Oregon Robotics Tournament and Outreach Program

Vex Workshop

2006

Opening doors to the worlds of science and technology for Oregon’s youth

Instructor Contacts

Ed C. Epp

Nick Brandt

Kevin W. Rudd

Glenn Graham

vex-questions@ortop.org

Today’s Goal

- Build an understanding of how to construct and program Vex robots.
- Build confidence - you can do it.
- Put you on the path to learning Vex.

Agenda

- Introductions
- Finish building a Vex robot
- EasyC overview
- Demo: one motor
- Exercise: two motors
- Demo: motor and sensors
- Demo: combining autonomous and radio control
- MPLab
- Exercise: Cando

Introductions

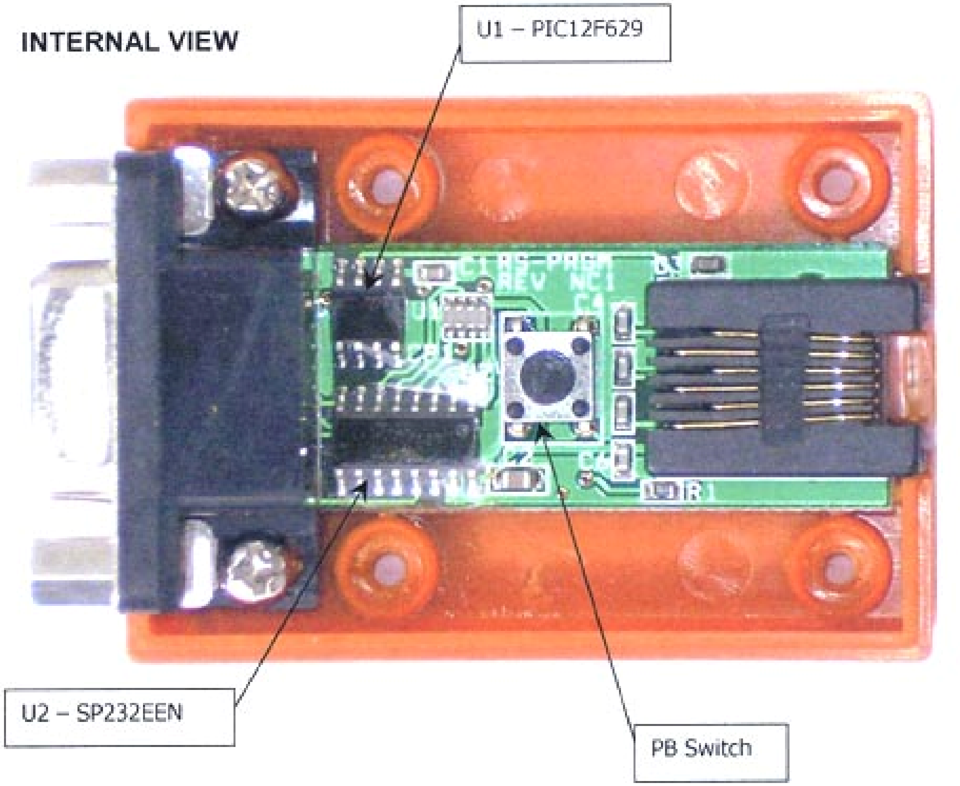
Hardware

Exercise

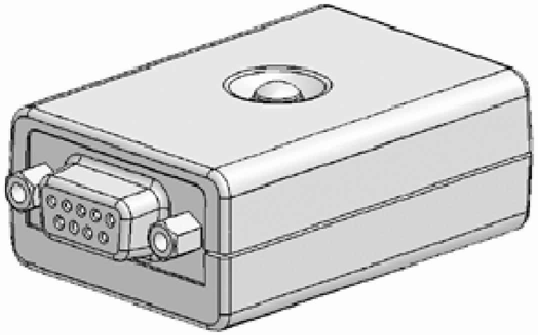
- Structure: 2.24
- Power: 4.6
- Sensor: 5.3
- Logic: 7.5 to 7.9

The Connection

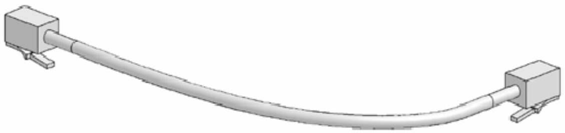
The Parts



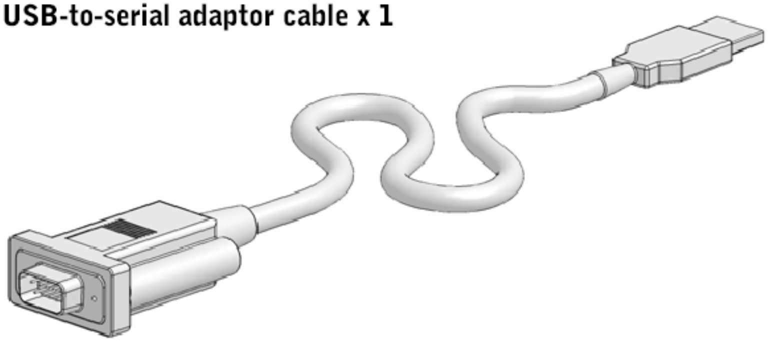
programming module x 1



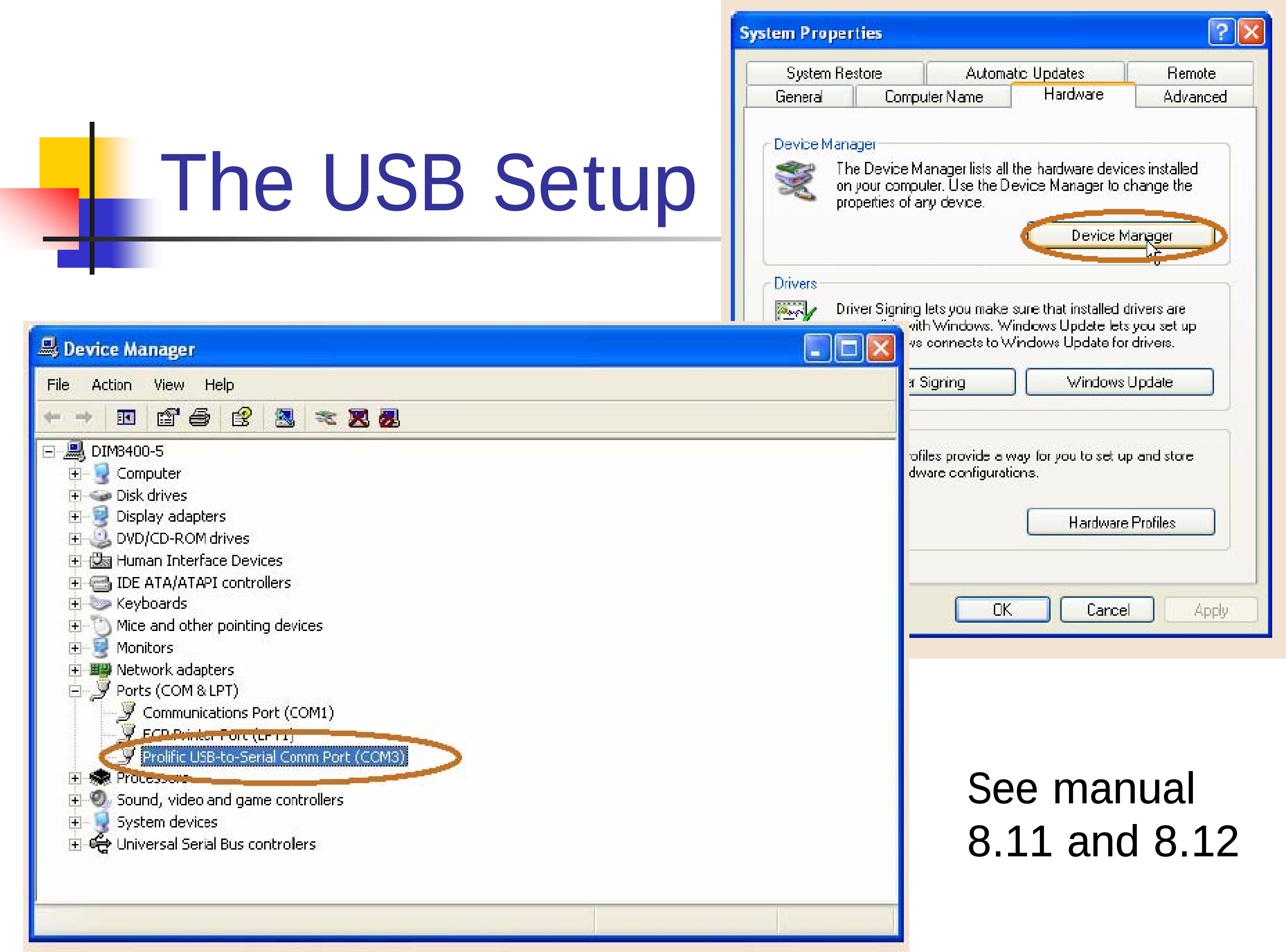
robot interface cable x 1



USB-to-serial adaptor cable x 1



The USB Setup



System Properties

Device Manager

The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Drivers

Driver Signing lets you make sure that installed drivers are ... with Windows. Windows Update lets you set up ...ws connects to Windows Update for drivers.

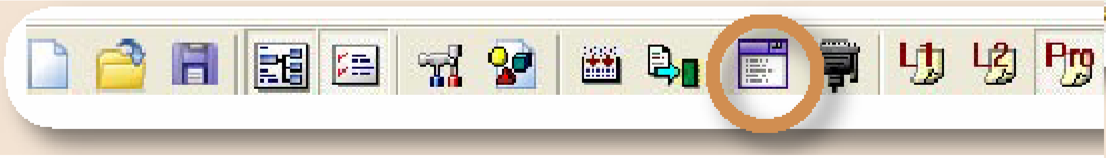
...ofiles provide a way for you to set up and store ...dware configurations.

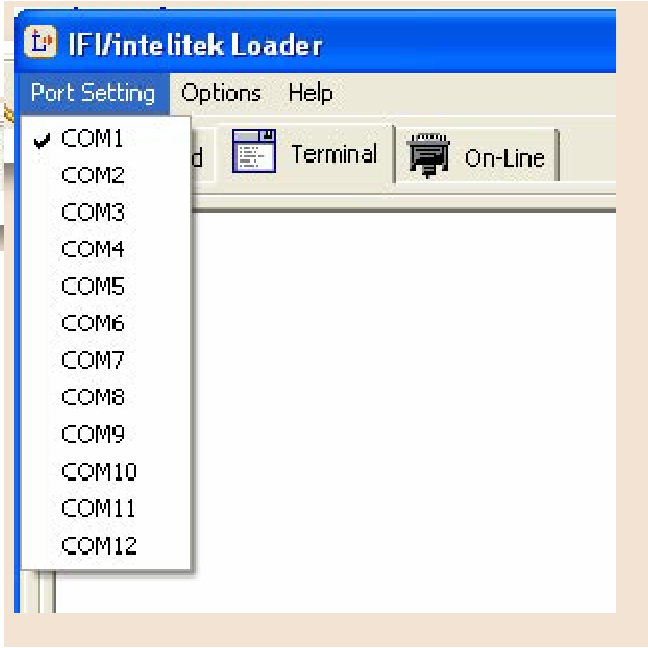
Device Manager

- DIM3400-5
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - Communications Port (COM1)
    - ECP Printer Port (LPT1)
    - Prolific USB-to-Serial Comm Port (COM3)
  - Processors
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers

See manual 8.11 and 8.12

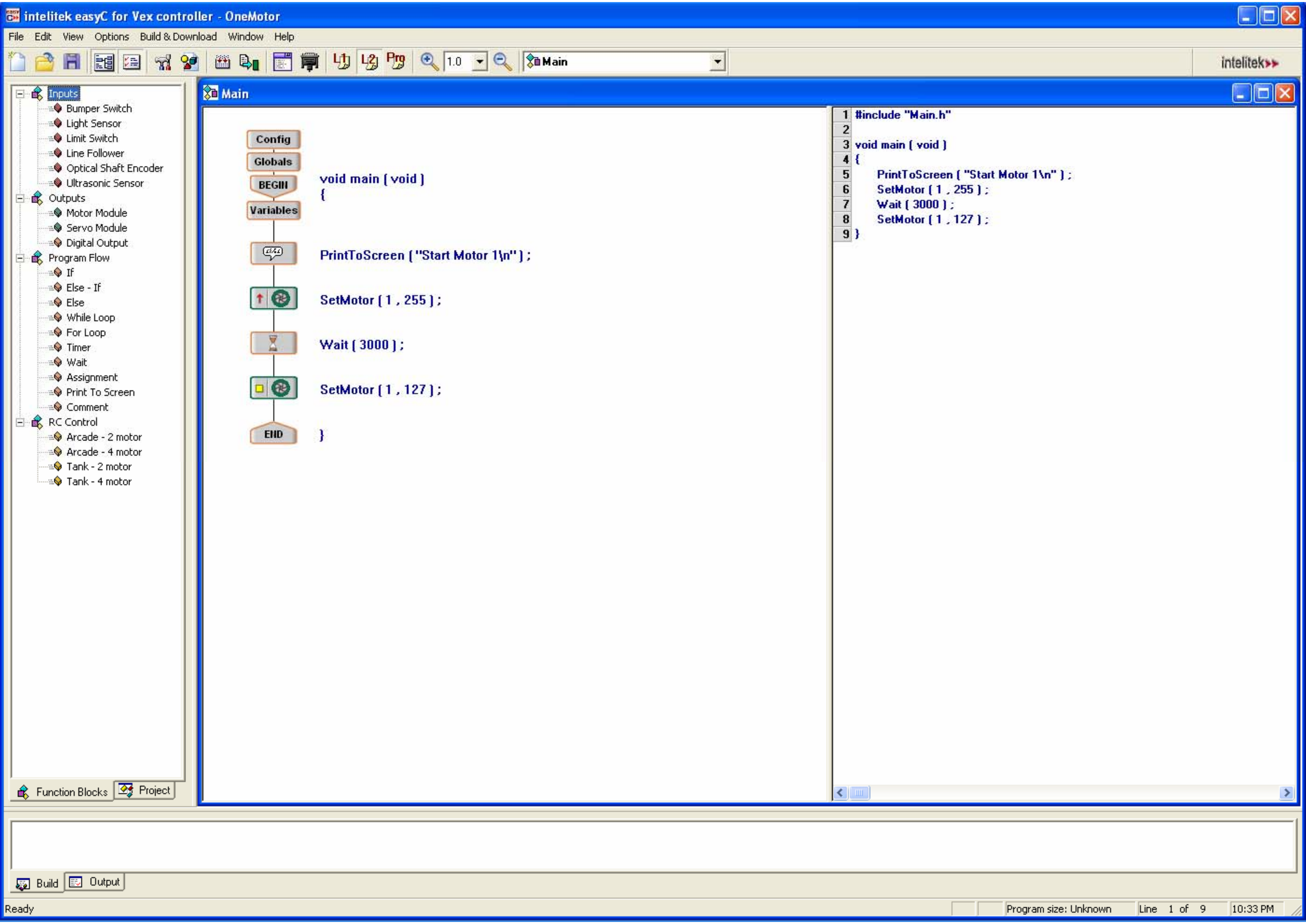
The USB Setup 2





See manual
8.16

EasyC



```
void main ( void )
{

PrintToScreen ( "Start Motor 1\n" ) ;

SetMotor ( 1 , 255 ) ;

Wait ( 3000 ) ;

SetMotor ( 1 , 127 ) ;

}
```

```
#include "Main.h"

void main ( void )
{
      PrintToScreen ( "Start Motor 1\n" ) ;
      SetMotor ( 1 , 255 ) ;
      Wait ( 3000 ) ;
      SetMotor ( 1 , 127 ) ;
}
```

Demo - Single Motor Program

- Select File > New Project.
- Select “L2” on the toolbar.
- Click and drag the Motor Module block into the program window between the Begin and End blocks.
- Release the block there.
  - Make sure “Motor Number” is 3 and “Clockwise” is selected. Click “OK.”

Load and Run the Program

- Select the “Build and Download” option in the “Build and Download” menu.
  - Choose Yes at the prompt to download your program.
  - As soon as your program finishes downloading, the left motor will turn on, making your robot spin in place.
- Turn your robot off
  - To run your program again, switch the robot back on.
- Save your program.

Exercise - A Second Motor

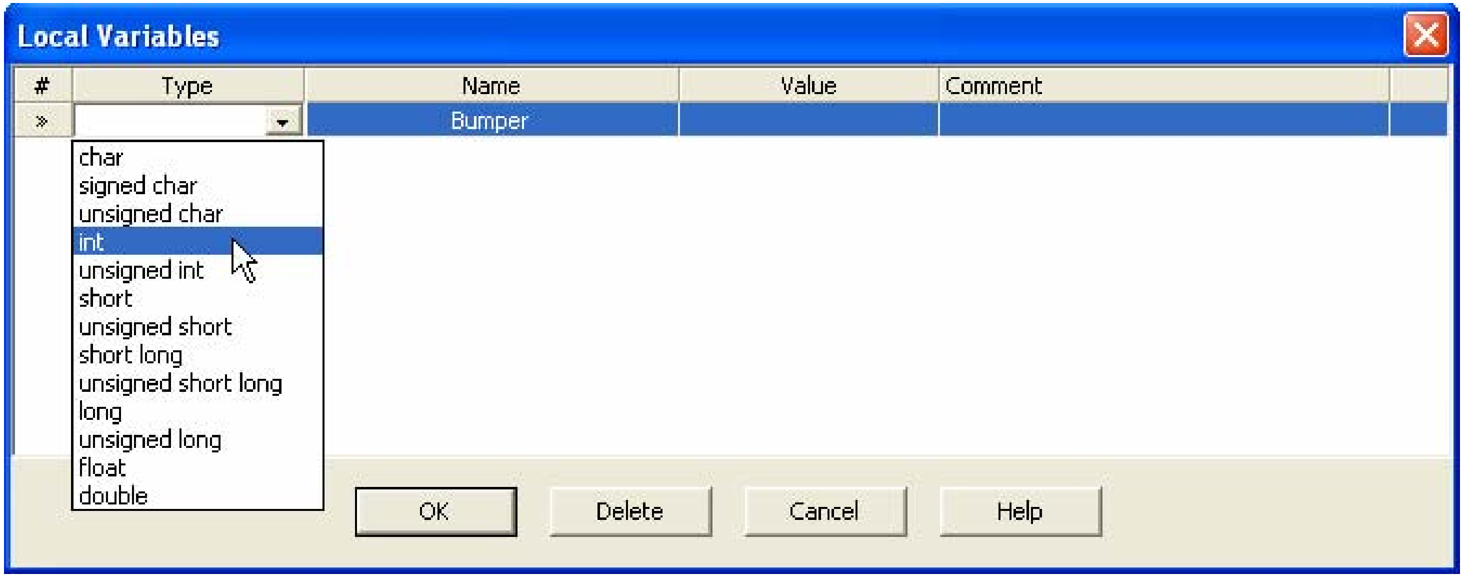
- Add a second motor
- Insert a wait
- Turn off both motors

Demo – Sensors and Motors

- If the bump switch is not pressed, the robot moves forward
- If the bump switch is pushed, the robot will stop

Add a Variable

- Select File > New Project
- Double-click the Variables block.
- Click "Type" and Select "int".
- Click "Name" column and type "Bumper", then click OK.

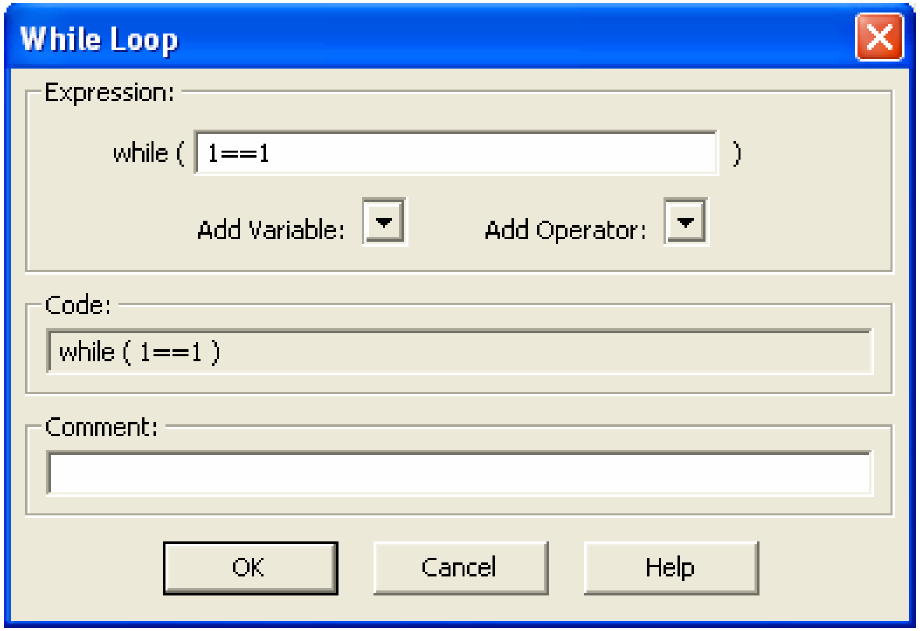


Local Variables

#	Type	Name	Value	Comment
»		Bumper		

Do Something Forever

- Drag and drop a While Loop function block between the variables and end icons.
- Set the condition of the while loop to be "1==1".



While Loop

Expression:

while (1==1)

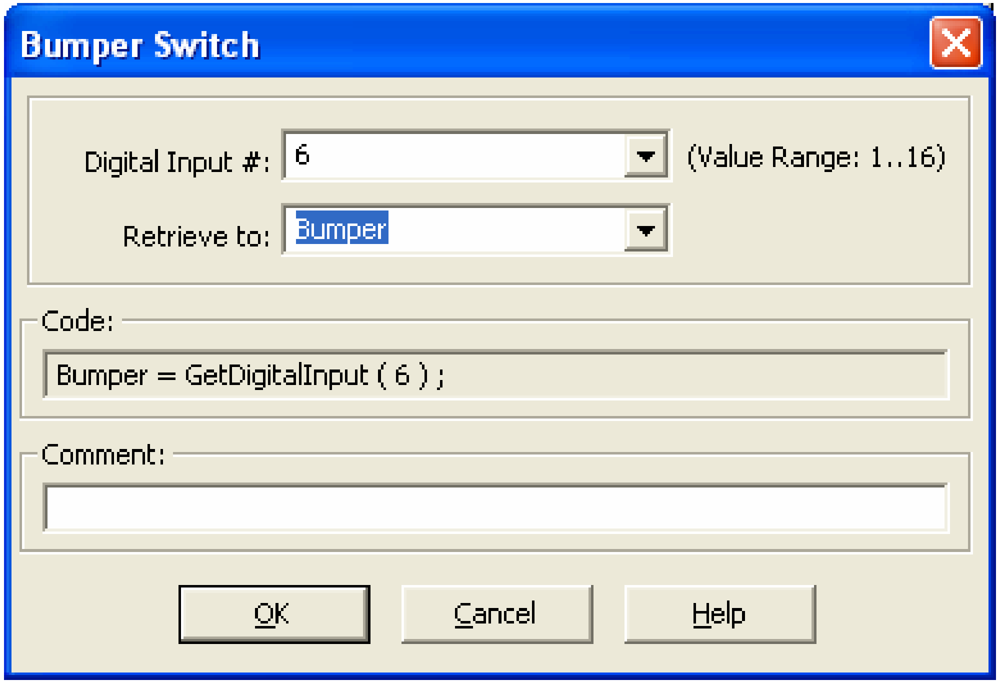
Code:

while (1==1)

Comment:

Add a Bumper Test

- Drag a Bumper Switch between the braces of the while loop.
- Set the "Digital Input #" to 6.
- Set the "Retrieve to" to "Bumper".



Digital Input #: 6 (Value Range: 1..16)

Retrieve to: Bumper

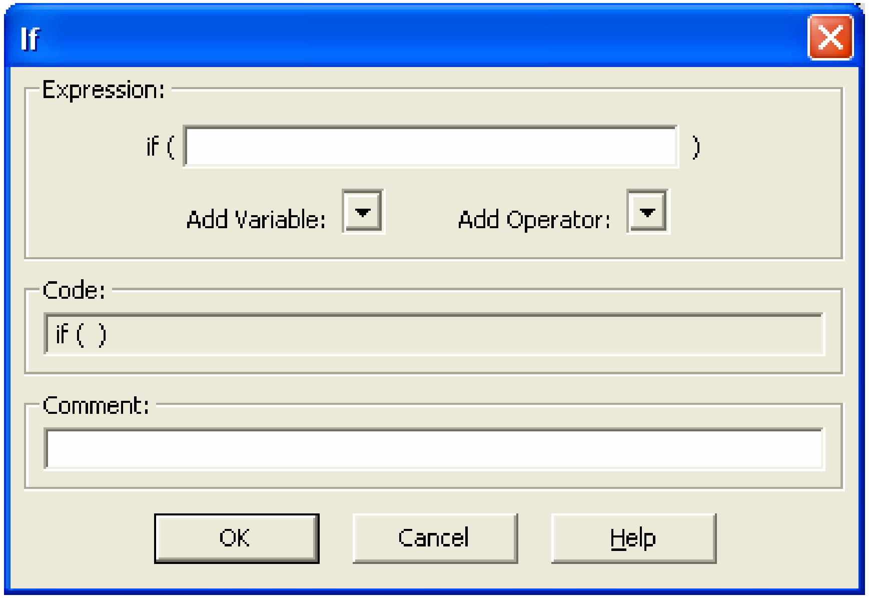
Code:

```
Bumper = GetDigitalInput ( 6 ) ;
```

Comment:

Add a Conditional Test

- Drag an If Condition underneath the Bumper Switch.
- Click the small arrow next to "Add Variable" and select "Bumper".
- Click the arrow next to "Add Operator" and select the "==" operation.
- After the "==" sign you will enter the number 1.



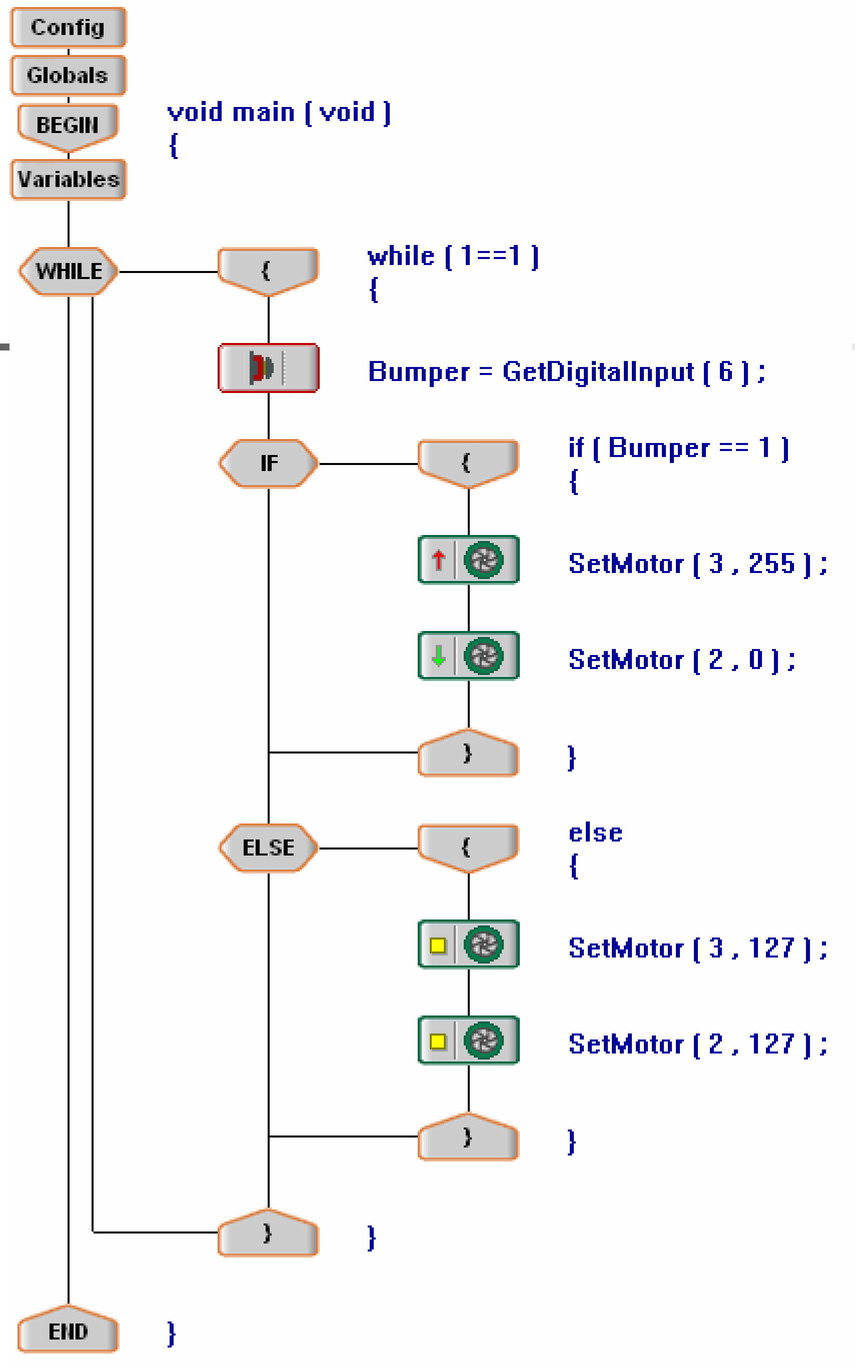
Code: if ( )

Add the Else Condition

- Drag and drop an Else Condition block into the while loop underneath the If Condition icon.

Fill in the If and Else Clauses

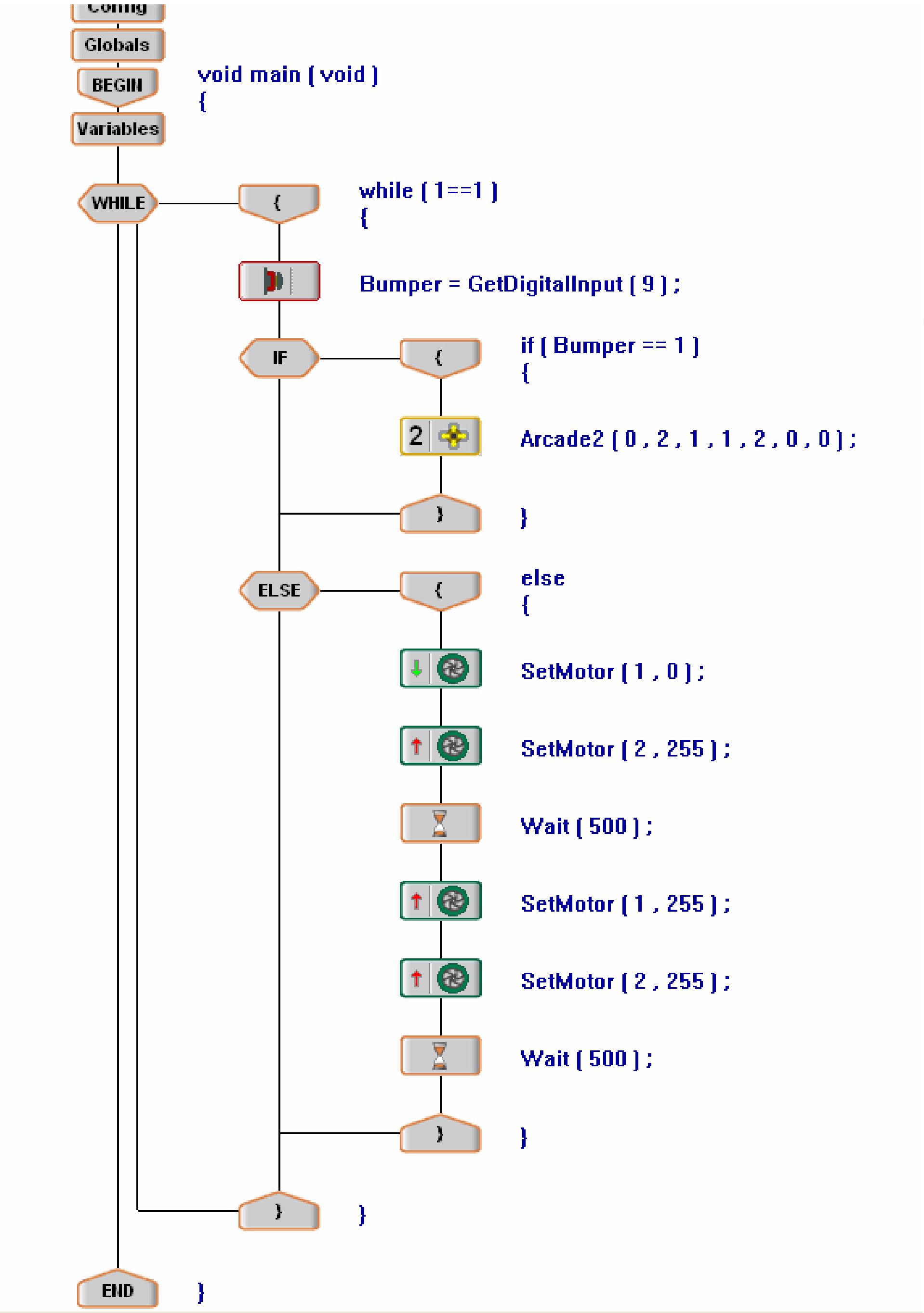
- If clause
  - Drag in two motor modules between the brackets under the "if" statement.
  - Set them up so that your robot will drive forward
- Else clause
  - Drag in two motor modules between the brackets under the "else" statement.
  - Configure them so that they form a stopping behavior.



```
void main ( void )
{
      while ( 1==1 )
      {
            Bumper = GetDigitalInput ( 6 ) ;
            if ( Bumper == 1 )
            {
                  SetMotor ( 3 , 255 ) ;
                  SetMotor ( 2 , 0 ) ;
            }
            else
            {
                  SetMotor ( 3 , 127 ) ;
                  SetMotor ( 2 , 127 ) ;
            }
      }
}
```

Demo: Autonomous and RC

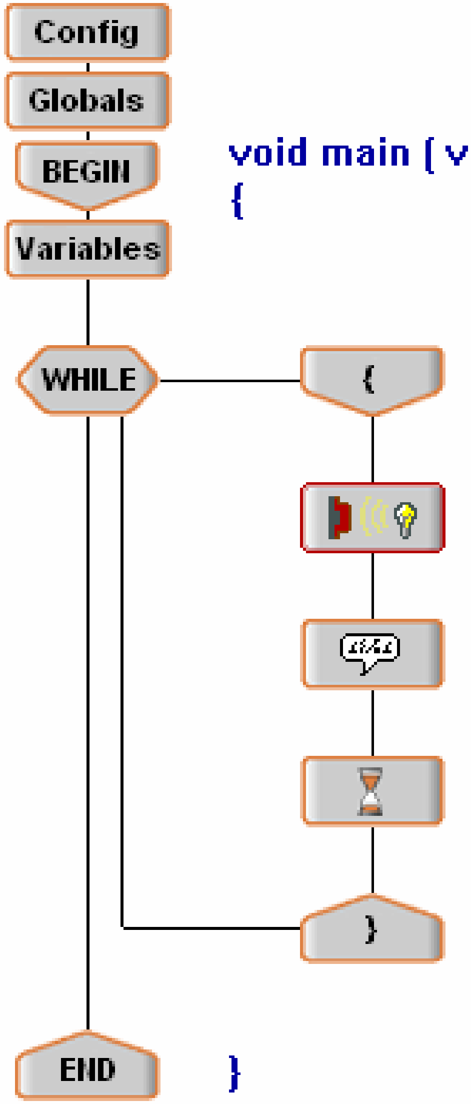
- Move under the remote control
- When an obstacle is hit
    - Back up
    - Turn
- Return to remote control



```
void main ( void )
{
    while ( 1==1 )
    {
        Bumper = GetDigitalInput ( 9 ) ;
        if ( Bumper == 1 )
        {
            Arcade2 ( 0 , 2 , 1 , 1 , 2 , 0 , 0 ) ;
        }
        else
        {
            SetMotor ( 1 , 0 ) ;
            SetMotor ( 2 , 255 ) ;
            Wait ( 500 ) ;
            SetMotor ( 1 , 255 ) ;
            SetMotor ( 2 , 255 ) ;
            Wait ( 500 ) ;
        }
    }
}
```

MPLab

Demo: Light Sensor



```
void main ( void )
{

      while ( 1==1 )
      {

      LightLevel = GetAnalogInput ( 1 ) ;

      PrintToScreen ( "Light Level: %d\n" , (int)LightLevel ) ;

      Wait ( 500 ) ;

      }

}
```

Cando Exercise