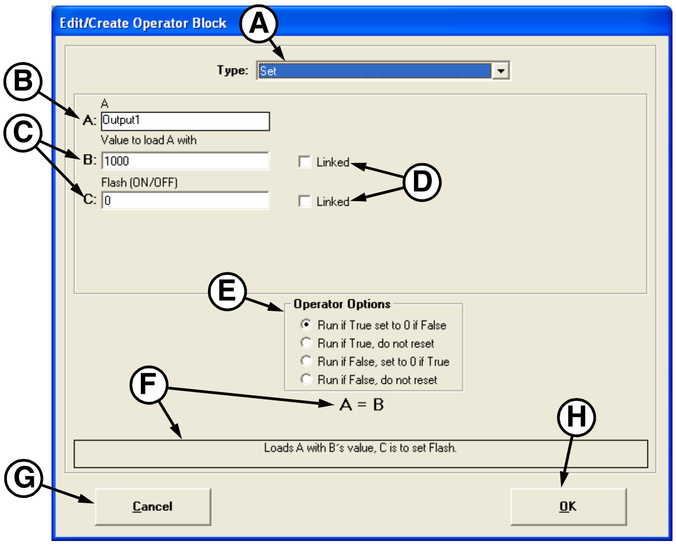
This will assign new or changed data to the *Comparison Block* and close the window.

Edit/Create Operator Block Window

Description

This window is accessed by double-clicking on an *Operator Block* in the **Rung Editor** window. It is used to define the *Operator Block* properties.

Operator Block windows differ greatly depending on the *Type* chosen. Shown is one *Type* of **Operator Block** window. *See Operator Block Window Types on page 67* for a complete list of examples.


Edit/Create Operator Block

Type: Set

A
A: Output1

Value to load A with
B: 1000 □ Linked

Flash (ON/OFF)
C: 0 □ Linked

Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = B

Loads A with B's value, C is to set Flash.

Figure 8

Controls

(A) Type Dropdown Menu

This defines what action the *Operator Block* should carry out.

The choices are:

Type	Boolean	Description
Set	A=B	Loads A with B's value, C is to set Flash
Dec	A=A-1	This will decrement the counter (A) by 1. The counter will not decrement again until the Rung turns False then True.
Inc	A=A+1	This will increment the counter (A) by 1. The counter will not increment again until the Rung turns False then True.
Percnt	A=(High Value, Low Value, Percent)	Calculates the percentage and stores it into A.
Add	A=B+C	Adds B to C and stores it into A.
Sub	A=B-C	Subtracts C from B and stores it into A.
Mult	A=BxC	Multiplies B and C and stores it into A.
Div	A=B/C	Divides B by C and stores it into A.
PID > T	A=(Input, Input Target, Input Deadband, Output Threshold, Output Max, P Gain, I Gain, D Gain)	Does a PID control on the Output.

Type	Boolean	Description
PID < T	A=(Input, Input Target, Input Deadband, Output Threshold, Output Max, P Gain, I Gain, D Gain)	Does a PID control on the Output.
PIDspd	A=(Input, Input Target, Input Deadband, Output Threshold, Output Max, P Gain, I Gain, D Gain)	Does a PID control on the Output.
Ramp	A=(Start Value, End Value, Ramp Time)	Loads A with B's value and will ramp up to C over D amount of time.
JOYabv	A=(Input, Input Center, Input Deadband, Input Max, Max+, Output Threshold, Output Max, Output Scale)	Converts the joystick or A/D Input value into an Output command. If the Input is greater than the center+deadband then the result is 0.
JOYblw	A=(((Center-Deadband)-Input) /((Center-Deadband)-Input Min))*(Output Max-Output Threshold)*(Output Scale/100))+Output)	Converts the joystick or A/D Input value into an Output command. If the Input is less than the center+deadband then the result is 0.
SETbit	A= A I B	Sets the bit(s) in A that are set in B.
CLRbit	A=A&B	Clears the bit(s) in A that are not set in B.
Lshift	A=B<<C	Left-shifts the value in A by B times.
Rshift	A=B>>C	Right-shifts the value in A by B times.
5 Volt	5-Volt supply	Turns the module 5-volt supply ON or OFF and sets the pull-up resistor to toggle.
CAN C	Send custom CAN message.	
CAN J	Send J1939 CAN message.	
Sort	Sort	Sort up to four numbers/items (low-to-high or high-to-low)
Wt AVG	Weighted AVG	Weighted AVG (item1*weight1+item2*weight2+item3*weight3+item4*weight 4)/Sum(weights)

Note: Items (B) through (F) will vary depending on the operation Type chosen (A). Each Variable that requires Input will be defined with a short description near the Input field.

(B) Text Box A:

This indicates the *Data Item* belonging to *Variable* A.

(C) Text Boxes

Enter *Variable* values in these. The quantity of text boxes will vary depending on the operation *Type* chosen.

(D) Linked Check Boxes

This enables the values to be assigned to a *Data Item*. By default it is unchecked. When the box is checked it will open the **Multi View** window where a *Data Item* can be selected.

(E) Operator Options

Check the appropriate button that will cause the operator to execute the assigned task.

(F) Formula Description

This displays the actual formula and a description of the formula action.

(G) Cancel Button

This will cancel all changes and close the window.

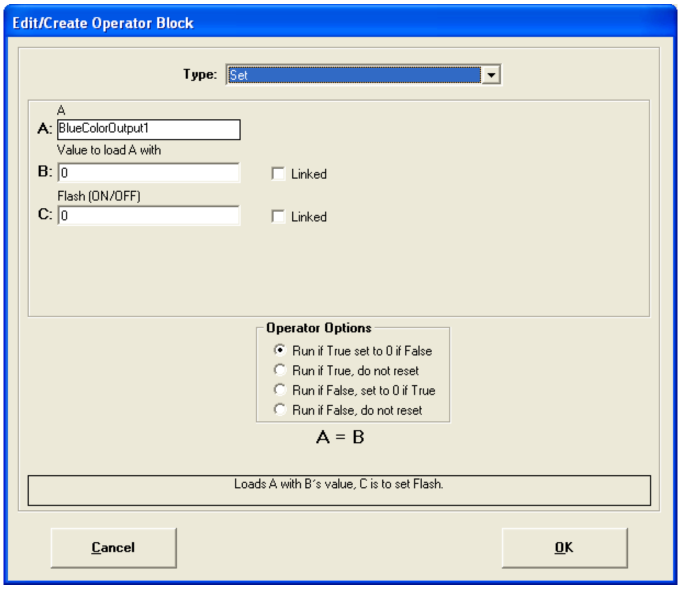
(H) OK Button

This will save the changes made and close the window.

Operator Blocks

OPERATOR BLOCK WINDOW TYPES

Set



Type: Set

A
A: BlueColorOutput1

Value to load A with
B: 0 □ Linked

Flash (ON/OFF)
C: 0 □ Linked

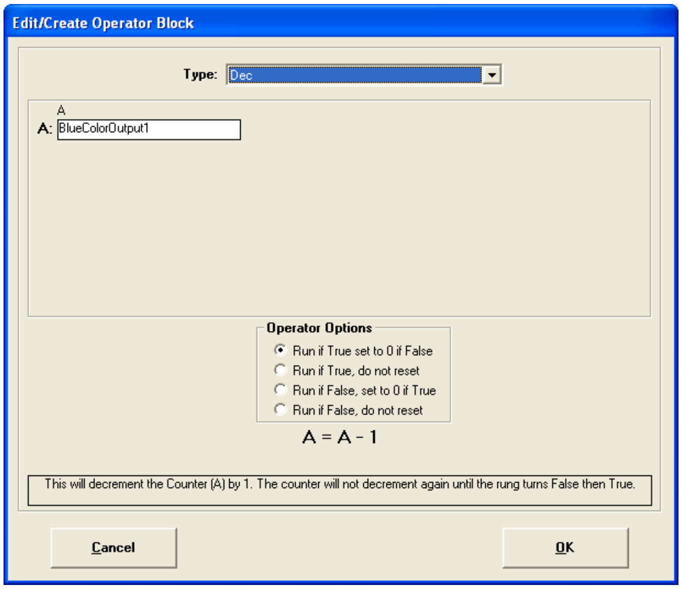
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = B

Loads A with B's value, C is to set Flash.

Figure 9

Dec



Type: Dec

A
A: BlueColorOutput1

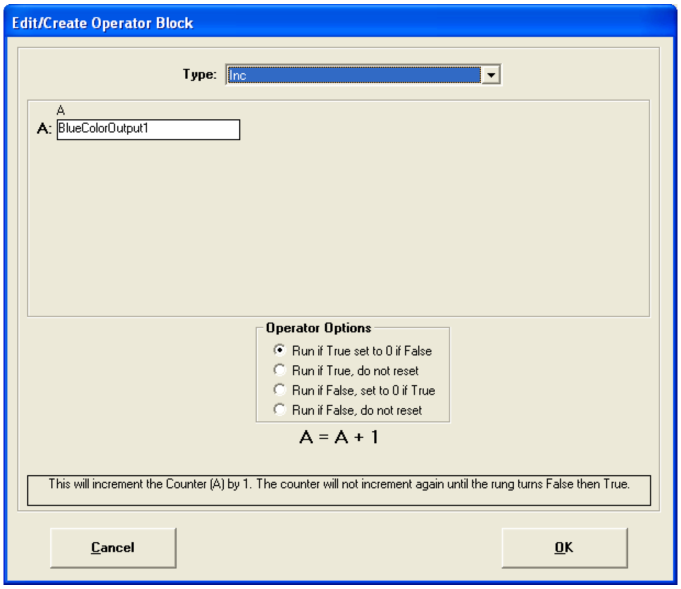
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = A - 1

This will decrement the Counter (A) by 1. The counter will not decrement again until the rung turns False then True.

Figure 10

Inc



Edit/Create Operator Block

Type: Inc

A
A: BlueColorOutput1

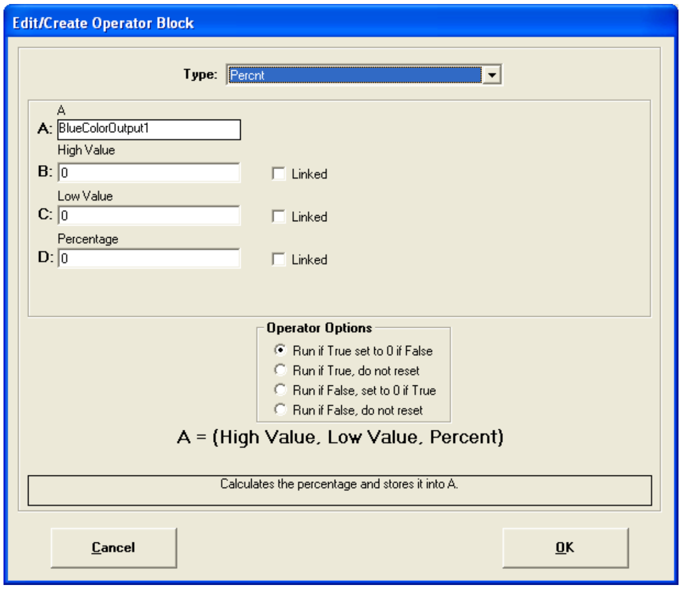
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = A + 1

This will increment the Counter (A) by 1. The counter will not increment again until the rung turns False then True.

Figure 11

Percnt



Edit/Create Operator Block

Type: Percnt

A
A: BlueColorOutput1

High Value
B: 0 □ Linked

Low Value
C: 0 □ Linked

Percentage
D: 0 □ Linked

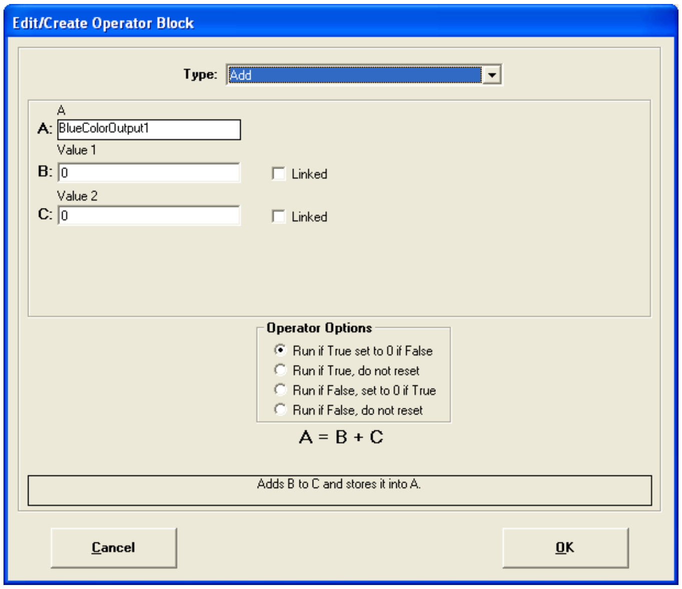
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = (High Value, Low Value, Percent)

Calculates the percentage and stores it into A.

Figure 12

Add



Type: Add

A
A: BlueColorOutput1

Value 1
B: 0 ☐ Linked

Value 2
C: 0 ☐ Linked

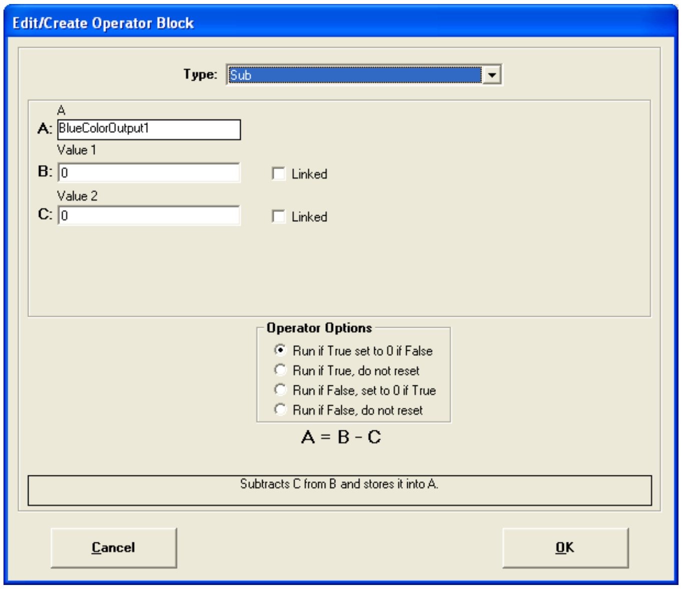
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = B + C

Adds B to C and stores it into A.

Figure 13

Sub



Type: Sub

A
A: BlueColorOutput1

Value 1
B: 0 ☐ Linked

Value 2
C: 0 ☐ Linked

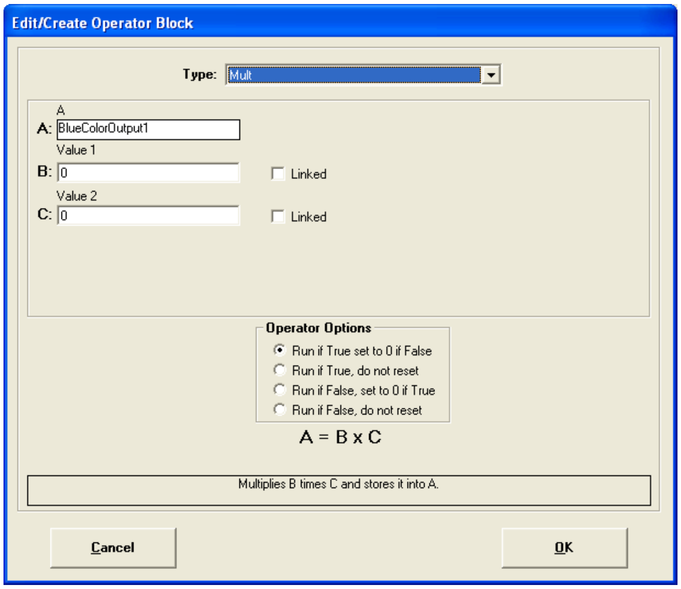
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = B - C

Subtracts C from B and stores it into A.

Figure 14

Mult



Edit/Create Operator Block

Type: Mult

A

A: BlueColorOutput1

Value 1

B: 0 □ Linked

Value 2

C: 0 □ Linked

Operator Options

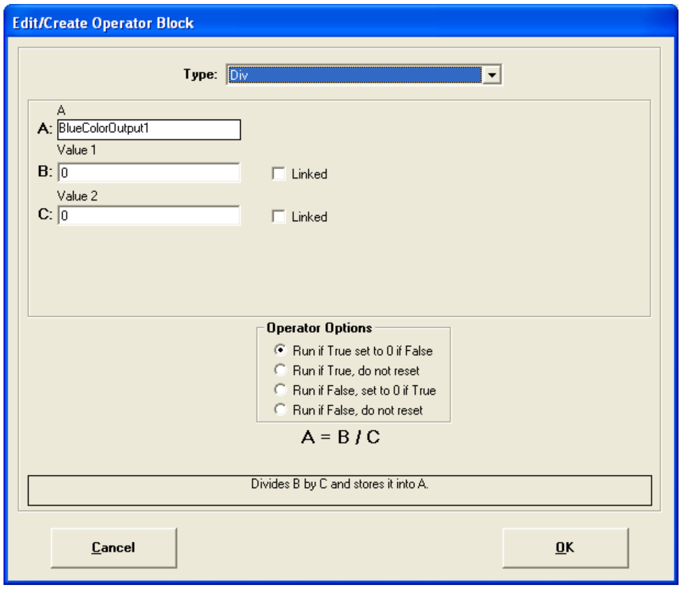
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = B x C

Multiplies B times C and stores it into A.

Figure 15

Div



Edit/Create Operator Block

Type: Div

A

A: BlueColorOutput1

Value 1

B: 0 □ Linked

Value 2

C: 0 □ Linked

Operator Options

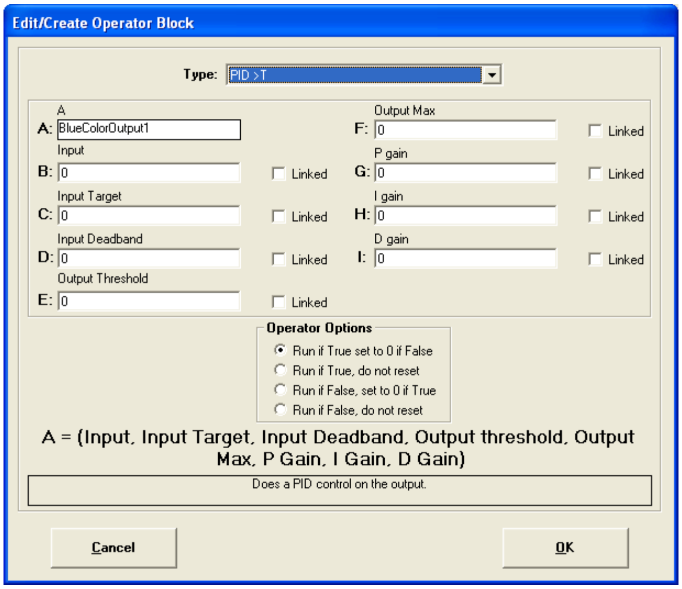
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = B / C

Divides B by C and stores it into A.

Figure 16

PID > T



Edit/Create Operator Block

Type: PID >T

A
A: BlueColorOutput1

Input
B: 0 □ Linked

Input Target
C: 0 □ Linked

Input Deadband
D: 0 □ Linked

Output Threshold
E: 0 □ Linked

Output Max
F: 0 □ Linked

P gain
G: 0 □ Linked

I gain
H: 0 □ Linked

D gain
I: 0 □ Linked

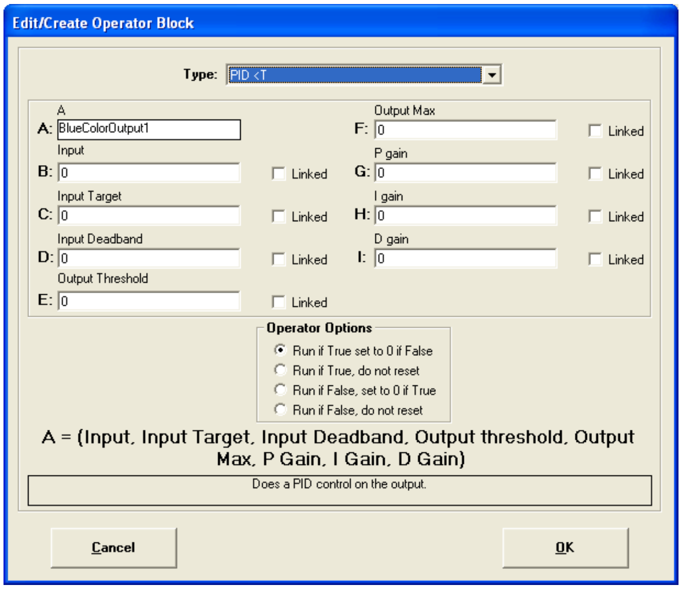
Operator Options
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = (Input, Input Target, Input Deadband, Output threshold, Output Max, P Gain, I Gain, D Gain)

Does a PID control on the output.

Figure 17

PID < T



Edit/Create Operator Block

Type: PID <T

A
A: BlueColorOutput1

Input
B: 0 □ Linked

Input Target
C: 0 □ Linked

Input Deadband
D: 0 □ Linked

Output Threshold
E: 0 □ Linked

Output Max
F: 0 □ Linked

P gain
G: 0 □ Linked

I gain
H: 0 □ Linked

D gain
I: 0 □ Linked

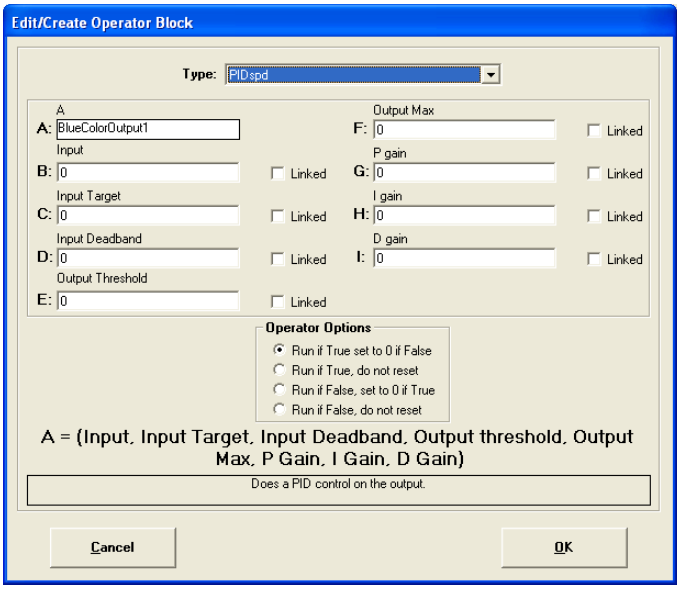
Operator Options
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = (Input, Input Target, Input Deadband, Output threshold, Output Max, P Gain, I Gain, D Gain)

Does a PID control on the output.

Figure 18

PIDspd



Edit/Create Operator Block

Type: PIDspd

A
A: BlueColorOutput1

Input
B: 0 □ Linked

Input Target
C: 0 □ Linked

Input Deadband
D: 0 □ Linked

Output Threshold
E: 0 □ Linked

Output Max
F: 0 □ Linked

P gain
G: 0 □ Linked

I gain
H: 0 □ Linked

D gain
I: 0 □ Linked

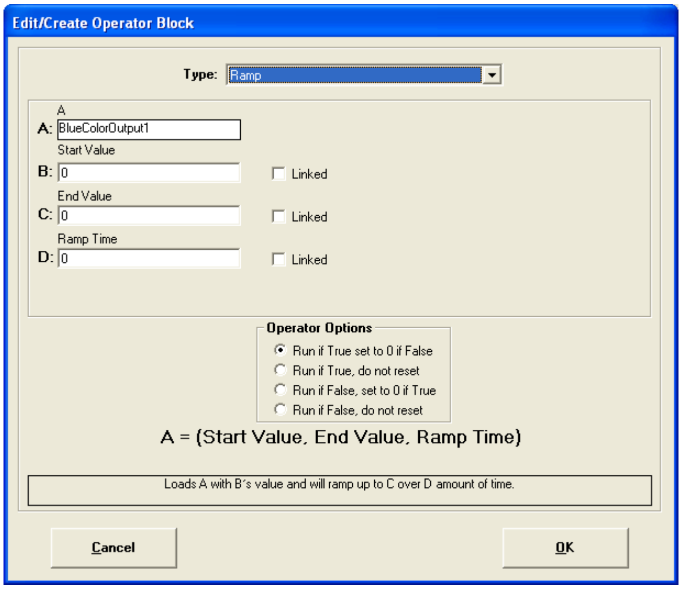
Operator Options
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = (Input, Input Target, Input Deadband, Output threshold, Output Max, P Gain, I Gain, D Gain)

Does a PID control on the output.

Figure 19

Ramp



Edit/Create Operator Block

Type: Ramp

A
A: BlueColorOutput1

Start Value
B: 0 □ Linked

End Value
C: 0 □ Linked

Ramp Time
D: 0 □ Linked

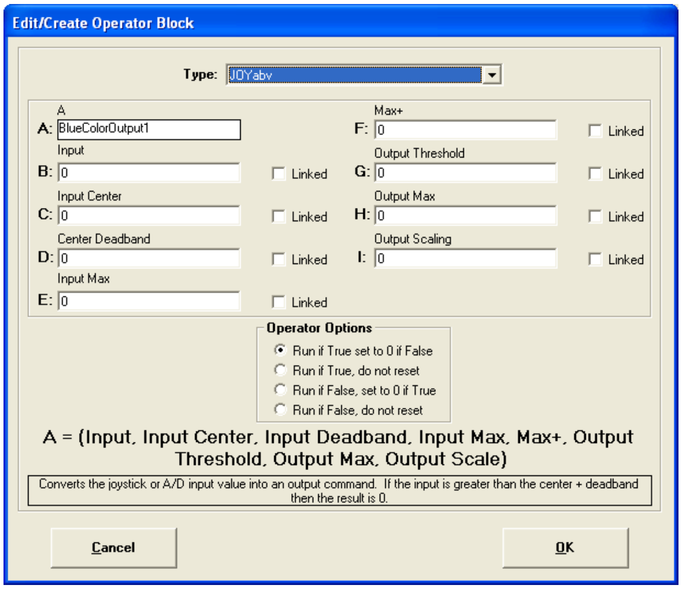
Operator Options
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = (Start Value, End Value, Ramp Time)

Loads A with B's value and will ramp up to C over D amount of time.

Figure 20

JOYabv



Edit/Create Operator Block

Type: JOYabv

A: BlueColorOutput1 (A)

B: 0 (Input) □ Linked

C: 0 (Input Center) □ Linked

D: 0 (Center Deadband) □ Linked

E: 0 (Input Max) □ Linked

F: 0 (Max+) □ Linked

G: 0 (Output Threshold) □ Linked

H: 0 (Output Max) □ Linked

I: 0 (Output Scaling) □ Linked

Operator Options

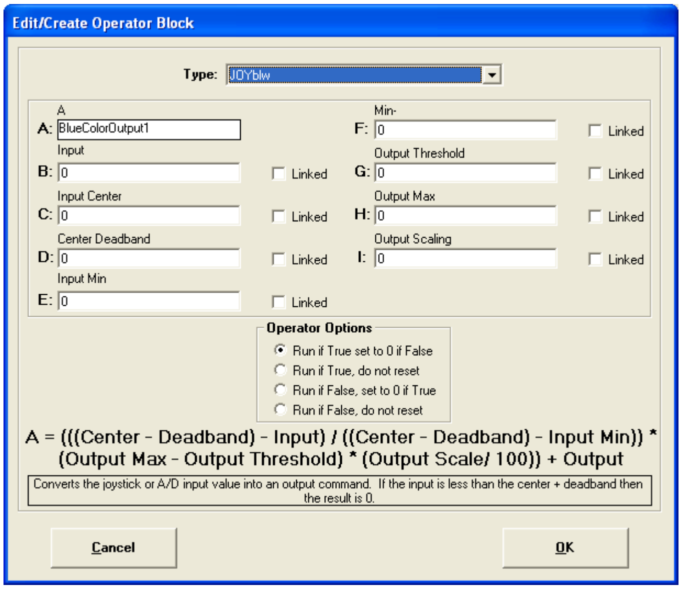
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = (Input, Input Center, Input Deadband, Input Max, Max+, Output Threshold, Output Max, Output Scale)

Converts the joystick or A/D input value into an output command. If the input is greater than the center + deadband then the result is 0.

Figure 21

JOYblw



Edit/Create Operator Block

Type: JOYblw

A: BlueColorOutput1 (A)

B: 0 (Input) □ Linked

C: 0 (Input Center) □ Linked

D: 0 (Center Deadband) □ Linked

E: 0 (Input Min) □ Linked

F: 0 (Min-) □ Linked

G: 0 (Output Threshold) □ Linked

H: 0 (Output Max) □ Linked

I: 0 (Output Scaling) □ Linked

Operator Options

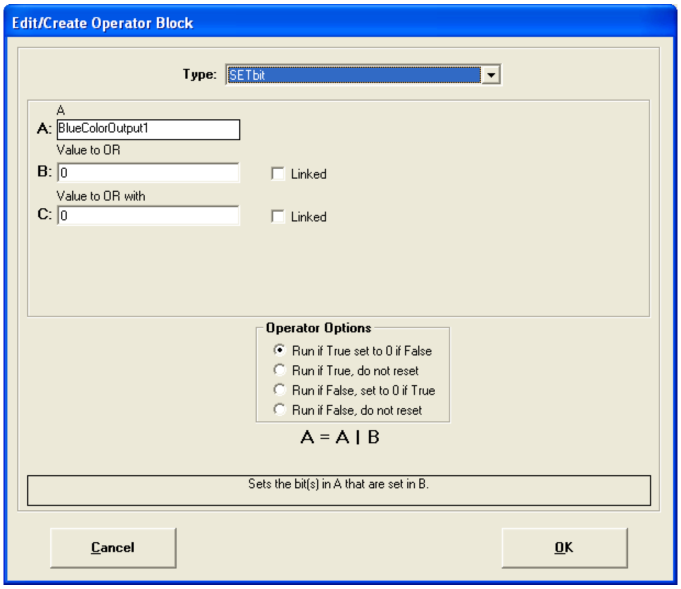
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = (((Center - Deadband) - Input) / ((Center - Deadband) - Input Min)) * (Output Max - Output Threshold) * (Output Scale/ 100)) + Output

Converts the joystick or A/D input value into an output command. If the input is less than the center + deadband then the result is 0.

Figure 22

SETbit



Edit/Create Operator Block

Type: SETbit

A

A: BlueColorOutput1

Value to OR

B: 0 □ Linked

Value to OR with

C: 0 □ Linked

Operator Options

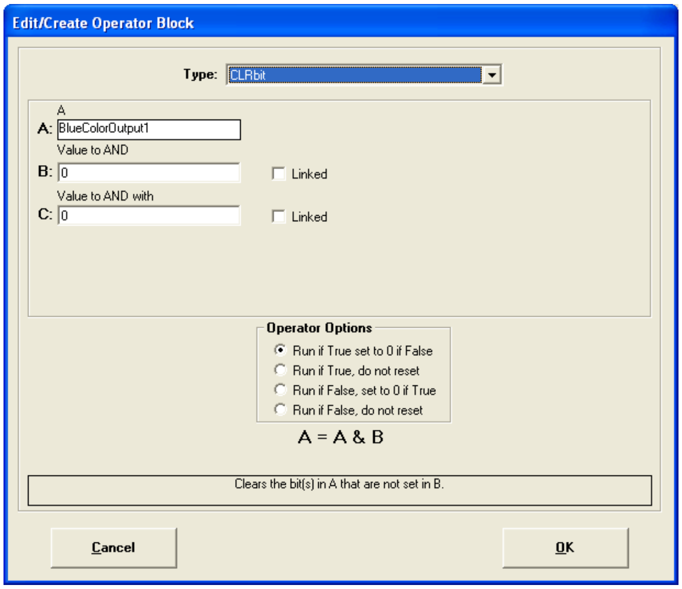
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = A | B

Sets the bit(s) in A that are set in B.

Figure 23

CLRbit



Edit/Create Operator Block

Type: CLRbit

A

A: BlueColorOutput1

Value to AND

B: 0 □ Linked

Value to AND with

C: 0 □ Linked

Operator Options

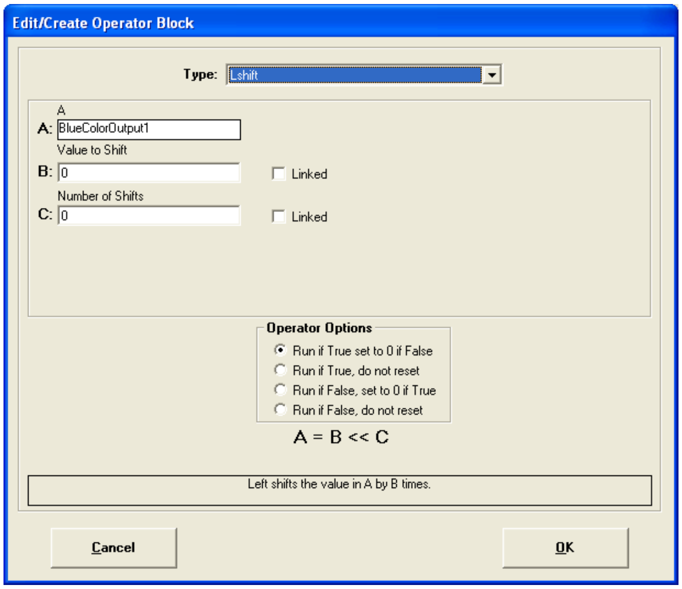
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

A = A & B

Clears the bit(s) in A that are not set in B.

Figure 24

Lshift



Edit/Create Operator Block

Type: Lshift

A
A: BlueColorOutput1

Value to Shift
B: 0 □ Linked

Number of Shifts
C: 0 □ Linked

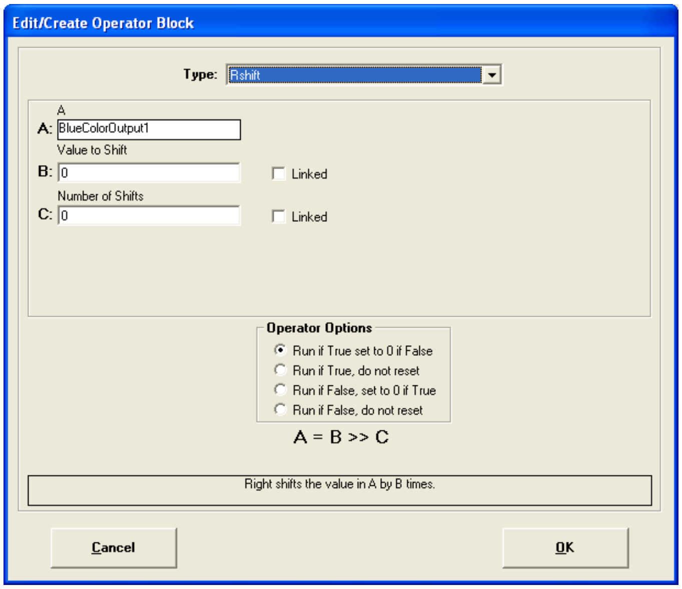
Operator Options
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = B << C

Left shifts the value in A by B times.

Figure 25

Rshift



Edit/Create Operator Block

Type: Rshift

A
A: BlueColorOutput1

Value to Shift
B: 0 □ Linked

Number of Shifts
C: 0 □ Linked

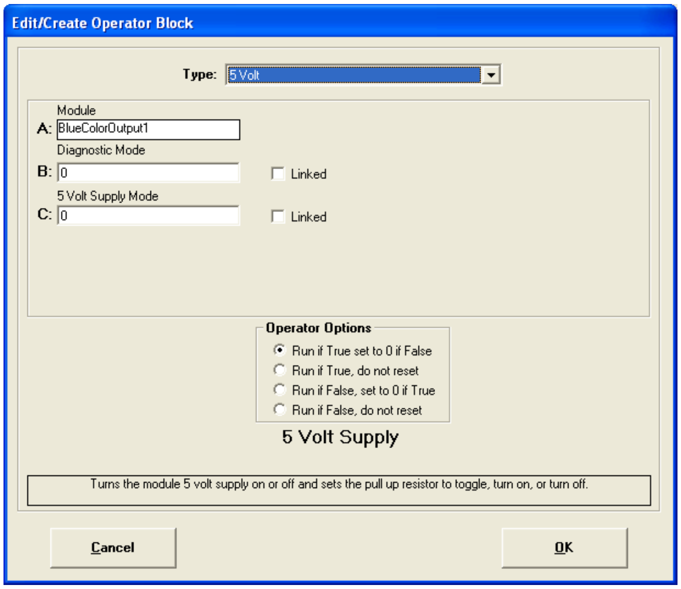
Operator Options
● Run if True set to 0 if False
○ Run if True, do not reset
○ Run if False, set to 0 if True
○ Run if False, do not reset

A = B >> C

Right shifts the value in A by B times.

Figure 26

5 Volt



Type: 5 Volt

Module
A: BlueColorOutput1

Diagnostic Mode
B: 0 □ Linked

5 Volt Supply Mode
C: 0 □ Linked

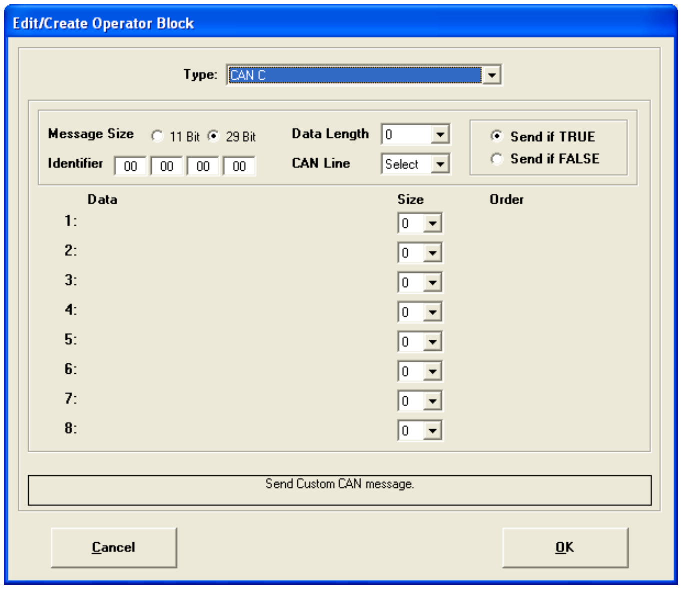
Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

5 Volt Supply

Turns the module 5 volt supply on or off and sets the pull up resistor to toggle, turn on, or turn off.

Figure 27

CAN C



Type: CAN C

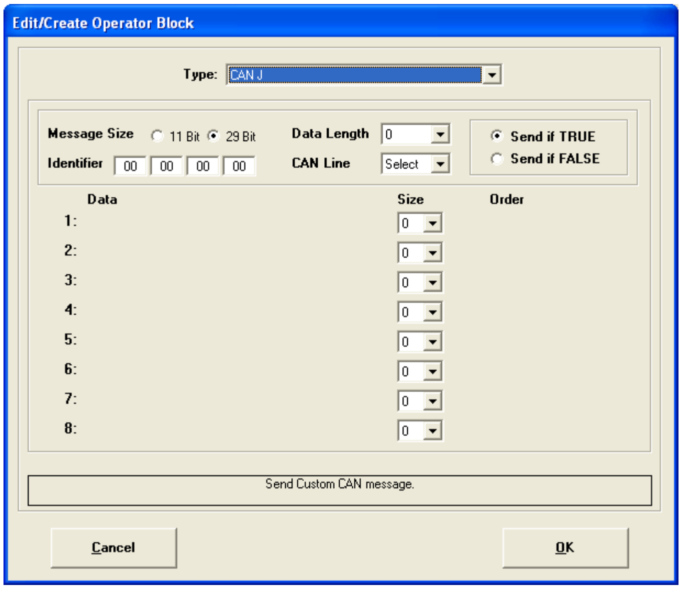
Message Size ○ 11 Bit ● 29 Bit   Data Length 0   ● Send if TRUE
Identifier 00 00 00 00   CAN Line Select   ○ Send if FALSE

	Data	Size	Order
1:		0	
2:		0	
3:		0	
4:		0	
5:		0	
6:		0	
7:		0	
8:		0	

Send Custom CAN message.

Figure 28

CAN J



Type: CAN J

Message Size ○ 11 Bit ● 29 Bit

Identifier 00 00 00 00

Data Length 0

CAN Line Select

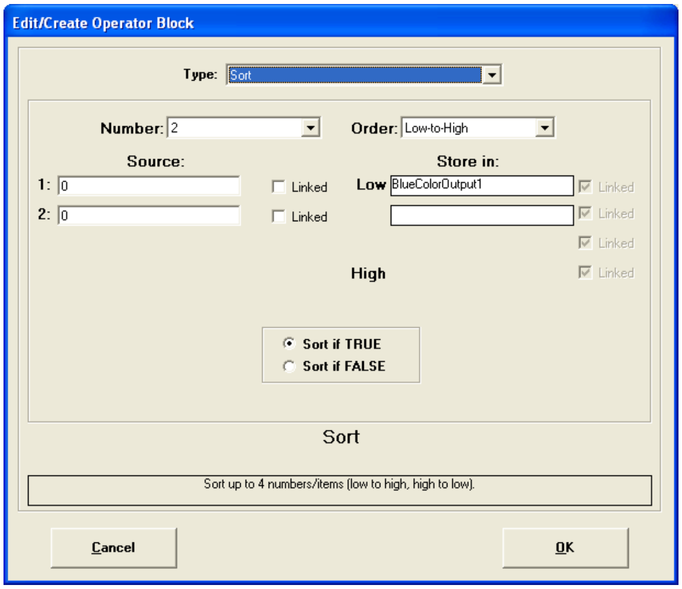
● Send if TRUE
○ Send if FALSE

Data	Size
1:	0
2:	0
3:	0
4:	0
5:	0
6:	0
7:	0
8:	0

Send Custom CAN message.

Figure 29

Sort



Type: Sort

Number: 2

Order: Low-to-High

1: 0 □ Linked

2: 0 □ Linked

Low BlueColorOutput1 ☑ Linked

☑ Linked

☑ Linked

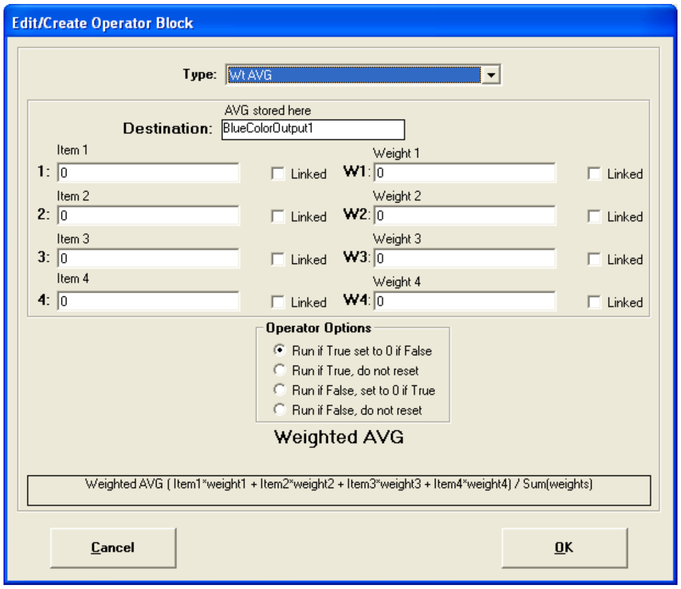
High ☑ Linked

● Sort if TRUE
○ Sort if FALSE

Sort up to 4 numbers/items (low to high, high to low).

Figure 30

Wt AVG



Edit/Create Operator Block

Type: Wt AVG

AVG stored here

Destination: BlueColorOutput1

Item 1
1: 0 □ Linked
Weight 1
W1: 0 □ Linked

Item 2
2: 0 □ Linked
Weight 2
W2: 0 □ Linked

Item 3
3: 0 □ Linked
Weight 3
W3: 0 □ Linked

Item 4
4: 0 □ Linked
Weight 4
W4: 0 □ Linked

Operator Options
- ● Run if True set to 0 if False
- ○ Run if True, do not reset
- ○ Run if False, set to 0 if True
- ○ Run if False, do not reset

Weighted AVG

Weighted AVG (Item1*weight1 + Item2*weight2 + Item3*weight3 + Item4*weight4) / Sum(weights)

Figure 31

Logic Blocks

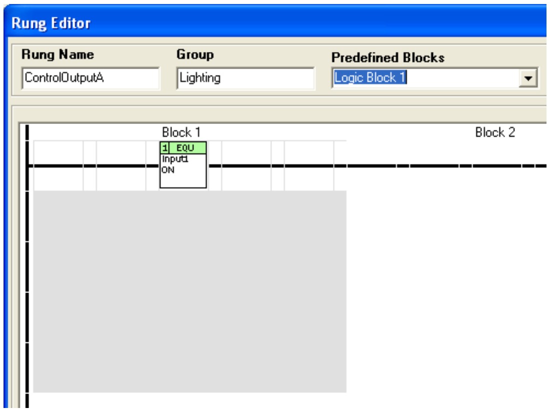
Logic Blocks will be displayed in varying numbers and varying order depending on the *Logic Block Type* chosen in *Predefined Blocks* dropdown menu in the **Rung Editor** window. Following is a graphic representation of each.

LOGIC BLOCK EXAMPLES

Logic Block 1

If A is true, then block is true.

Boolean: (A)

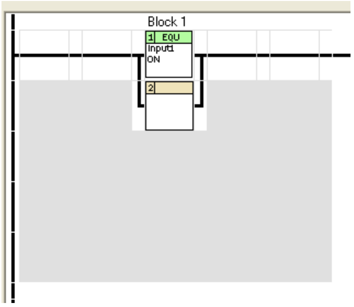


Rung Name: ControlOutputA
Group: Lighting
Predefined Blocks: Logic Block 1

Logic Block 2

If A is true or if B is true, then block is true.

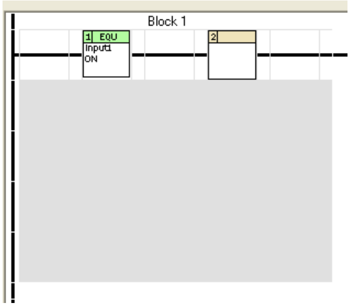
Boolean: (A) OR (B)



Logic Block 3

If A and B are true, then block is true.

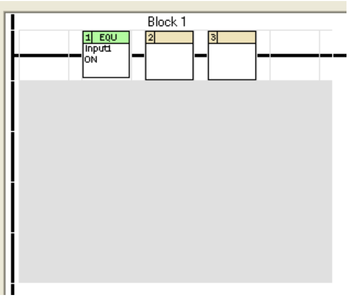
Boolean: (A) AND (B)



Logic Block 4

If A, B and C are true, then block is true.

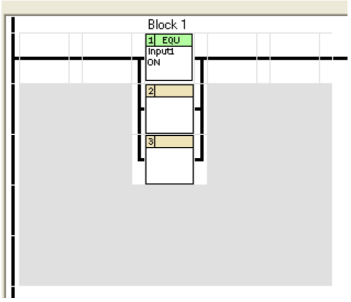
Boolean: (A) AND (B) AND (C)



Logic Block 5

If A is true or B is true or C is true, then block is true.

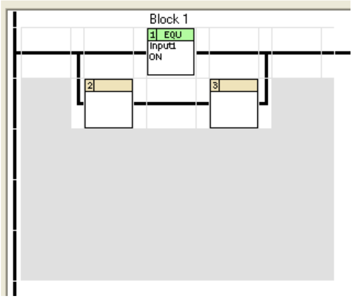
Boolean: (A) OR (B) OR (C)



Logic Block 6

If A is true or B and C are true, then block is true.

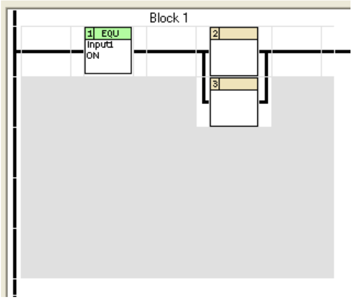
Boolean: (A) OR ((B) AND (C))



Logic Block 7

If A is true and B or C is true, then block is true.

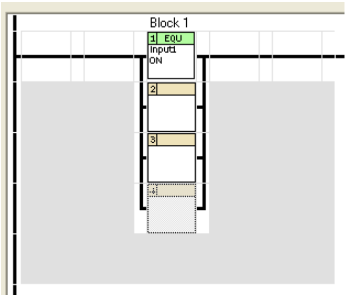
Boolean: (A) AND ((B) OR (C))



Logic Block 8

If A, B, C, or D is true, then block is true.

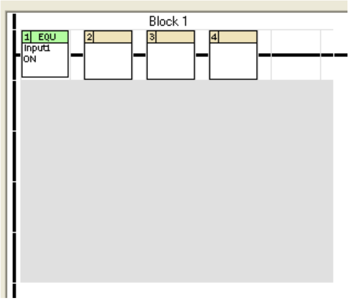
Boolean: (A) OR (B) OR (C) OR (D)



Logic Block 9

If A, B, C, and D are true, then block is true.

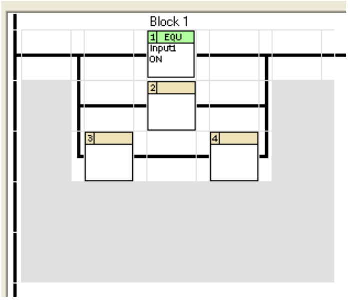
Boolean: (A) AND (B) AND (C) AND (D)



Logic Block 10

If A is true or B is true or C and D are true, then block is true.

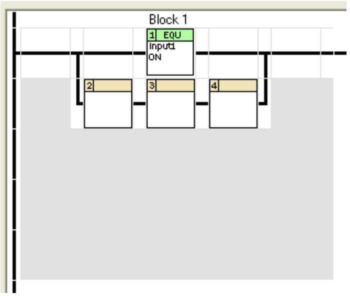
Boolean: (A) OR (B) OR ((C) AND (D))



Logic Block 11

If A is true or B, C and D are true, then block is true.

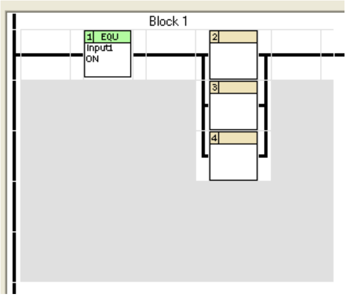
Boolean: (A) OR ((B) AND (C) AND (D))



Logic Block 12

If A is true and B, C or D is true, then block is true.

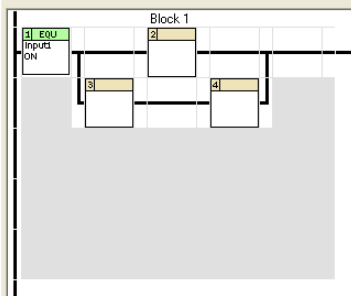
Boolean: (A) AND ((B) OR (C) OR (D))



Logic Block 13

If A is true and B is true or C and D are true, then block is true.

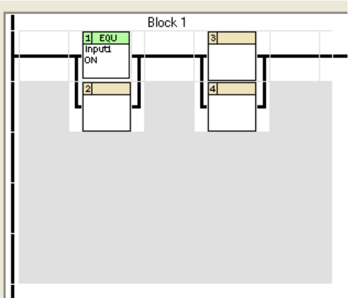
Boolean: (A) AND ((B) OR ((C) AND (D)))



Logic Block 14

If A or B is true and C or D is true, then block is true.

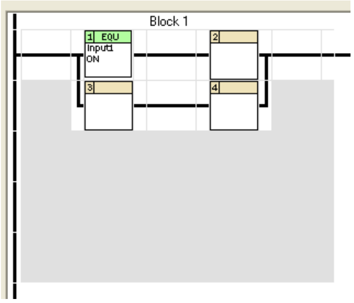
Boolean: ((A) OR (B)) AND ((C) OR (D))



Logic Block 15

If A and B are true or C and D are true, then block is true.

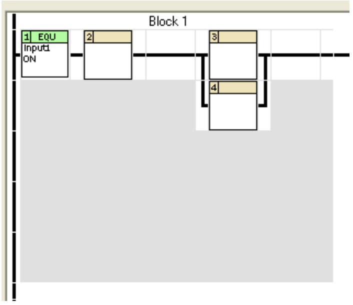
Boolean: ((A) AND (B)) OR ((C) AND (D))



Logic Block 16

If A and B are true and C or D is true, then block is true.

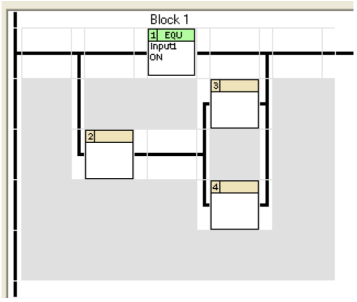
Boolean: (A) AND (B) AND ((C) OR (D))



Logic Block 17

If A is true or B and C or D are true, then block is true.

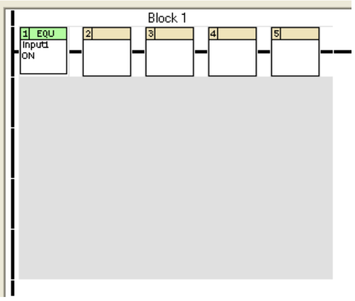
Boolean: (A) OR ((B) AND ((C) OR (D)))



Logic Block 18

If A, B, C, D and E are true, then block is true.

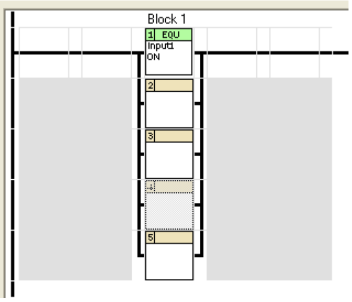
Boolean: (A) AND (B) AND (C) AND (D) AND (E)



Logic Block 19

If A, B, C, D or E are true, then block is true.

Boolean: (A) OR (B) OR (C) OR (D) OR (E)



This Page Intentionally Left Blank

Compiling Database

COMPILER WINDOW

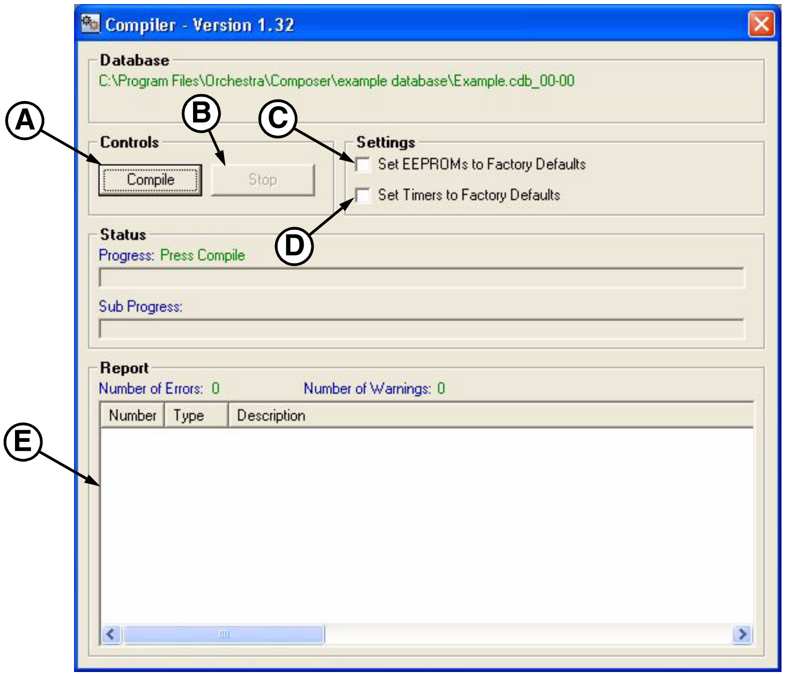
Description

Compiles the user's *Data Items, Modules, Functions,* and *Rungs* into a downloadable format.

Files generated during download are located in the same directory that the database was saved to. Do not save the database into the composer installation path.

Several file extensions are created from the database name:

- name.cdb_00-00 = User database.
- name.clc1_00-00 = Master download file. Must be selected by Downloader.
- name.clc2_00-00 = Secondary file called from Downloader.
- name.clc3_00-00 = Third file called from Downloader.
- name.ioc_00-00 = This file is used by Conductor for diagnostics.
- name.dsp_00-00 = List created to be used with displays.
- name.err_00-00 = Tracks *Errors* and *Warnings* of the last compile.



Compiler - Version 1.32

Database
C:\Program Files\Orchestra\Composer\example database\Example.cdb_00-00

Controls

Settings
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

Status
Progress: Press Compile

Sub Progress:

Report
Number of Errors: 0 Number of Warnings: 0

Number	Type	Description

Figure 1

Controls

(A) Compile Button

Clicking this button will begin the Compile process.

(B) Stop Button

Clicking this button will stop the Compile process.

(C) Set EEPROMs to Factory Defaults Check Box

When this is checked, the compile files will force the application Downloader to set the *EEPROMs* to factory defaults.

(D) Set Timers to Factory Defaults Check Box

When this is checked, the compile files will force the application Downloader to set the *Timers* to factory defaults.

(E) Report Text Box

Any *Warnings* or *Errors* are displayed here. *Warnings* and *Errors* are also written to a .err file located in the same file path as the user database.

OVERVIEW

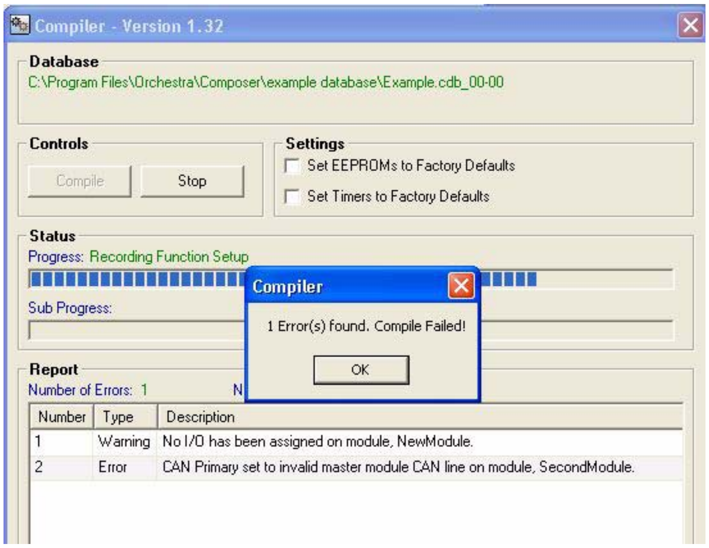
Compiling will generate the files that will be downloaded to the Master Module and an IOC file for Conductor.

The general order of processes is as follows:

1. Click the ⚙ *Compile Database* button to open the **Compiler** window.
2. You will be prompted to save existing database files. Selecting *Yes* will continue the compile. Selecting *No* will cancel the operation.
3. Select the *Compile* button from the **Compiler Setup** window.
4. All *Modules* are checked for *Errors* or *Warnings*. *Errors* are fatal and must be corrected. *Warnings* are non-fatal and correction is optional.
   (a) If *Errors* are found, the compile will stop and notify you of what needs to be corrected.
   (b) If *Warnings* are found, you will be prompted to continue or not. If *Yes* is selected, the compile will continue and the *Warnings* will be listed in the *Text Box*. If *No* is selected, the compile will stop, allowing you to make the necessary corrections.
5. All *Data Items* are checked for *Errors* or *Warnings*.
   (a) If *Errors* are found, the compile will stop and notify you of what needs to be corrected.
   (b) If *Warnings* are found, you will be prompted to continue or not. If *Yes* is selected, the compile will continue and the *Warnings* will be listed in the *Text Box*. If *No* is selected, the compile will stop, allowing you to make the necessary corrections.
6. All *Function Blocks* are checked for *Errors* or *Warnings*.
   (a) If *Errors* are found, the compile will stop and notify you of what needs to be corrected.
   (b) If *Warnings* are found, you will be prompted to continue or not. If *Yes* is selected, the compile will continue and the *Warnings* will be listed in the *Text Box*. If *No* is selected, the compile will stop, allowing you to make the necessary corrections.
7. All files will be saved to the directory where the database currently resides.
8. Composer will then create a series of files. *See Description on page 87* for a list of these files.

Errors and Warnings

If an *Error* is found during compile, you will be notified and the compile will fail **(Figure 2)**.


Compiler - Version 1.32

Database
C:\Program Files\Orchestra\Composer\example database\Example.cdb_00-00

Settings
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

Status
Progress: Recording Function Setup

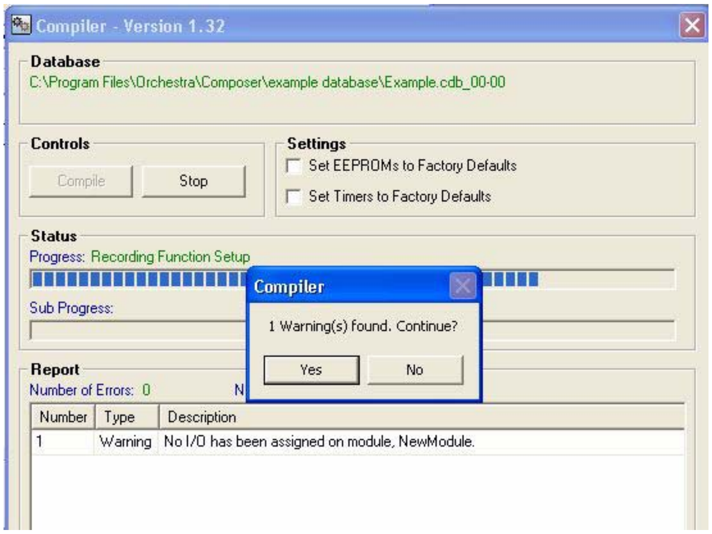
Compiler
1 Error(s) found. Compile Failed!

Report
Number of Errors: 1

Number	Type	Description
1	Warning	No I/O has been assigned on module, NewModule.
2	Error	CAN Primary set to invalid master module CAN line on module, SecondModule.

Figure 2

If a *Warning* is found during compile, you will be notified and have the option to continue or correct the problem **(Figure 3)**.


Compiler - Version 1.32

Database
C:\Program Files\Orchestra\Composer\example database\Example.cdb_00-00

Settings
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

Status
Progress: Recording Function Setup

Compiler
1 Warning(s) found. Continue?

Report
Number of Errors: 0

Number	Type	Description
1	Warning	No I/O has been assigned on module, NewModule.

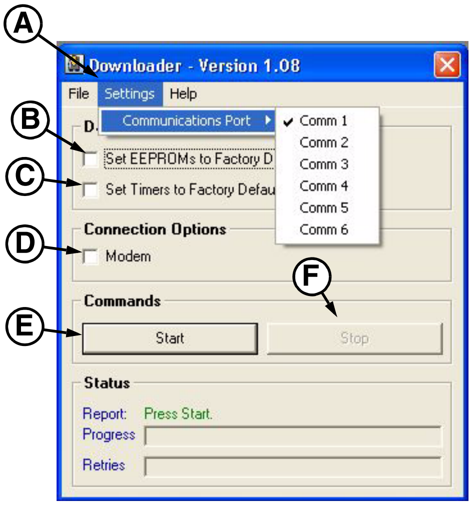
Figure 3

Download Database

DOWNLOADER WINDOW

Description

Downloader will send the compiled files to the Master Module.



Report: Press Start.

Figure 4

Controls

(A) Settings File Menu

This allows you to select the appropriate comm port for communications.

(B) Set EEPROMs to Factory Defaults Check Box

Checking this box will set the *EEPROMs* in the Master Module back to the settings that were defined by Composer.

When using Conductor, you can change the *EEPROMs* to enhance the machine's functionality. Resetting the *EEPROMs* by mistake would remove these changes.

If the settings were changed to perform an experimental running test and you don't want the machine to go out with the new setting, this will reset all *EEPROMs* to default. If any *EEPROMs* were added or removed, all *EEPROMs* will be set to default automatically.

In the **Project Setup** window in Composer, a user can select these to be checked ON automatically.

(C) Set Timers to Factory Defaults Check Box

Checking this box will set the *Timers* in the Master Module back to the settings that were defined by Composer.

For example, if you want to retain the running hours of an engine, you would not want to check this box. However, if you were performing a procedure in which the engine hours needed to be reset to Composer default, you would check this box. If any *Timers* were added or removed, all *Timers* will be set to default automatically.

In the **Project Setup** window in Composer, a user can select these to be checked ON automatically.

(D) Modem Check Box

When this box is checked, the **Phone Number** box will open. A modem on the PC and also on the HED CANLink bus are required for this function to work.

(E) Start Button

Clicking this button will begin the download process.

(F) Stop Button

Clicking this button will stop the download process.

OVERVIEW

This will download the files created during Compile into the module, and allow the module to perform the tasks defined during programming.

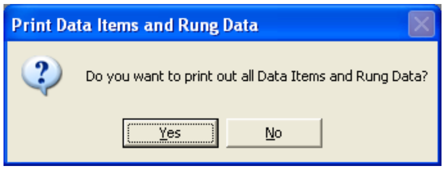
The general order of processes is as follows:

1. Click the *Download Composer Application* button.
2. The **Open** window will appear. Choose the database to download and click the *Open* button.
3. The **Downloader** window will open.
4. Select the appropriate comm port from *Settings>Communications Port.*
5. Choose the appropriate *Data Options* and *Connection Options.*
6. If the *Set EEPROMs to Factory Defaults* check box is checked or the *EEPROM* time stamp is different, the current *EEPROMs* will be over-written.
7. If you have selected the *Set Timers to Factory Defaults* or the *Timers* time stamp is different, the current *Timers* will be over-written.
8. Click the *Start* button.
9. The PC will connect with the module and the software will verify the firmware version. If the firmware version is not compatible, the firmware will need to be updated before the download can continue.
10. When communication is established and the firmware version is confirmed, the download will begin.

Printing

PRINT

Click the *Print* button to print the contents of the database. You will be prompted to continue.



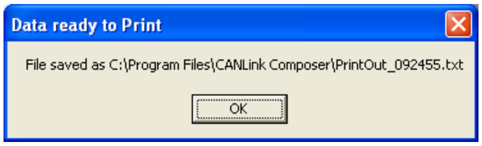
Print Data Items and Rung Data

Do you want to print out all Data Items and Rung Data?

Figure 5

If you click the *No* button, the print process will be halted.

If you click the *Yes* button, a text file will be created and a window showing the file path will appear.

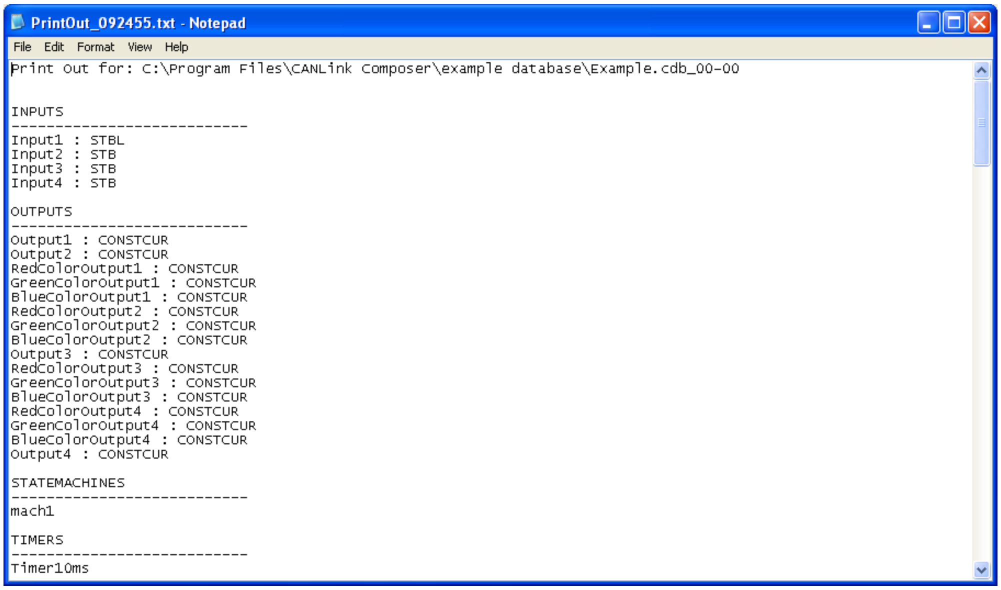


Data ready to Print

File saved as C:\Program Files\CANLink Composer\PrintOut_092455.txt

Figure 6

Locate and open this text file. Print it from the text application.



PrintOut_092455.txt - Notepad

```
Print out for: C:\Program Files\CANLink Composer\example database\Example.cdb_00-00

INPUTS
---------------------------
Input1 : STBL
Input2 : STB
Input3 : STB
Input4 : STB

OUTPUTS
---------------------------
Output1 : CONSTCUR
Output2 : CONSTCUR
RedColorOutput1 : CONSTCUR
GreenColorOutput1 : CONSTCUR
BlueColorOutput1 : CONSTCUR
RedColorOutput2 : CONSTCUR
GreenColorOutput2 : CONSTCUR
BlueColorOutput2 : CONSTCUR
Output3 : CONSTCUR
RedColorOutput3 : CONSTCUR
GreenColorOutput3 : CONSTCUR
BlueColorOutput3 : CONSTCUR
RedColorOutput4 : CONSTCUR
GreenColorOutput4 : CONSTCUR
BlueColorOutput4 : CONSTCUR
Output4 : CONSTCUR

STATEMACHINES
---------------------------
mach1

TIMERS
---------------------------
Timer10ms
```

Figure 7

Supplemental Information

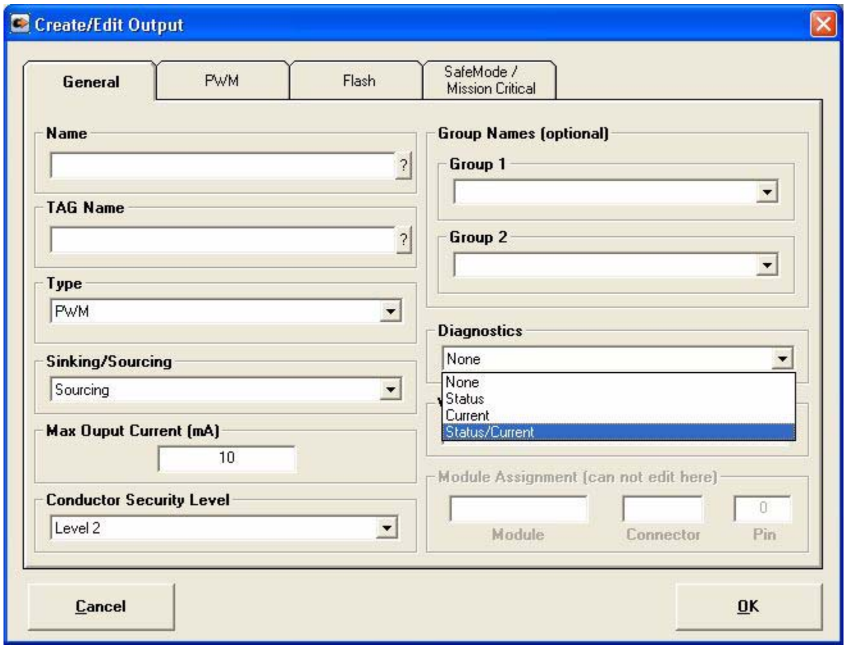
FLYBACK CALCULATION PROCEDURE

Items required for procedure:

- Microsoft® Excel (or similar plotting software)
- Module(s) to be used by system
- Meter capable of reading current
- Composer Software
- Conductor Software
- Coil (this is the coil that will be used by the *Output* on the system).

Steps:

1. Use only one *Output* from the coil for this test (This *Output* must be capable of being a *PWM Output* with *Current Feedback*).
2. Connect the coil to the *Output* of the module.
3. Configure the appropriate *Output*, using Composer (See *Create/Edit Output Window - Unique Fields on page 35*).
   (a) Under *General Tab* **(Figure 8)**
   - Set the *Type* to *PWM*.
   - Set *Diagnostics* to *Status/Current*.


Create/Edit Output

Name:

TAG Name:

Type: PWM

Sinking/Sourcing: Sourcing

Max Ouput Current (mA): 10

Conductor Security Level: Level 2

Group Names (optional)

Group 1:

Group 2:

Diagnostics: None

Module Assignment (can not edit here)

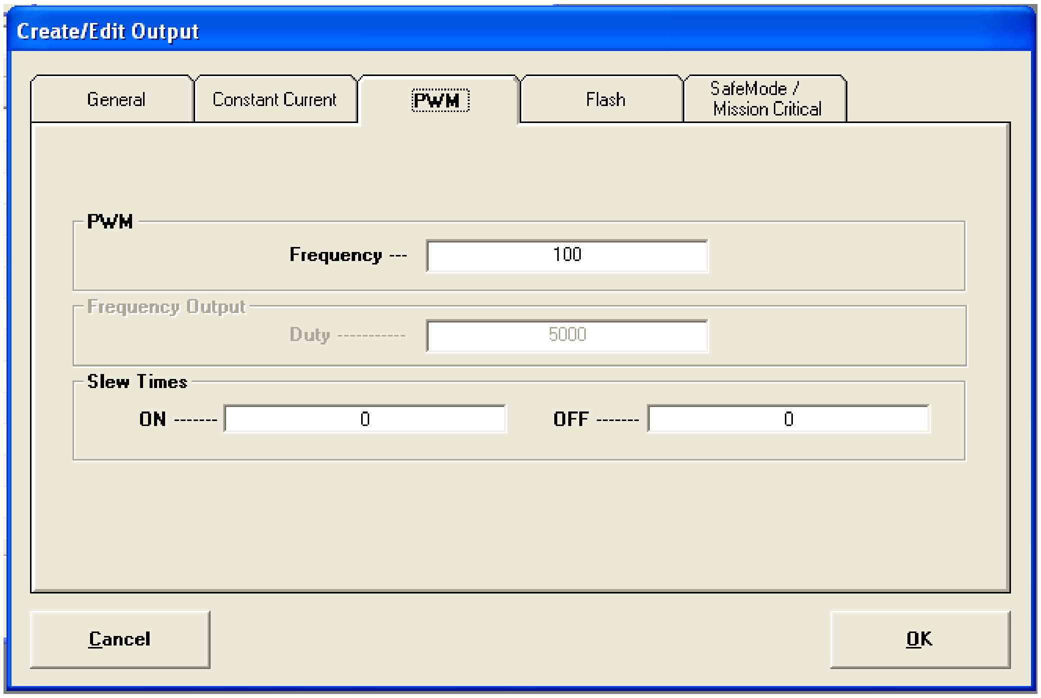
Module:

Connector:

Pin: 0

Figure 8

(b) Under the *PWM Tab* **(Figure 9)**, set the desired *Frequency*, based on the characteristics of the coil.



Create/Edit Output

PWM

Frequency --- 100

Frequency Output

Duty ---------- 5000

Slew Times

ON ------- 0 OFF ------- 0

Figure 9

4. Either using the existing system database or creating a new database in Composer, set up one *Rung* to turn the *Output* off.
5. Download database to the module.
6. Using Conductor, add the *Output* to the debug window.

Note: The following steps (7 through 11) are time- and heat-dependent. Proceed in a timely manner as not to heat up the coil enough to affect the readings. Turn off the Output as soon as possible.

7. In the **Debug** window, turn the *Output* on to 100 (10% duty).
8. Using a meter, record the *Coil Current*.
9. Record the *Current* value in the *Cur Fdbk* column of the **Output** window in Conductor.
10. Calculate and record the *Flyback Current* (*Coil Current - Board Current = Flyback Current*).
11. Turn the *Output* off and let the coil cool down.

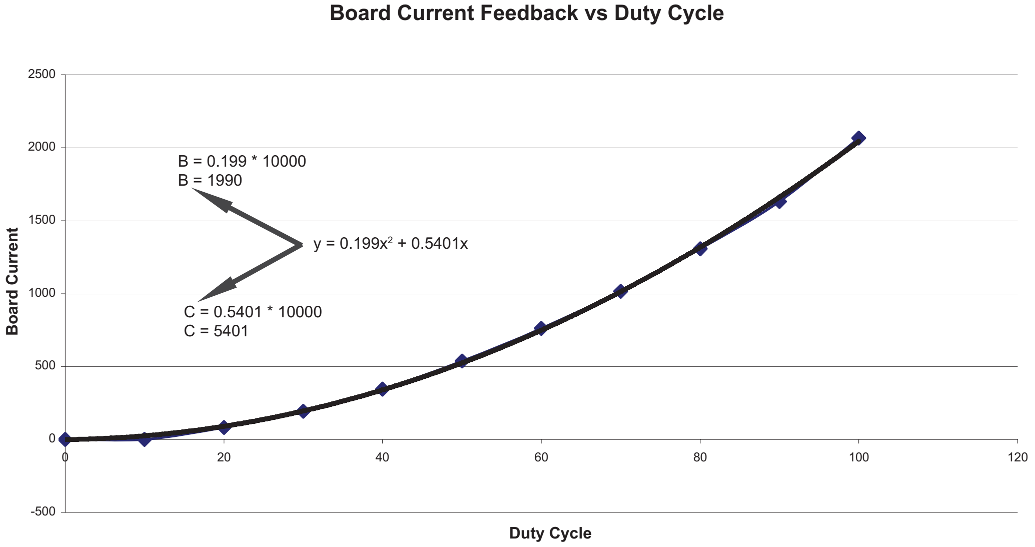
12. Repeat steps 7 through 11, each time adding 10 percent to the previous duty cycle. This should give data points for 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100% duty as found in the *Example Data Points* table below **(Figure 10)**.

Example Data Points

Duty Cycle (Commanded)	Coil Current (Measured)	Board Current Feedback (Recorded Using the Conductor)	Flyback Current (Calculated)
0	0	0	0
10	110	0	110
20	319	83	236
30	536	194	342
40	753	345	408
50	971	537	434
60	1189	762	427
70	1409	1015	394
80	1610	1307	303
90	1830	1631	199
100	2065	2065	0

Figure 10

13. Once the data is recorded, plot two X-Y graphs to view as the examples below **(Figure 11)** *Board Current Feedback vs Duty Cycle* and **(Figure 12)** *Flyback Current vs Duty Cycle*.

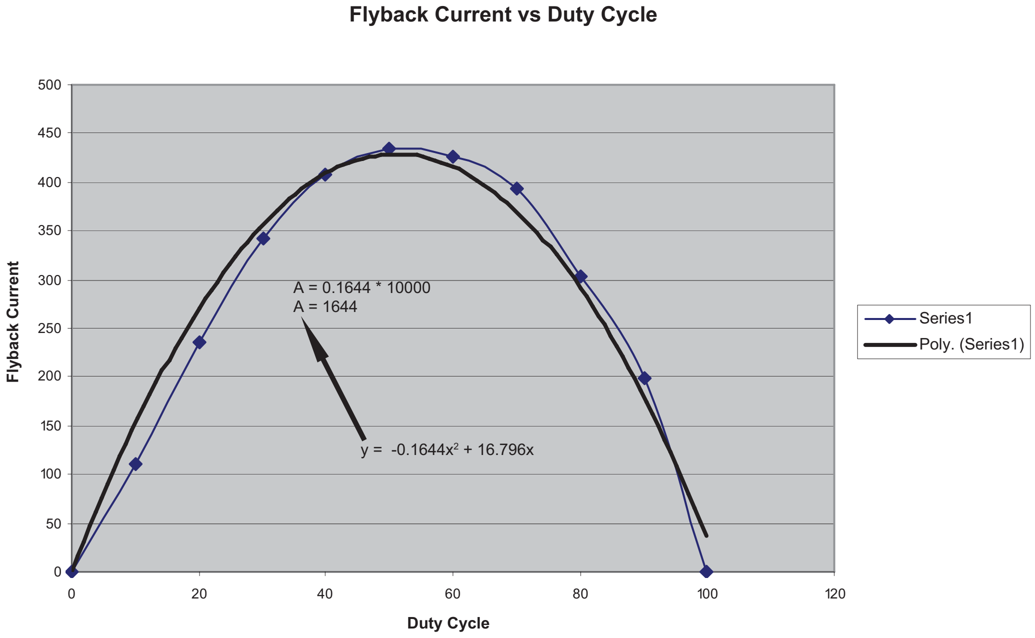


B = 0.199 * 10000
B = 1990

y = 0.199x² + 0.5401x

C = 0.5401 * 10000
C = 5401

Figure 11

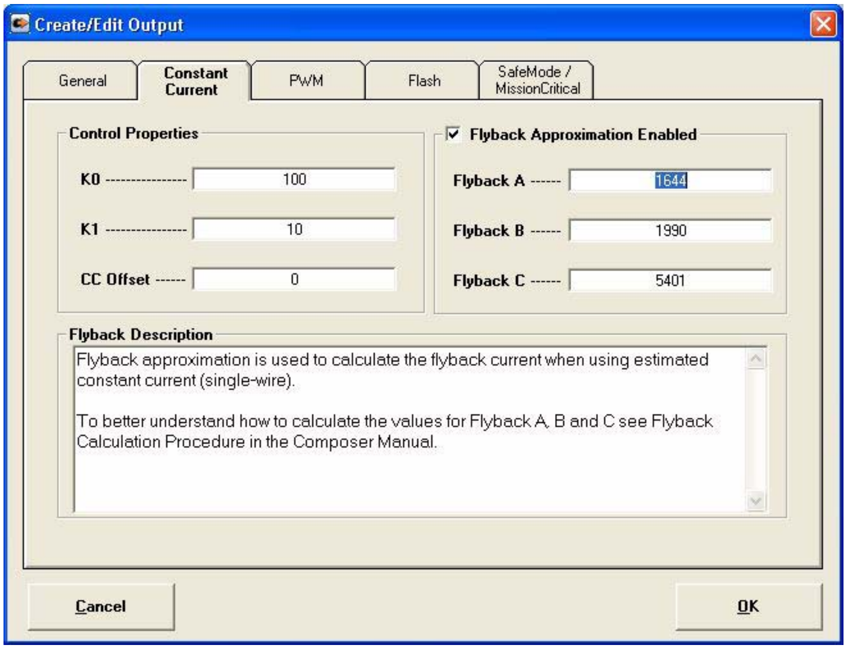


A = 0.1644 * 10000
A = 1644

y = -0.1644x² + 16.796x

Figure 12

14. Create a trend line for each graph with an equation of it.
15. In this example **(Figure 11)**, the equation of the trend line from the *Board Current Feedback vs Duty Cycle* ($y = 0.199x^2 + 0.5401x$).
    (a) Disregard the sign of the numbers and multiply each by 10,000.
    (b) *Factor B* = 0.199 * 10000 = 1990
    (c) *Factor C* = 0.5401 * 10000 = 5401
    (d) *Factors B* and *C* will be used in Composer, later in this example.
16. In this example **(Figure 12)**, the equation of the trend line from the *Flyback Current vs Duty Cycle* ($y = -0.1644x^2 + 16.796x$).
    (a) Disregard the sign of the number and multiply it by 10,000.
    (b) *Factor A* = 0.1644 * 10000 = 1644
    (c) Factor A will be used in Composer, later in this example.
17. Using/Entering Results
    (a) Using the same database, in Composer, reconfigure the *Output Type* of *Output 1* to be *Constant Current*.
    (b) Under the *Constant Current Tab* **(Figure 13)**:
    - Check the *Flyback Approximation Enabled* check box.
    - Enter the three *Factors A, B* and *C* found above.



Control Properties

K0 ---------------- 100

K1 ---------------- 10

CC Offset ------ 0

☑ Flyback Approximation Enabled

Flyback A ------ 1644

Flyback B ------ 1990

Flyback C ------ 5401

Flyback Description

Flyback approximation is used to calculate the flyback current when using estimated constant current (single-wire).

To better understand how to calculate the values for Flyback A, B and C see Flyback Calculation Procedure in the Composer Manual.

Figure 13

(c) Change the one *Rung* to set the current somewhere in the middle of the range of the coil. Example: If the range of the coil is 0-3A, then set it to 1.5A (1500 in Conductor).

(d) Compile and download the database to the module.

(e) The current should remain constant when the voltage is turned up or down. If it does not, adjust *Factor A*.

- When the voltage goes down and the current goes up, *Factor A* is set too high.
- When the voltage goes down and the current goes down, *Factor A* is set too low.
- Keep adjusting *Factor A* until current remains relatively stable as voltage is raised and lowered.
- If you cannot get the current to stabilize, try adjusting K_0 and K_1 (See *Constant Current Tab on page 36*).

Licensing

DONGLE

A dongle is required for Composer to operate. If the dongle is lost or damaged, a temporary license file can be requested by clicking *Help>Request Temp License File* menu option from the **Main Window**. This will create a file called *HEDLicenseFile.dat*.

HEDLicenseFile.dat must be sent to HED where it will be read and a file called *LicenseRequest.dat* will be created and sent to you. *LicenseRequest.dat* must be placed in Composer installation path.

If no dongle or license request file are present, Composer will run in demo mode, with limited use, allowing you to create *Data Items* and select *Modules* for up to 10 *Rungs*. You will also be able to compile and download.

Company Information

ADDRESS

HED (Hydro Electronic Devices, Inc.)
2120 Constitution Ave.
Hartford, WI 53207

WEB SITE

www.hedonline.com

This Page Intentionally Left Blank

Index

CANLink Software Products

HED®

Hydro Electronic Devices, Inc.