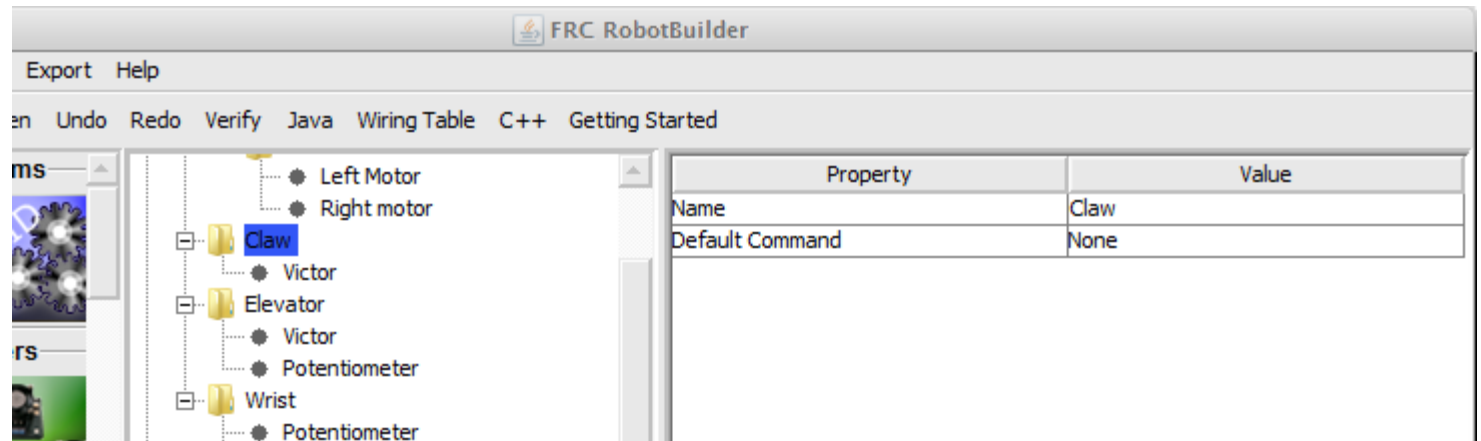


Writing the C++ code for a subsystem

Writing the C++ code for a subsystem

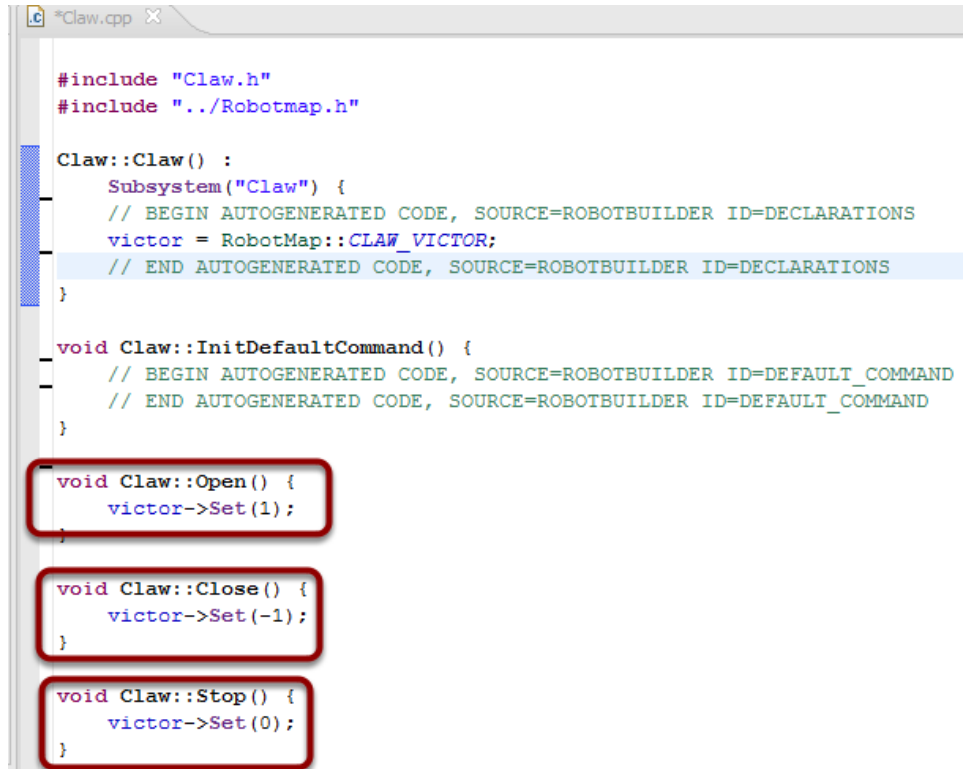
RobotBuilder representation of the Claw subsystem



The claw at the end of a robot arm is a subsystem operated by a single Victor speed controller. There are three things we want the claw to do, start opening, start closing, and stop moving. This is the responsibility of the subsystem. The timing for opening and closing will be handled by a command later in this tutorial.

Writing the C++ code for a subsystem

Adding capabilities to a simple subsystem



```
#include "Claw.h"
#include "../Robotmap.h"

Claw::Claw() :
    Subsystem("Claw") {
    // BEGIN AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DECLARATIONS
    victor = RobotMap::CLAW_VICTOR;
    // END AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DECLARATIONS
}

void Claw::InitDefaultCommand() {
    // BEGIN AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DEFAULT_COMMAND
    // END AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DEFAULT_COMMAND
}

void Claw::Open() {
    victor->Set(1);
}

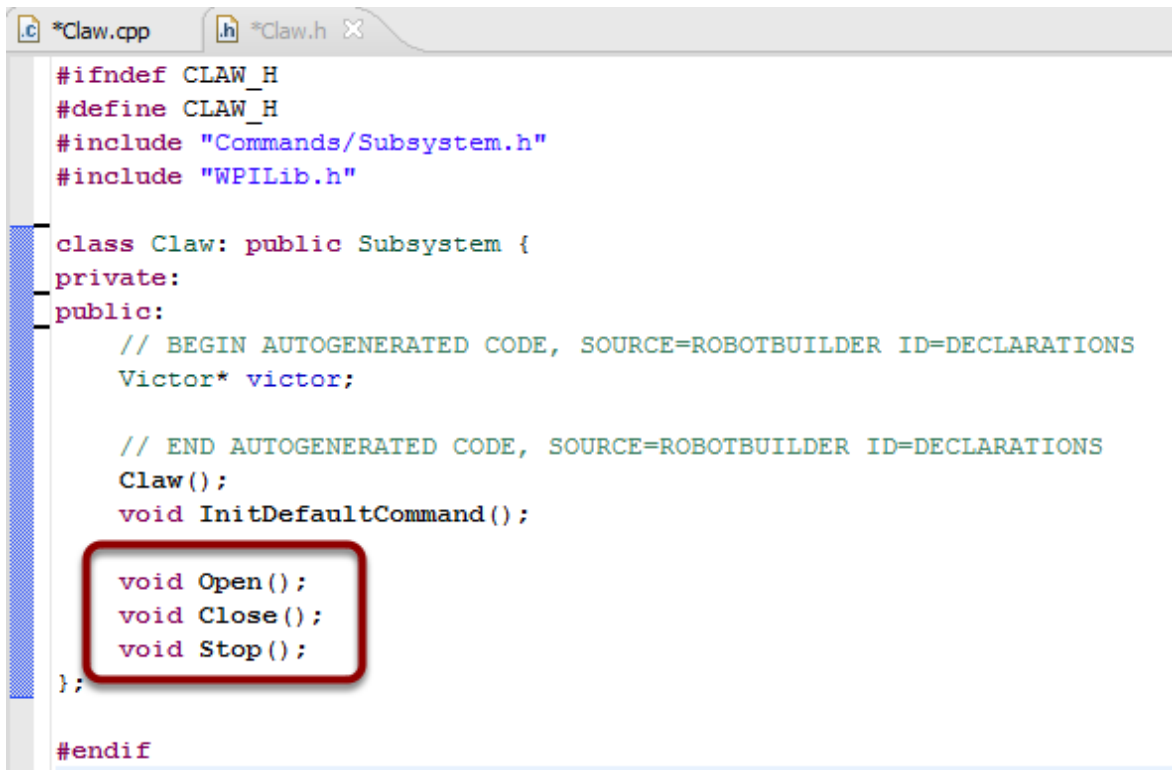
void Claw::Close() {
    victor->Set(-1);
}

void Claw::Stop() {
    victor->Set(0);
}
```

The subsystem generated by RobotBuilder has the code to create the Claw class that corresponds to the subsystem and the code to copy the reference to the generated Victor object into the class to make it easily referenced. To add the capabilities to the subsystem to actually operate the motor add 3 methods, Open(), Close(), and Stop() to start the motor moving in the open or close direction or stop the motor.

Writing the C++ code for a subsystem

Adding the method declarations to the generated header file - Claw.h



```
#ifndef CLAW_H
#define CLAW_H
#include "Commands/Subsystem.h"
#include "WPILib.h"

class Claw: public Subsystem {
private:
public:
    // BEGIN AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DECLARATIONS
    Victor* victor;

    // END AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DECLARATIONS
    Claw();
    void InitDefaultCommand();

    void Open();
    void Close();
    void Stop();
};

#endif
```

In addition to adding the methods to the class implementation file, Claw.cpp, the declarations for the methods need to be added to the header file, Claw.h. Those declarations that must be added are shown here.

To add the behavior to the claw subsystem to operate it to handle opening and closing you [need to define commands](#).