

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



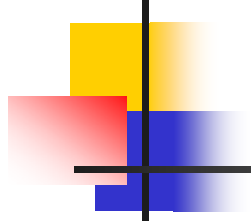
Today's Goal

- 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.

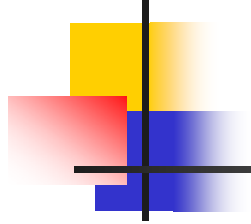


Agenda

- 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

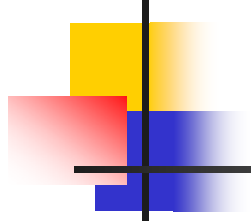


Hardware



Exercise

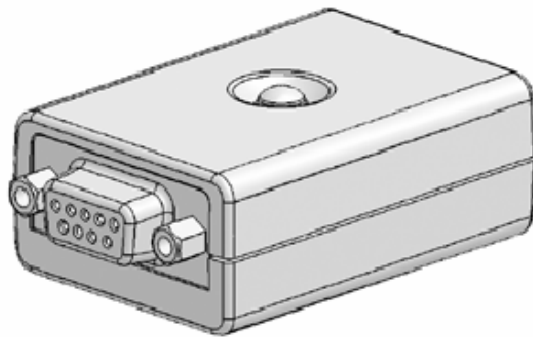
- Structure: 2.24
- Power: 4.6
- Sensor: 5.3
- Logic: 7.5 to 7.9



The Connection

The Parts

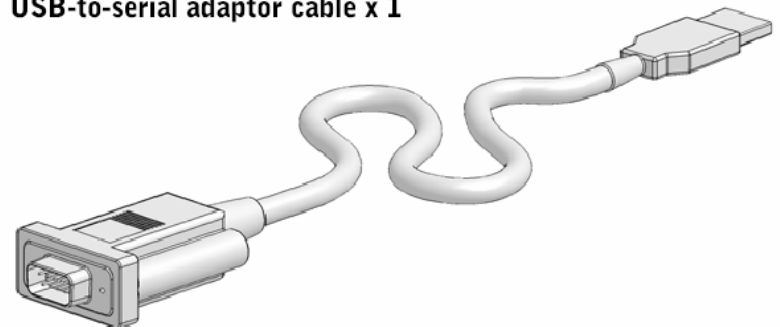
programming module x 1



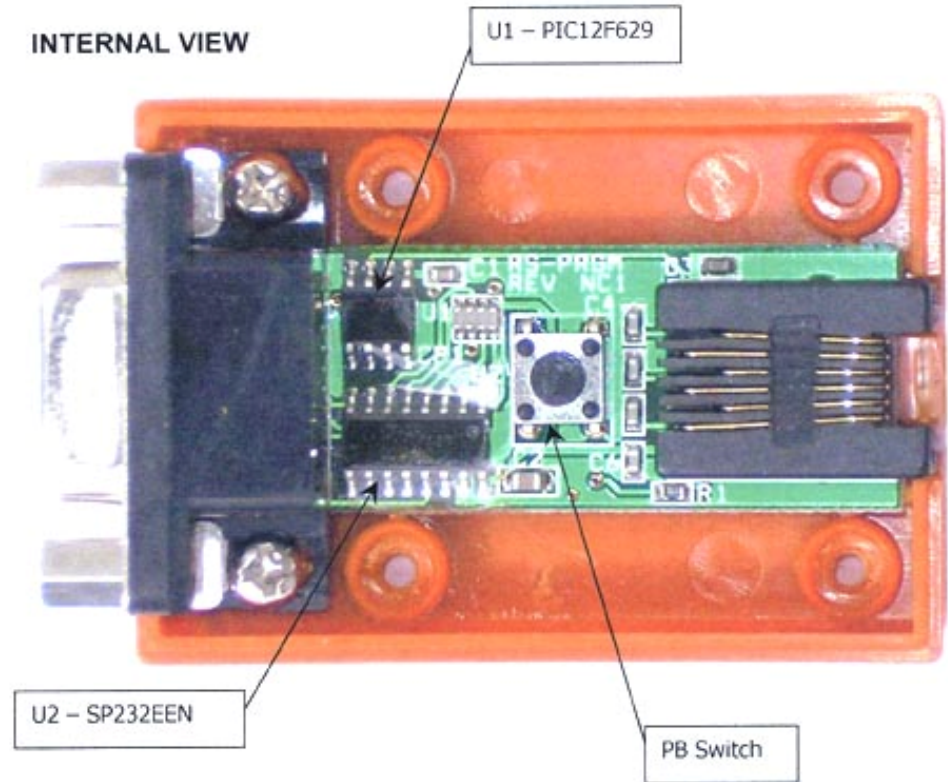
robot interface cable x 1



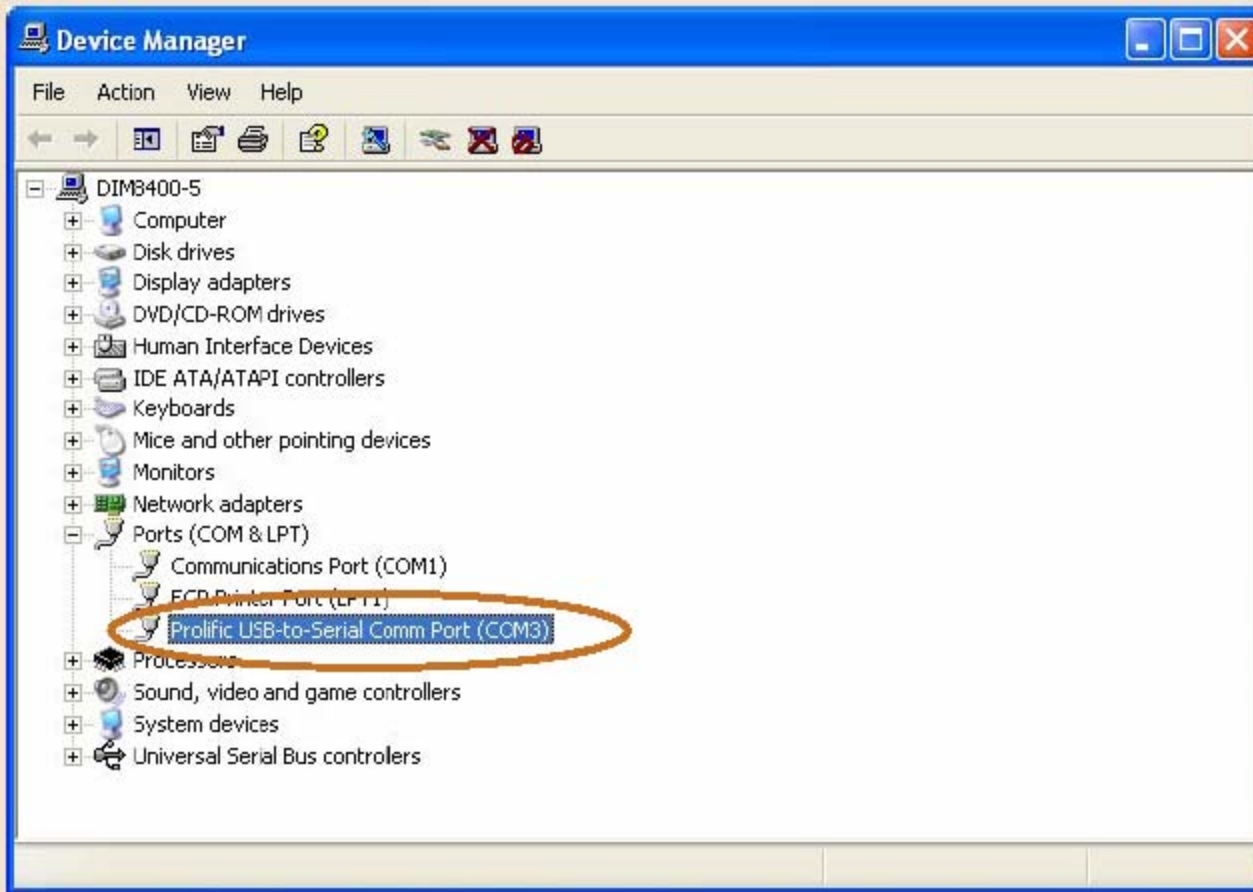
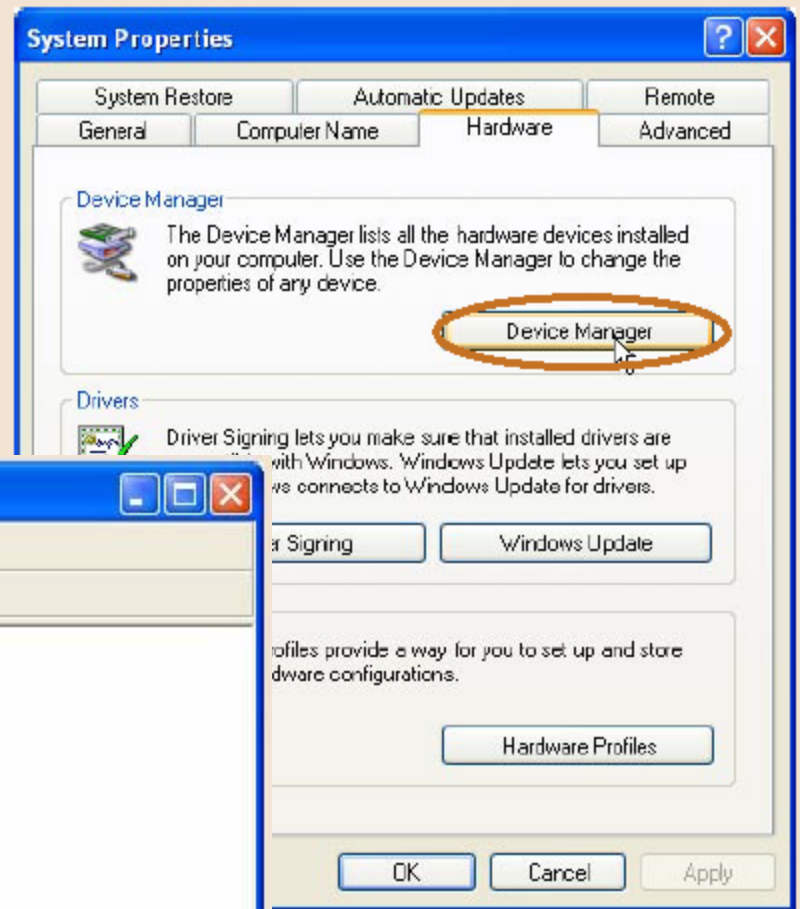
USB-to-serial adaptor cable x 1



INTERNAL VIEW

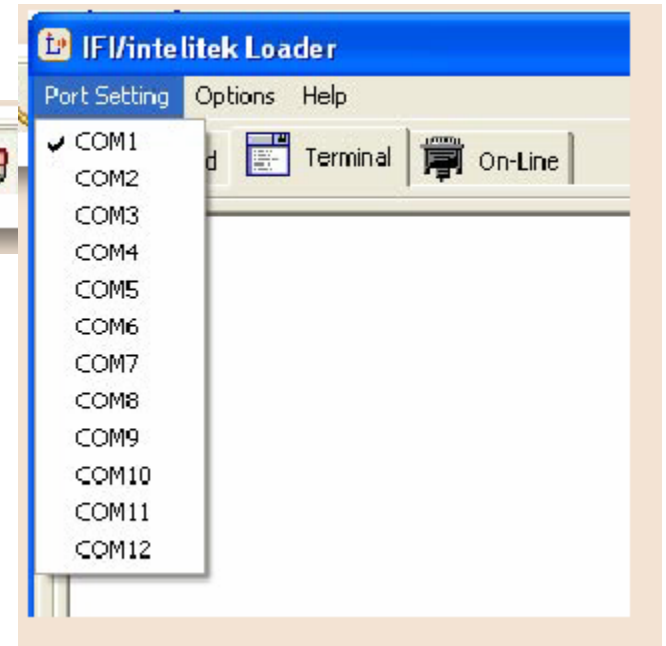
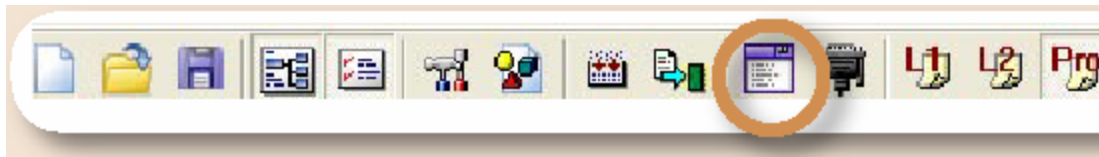


The USB Setup

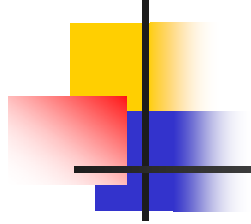


See manual
8.11 and 8.12

The USB Setup 2



See manual
8.16



EasyC

- Inputs
 - Bumper Switch
 - Light Sensor
 - Limit Switch
 - Line Follower
 - Optical Shaft Encoder
 - Ultrasonic Sensor
- Outputs
 - Motor Module
 - Servo Module
 - Digital Output
- Program Flow
 - If
 - Else - If
 - Else
 - While Loop
 - For Loop
 - Timer
 - Wait
 - Assignment
 - Print To Screen
 - Comment
- RC Control
 - Arcade - 2 motor
 - Arcade - 4 motor
 - Tank - 2 motor
 - Tank - 4 motor



```

1 #include "Main.h"
2
3 void main ( void )
4 {
5     PrintToScreen ( "Start Motor 1\n" );
6     SetMotor ( 1 , 255 );
7     Wait ( 3000 );
8     SetMotor ( 1 , 127 );
9 }
    
```

Build Output



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

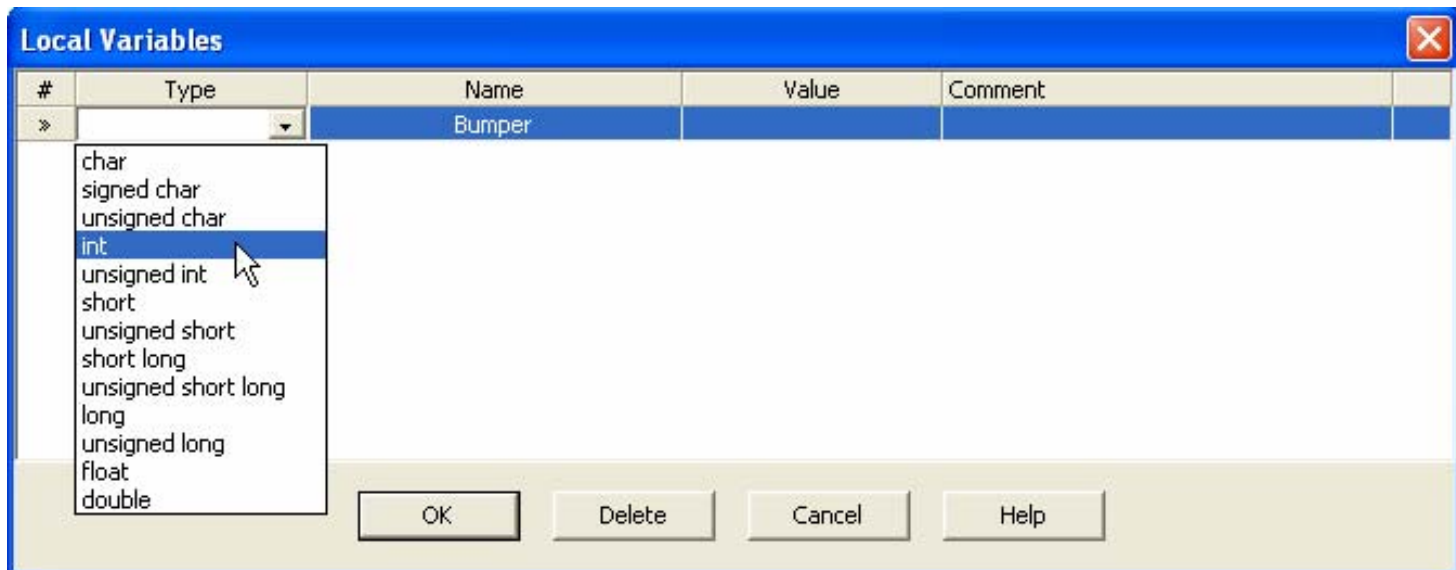


Demo – Sensors and Motors

- 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.



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".

The screenshot shows a dialog box titled "While Loop" with a blue header and a close button in the top right corner. The dialog is divided into three main sections:

- Expression:** A text input field containing the code `while (1==1)`. Below this field are two dropdown menus labeled "Add Variable:" and "Add Operator:".
- Code:** A text input field containing the code `while (1==1)`.
- Comment:** An empty text input field.

At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

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".

Bumper Switch

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

Retrieve to: Bumper

Code:
Bumper = GetDigitalInput (6) ;

Comment:

OK Cancel Help

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.

The screenshot shows a dialog box titled "If" with a close button in the top right corner. The dialog is divided into three sections:

- Expression:** This section contains the text "if (" followed by a text input field, a closing parenthesis ")", and two dropdown menus labeled "Add Variable:" and "Add Operator:".
- Code:** This section contains a text input field with the text "if ()".
- Comment:** This section contains an empty text input field.

At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".



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.

Config
Globals
BEGIN
Variables

```
void main ( void )  
{
```

WHILE

{

```
while ( 1==1 )  
{
```



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

IF

{

```
if ( Bumper == 1 )  
{
```



```
SetMotor ( 3 , 255 );
```



```
SetMotor ( 2 , 0 );
```

}

```
}
```

ELSE

{

```
else  
{
```



```
SetMotor ( 3 , 127 );
```



```
SetMotor ( 2 , 127 );
```

}

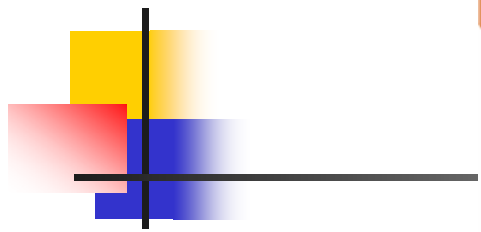
```
}
```

}

```
}
```

END

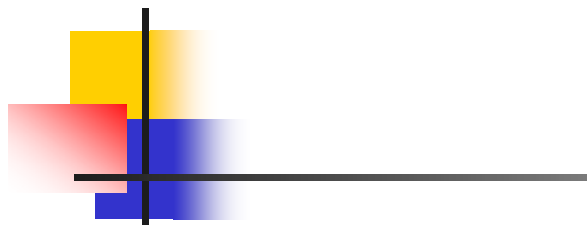
```
}
```





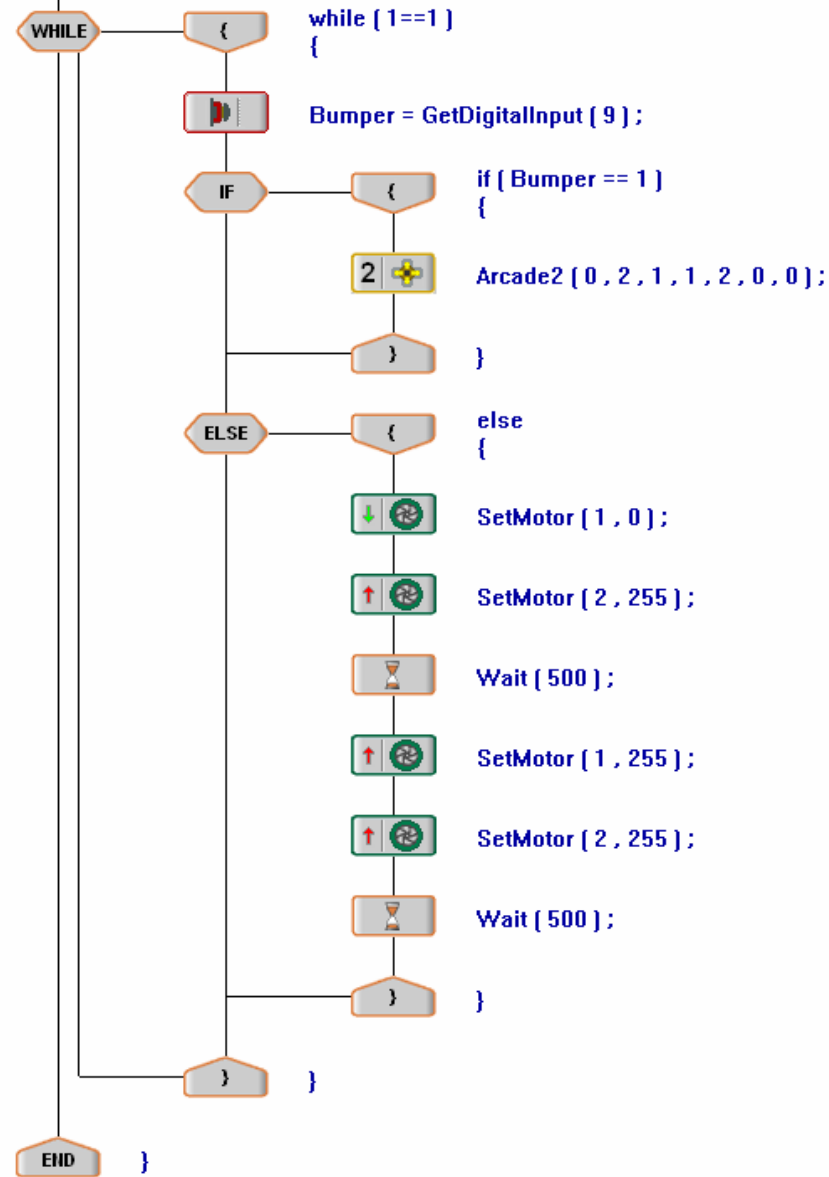
Demo: Autonomous and RC

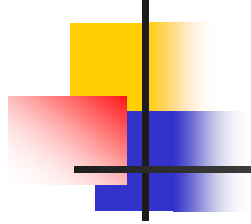
- Move under the remote control
- When an obstacle is hit
 - Back up
 - Turn
- Return to remote control



Config
Globals
BEGIN
Variables

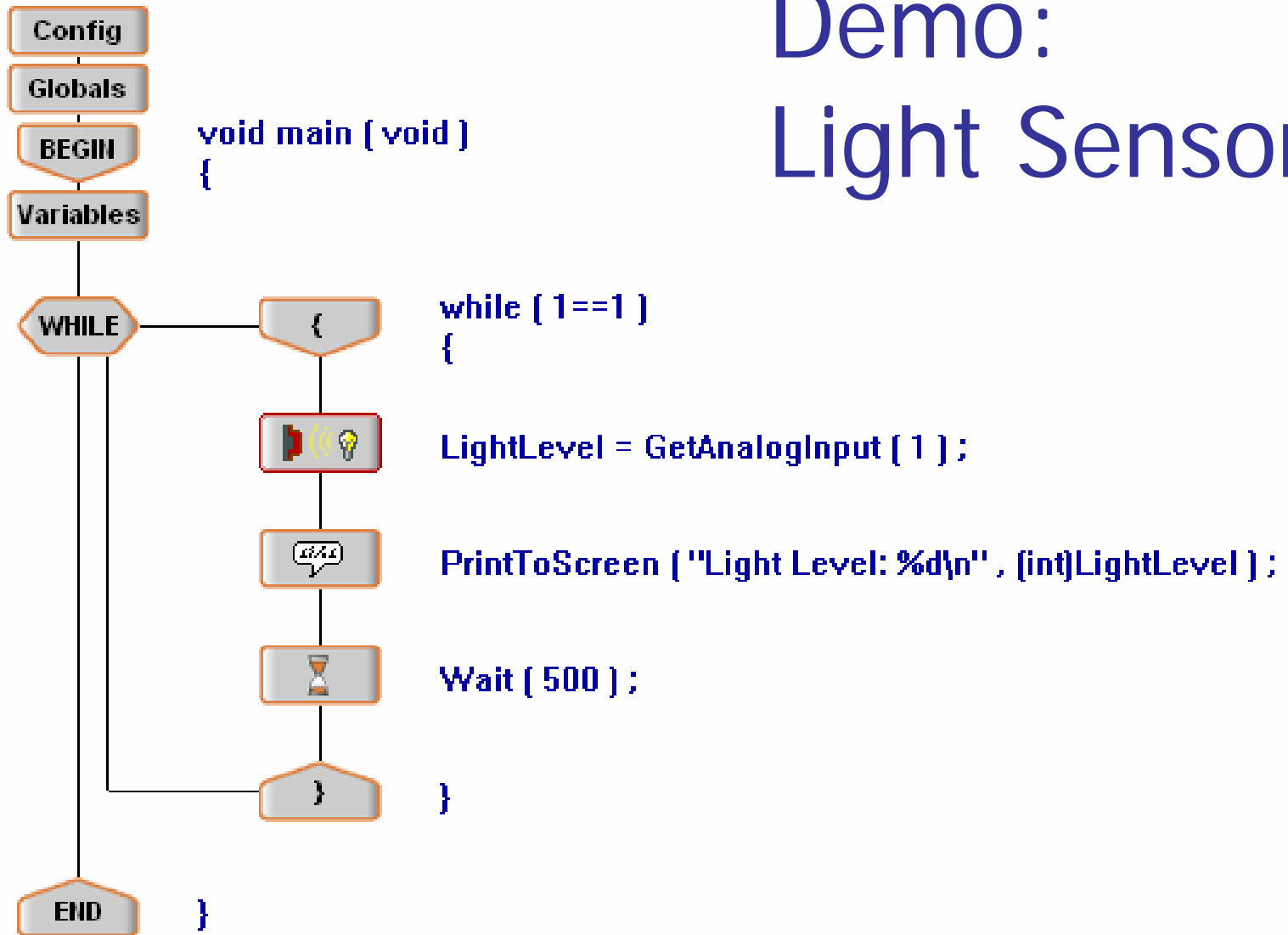
```
void main ( void )  
{
```

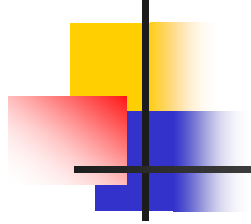




MPLab

Demo: Light Sensor





Cando Exercise