# Operating pneumatic cylinders - Solenoids

There are two ways to connect and operate pneumatic solenoid valves to trigger pneumatic cylinder movement using the current control system. One option is to hook the solenoids up to a Spike relay; to learn how to utilize solenoids connected in this manner in code see the article on Relays. The second option is to connect the solenoids to a Cross the Road Electronics Pneumatics Control Module. To solenoids connected to a PCM in code, use the WPILib "Solenoid" and/or "Double Solenoid" classes, detailed below.

## Solenoid Overview

The pneumatic solenoid valves used in FRC are internally piloted valves. For more details on the operation of internally piloted solenoid valves, see this Wikipedia article. One consequence of this type of valve is that there is a minimum input pressure required for the valve to actuate. For many of the valves commonly used by FRC teams this is between 20 and 30 psi. Looking at the LEDs on the PCM itself is the best way to verify that code is behaving the way you expect in order to eliminate electrical or air pressure input issues.

Single acting solenoids apply or vent pressure from a single output port. They are typically used either when an external force will provide the return action of the cylinder (spring, gravity, separate mechanism) or in pairs to act as a double solenoid. A double solenoid switches air flow between two output ports (many also have a center position where neither output is vented or connected to the input). Double solenoid valves are commonly used when you wish to control both the extend and retract actions of a cylinder using air pressure. Double solenoid valves have two electrical inputs which connect back to two separate channels on the solenoid breakout.

## PCM Module Numbers

PCM Modules are identified by their Node ID. The default Node ID for PCMs is 0. If using a single PCM on the bus it is recommended to leave it at the default Node ID. For more information about setting PCM Node IDs see this article in the Getting Started with the 2015 Control System manual.

## Single Solenoids in WPILib

**C++**

```
Solenoid *exampleSolenoid = new Solenoid(1);
```

```
exampleSolenoid->Set(true);
exampleSolenoid->Set(false);
```

```
Java

exampleSolenoid = new Solenoid(1);

exampleSolenoid.set(true);
exampleSolenoid.set(false);
```

Single solenoids in WPILib are controlled using the Solenoid class. To construct a Solenoid object, simply pass the desired port number (assumes Node ID 0) or Node ID and port number to the constructor. To set the value of the solenoid call set(true) to enable or set(false) to disable the solenoid output.

## Double Solenoids in WPILib

```
C++

DoubleSolenoid exampleDouble = new DoubleSolenoid(1, 2);

exampleDouble->Set(DoubleSolenoid::Value::kOff);
exampleDouble->Set(DoubleSolenoid::Value::kForward);
exampleDouble->Set(DoubleSolenoid::Value::kReverse);
```

```
Java

exampleDouble = new DoubleSolenoid(1, 2);
```

## Operating pneumatic cylinders - Solenoids

```
exampleDouble.set(DoubleSolenoid.Value.kOff);
exampleDouble.set(DoubleSolenoid.Value.kForward);
exampleDouble.set(DoubleSolenoid.Value.kReverse);
```

Double solenoids are controlled by the DoubleSolenoid class in WPILib. These are constructed similarly to the single solenoid but there are now two port numbers to pass to the constructor, a forward channel (first) and a reverse channel (second). The state of the valve can then be set to kOff (neither output activated), kForward (forward channel enabled) or kReverse (reverse channel enabled).