Operating a compressor for pneumatics

The Pneumatics Control Module from Cross the Road Electronics allows for integrated closed loop control of a compressor. Creating any instance of a Solenoid or Double Solenoid object will enable the Compressor control on the corresponding PCM. **The Compressor object is only needed if you want to have greater control over the compressor or query compressor status.**

Instantiating, Starting and Stopping a Compressor

```
C++

Compressor *c = new Compressor(0);

c->SetClosedLoopControl(true);
c->SetClosedLoopControl(false);
```

```
Java

Compressor c = new Compressor(0);

c.setClosedLoopControl(true);
c.setClosedLoopControl(false);
```

To use the Compressor class create an instance of the Compressor object by passing in the PCM Node ID (default 0). For more information about PCM Node IDs see the Solenoid article and the Updating and Configuring Pneumatics Control Module and Power Distribution Panel article.

The compressor closed loop control can be enabled and disabled by using the SetClosedLoopControl() method. When closed loop control is enabled the PCM will automatically turn the compressor on when the pressure switch is closed (below the pressure threshold) and turn it off when the pressure switch is open (~120PSI). When closed loop control is disabled the compressor will not be turned on.

Compressor Status

C++

```
bool enabled = c->Enabled();
bool pressureSwitch = c->GetPressureSwitchValue();
float current = c->GetCompressorCurrent();
```

Java

```
boolean enabled = c.enabled();
boolean pressureSwitch = c.getPressureSwitchValue();
float current = c.getCompressorCurrent();
```

The other reason to create a compressor object would be to query the status of the compressor. The state (currently on or not), pressure switch state, and compressor current can all be queried from the Compressor object.