



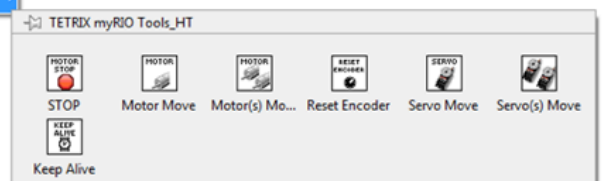
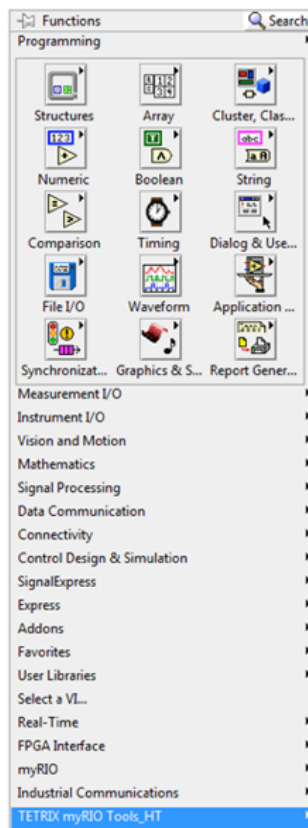
## TETRIX® myRIO Programming Tools for LabVIEW

### Overview

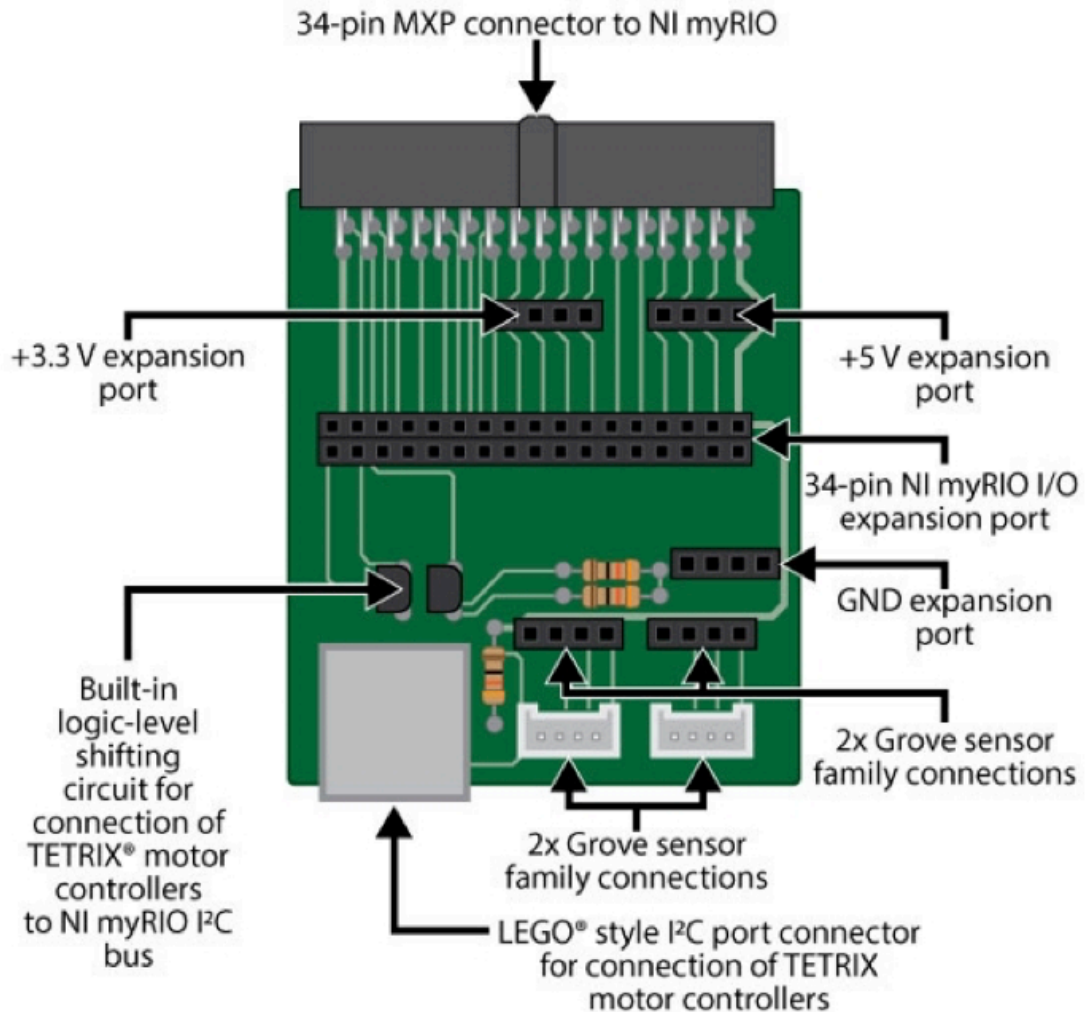
This document provides guidance on installation of the TETRIX myRIO programming tools for *LabVIEW*, and a brief overview of each VI. You must first have installed *LabVIEW* and configured your myRIO. Please reference the instructions and resources included with myRIO for help on completing these items. After that has been done, download and unzip the file “TETRIX\_HT WRO VI’s” to a folder on your computer. Open the folder and find the VI Package file “pitsco\_education\_lib\_tetrix\_myrio\_tools\_ht.” There will be a version number at the end of the filename; however, that can change as new releases become available.

### Installing the programming tools

To begin installation, double click the VI package file. If you have not already installed the VI Package manager software, you will be prompted to do so. Simply follow the prompts on the screen to guide you through the process. After the VI Package manager software is installed, double click the VI Package file once more to start the installation of the tools. Follow the prompts on the screen to complete the installation. After the installation is finished, the installed TETRIX myRIO programming VIs can be located on the *LabVIEW* programming palette.



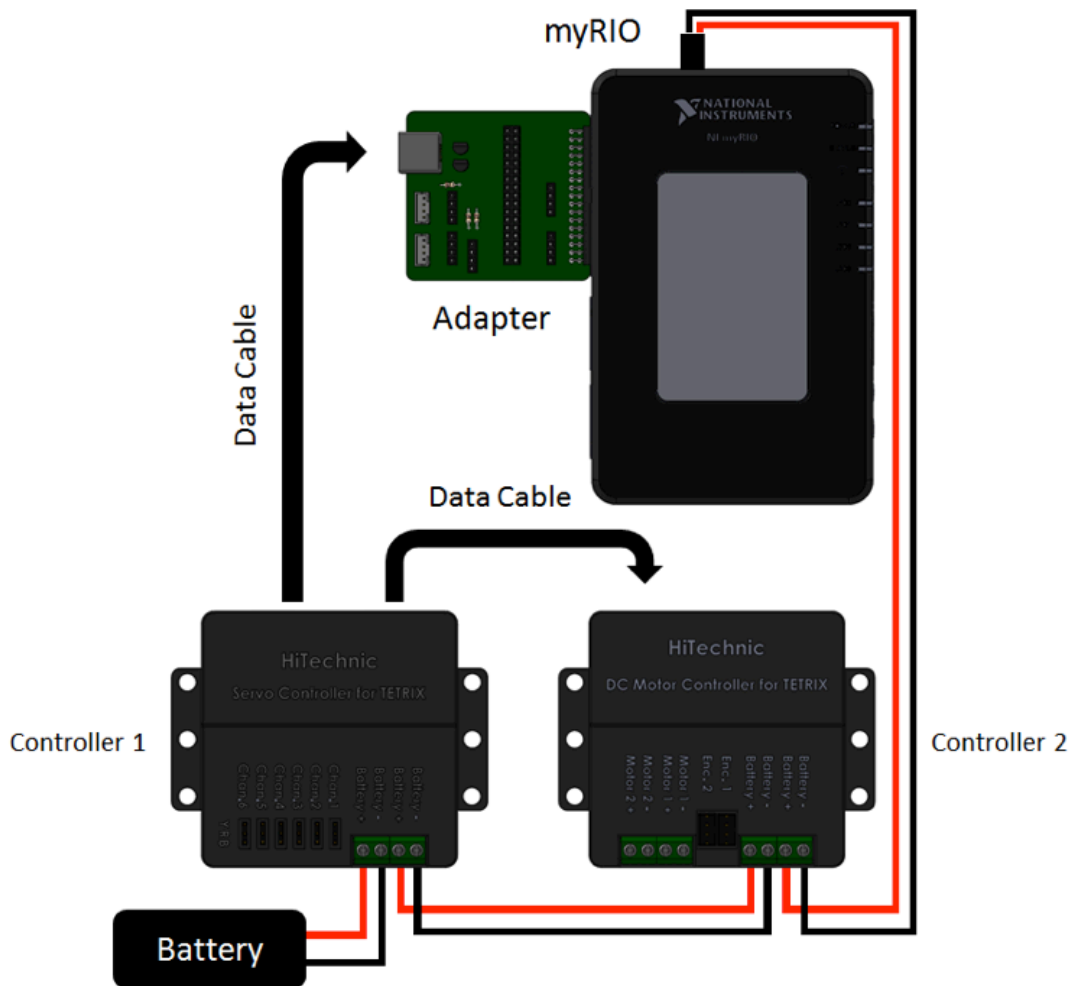
Each of these VI tools can be used to construct a *LabVIEW* program for controlling a TETRIX robot equipped with a HiTechnic motor and servo controllers and myRIO. An adapter interface board developed by Pitsco and included in each WRO TETRIX myRIO bundled kit must be used to ensure proper connection and I2C communication between the TETRIX HiTechnic motor controllers and the myRIO controller.



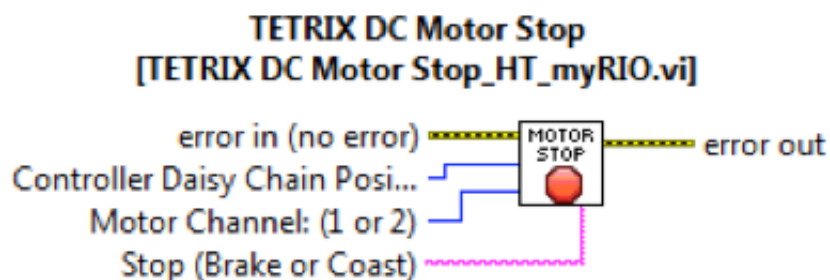
The HiTechnic controllers connect via the Pitsco adapter to the I2C bus of the myRIO controller. The motor controllers are connected in a daisy chain hookup with the first controller in the chain being designated as Position 1. Subsequent controllers will be set to Positions 2, 3, and 4. A maximum of four controllers can be connected in a single daisy chain to each myRIO MXP port.

The I2C connection cables are the LEGO NXT style cables, which are included with each HiTechnic controller. Power to the myRIO is fed from the open battery position terminals on the last controller box in the daisy chain. A DC barrel connector to terminal adapter is included in each TETRIS myRIO bundled set.

To connect power, wire a RED jumper wire between the positive terminals on the motor controller battery terminal strip and the DC barrel jack adapter, then wire a BLACK jumper wire between the negative terminals on the motor controller and DC barrel jack adapter. Plug the DC barrel jack into the myRIO power input port. This will provide battery power to the myRIO controller.

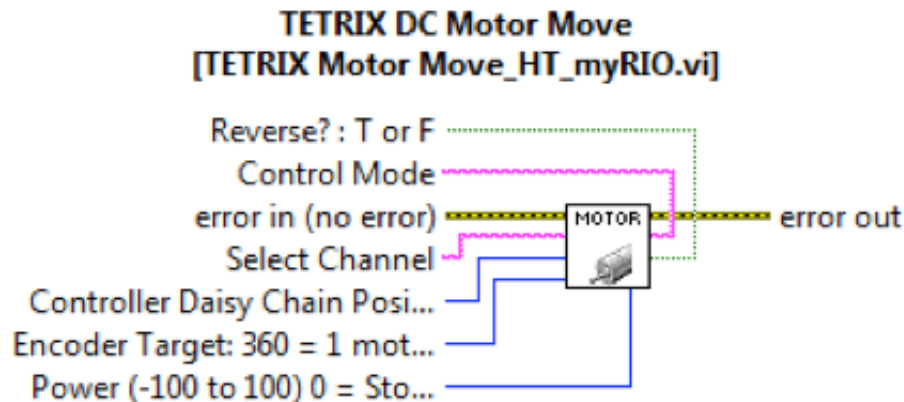


Typical Hookup Diagram. A maximum of four controllers can be daisy chained together.



Stops DC motors with a brake or coast characteristic.

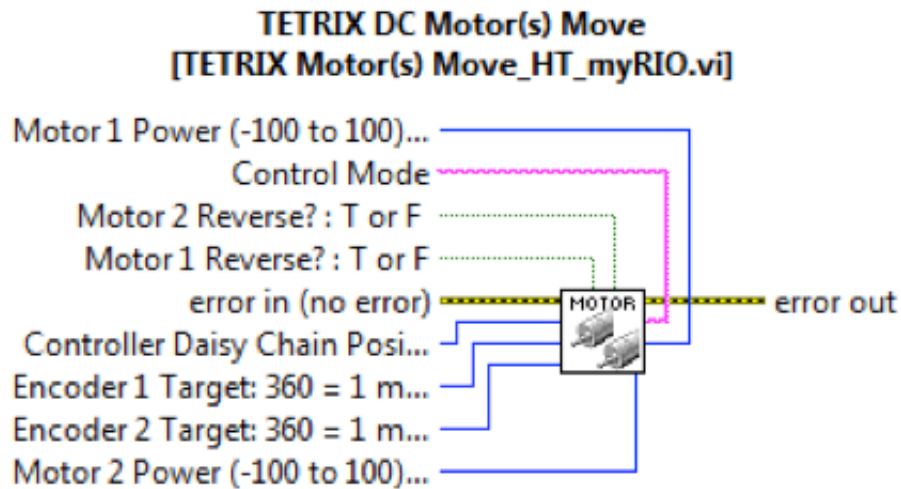
- Error in – Error data in.
- Error out – Error data out.
- Controller Daisy Chain Position –The position in the daisy chain that corresponds to the motor controller box that you wish to communicate to on the bus. Valid positions are 1, 2, 3, or 4.
- Motor Channel – Target motor channel to control (1 or 2).
- Stop – Brake is a hard stop. Coast is a coast to a stop.



Move a TETRIX DC motor with Power/Direction, Constant Speed, or Run to position mode command.

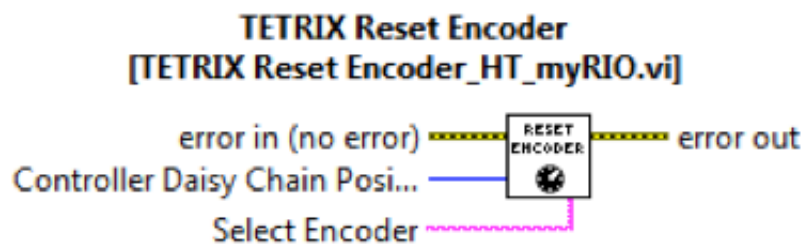
- Error in – Error data in
- Error out – Error data out
- Reverse? : True or False – Selecting true will reverse the direction of the target DC motor output channel and encoder input. This function is primarily intended to harmonize the forward and reverse directions for motors on opposite sides of a skid-steer chassis for example.
- Control Mode:
  - o Power Only – Sets the target motor channel to run in open loop mode at the power level and direction value wired to the power terminal.
  - o Constant Speed – Uses built in firmware PID which results in the target motor channel to run at a constant speed by using encoder feedback. The speed of the motor will compensate for changing loads in order to maintain a constant motor speed. Motor must be equipped with encoder. The rotation speed and direction is set by the value wired to the power terminal of the VI.
  - o Run to Position – Uses built in firmware PID which results in the target motor channel to run the motor to make the current encoder value to become equal to the target encoder value. It will do this by using a maximum rotation rate as set by the value wired to the power terminal.
- Select Channel: Target motor channel to control (Motor1 or Motor2).
- Controller Daisy Chain Position –The position in the daisy chain that corresponds to the motor controller box that you wish to communicate to on the bus. Valid positions are 1, 2, 3, or 4.
- Encoder Target: When using Run to Position Mode, this value sets the target position for which to rotate the motor and hold position. For example, one rotation of the motor shaft corresponds to a value of 360. Setting a value of 360 on this terminal will spin the corresponding DC motor one shaft rotation and hold. Setting it to 720 will result in two revolutions.

- Power: Sets the motor power level and direction for the target motor channel. Power level is represented in percent of power (0 – 100%). Direction is set by the sign of the power value. Positive rotates the motor forward; negative value rotates the motor in reverse. Entering zero will stop “brake” of the corresponding motor channel. Entering -128 will stop “coast” of the corresponding motor channel.



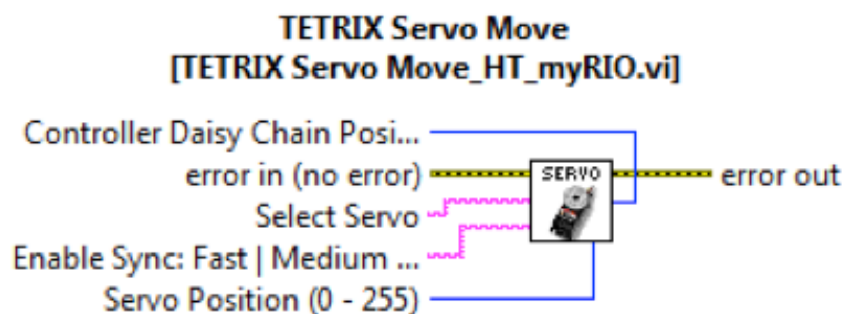
Move two TETRIX motors simultaneously using a single block VI.

- The motor(s) VI block does the same thing as the single motor VI block, only this VI will command both motors with a single I2C data write at once resulting in better start/stop syncing of the motor channels.



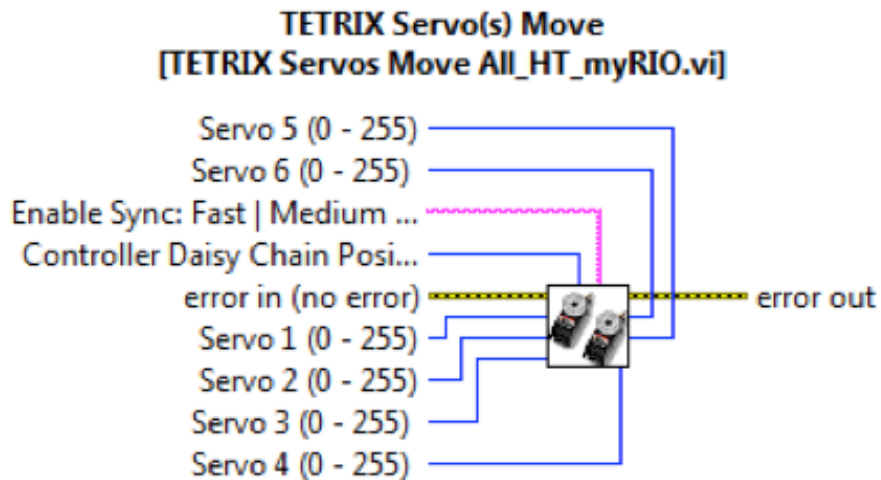
Resets the current target encoder value in firmware to zero. This value is also set to zero upon power up.

- Error in – Error data in
- Error out – Error data out
- Controller Daisy Chain Position –The position in the daisy chain that corresponds to the motor controller box you wish to communicate to on the bus. Valid positions are 1, 2, 3, or 4.
- Select Encoder: Specific encoder value to reset. Encoder 1 or Encoder 2.



Rotates a servo to the position wired to the Servo Position terminal.

- Error in – Error data in
- Error out – Error data out
- Controller Daisy Chain Position – The position in the daisy chain that corresponds to the motor controller box you wish to communicate to on the bus. Valid positions are 1, 2, 3, or 4.
- Select Servo: Servo channels (1 – 6).
- Enable Sync: Sets a step time for the servo channel which has the furthest to move. Other servo channels which are not at their designated positions yet will run at a slower rate to ensure they reach their destination positions at the same time. If “None” is selected, all servos move at their maximum rate.
- Servo Position: Set this value to the angle of rotation for which to position the servo motor. For example, a value of 128 rotates the servo shaft to its center position. A value of zero rotates the servo 90 degrees from center, while a value of 255 will rotate the motor 90 degrees from center in the opposite direction. The range of 0 – 255 represents 0 – 180 degrees of servo movement.

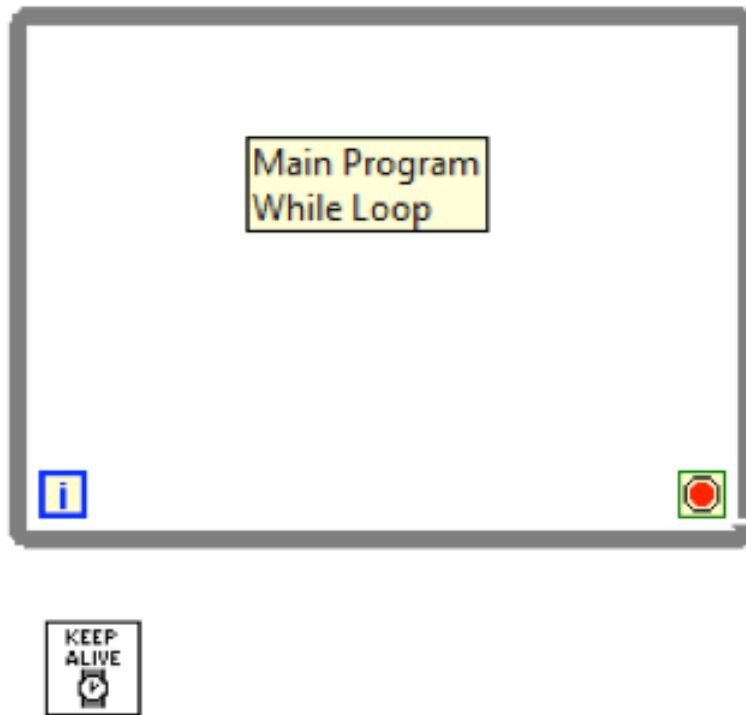


Rotates all six servo channels to a rotational position simultaneously using a single VI block.

- The Servo(s) Move VI block does the same thing as the single Servo Move VI block, only this VI will command all six servo motors with a single I2C data write at once resulting in better start/stop syncing of the motor channels.



Keeps the HiTechnic DC motor controllers internal firmware watchdog timer from expiring, causing the a shutdown of the motor channels. The watchdog timer is a safety feature to ensure rotating motors shut down in the event of a malfunction or loss of communication with myRIO. This VI block sends a byte of data to the I2C bus once every two seconds to keep the watchdog timer from timing out. To use this VI simply place it on the block diagram anywhere outside of a structured loop (see below).



## Additional Resources

TETRIX WRO building sets: [www.tetrixrobotics.com/Competition/WRO](http://www.tetrixrobotics.com/Competition/WRO)

TETRIX Robotics Home Page: [www.TETRIXRobotics.com](http://www.TETRIXRobotics.com)

myRIO information: [www.NI.com/myrio](http://www.NI.com/myrio)

Grove Sensors: [www.seeedstudio.com/depot/s/grovefamily.html](http://www.seeedstudio.com/depot/s/grovefamily.html)

Digilent Sensors and myRIO Accessories: [www.digilentinc.com](http://www.digilentinc.com)

Programming Wiki for Grove Sensors: [www.seeedstudio.com/wiki/GROVE\\_System](http://www.seeedstudio.com/wiki/GROVE_System)

WRO 2015 Bowling Game Challenge: [www.wro2015.org](http://www.wro2015.org)

**Call Toll-Free**  
**800•835•0686**

**FreeFax**  
**800•533•8104**

**Visit Us Online at**  
**shop.pitsco.com**

