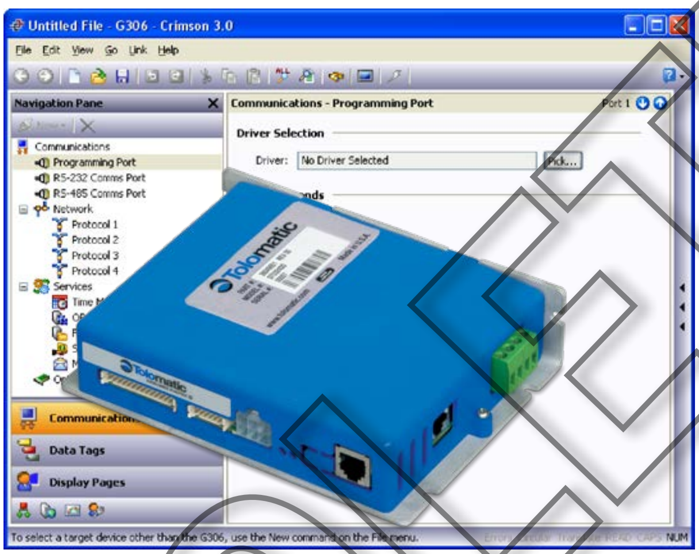
Tolomatic™
EXCELLENCE IN MOTION

Technical Note

Setting up the Red Lion® Crimson Software for Modbus TCP Communication to Tolomatic's ACS Drive



Driver: No Driver Selected

To select a target device other than the G306, use the New command on the File menu.

Contents

1 System Requirements 2
2 Cabling 2
3 Setting up Tolomatic ACS Drive 2
4 Setup Redlion HMI 3
5.1 Communication 4
5.2 Data Tags 5
5.3 Display Pages 6
5.4 Download to HMI 8
5 Programming HMI 8
5.1 Entering a Motion Profile 8
5.2 Perform an Absolute Move 8
5.3 Button Action 9
5.4 Page Navigation 10
6 Complex Programs 11
6.1 Incremental Moves 11
6.2 Index Moves 12

1 System Requirements

Red Lion HMI HARDWARE	MINIMUM REQUIREMENTS
Processor	Refer to OS Requirements
Software Requirements	Windows XP with Service Pack 2 (or above)
RAM	Refer to OS RAM requirements
Disk Space	150MB needed for intallation plus an OS disk requirements

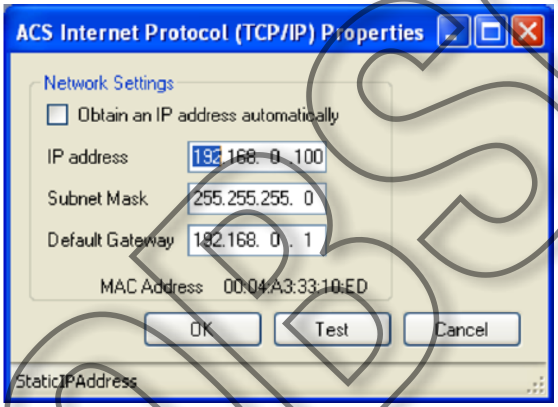
Tolomatic's ACS Drive used the Tolomatic Motion Interface (TMI) software which is dependant on .NET4. Please reference the TMI User Guide 3600-4164 for minimum requirements.

2 Cabling

Connect the HMI power connector to an appropriate power supply. Next, connect the HMI to your computer though a USB cable. Lastly, connect the HMI to the ACS drive using an Ethernet cable and the Ethernet ports.

3 Setting Up the Tolomatic ACS Drive IP Address

Run Tolomatic Motion Interface and connect to the ACS drive. Choose the Tools -> Ethernet menu selection or press the Configure Ethernet button in the toolstrip.



ACS Internet Protocol (TCP/IP) Properties

Network Settings

□ Obtain an IP address automatically

IP address: 192.168. 0 .100

Subnet Mask: 255.255.255. 0

Default Gateway: 192.168. 0 . 1

MAC Address: 00:04:A3:33:10:ED

StaticIPAddress

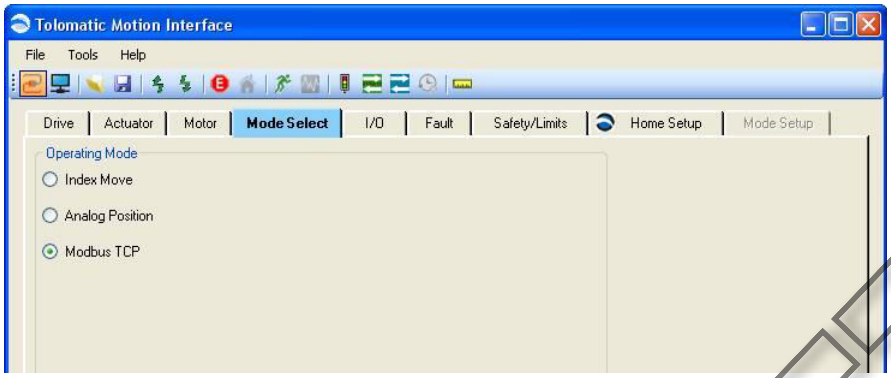
Figure 1: Network Settings

Enter the IP address, Subnet Mask and Default Gateway parameters for your network. (The values shown above are the factory default values for the ACS drive. Typically at least the IP address and Default Gateway will need to be changed for your network.)

Configure ACS drive for the desired Actuator and Motor.

Configure ACS drive for Modbus TCP control.

Configure ACS drive with desired 16-position index moves.



Operating Mode

- ○ Index Move
- ○ Analog Position
- ● Modbus TCP

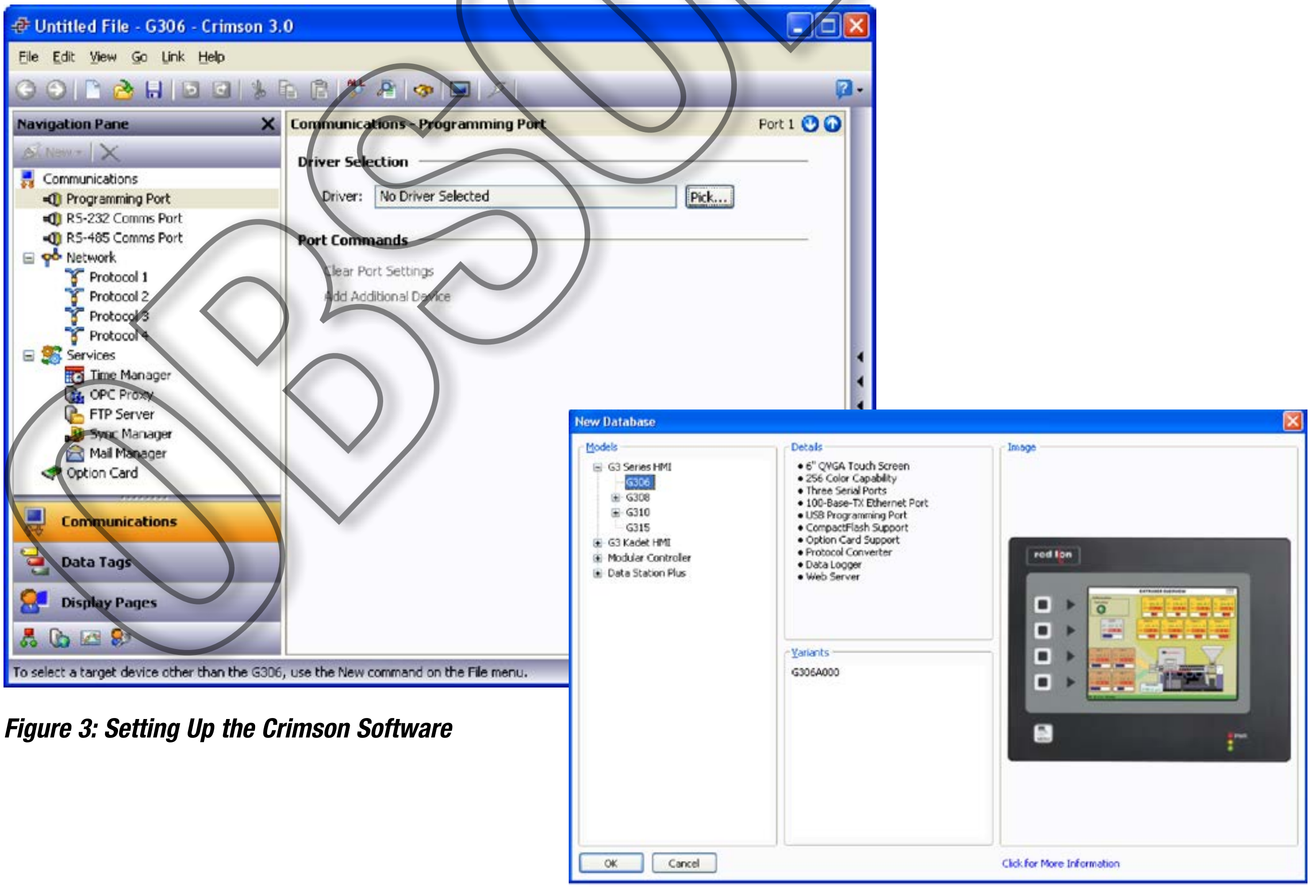
Figure 2: Select Modbus TCP in the Mode Select Tab

4 Setting Up the Red Lion HMI Using Crimson Software

Download and install Crimson 3.0 from the Red Lion download website:

http://www.redlion.net/Support/Downloads/SoftwareLibrary/Crimson3.html

Open Crimson 3.0 software. Select 'File' -> 'New' to create a new database project. Select the model of the HMI. In this example a G306 is used. Click 'OK'.



Driver: No Driver Selected

To select a target device other than the G306, use the New command on the File menu.

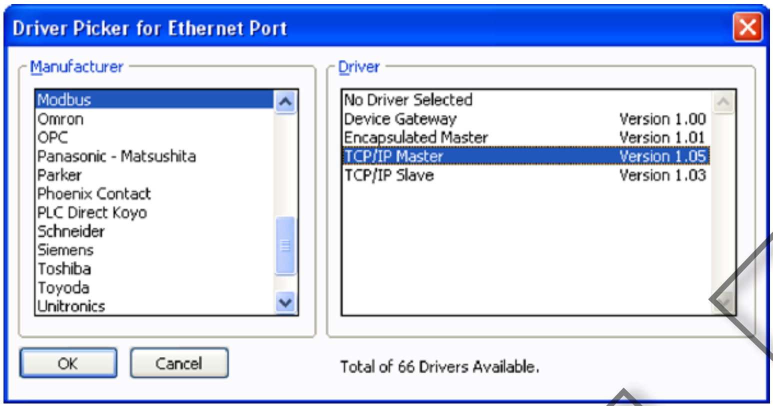
- 6" QVGA Touch Screen
- 256 Color Capability
- Three Serial Ports
- 100-Base-TX Ethernet Port
- USB Programming Port
- CompactFlash Support
- Option Card Support
- Protocol Converter
- Data Logger
- Web Server

Variants: G306A000

Figure 3: Setting Up the Crimson Software

4.1 Communication

Next, select ‘Protocol 1’ in the Navigation Pane. Click on ‘Pick’ under Driver Selection. A list of all communication options appears. Select ‘Modbus’ and ‘TCP/IP Master’ then click ‘OK’.



Driver Picker for Ethernet Port

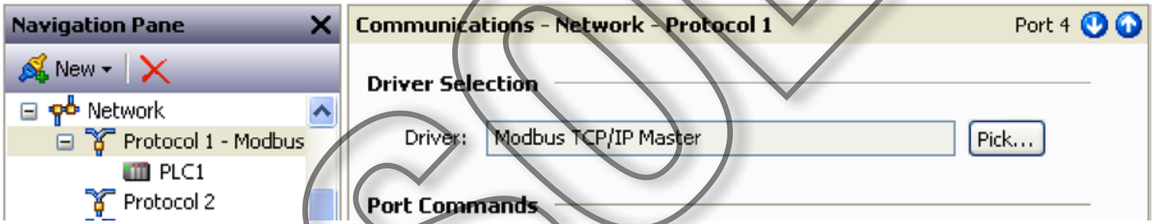
Manufacturer: Modbus (selected), Omron, OPC, Panasonic - Matsushita, Parker, Phoenix Contact, PLC Direct Koyo, Schneider, Siemens, Toshiba, Toyoda, Unitronics

Driver	
No Driver Selected	
Device Gateway	Version 1.00
Encapsulated Master	Version 1.01
TCP/IP Master (selected)	Version 1.05
TCP/IP Slave	Version 1.03

Total of 66 Drivers Available.

Figure 4: Driver Selection

The screen should now look like Figure 5.



Navigation Pane: Network, Protocol 1 - Modbus, PLC1, Protocol 2

Communications - Network - Protocol 1 — Port 4

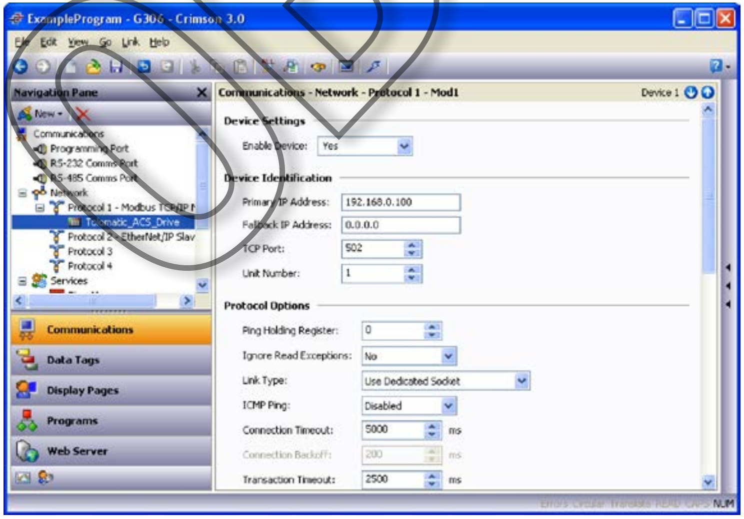
Driver Selection

Driver: Modbus TCP/IP Master

Port Commands

Figure 5: Navigation and Communications Screen After Driver Selection

Select ‘PLC1’ in the Navigation Pane and change the name of the drive. This drive is named Tolomatic_ACS_Drive. Select the drive and enter communication information including the IP address of the ACS Drive.



ExampleProgram - G306 - Crimson 3.0

Navigation Pane: Communications, Programming Port, RS-232 Comms Port, RS-485 Comms Port, Network, Protocol 1 - Modbus TCP/IP M, Tolomatic_ACS_Drive (selected), Protocol 2 - EtherNet/IP Slav, Protocol 3, Protocol 4, Services

Communications - Network - Protocol 1 - Mod1 — Device 1

Device Settings

Enable Device: Yes

Device Identification

Primary IP Address: 192.168.0.100

Fallback IP Address: 0.0.0.0

TCP Port: 502

Unit Number: 1

Protocol Options

Ping Holding Register: 0

Ignore Read Exceptions: No

Link Type: Use Dedicated Socket

ICMP Ping: Disabled

Connection Timeout: 5000 ms

Connection Backoff: 200 ms

Transaction Timeout: 2500 ms

Figure 6: Changing the Drive Name

4.2 Data Tags

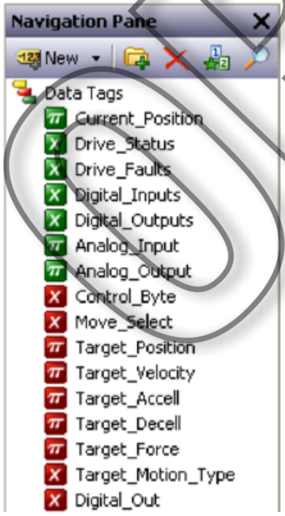
Create data tags with addresses that correspond to the ACS drive's Modbus Register Addresses.

The Modbus Register Addresses are summarized in the following table. Pay close attention to changes in settings, and add them all to Data Tags. The tags used in this tutorial are identified by bold font.

TAG NAME	ADDRESS	MANIPULATION	TREAT AS
READ ONLY			
Current_Position	304001.REAL	Reverse Words	Floating Point
Drive_Status	304003.LONG	Reverse Words	Signed Integer
Drive_Faults	304005.LONG	Reverse Words	Signed Integer
Digital_Inputs	304007.LONG	Reverse Words	Signed Integer
Digital_Outputs	304009.LONG	Reverse Words	Signed Integer
Analog_Input	304011.REAL	Reverse Words	Floating Point
Analog_Output	304013.REAL	Reverse Words	Floating Point
READ/WRITE			
Control_Byte	414001.WORD	None	Unsigned Integer
Move_Select	414002.WORD	None	Unsigned Integer
Target_Postion	414003.REAL	Reverse Words	Floating Point
Target_Velocity	414005.REAL	Reverse Words	Floating Point
Target_Accel	414007.REAL	Reverse Words	Floating Point
Target_Deccel	414009.REAL	Reverse Words	Floating Point
Target_Force	414011.REAL	Reverse Words	Floating Point
Target_Motion_Type	414014.WORD	None	Unsigned Integer
Digital_Output	414016.WORD	None	Unsigned Integer

NOTE: Refer to the Tolomatic Mobdus TCP Programmer's Guide 3600-4169 for information on faults and status bits.

After all of the tags are entered, the names, colors, and symbols should look like Figure 7.



Data Tags
- Current_Position
- Drive_Status
- Drive_Faults
- Digital_Inputs
- Digital_Outputs
- Analog_Input
- Analog_Output
- Control_Byte
- Move_Select
- Target_Position
- Target_Velocity
- Target_Accell
- Target_Decell
- Target_Force
- Target_Motion_Type
- Digital_Out

Figure 7: Data Tags

4.3 Display Pages

The tags are now ready to be used in the display pages. First, select the ‘Display Pages’ tab on the Navigation Pane. Click the icon in the upper left to add a new page. An image of the Red Lion HMI screen should appear. Right click this new page and select ‘Rename’ to give it a useful name. Now select ‘Data Tags’ on the ‘Resource Pane’. Your screen should look like Figure 8.

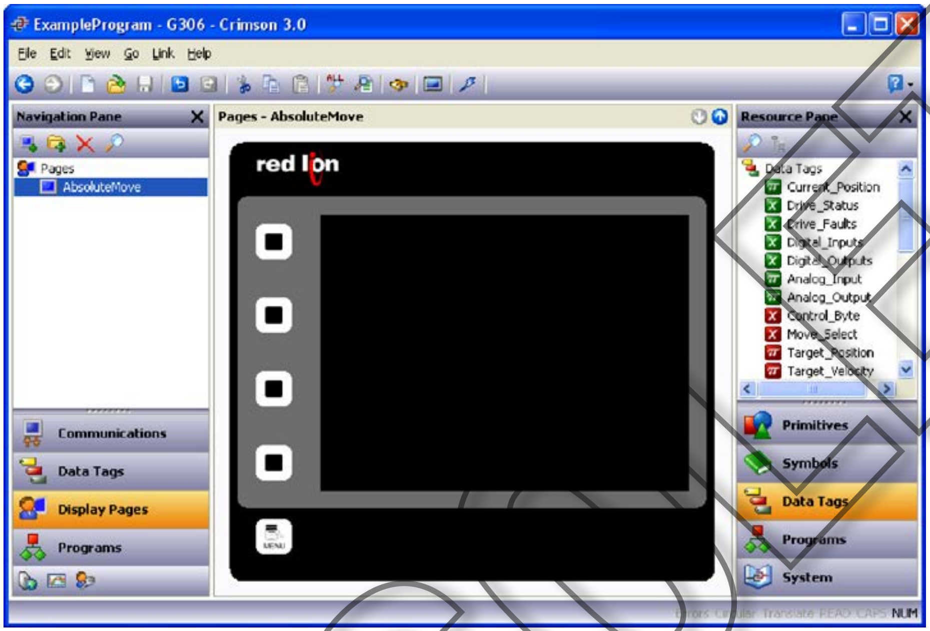


Figure 8: Using Data Tags in the Display Pages

Place a tag to show the current position value. To do this, either double-click on the tag in the Resource Pane to place it or click on the tag and drag it onto the page. Click-and-hold on the tag to move it around the screen.

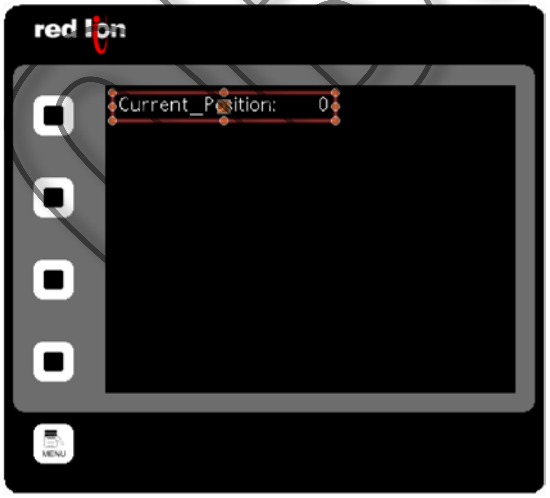
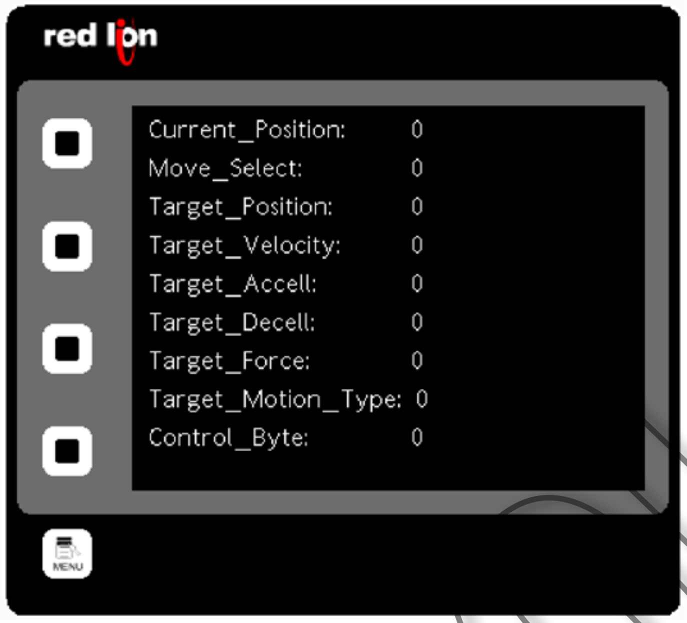


Figure 9: Placing a Data Tag

Double click on the tag to access its properties. Nothing needs to be changed in the properties for this tag.

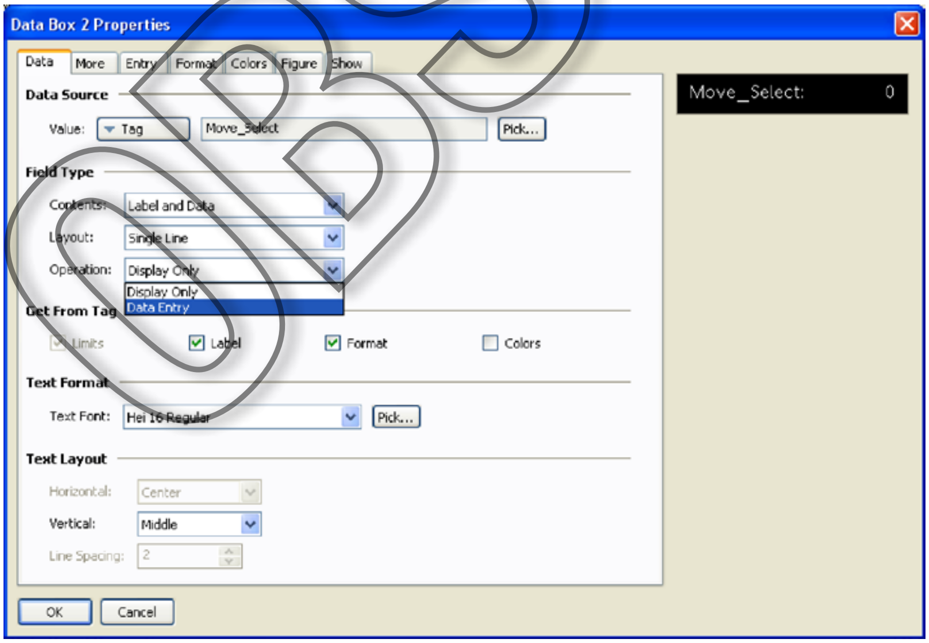
Add additional tags as necessary to command the drive to make an absolute move. The tags needed to setup a motion profile are shown in Figure 10. Stretch them out horizontally so they are aligned.



```
Current_Position:      0
Move_Select:           0
Target_Position:       0
Target_Velocity:       0
Target_Accell:         0
Target_Decell:         0
Target_Force:          0
Target_Motion_Type:  0
Control_Byte:          0
```

Figure 10: Setting up a Motion Profile

For the remaining tags, double-click the tag to see its properties. Under 'Field Type' -> 'Operation' select 'Data Entry'.



Data Box 2 Properties

Data Source
Value: Tag | Move_Select

Field Type
Contents: Label and Data
Layout: Single Line
Operation: Display Only
Display Only
Data Entry

Get From Tag
☑ Limits ☑ Label ☑ Format □ Colors

Text Format
Text Font: Hei 16 Regular

Text Layout
Horizontal: Center
Vertical: Middle
Line Spacing: 2

Move_Select: 0

Figure 11: Editing Data Tag Values

4.4 Download to HMI

Be sure to save your project first, and then download the project to the HMI. Go to menu item 'Link' -> 'Update' or click the icon on the toolbar. If the program downloads correctly there will be no pop-up messages and the page created should show on the HMI screen.

5 Programming the HMI

To perform a move on the ACS drive, a motion profile must be assigned using data tags, and a button must be added to the display to call an action upon being pressed.

5.1 Entering a motion profile

Power-on the Tolomatic ACS Drive and verify it is connected to the HMI with the Ethernet cable. When communication begins, the tag values should change from '---' to some number. Now values for a motion profile can be entered by touching the tag on the HMI screen. A popup keypad should appear for entering numbers.

5.2 Perform an Absolute Move

Enter a motion profile similar to the one in the following table. The parameters should be kept less than the maximum ratings of the actuator.

DATA TAG	VALUE (UNITS)	MAX ALLOWABLE (UNITS)
Move_Select	0 (for immediate move)	16 (0 for immediate move, 1-16 for index)
Target_Position	30.0 (mm)	Maximum stroke length
Target_Velocity	15.0 (mm/sec)	Refer to actuator documentation
Target_Accell	100.0 (mm/sec/sec)	Refer to actuator documentation
Target_Decel	100.0 (mm/sec/sec)	Refer to actuator documentation
Target_Force	100.0 (%)	10-100%
Target_Motion_Type	0 (for absolute move)	0 (for absolute), 1 (for incremental forward), 2 (for incremental reverse)

The 'Control_Byte' data tag must be manipulated to instruct the drive to execute moves. First, the actuator must find its home position. The Control Byte tag must be set to '5'. This sets the drive enable bit and the home bit. The drive should immediately home when a '5' is entered.

An absolute move is made by writing '3' to the Control Byte. The actuator will move to the value in 'Target_Position'. No motion will occur if the Target Position value has not changed since the last move. Writing a '1' to the Control Byte clears the Start Motion bit and keeps the enable bit set. Writing a '3' sets the start motion bit. For motion to occur, the Start Motion bit must transition from low to high while the drive is still enabled.

5.3 Button Action

Next, an 'Enable' button will be created to enable the drive when it is pressed.

Start by creating another page by selecting Display Pages tab on the Navigation Pane and clicking the new page icon.

Some button options are under ‘Primitives’ on the Resource Pane. Select the shape to add, and drag it onto the page. Figure 12 shows a green bevel button.

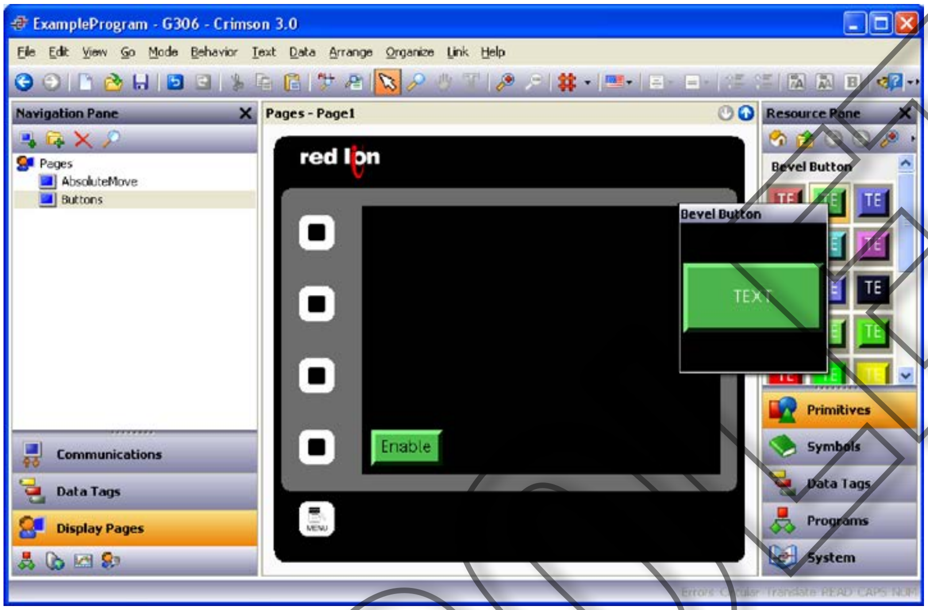
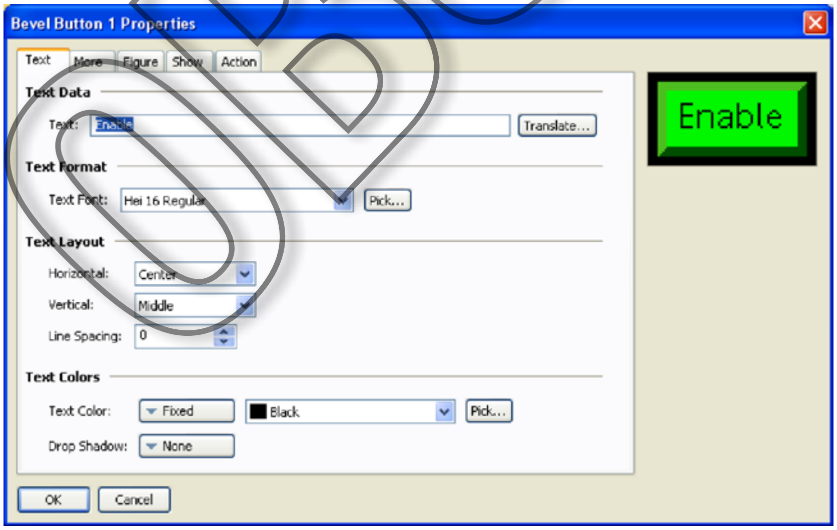


Figure 12: Adding Buttons

To add text, select the shape and select ‘Text’ -> ‘add’ from the menu. Modify the color and size for the button’s purpose. To modify the text color and size, select the button and go to ‘Text’ in the menu and ‘Properties’. This button’s text was changed to black.


Bevel Button 1 Properties

Text | More | Figure | Show | Action

Text Data

Text: Enable

Text Format

Text Font: Hei 16 Regular

Text Layout

Horizontal: Center

Vertical: Middle

Line Spacing: 0

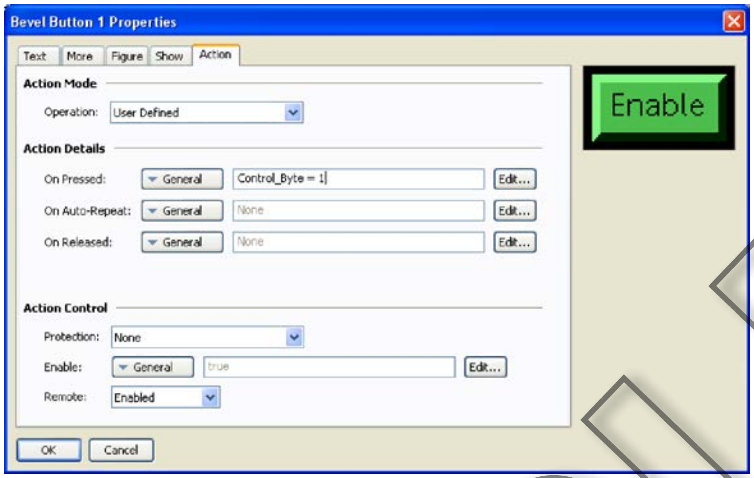
Text Colors

Text Color: Fixed, Black

Drop Shadow: None

Figure 13: Modifying Buttons

To make this button perform an action when it is pressed, click on the ‘Action’ tab in the button properties window. If there is no Action tab, then it can be added by highlighting the button and select ‘Behavior’ -> ‘Add Action’ from the menu. Under the ‘Action’ tab and ‘Action Mode’ -> ‘Operation’, select ‘User Defined’. Now write a simple expression in the textbox next to ‘On Pressed’. Writing a ‘1’ to the Control_Byte tag sets the enable bit and clears the other controls.


Bevel Button 1 Properties

Action Mode
Operation: User Defined

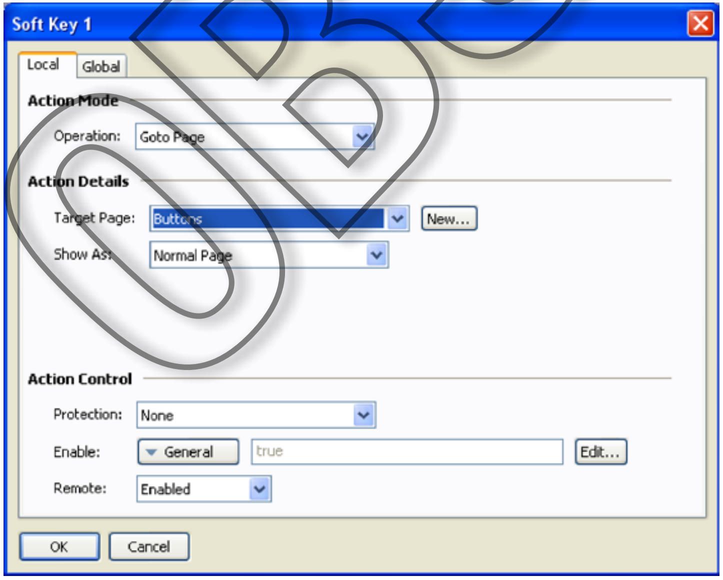
Action Details
On Pressed: General | Control_Byte = 1
On Auto-Repeat: General | None
On Released: General | None

Action Control
Protection: None
Enable: General | true
Remote: Enabled

Figure 14: Creating Button Actions

5.4 Page Navigation

Once multiple pages are created, create the navigation to the buttons pages. On the display page, there the 4 buttons on the left-hand side of the screen. There is also a fifth ‘Menu’ button. Double click on the top button (Soft Key 1). In the properties, select ‘Goto Page’ for the operation. Select ‘Buttons’ for the target page to navigate to the Buttons display page. Double click the top button, and repeat the above steps selecting ‘AbsoluteMove’ as the target page. The Soft Key button’s function must be specified for each display page.


Soft Key 1

Action Mode
Operation: Goto Page

Action Details
Target Page: Buttons
Show As: Normal Page

Action Control
Protection: None
Enable: General | true
Remote: Enabled

Figure 15: Creating Button Navigation

Click ‘OK’, save the program, and download to the HMI. The Enable buttons should be functional.

Try adding more buttons in the same manner. Add 'Disable', 'Home', and 'E-Stop' buttons. The expressions for each button are as follows:

BUTTON	EXPRESSION
Enable	Control_Byte = 1
Disable	Control_Byte = 0
Home	Control_Byte = 5
E-Stop	Control_Byte = 9

Here is an example of adding other buttons.

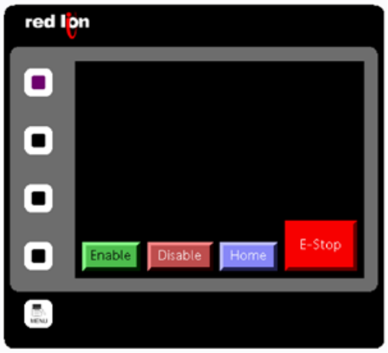


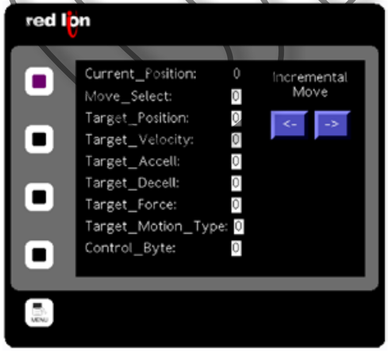
Figure 16: Button Examples

6 Complex Programs

The Red Lion HMI has programming ability. These next two examples will show how to perform more advanced moves.

6.1 Incremental Moves

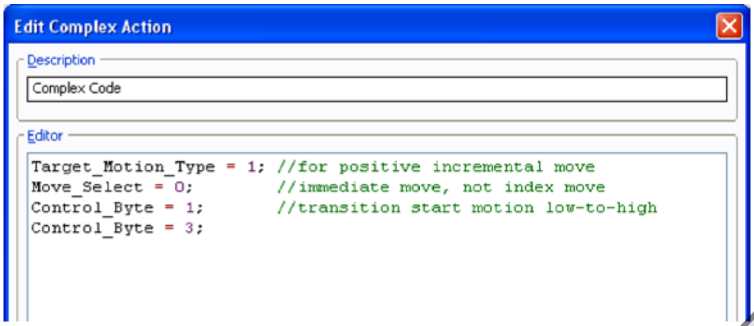
Add two more buttons to the Absolute Moves page for positive and negative incremental moves. Place a label to identify the action to be performed.



Current_Position: 0
Move_Select: 0
Target_Position: 0
Target_Velocity: 0
Target_Accell: 0
Target_Decell: 0
Target_Force: 0
Target_Motion_Type: 0
Control_Byte: 0

Figure 17: Adding Positive and Negative Incremental Move Buttons

In button properties, select a 'User Defined' operation and select 'Complex' under 'On Pressed'. A window will pop up to edit complex actions. This is the place to write several expressions (program code). The motion type is set to '1' for positive incremental moves for the right arrow button. Click ‘OK’ when the code in the following figure is entered correctly.



Complex Code

```
Target_Motion_Type = 1; //for positive incremental move
Move_Select = 0;        //immediate move, not index move
Control_Byte = 1;       //transition start motion low-to-high
Control_Byte = 3;
```

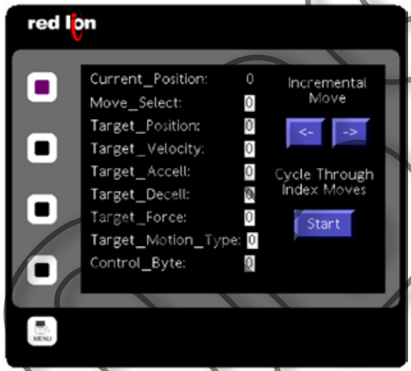
Figure 18: Edting Complex Actions

Add the same complex action to the left arrow button, but make the motion type ‘2’ for negative incremental moves. Download the program to the HMI and verify it is working. Be sure to have non-zero values for the Target_Postion, Velocity, Accel, Decel, and Force date tags.

6.2 Index Moves

It may be useful to program the HMI to step through all 16 of the index moves. Setup the 16 move profiles using TMI and save them to flash memory.

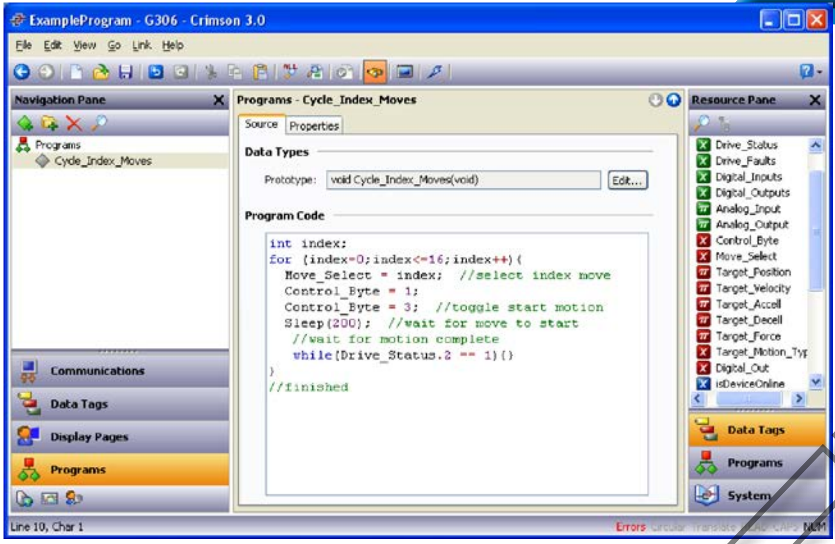
Make a ‘Start’ button. Add text describing its function.



Current_Position: 0
Move_Select: 0
Target_Position: 0
Target_Velocity: 0
Target_Accell: 0
Target_Decell: 0
Target_Force: 0
Target_Motion_Type: 0
Control_Byte: 0

Figure 19: Index Move Step Through Setup

In the Navigation Pane, select the ‘Programs’ tab. Select the icon in the upper left for a new program. Rename it to ‘Cycle_Index_Moves’. Add the code shown in the next figure. When the code is entered, click the ‘Translate’ icon on the toolbar. Correct any syntax errors.



Data Types

Prototype: void Cycle_Index_Moves(void) Edit...

Program Code

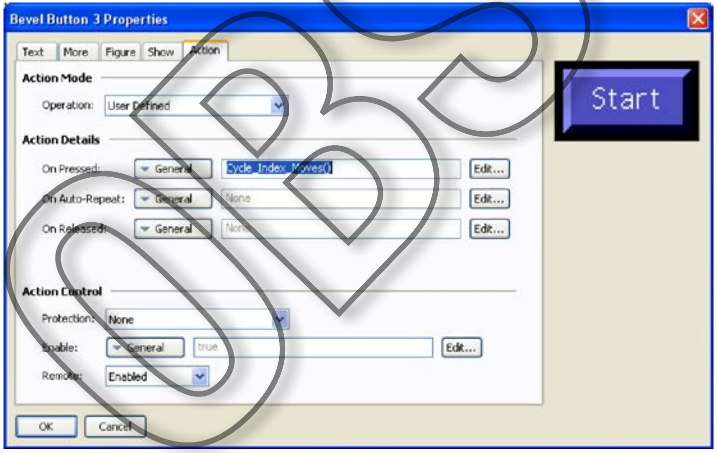
```
int index;
for (index=0;index<=16;index++){
  Move_Select = index;  //select index move
  Control_Byte = 1;
  Control_Byte = 3;  //toggle start motion
  Sleep(200);  //wait for move to start
   //wait for motion complete
   while(Drive_Status.2 == 1){}
}
//finished
```

Figure 20 : Creating a Program to Cycle Index Moves

This program loops through each of the 16 index moves (1-16). The counter is an integer called 'index'. The 'Move_ Select' tag tells the drive which move to make. Next, toggle the Start Motion bit to start the move. Insert a short delay so that the move is started before the next instruction executes. Insert a 'while' loop that will wait until motion is complete before executing the next move. Motion complete is indicated by bit 2 of the 'Drive_Status' tag. The program will exit when index 16 is completed.

Click on properties on the programs tab. Set the execution task to 'Background 1'. This will prevent the HMI from thinking it is stuck in a loop and resetting power. It also helps to update tags displayed on the screen.

Now that the program is written, it can be added to the button action. Add the name of the program ending with open and closed braces: Cycle_Index_Moves( ).



Action Mode

Operation: User Defined

Action Details

On Pressed: General Cycle_Index_Moves() Edit...

On Auto-Repeat: General None Edit...

On Released: General None Edit...

Action Control

Protection: None

Enable: General true Edit...

Remote: Enabled

Figure 21: Adding a Program to the Button Action¸

Click 'OK', save project, and then click 'Update' icon to download to the HMI. The new button should execute the program when pressed.