DINGS' MOTION USA™

....Precision Motion Specialists

SnapTrack™ Software Manual

Programming and Software Reference for all *ServoTrack™* Motion Controllers

SnapTrack Software Manual Change Log		
Date	**Revision**	**Changes**
01/09/2016	00	Initial Release
04/26/2017	01	Added Block Type Reference Table
1/24/2018	02	Added More Detail
1/20/2023	03	Minor Updates

The information contained in the DINGS' Motion USA product manuals and publications are carefully checked and are believed to be accurate. However, DINGS' Motion USA assumes no responsibility for any inaccuracies found in said content.

DINGS' Motion USA reserves the right to make changes without further notice to any products for the purpose of improving reliability, function and / or design. DINGS' Motion USA neither assumes any liability arising from application, or use of any product or circuit described herein; nor does it convey any license under its patent rights of others.

DINGS' Motion USA general policy does not recommend use of its products in life support- or aircraft-related applications wherein product failure or malfunction may directly threaten life and / or result in injury. Per DINGS' Motion USA terms and conditions of sale, users of DINGS' Motion USA products for life support- or aircraft-related applications assumes all risk of such use and indemnifies DINGS' Motion USA from all damages.

Manual Revision Rev 3.0, January 2023

www.dingsmotionusa.com

Table of Contents

1 INTRODUCTION

1.1 About This Manual

This manual is for ServoTrack™ devices using SnapTrack programming software.

1.2 Overview

SnapTrack is a program that utilizes block programming to allow simple and quick use of ServoTrack™ motion controllers. In block programming, blocks can be linked together to create applications, and no experience with scripting languages are required.

This section will acquaint users with the general functions of SnapTrack, including:

- the (3) primary tabs used to create an application
- introduction to block programming

1.3 Opening Window

Selectable Tabs On opening SnapTrack, the window in *Figure 1.1* appears. Near the top, there are (3) primary **tabs** available for creating a motion application: *Build*, *Setup*, and *Dashboard*. Each tabs contains different tools to create and edit a program.

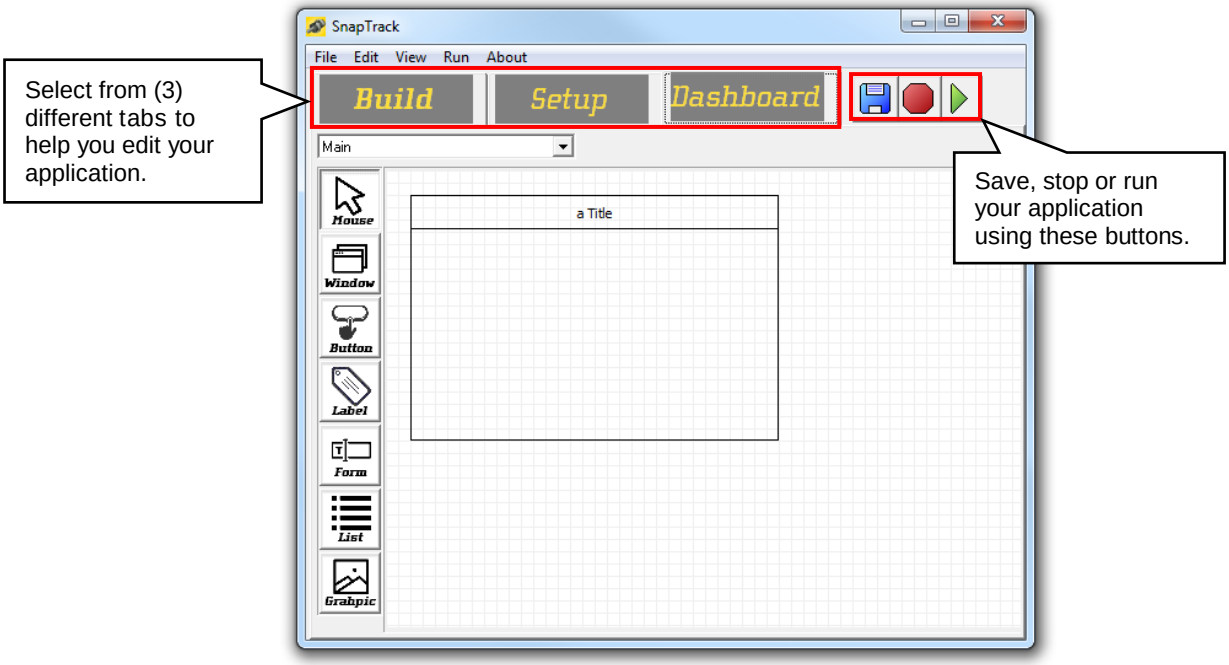


Figure 1.1 Opening window.

Save, Stop, or Run Buttons to save, stop, or run programs are placed next to the selectable tabs. As programs are being edited, save often to prevent loss of work. SnapTrack programs do not save automatically when run.

Power-Cycling Occasionally an error message may pop up upon running a program. In this event, power-cycle the ST484 controller by first turning off the power supply voltage to the unit. Then check all wire connections before reapplying power.

1.4 Dashboard

The *Dashboard* tab is open by default when launching SnapTrack (shown in *Figure 1.1 and 1.2*). This tab provides the starting point in creating programs. The graphed area provides a workspace to create the program's interface using tools, such as buttons, labels, and more.

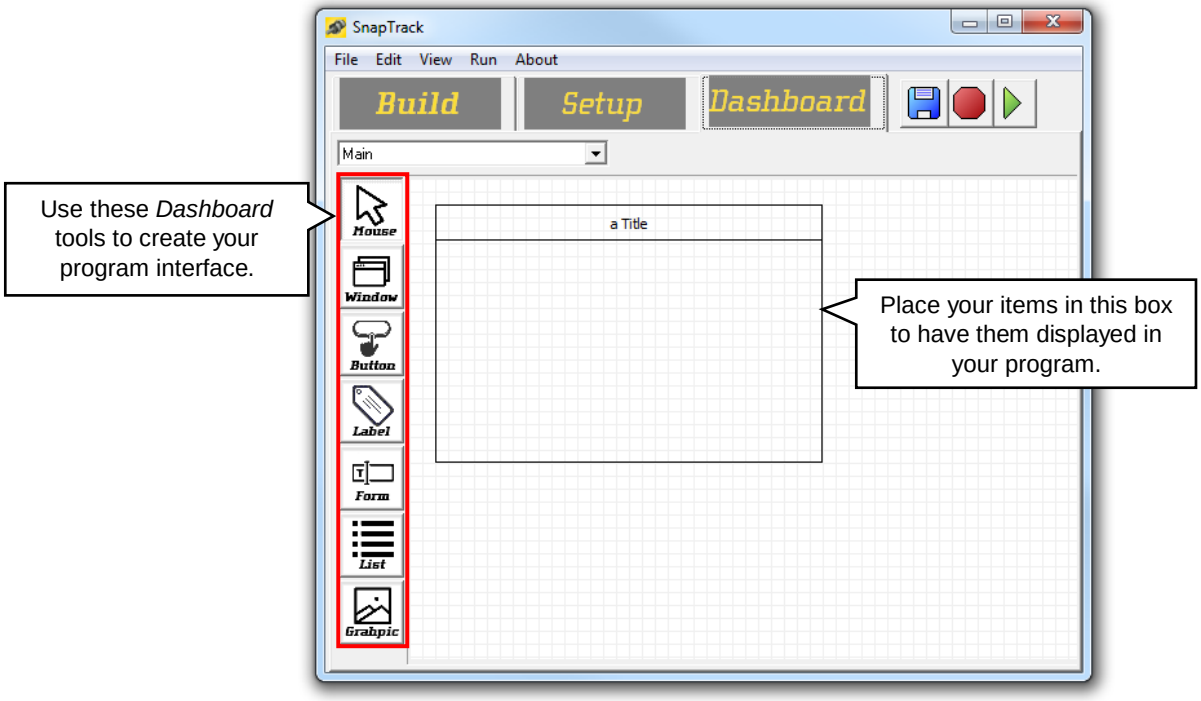


Figure 1.2 The *Dashboard* screen with different tools and items for creating a user interface.

Dashboard Tools

A list of different items can be found on the left-hand side of the *Dashboard*, as shown in Figure 1.2. Table 1.1 briefly describes each of these tools and their functions. These tools allow a user to create different *Dashboard* items to interact within a program. Once items have been created, they may be programmed with actions or displays. Be sure to place these items within the box labeled “a Title” (by default). Your items will not appear unless they are placed within a display window.

Table 1.1 *Dashboard* tools and description.

Tool	Description	
	Cursor:	Select, move, or resize *Dashboard* items.
	Window:	Create new windows for program interface.
	Button:	Create a programmable button.
	Label:	Create s user-defined text label. Can display an output value as well.
	Form:	Create a field to read user input from keyboard
	List:	Create a dropdown list.
	Graphic:	Insert an image or graphic to window

**See Section TBD for more detailed information one each tool.*

1.5 Setup

The Setup tab will be used to set the fundamental parameters of a motor. These parameters include naming different motors (for distributed controls), setting a motor's user units, etc. Any inputs and outputs (I/O's) wired up to a ServoTrack™ controller will also be edited here, including names of I/O's, I/O direction, digital vs analog, etc. The Setup tab essentially functions as the *hardware settings* for your controller.

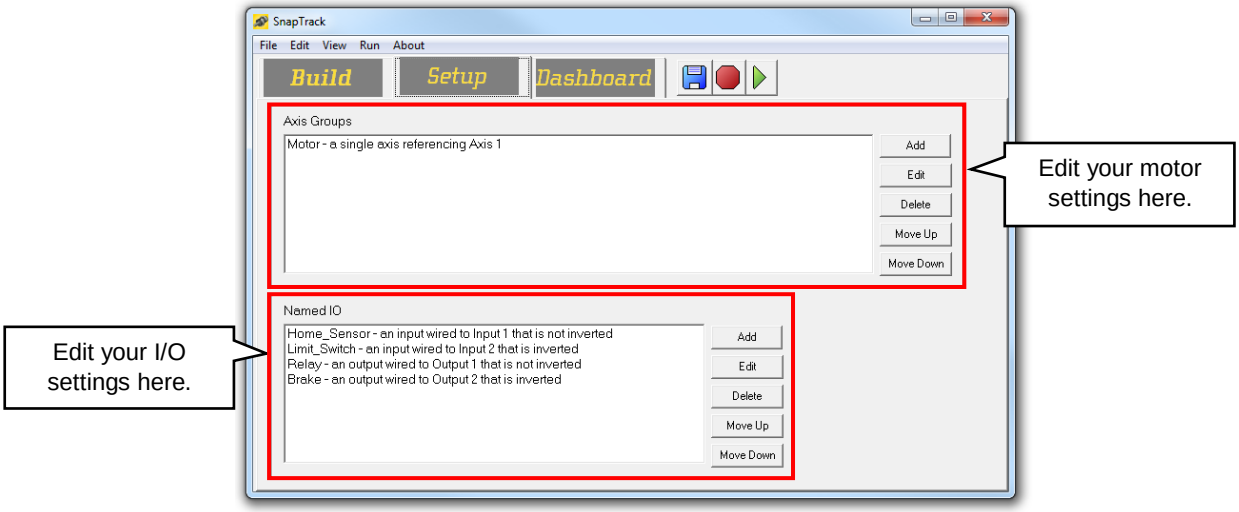


Figure 1.3 Setup tab for motor and I/O settings.

The Build tab can be seen in *Figure 1.3*. In this tab are two main sections: "Axis Groups" for motor parameters and "Named IO" for I/O parameters. To edit, first select the motor or I/O to be edited. Then click Edit to set its corresponding parameters. Motors and I/O's may be added, renamed or organized per user preference.

1.6 Build

Items created on the *Dashboard* will be edited in the Build tab. The main parts of this tab are:

- a dropdown menu for *Dashboard* items
- a selection menu of block types
- blocks corresponding to the selected block type

Figure 1.4 highlights the main sections of the Build tab. The Motion block type has been selected (in blue). All Motion block are shown below it.

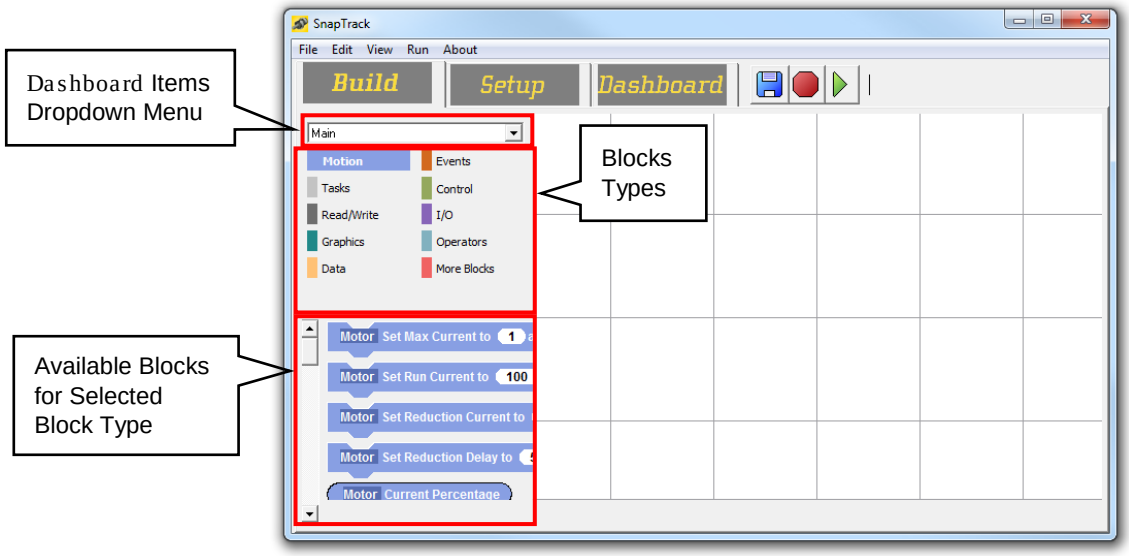


Figure 1.4 Use *Build* tab to program *Dashboard* items.

To begin programming, start by selecting a *Dashboard* item. This action can be done in one of two ways:

1. Click on an item from *Dashboard* before moving to *Build*, OR
2. Use the *Dashboard* Items dropdown menu in the upper left hand corner of *Build*.

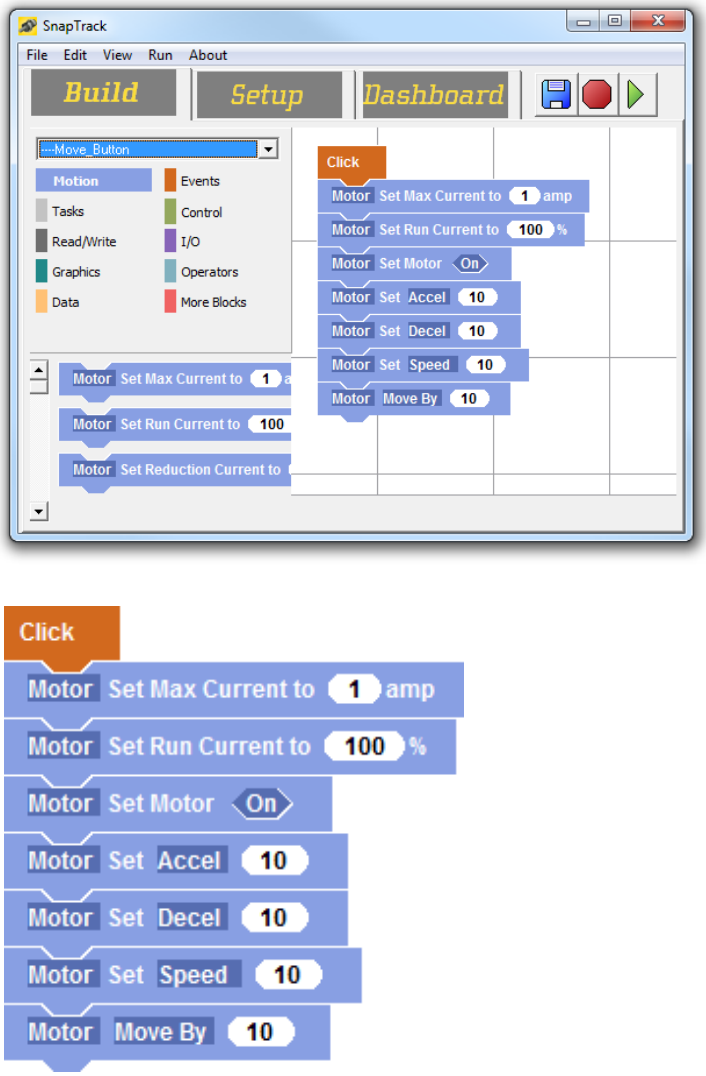


Figure 1.5 Example of a Block program.

Action Lists

In SnapTrack, actions are programmed by linking blocks together, to form *Action Lists*. SnapTrack features many different block types, each with many different blocks. The blocks are color coded to easily differentiate between block types. This type of programming removes the complexity of scripting languages and reduces the likelihood of syntax errors by allowing users to simply link blocks together. Standard coding features, such as if-else statements and while loops, are all still available for limitless functionality.

Figure 1.5 shows an example of a block program where blocks have been assembled into an Action List. The Action List, shown, corresponds to the actions for the item: **Button1**.

In this program, a button is programmed to activate by click, and triggers the actions below it. The motor's current is first set to 1 Amp. Then, the motor is turned on (or energized), and the acceleration, deceleration, and speed for the motor are all set to 10 user units. Finally, the motor will move 10 user

units. These user units can be full steps, full revolutions, and more depending on the user-determined parameters in the Setup tab. The general structure of any Action List is an Events block, followed by motion parameters, and finally, an action, as shown in *Figure 1.6*. Action List may deviate from this structure. However, it is good practice to define all parameters prior to executing an action.

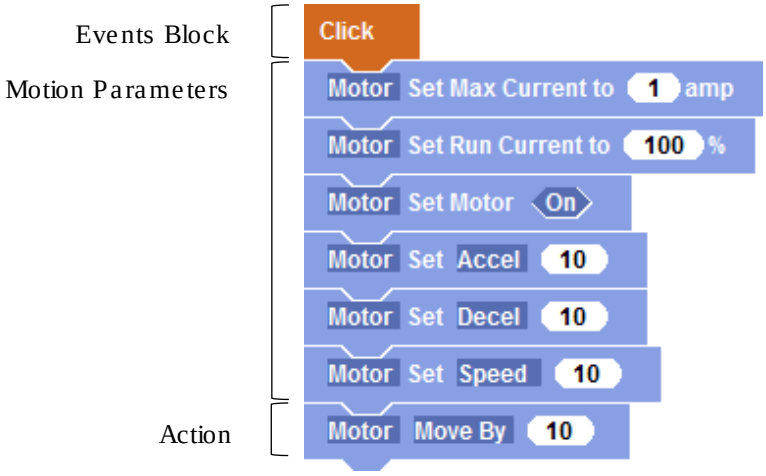


Figure 1.6 General Structure of an *Action List*.

2 Dashboard

Dashboard functions as the starting point for creating new applications.

Here, users can create buttons, labels, and other items to use for a user-created application. Applications are first sketched on the *Dashboard* screen. SnapTrack then uses this sketch to create the user's application. *Figure 2.1* shows an example of a sketch, and its resultant application. The application features (3) buttons.

On the top screen, we see a sketch of a user's application. Upon clicking *Run*, SnapTrack builds the application and displays it in a new window reflecting the user's sketch.

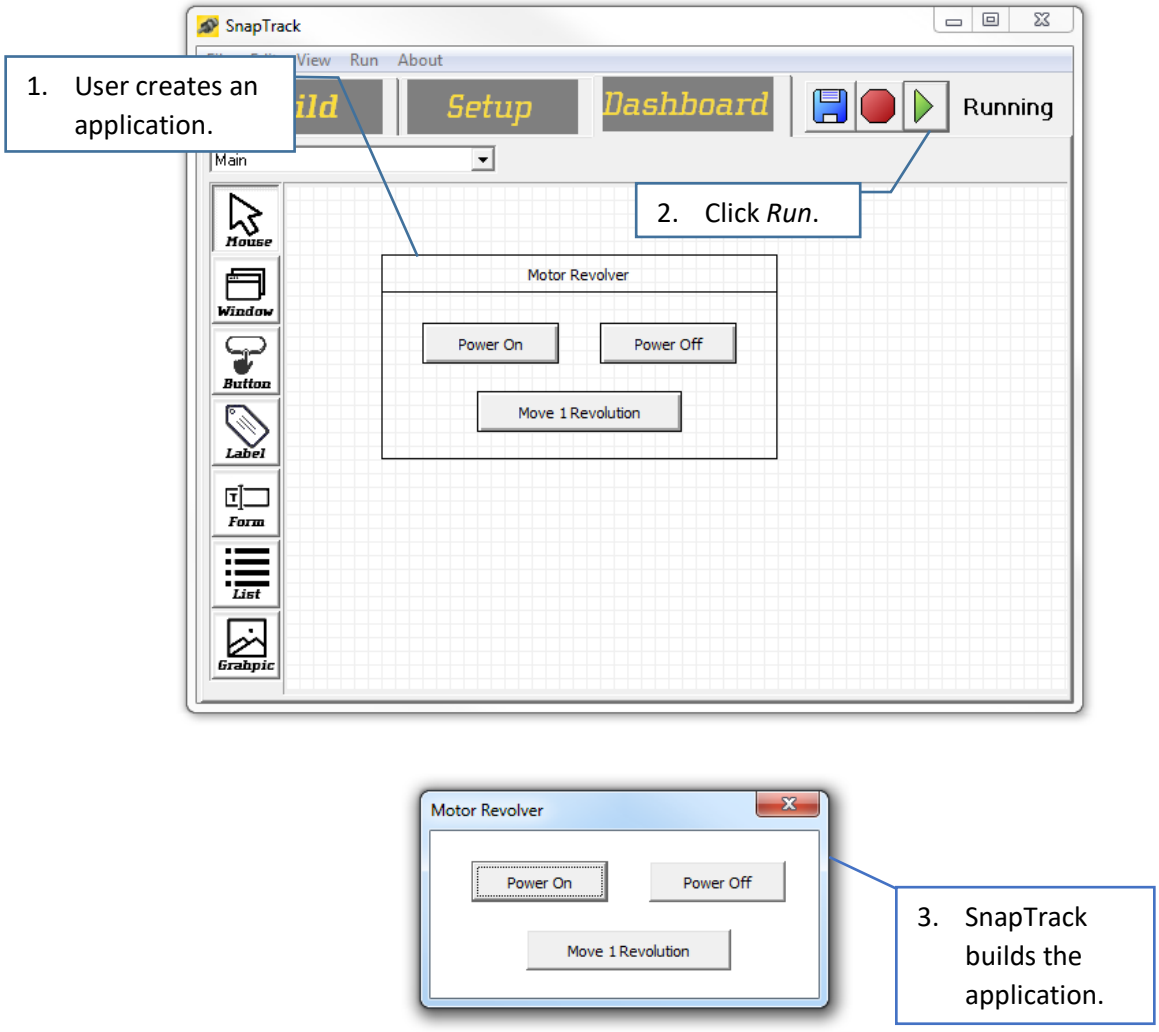


Figure 2.1 A user created application, and final output.

Getting Started

To start sketching an application, use one of the (7) tools provided on the left-hand side of the *Dashboard* panel. Each tool allows users to create or edit a different item.

Start sketching by populating the provided window with items to be shown in the final application. Each item can be programmed through Action Lists on the Build screen. To build, select an item before moving to the Build tab. It may be helpful to first sketch all items and displays for the application.

See below for a description and guide on each tool.

2.1 Cursor

Figure 2.2
Cursor Tool

The Cursor tool is the first and top most item available to users.

Use the cursor to position items, resize windows and buttons, or edit created fields.

Double-clicking buttons, windows, labels and forms with the cursor will also open a properties menu for the item.

2.2 Window

Figure 2.3
Window Tool

The Window tool allows users to create new display windows.

While SnapTrack provides a primary window to begin with, users may wish to create additional windows.

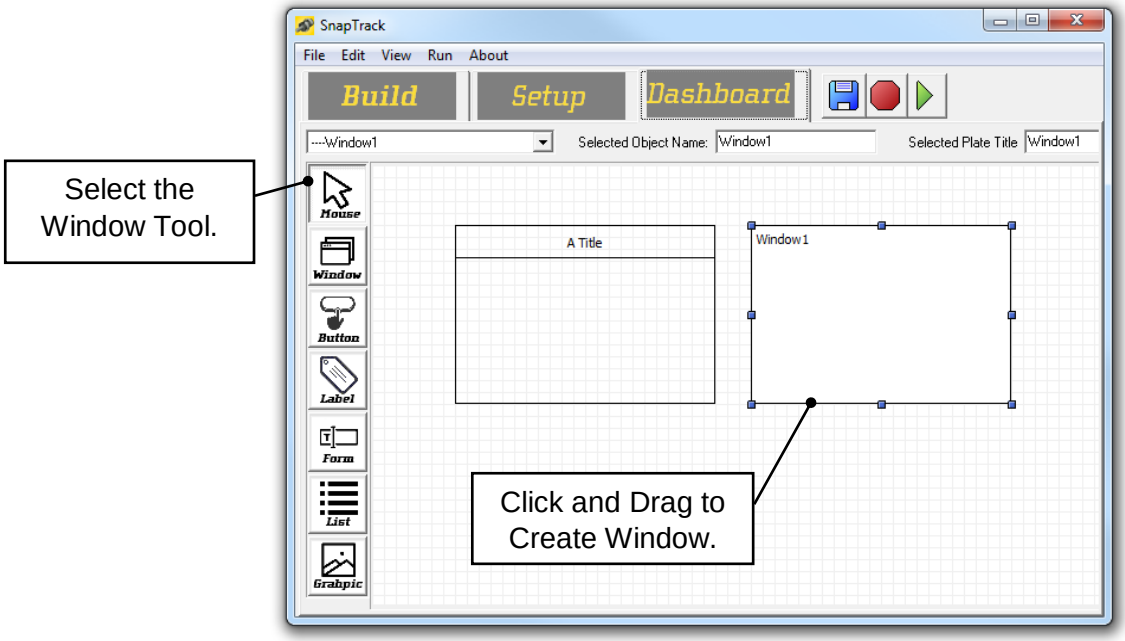
As with the first window, new windows can also be populated with *Dashboard* items, allowing users to organize as they see fit.

To make a new window, select the Window tool and drag out a boxed area for the new window. Double clicking the new window will open up an area ready for editing. To return to the original window, simply double-click outside of the boxed area.

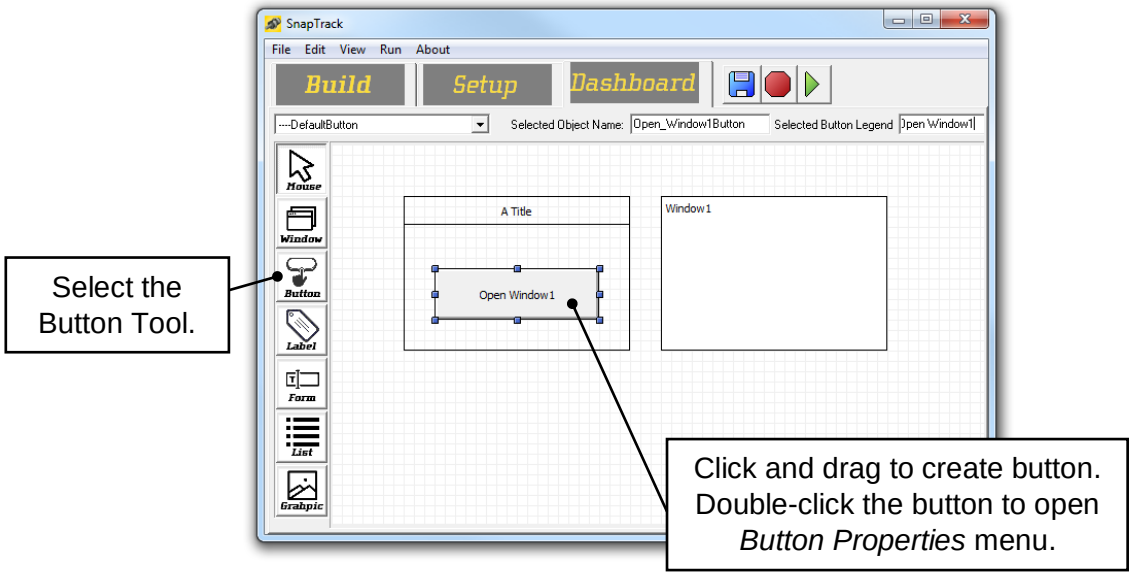
Please note, windows must be programmed to open with a button.
See *Creating a New Window* (Figure 2.2) for step-by-step instructions.

Creating a New Window

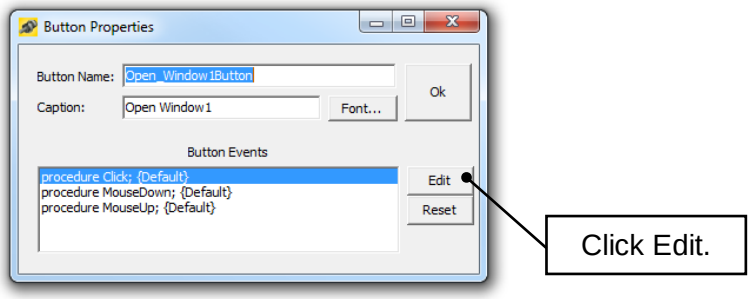
Step 1 Create a new window. Remember the object name: “Window1”.



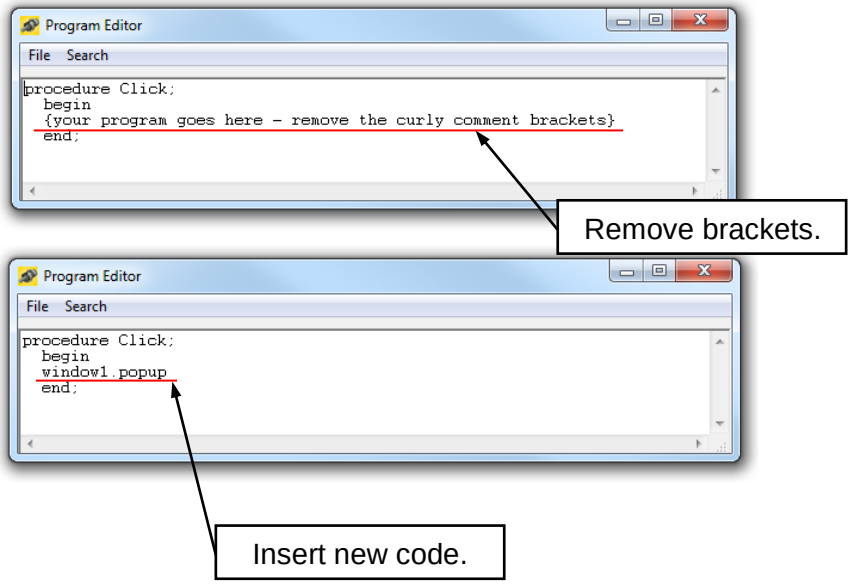
Step 2 Create a button to open the new window.



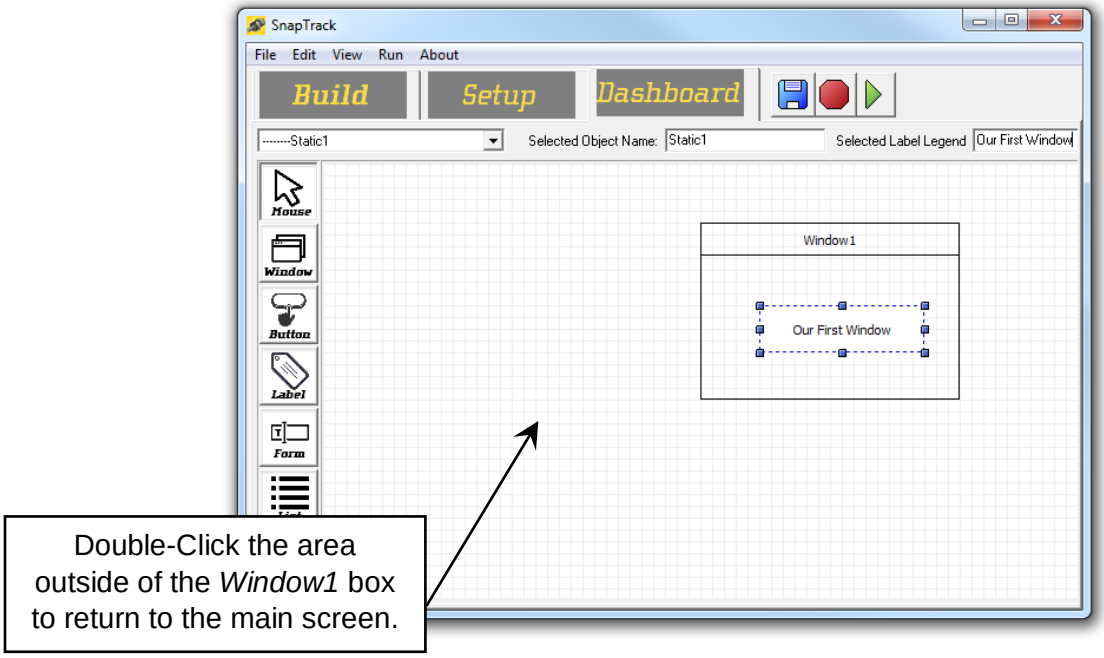
Step 3 Double-Click the button to open the *Button Properties* menu. Click Edit on the right-hand side.



Step 4 Replace the bracketed line in the *Program* Editor with "*window_name.popup"*.
For the window in this example, it will be "window1.popup".
Close the *Button Editor* and *Program Editor* screens.



Step 5 Double-Click *Window1*. The following screen will appear.
Here, the label “Our First Window” was added for reference. (optional)
Double-Click any area outside of the *Window1* box to return to the main screen.



Step 6 Click *Run* on the main screen.
The following window will appear.

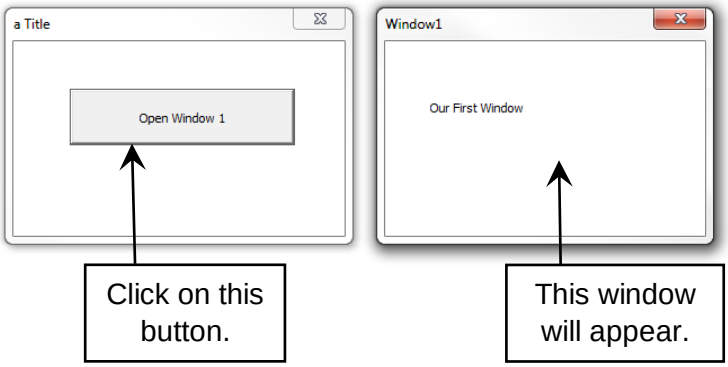


Figure 2.4 Procedures for creating a new window.

2.3 Button

Figure 2.5
Button Tool

The Button tool is used to create programmable buttons. These buttons can perform actions when they are either clicked or moused down.

Buttons will be the *primary* tool for creating interactive applications. They may be built to turn a motor on or off, continuously move a motor, or move a specific number of steps per button click, among many other tasks.

See Figure 2.5: Step 2 for a recap on how to create a button.

To start creating buttons, first select the button tool. Once selected, move to the graphed area to the right, click-hold and drag a boxed area for the desired button size and location. Buttons sizes and location may be changed to the user's preference.

After the button has been created in *Dashboard*, move to the Build tab to create the Action List for that button.

2.4 Label

Figure 2.6
Label Tool

The Label tool is used to create text or data labels. Labels may display text or motion data from the application or ServoTrack™ unit. For example, the label may be static text that read, "Power buttons here:" OR it may display a current read-out, to show a user the current percentage of the ServoTrack™ unit. These displays may be built to show many different parameters, such as, motor speed, motor position, or current percentage, etc.

To create a display label, select the tool and drag a boxed area for the appropriate size and location. To build the label's tasks, first select the label on the *Dashboard* screen. Then move to the Build tab.

2.5 Form

Figure 2.7
Form Tool

The Form tool is used to create a field that can read text inputs. This tool allows apps to take in typed data, such as inputting motor current, or move distance. Data taken from forms may also be stored and referenced for later use.

Per the standard procedure for the previous tools, create a form by first selecting the tool, then dragging open an appropriate area for the field.

To program the form, select the form and move to the Build tab.

2.6 List

Figure 2.8
List Tool

Create menus and lists with the List tool. Per the standard procedure, select the List tool and drag open a box for the desired list size. Lists may be displayed with all elements shown at once, or within a drop-down menu. Fonts and items may be edited by double clicking the created list box.

2.7 Graphic

Figure 2.9
Graphic Tool

The Graphic tool allows users to insert images into an application. These images may include custom logos or diagrams to attach to an application.

To do so, select the graphics tool and select the image location in the file explorer. Images must be in .bmp format and cannot be scaled when importing to SnapTrack. Resize images to an appropriate file size before uploading.

Once selected, move to the graphed area and drag an allotted area for the selected image.

Naming Items

Name newly created items in the “Selected Object Name” field, located near the top of the *Dashboard* screen, immediately below “*Dashboard*”.

When working from certain screens, like editing a window or editing from the Build tab, items may not be visible for selection. However, in the drop-down menu located near the top left of the *Dashboard* screen, all items will be listed by name for easy access.

3 Setup

The parameters set in the Setup screen will determine the preliminary and default motion and hardware parameters for a SnapTrack application.

Here, users can organize motors for multi-drop systems, and organize hardware inputs and outputs (I/Os). Users may name motors (called axes), name I/Os, set preliminary motor speed, acceleration, and deceleration, set user units for a motor, or designate if an external device is an input or output, or a digital or analog I/O among other options.

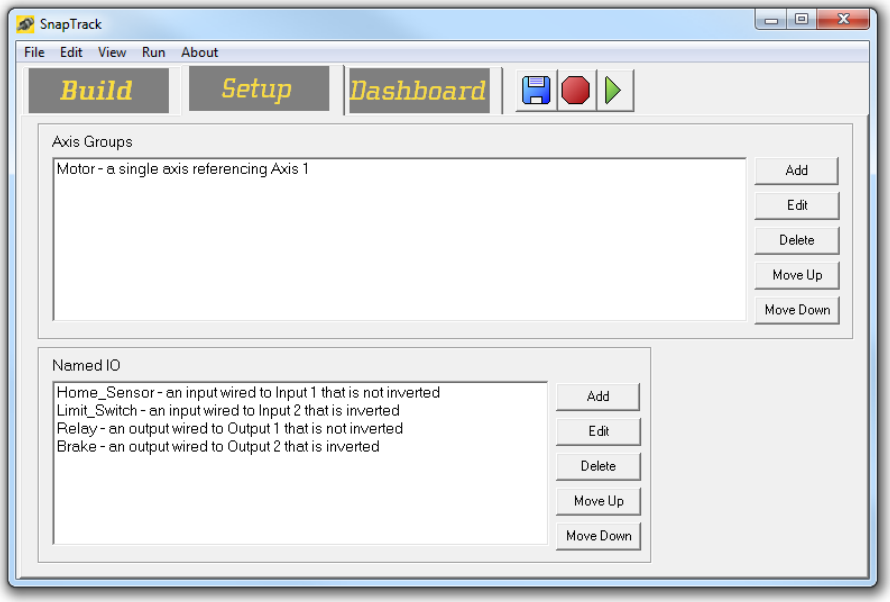


Figure 3.1 The Setup screen.

Figure 3.1 shows the Setup screen with two primary parts: *Axis Group*, and *Named IO*. The default Setup screen lists (1) motor in the *Axis Group* section and (4) I/Os in the *Named IO* section.

The (4) listed I/Os are only a few examples of many different devices that may be programmed with ServoTrack™. Any of the default device names may be modified or deleted, and new ones may also be added.

3.1 *Axis Groups*: Setting Motor Parameters

To begin, select the item labeled *Motor – a single axis referencing Axis 1* within the *Axis Groups* section. This item will typically refer to the first ServoTrack™ - motor pair that is connected to the programming PC. To rename this motor, simply double-click the motor, or click to highlight the item, then click Edit. Once done, the Axis Group Editor screen will appear (Figure 3.2). Motor axes may also be added, deleted, or reorganized using the buttons on the right-hand side of the screen.

Once *Motor* has been selected and double-clicked, an *Axis Group Editor* screen, shown in Figure 3.2, should appear. Here, the starting parameters for the first motor in the system can be set.

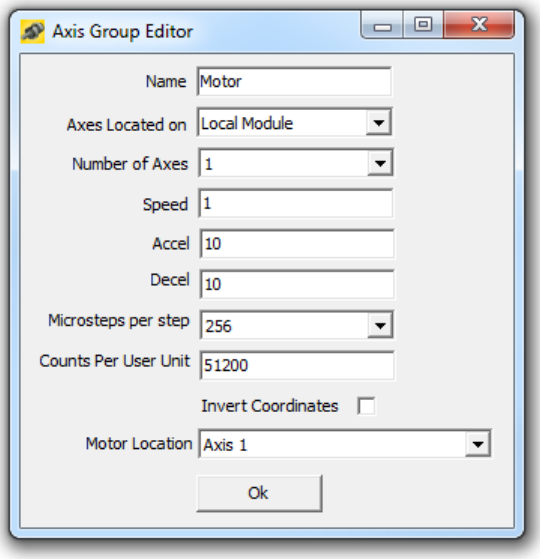


Figure 3.2 *Axis Group Editor* screen for motor parameters.

Parameters include:

- *Name:*
- *Axes Located on:*
- *Number of Axes:*
- *Speed:* (in user units per second)
- *Accel:* (motor acceleration in user units per second per second*)*
- *Decel:* (motor deceleration in user units per second per second)
- *Microsteps per step* (microstep resolution)
- *Counts Per User Units*
- *Invert Coordinates* (to reverse ServoTrack™'s default direction convention)

Speed, Accel & Decel Set initial motion parameters for a motor by entering desired values in the appropriate fields. The speed value will be in user units per second for speed, and user units per second per second for acceleration and deceleration. These settings will be the default ramp and slew speeds for the

motor in use. Please note, these motion conditions can be changed any time, in an application. A single application may have several buttons and tasks with different speed, acceleration and deceleration. The settings in the *Axis Group Editor* screen only sets default parameters, or starting conditions.

Microstep Resolutions SnapTrack offers (7) different options for microstep resolutions. This setting determines how many counts each motor step will be divided by, offering greater resolution for higher microstep settings.

As an example, for a standard 200 step per revolution stepper motor, each full step the motor takes it will turn is 1.8 degrees. For a microstep setting of *8*, this motor may commutate 8 times, or 8 *counts*, for each full step it takes. Hence, this motor will move 0.225 degrees per count.

Higher microstep settings will allow a motor to run smoother and more precise. As a result, the microstep setting is set to the highest resolution, at *256 microsteps per step*. Lower microstep resolutions are only recommended for higher speed applications where lower signal frequencies are required.

Counts per User Unit This setting is a conversion factor for the units to be used in SnapTrack. It determines the number of microsteps counted for each unit in the application.

Users may set a value of user units to reflect an easier-to-work-with value. For example, users may prefer to work with units of revolution rather than microsteps. This setting is the default value in SnapTrack.

Conversely, users may also set *Counts per User Unit* to reflect inches per unit with reference to the screw lead of the stepper linear actuator in use.

To give an example, a user has a linear actuator with a 0.1 inch lead(0.1 inches per revolution) and wish to work in units of full inches. With a microstep setting of 256 microstep, and a standard bipolar 200-step motor, the calculations, below, will determine the *Counts per User Unit* needed to give inches per unit.

$$\left(\frac{256\ \text{counts}}{\text{step}}\right)\left(\frac{200\ \text{steps}}{\text{revolution}}\right)\left(\frac{1\ \text{revolution}}{0.1\ \text{inches}}\right) = 512{,}000\ \text{counts}/\text{inch}$$

Given the above lead screw, motor, and microstep settings, a user may set the *Counts per User Unit* to 512,000 in order to work in inches per unit. This means that when a user sets motor speed to a value of 1, it will represent 1 inch per second.

Set user units and microstep resolution settings to values which will be most intuitive for the current application.

Invert Coordinates Stepper motor wiring conventions and sequence charts may vary between manufacturers and may even vary among motor sizes for one manufacturer. This variation may lead to some confusion in step direction when wiring a motor to the driver.

To alleviate this difficulty, users may switch the direction convention of their system by checking or unchecking the *Invert Coordinates* box. If a user is expecting a motor to rotate clockwise, but notices the motor rotate otherwise, checking the box will reverse the motor's direction and align it to the user's expectation.

3.2 *Named IO*: Setting Device Parameters

A variety of input and output devices may be linked to ServoTrack™ and applied to custom applications. In the *Named IO* section of the *Setup* Screen, users may organize, label and set signal direction for each input and output (IO).

To start, first select one of four default inputs and outputs (IOs) listed in the *Named IO* section. Double-click the selected item, or highlight and click the *Edit* button on the right-hand side. Once done, the screen in Figure 3.3 will appear.

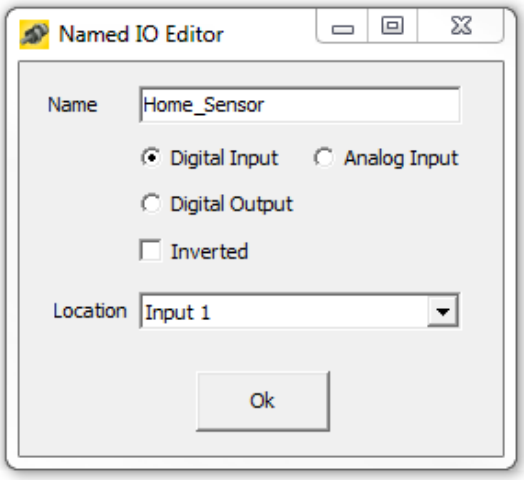


Figure 3.3 *Named IO Editor* screen for IO parameters.

In this window, rename the IO to match the device being used. Also set the device's signal direction type: *Digital Input, Digital Output,* or *Analog Input.*

In total, each SnapTrack controller may also interface with 3 different IOs. Each IO will have an associated index number, specified in the *Location* window.

This device name and index number will be used to reference IO devices when creating applications and building Action Lists. Figure 3.4 shows two blocks. The left block is referencing the device, *Home Sensor*, and the right block is referencing sensor *Input 1* connected to *Motor,* the name of a ServoTrack motor axis. An explanation on blocks will be provided in the next section: *4. Build*.

Figure 3.4 Blocks referencing an input device

4 Build

This section will cover programming instructions for SnapTrack.

Before an application can perform any tasks, *Dashboard* items must be created programmed using Action Lists to govern what these items will do. To do so, this section will address the following topics:

- Introduction to Action Lists *(Section 4.1)*
- Block Shapes *(Section 4.2)*
- Block Types *(Section 4.3)*
- Block Types Reference Table *(Section 4.4)*

4.1 Introduction to *Action Lists*

Figure 4.1 shows the *Build* screen for a *Power On* button.

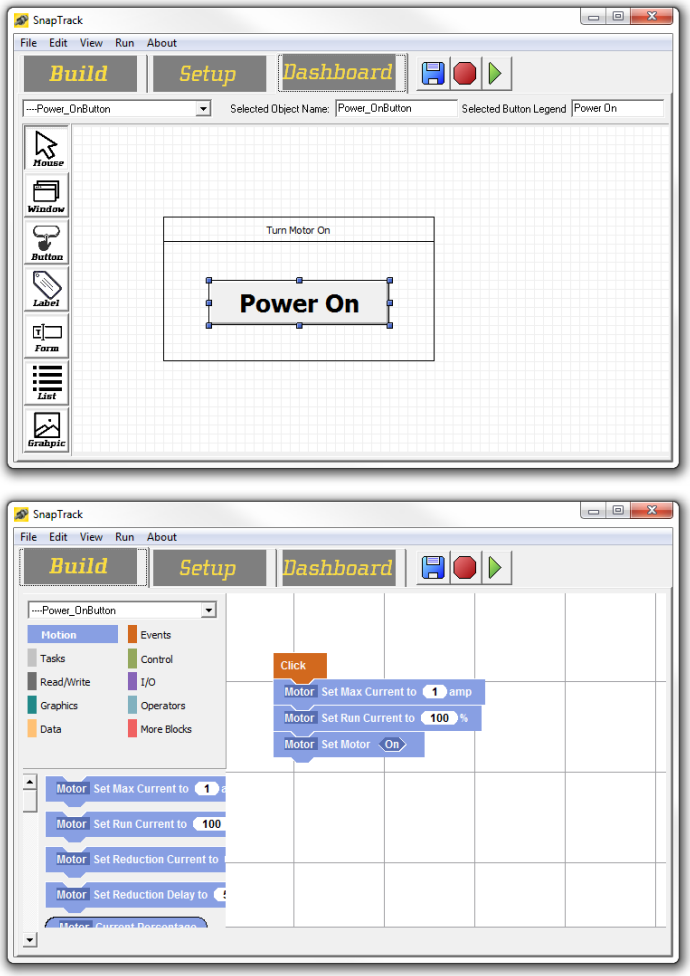


Figure 4.1 A *Dashboard* Item and the *Action List* for the item

The top screen shows a *Dashboard* item being selected.
The bottom screen shows *Build* screen, where the *Action List* for the *Power On* button is found.

This *Action List* is a program to determine what function and tasks the *Power On* button will perform.

Likewise, each *Dashboard* item must have its own *Action List*, accessible from the *Build* screen. *Action Lists* will essentially be the codes or programs for all user-created applications.

To program *Dashboard* Items, follow the steps below:

1. Create a *Dashboard* Item
2. Select the *Dashboard* Item
3. Move to the *Build* screen
4. Create an *Action List*

Hardware parameters in the *Setup* tab must also be set. However, these parameters can be set independently from creating *Dashboard* items and *Action Lists*.

Scripting code will not be used very often in SnapTrack. Users will only interact with code in very key instances, such as creating a pop-up window. Conversely, all programming will be done with blocks and *Action Lists*.

Knowledge of block shapes and block types will be crucial before exploring *Action Lists* further *(Section 4.4)*.

4.2 Block Shapes

To program in SnapTrack, link blocks together to form Action Lists. These Action Lists will determine the commands for a single *Dashboard* item.

Blocks may be linked together when two blocks have matching geometry, and each one will have different shapes to determine which blocks it may connect with. This shapes will also determine the function of the block. See Figure 4.2 as an example of two mating blocks. These blocks have shapes that allow them to mate.

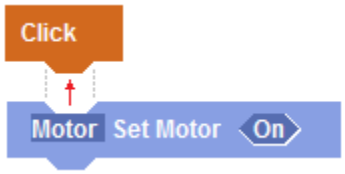


Figure 4.2 Two blocks with mating shapes.

To notify users that two blocks may connect, SnapTrack will show an orange bar when two blocks are placed near each other, as shown in Figure 4.3. This bar will now show if two blocks may not connect. Blocks will snap together when the mouse click is released.

Figure 4.3 Compatible blocks. A link bar being shown on the left, to indicate two compatible blocks. On the right, the two blocks that have been linked together.

In addition, some blocks may also be inserted into other blocks. Figure 4.4 shows an example of this type of link. The blocks, here, will command a label to show the motor speed.

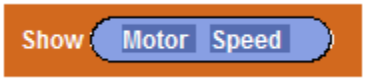
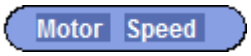


Figure 4.4 A is inserted into another block.

There are four main block shapes. Table 4.1 lists these block shapes and provides more detail on their function, appearance, and use.

Table 4.1 List of different block shapes

Shape	Example	Function	Description
Square		Action	Performs a single task for a *Dashboard* Item
Tabbed Square		Action	Perform a single task within an Action List for a *Dashboard* item
Diamond		Boolean	Holds a logic True or logic False value to be placed within a block
Round		Variable	Holds a numeric value

In general, block shapes will determine how a block may interact with other blocks beyond just connections. For example, the *Show* block in Figure 4.4 cannot chain with other blocks to form Action Lists. It must stand alone, and may only accept round Variable blocks within the space provided.

Likewise, round blocks and diamond blocks may not form Action Lists with other blocks either unless the Action List has a spaces to receive it. Only tabbed square blocks may form Action Lists.

Use these shapes as a guide to program commands for *Dashboard* items.

4.3 Block Types

Along with shapes, SnapTrack also features different block types. Each block type is color coded for easy identification.

Different block types may be linked together if their block shapes are compatible.

Figure 4.4 shows a screenshot of the block types menu in the *Build* tab. Block types are listed at all times. In this figure, the *Motion* block type has been selected, with all motion blocks listed below it. Each time a block type is selected, blocks of that type are listed.

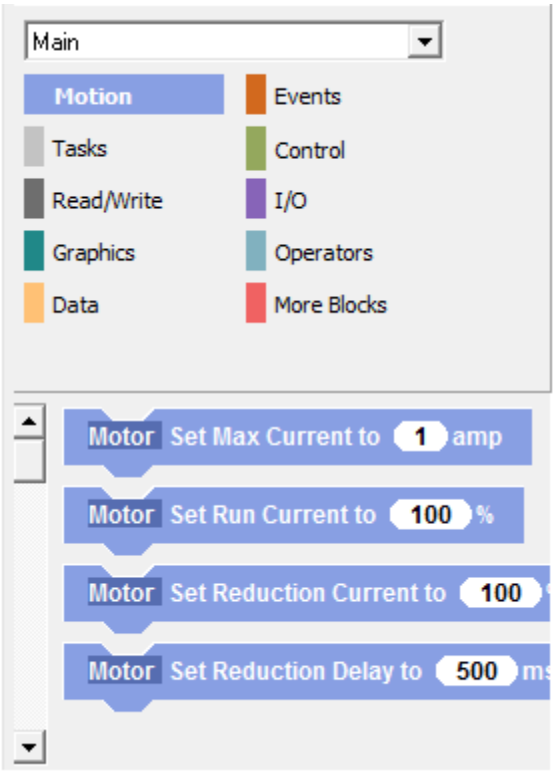


Figure 4.5 Block Types Menu with Color Legend. Blocks for the selected block type appear below in the *Build* tab.

See below for details on each block type.

4.4 Block Types Reference Table

Block Type	Description / Use
Motion	Set motion parameters, commands, encoder settings, current settings, lead / lag output, motor on / off setting, etc. All motor parameters and commands will involve these blocks.
Events	Determine how a user may interact with Dashboard items. Set triggers to activate a Dshboard item and commence an Action List. For example, trigger a Dashboard item via click, when a mouse button is held down, or display a value to the user interface.
Tasks	Organize procedures to determine when specific actions will be performed within an action list. Task blocks are nested within Action Lists.
Control	Set time delays or loop actions within an Action List. Also set conditions before executing specific actions.
Read/Write	Take in variable or string information from the application. For example, take in a user's keyboard input to set motor current.
I/O	Interact with I/O's connected to the ServoTrack unit. Can turn on/off, set sample rates, or take in I/O information.
Graphics	Draw a line on the user interface. Typically used to graph information from I/O's or motor parameters.
Operators	Conditional operators used to compare two variables or do simple math. These blocks are all variables or Booleans, and need to be nested in other blocks.
Data	Modify and edit variables or Booleans.
More Blocks	Create custom blocks that perform an Action List. Can allow commonly used actions to be placed in a single block versus recreating the same Action List sequence in several program locations.

COMMUNICATION PROTOCOLS AVAILABLE:

1. **RS485 Half Duplex**
2. **Distributed Motion (Multi-Drop with Master and Slave)**
3. **Modbus TCP / IP**
4. **ASCI via RS485 (ASCI library available upon request)**

DISTRIBUTED CONTROL
(Multi-Drop with Master and Slave)

Overview

Make sure you have the right files.

See Diagram Below

Connect D+ to D+
Connect D- to D-

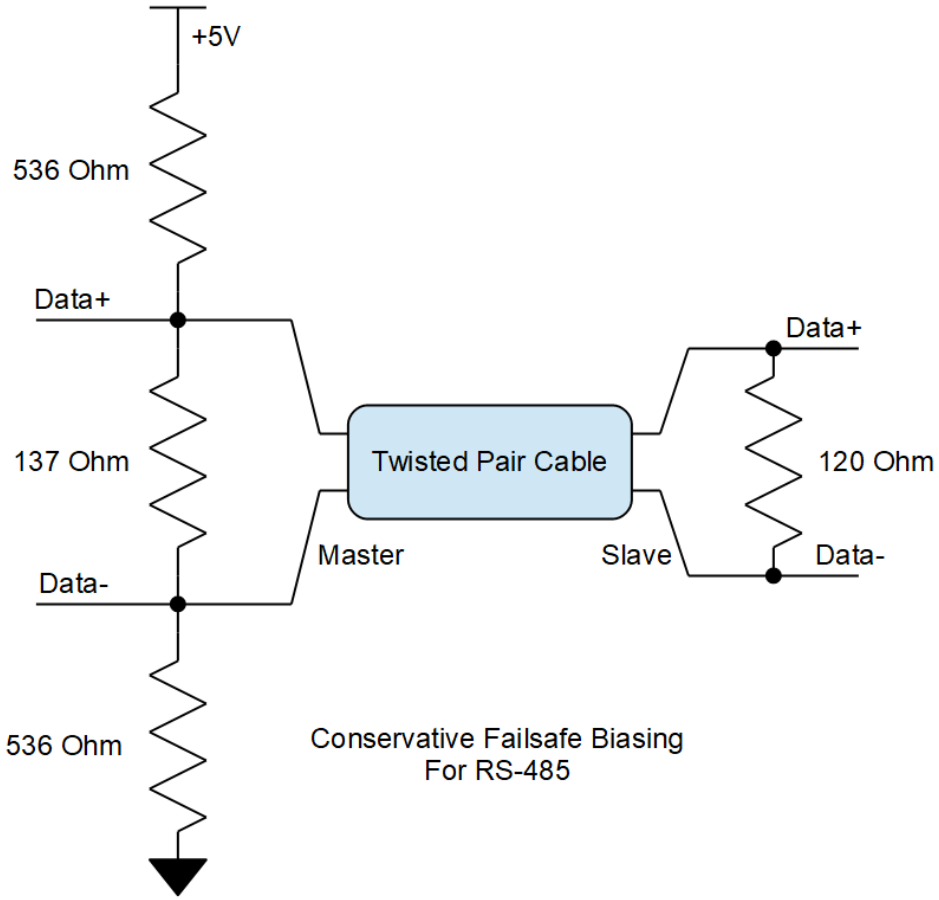
Failsafe biasing resistors recommended. Use the ground in the 3 pin connector and 5v from the 8 pin connector. Include diagram Randy created.

Slave Program to prepare the other axes (not master). Module number corresponds to node number.
Create Master program – control 2nd axis by switching to that name.

When running the master, it should detect that you are running distributed control – gives you a "compiling master" message in the status display.

Hardware
May want to add more obvious hardware like power supply, motor, etc?

1. Windows computer to run SnapTrack
2. Vector or ST484 units. They can connect to each other. You may need the breakout board for the Vector to easily access the D+ and D- pins.
3. Wires to connect each units' D+ and D- together.
   For a 3rd unit, would the "master" connect D+ and D- to all slaves? I think this is the case.
4. Failsafe biasing resistor recommendation.



Software

1. SnapTrack
2. Files

   ST486Master.enc
   RS485_Slave.enc
   Slave Program.DSM

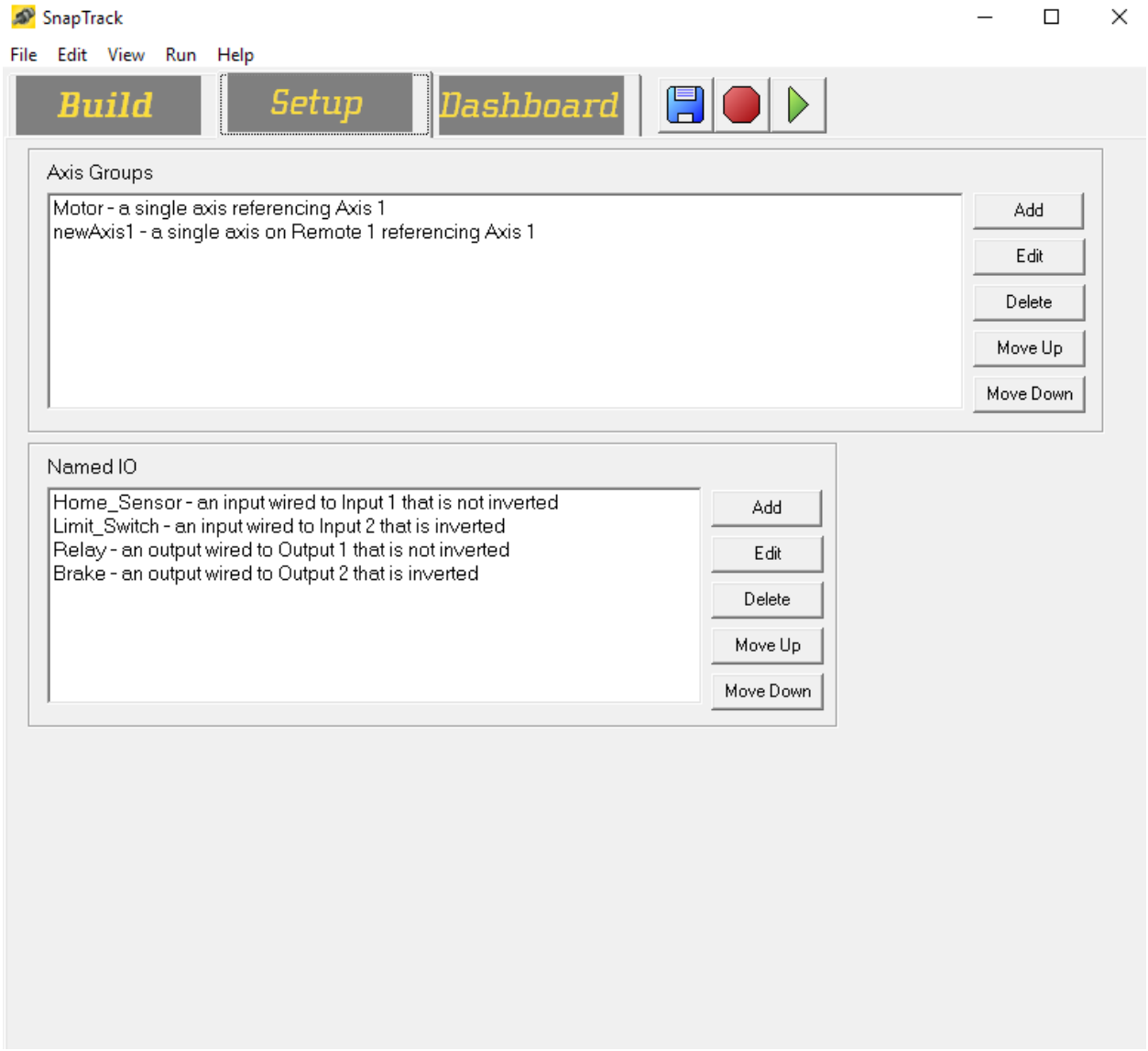
Slave Program

1. Open SnapTrack
2. Open “Slave Program.DSM” with SnapTrack
3. Run
4. Set Node #
5. Save
6. Repeat for each slave, making sure node # is unique.

Master Program

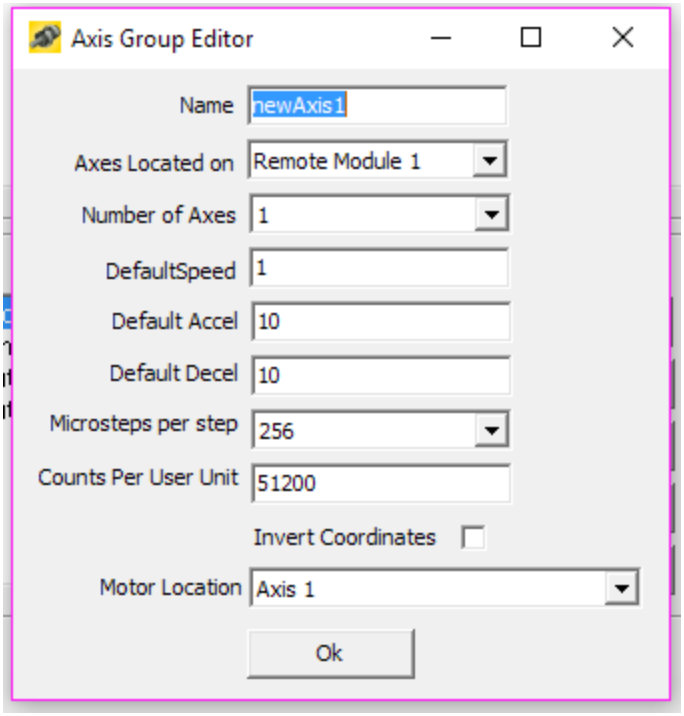
Once the slave programs are loaded into the slaves and everything is connected, you can create and run a custom program.

To create multiple axes on the master program, just add a to the Axis Groups found in the Setup Tab.



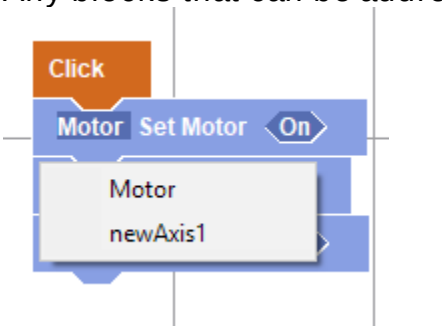
Each axis can have its own settings

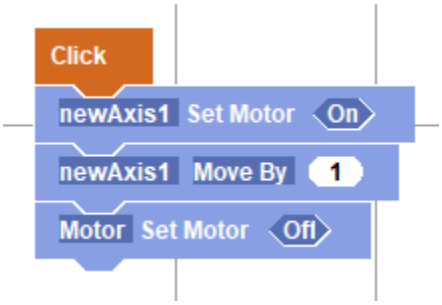
Each axis can have its own settings



Note that the added slave node # needs to match **“Axis Located on” #.**

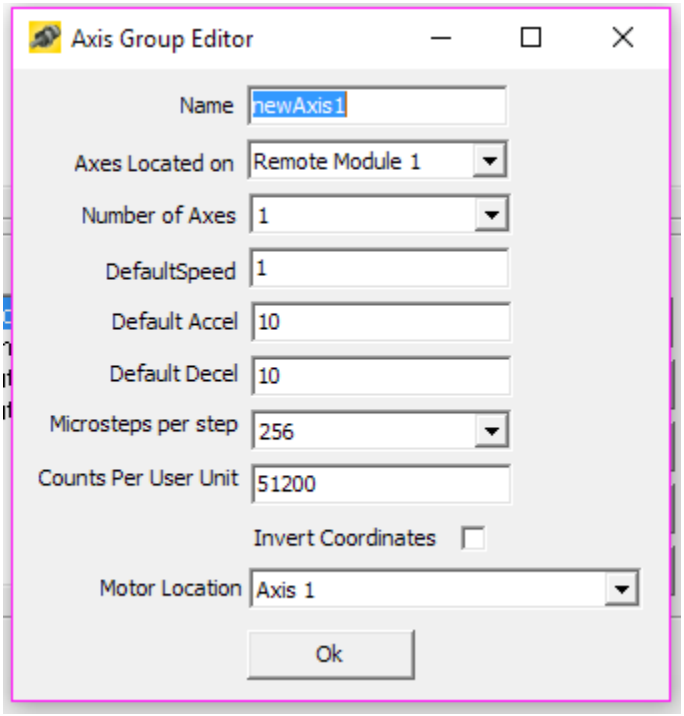
Any blocks that can be addressed to a slave has a drop down menu to specify the axis.





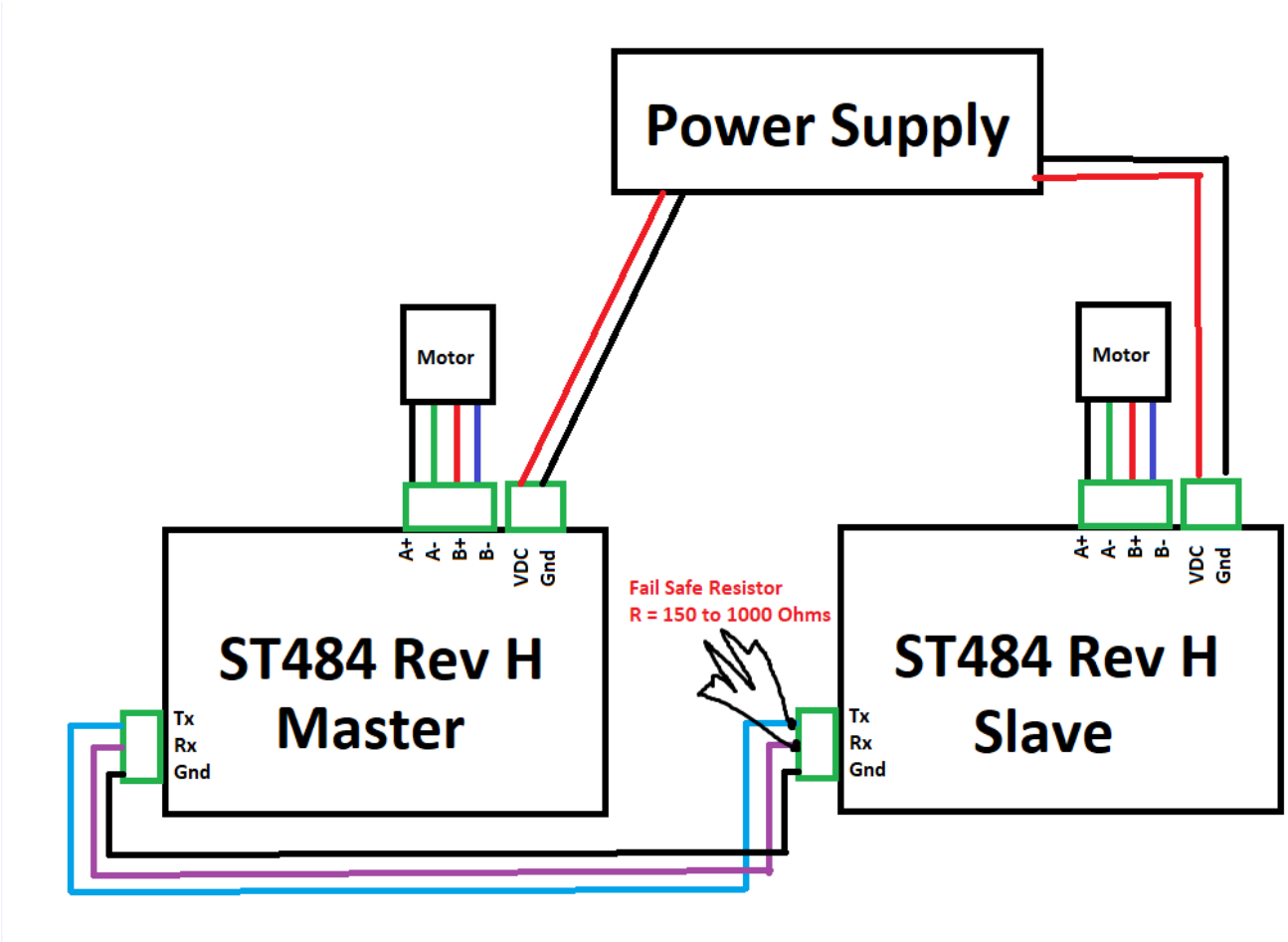
When compiling a distributed control program, one indication that SnapTrack **recognizes this is it will state in the status bar "Compiling Master.." rather than just "Compiling".**

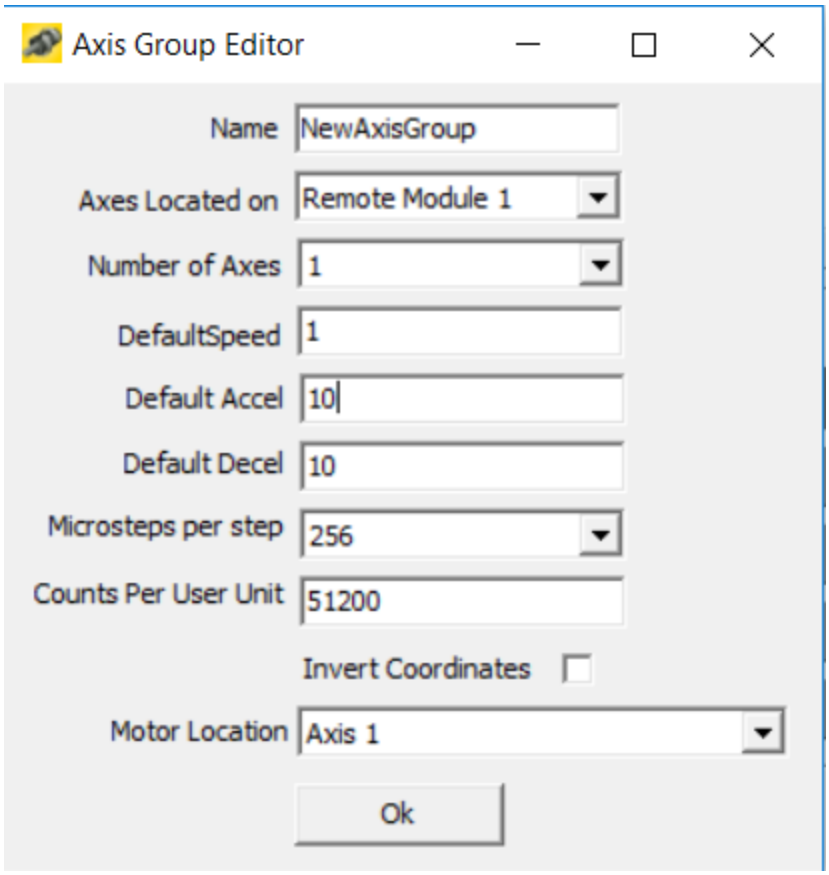
For example……



Note that the added slave node # needs to match "Axis Located on" #.

Typical Connectivity Diagram




Axis Group Editor
Name
NewAxisGroup
Axes Located on
Remote Module 1
Number of Axes
1
DefaultSpeed
1
Default Accel
10
Default Decel
10
Microsteps per step
256
Counts Per User Unit
51200
Invert Coordinates
Motor Location
Axis 1
Ok

RS485 HALF DUPLEX IMPLEMENTATION

Modbus TCP / IP