Phoenix Framework

The software framework for all CTR-Electronics robot components.

See what control components make sense for your robotic needs at CTR-Electronics.

Need info? Check the Wiki | or Create an issue | Check our Store

Table of Contents

- Hardware setup
- Before you write any software!
  - Installing Phoenix Framework into PC
  - Installing Phoenix Framework into your robot
  - Test the install
- Everything is installed, can I write software now?
  - Check the web-based configuration
  - Set your device IDs
  - Update your CAN Devices
  - Pick the device names
  - Self-Test the hardware
- Where is the API?
  - C++ - How to intellisense/What header
  - Java - How to intellisense/What to import
  - LabVIEW - Where are the VIs?
- Hardware Object Model
  - Motor Controllers
    - Where to begin?
    - Open-Loop (No Sensor) Control
      - Pick your direction
      - Pick your neutral mode
      - Current limiting
      - Ramping
      - Follower
    - Limit Switches
    - Closed-Loop (Using Sensor) Control
      - Sensors
        - Why bother with sensors?
        - How do I choose the sensor?
        - How do I know the sensor works?
        - Sensor phase and why it matters
        - What are the units?

        - Closed-Loop Control Modes
    - Status Frames and how to tweak them
  - Sensors
    - Pigeon IMU
    - CANifier

- Drivetrain Model
- Servo Object Model
- CRF Firmware Version

Hardware setup

Text here.

Motor Controllers

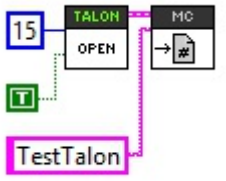
CTRE provides two CAN/PWM motor controller options...

- Victor SPX
- Talon SRX

Where to begin? Create the object.

The first step of controlling a motor controller is to instantiate the controller in your robot controller software.

If using LabVIEW, use the Open VI corresponding to your motor controller. Here we are creating an object for the Talon with device ID 15.



If using a programming language, create a TalonSRX object using the appropriate class name. `private TalonSRX m_Wheel = new TalonSRX(15);`

Regardless of the what the motor controller is used for, the next step is usually open-loop (no sensor) control. This is to ensure the mechanism is functional and that the motor and motor controller is wired correctly.

Start with the open-loop features below and configure each setting that is applicable.

Open-Loop (No Sensor) Control

These features and configurations influence the behavior of the motor controller when it is directly controlled by the robot controller.

Pick your direction

The

Pick your neutral mode

Text here.

Current limiting

Text here.

Ramping

Text here.

Follower

Text here.