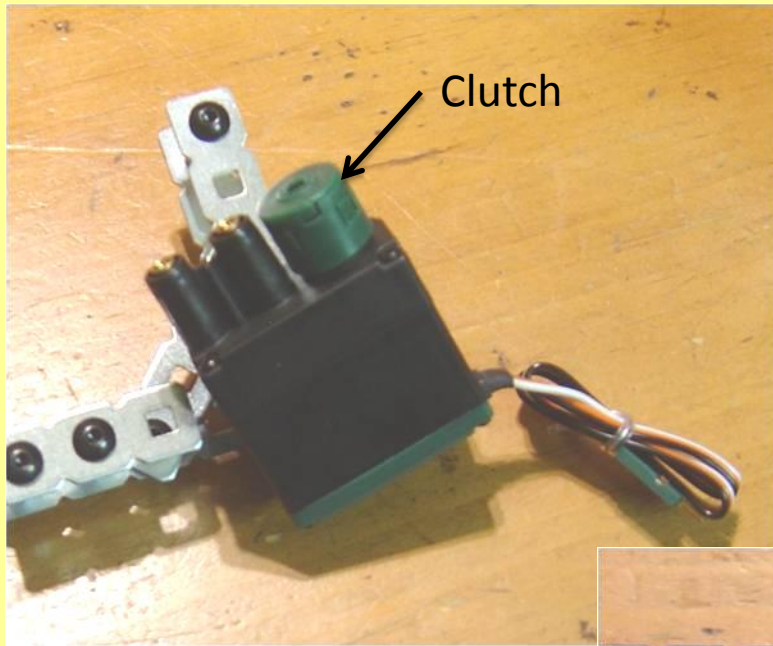




7.2 V battery pack
and charger

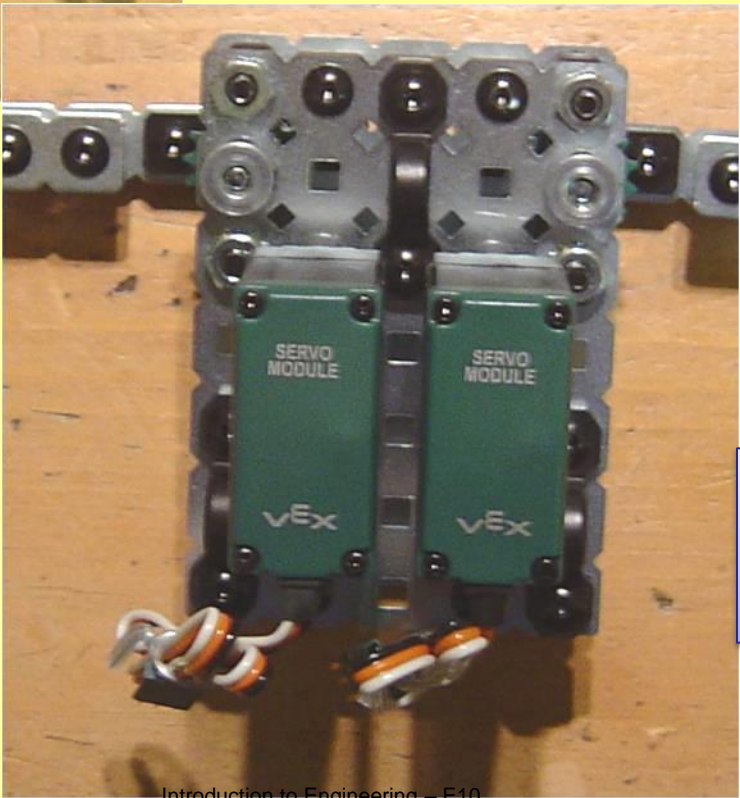
Put the battery in the charger station at the end of the lab. period. Red light indicates charging.





Motors

Continuous motor



Servo motor, rotates through a specified angle.

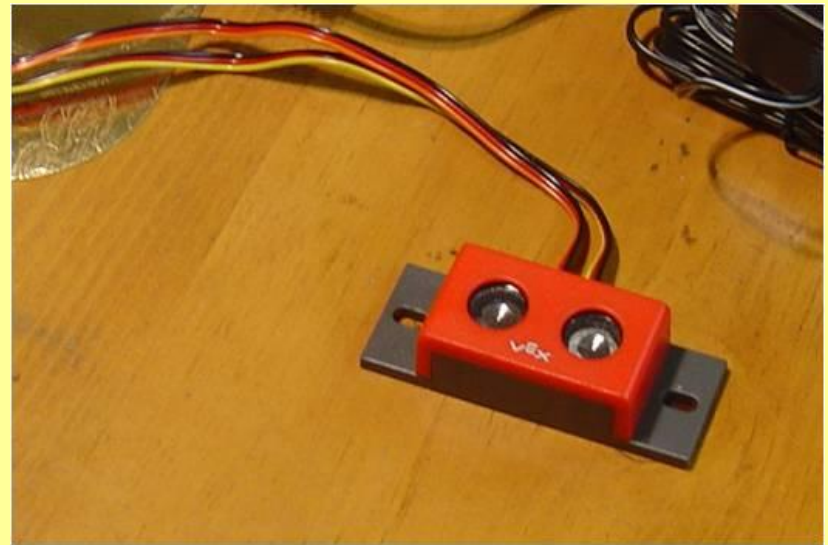
Make sure the clutch module is always on. It prevents damage to the motor in case of large torque (resistance to motion).

Sensors

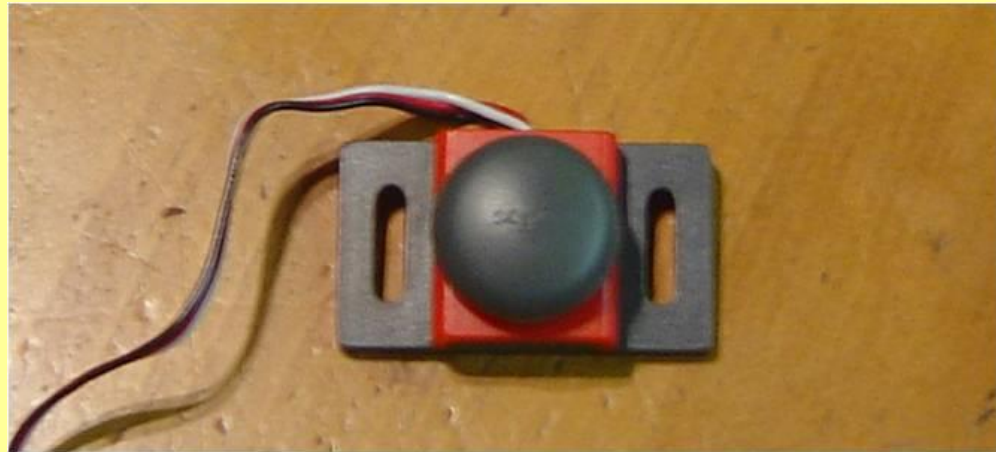


Limit Switch Sensor

attach all sensors to 5-8 ports
on Analog/Digital section

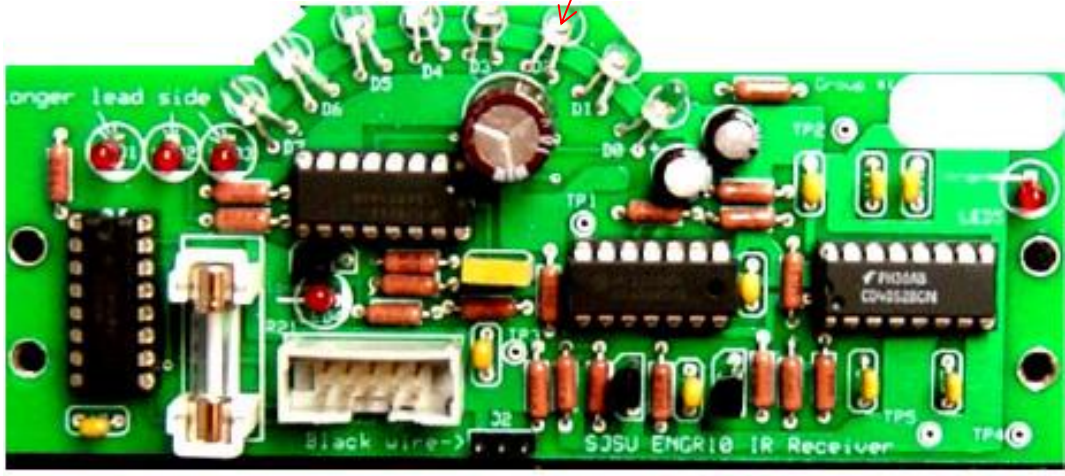


Ultrasonic Sensor



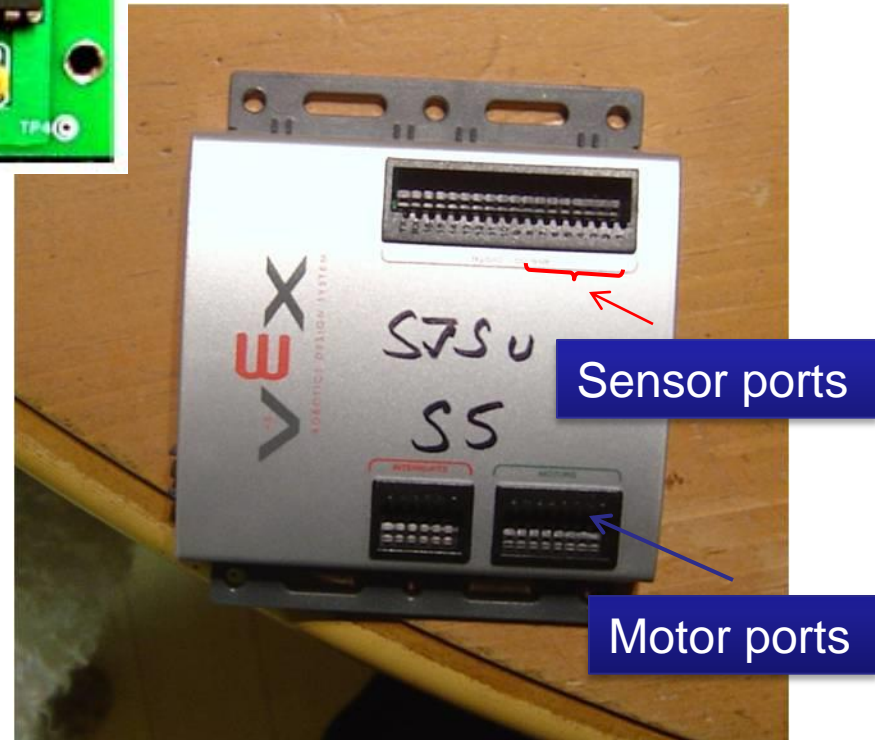
Bumper Switch Sensor

Detectors, cover a field of view of 100°



Infrared Receiver Board, the eyes of the robot.

Micro Controller



EasyC

Select File and choose New Project



Inputs

- Bumper Switch
- Light Sensor
- Limit Switch
- Line Follower
- Optical Shaft Encoder
- Ultrasonic Sensor

Outputs

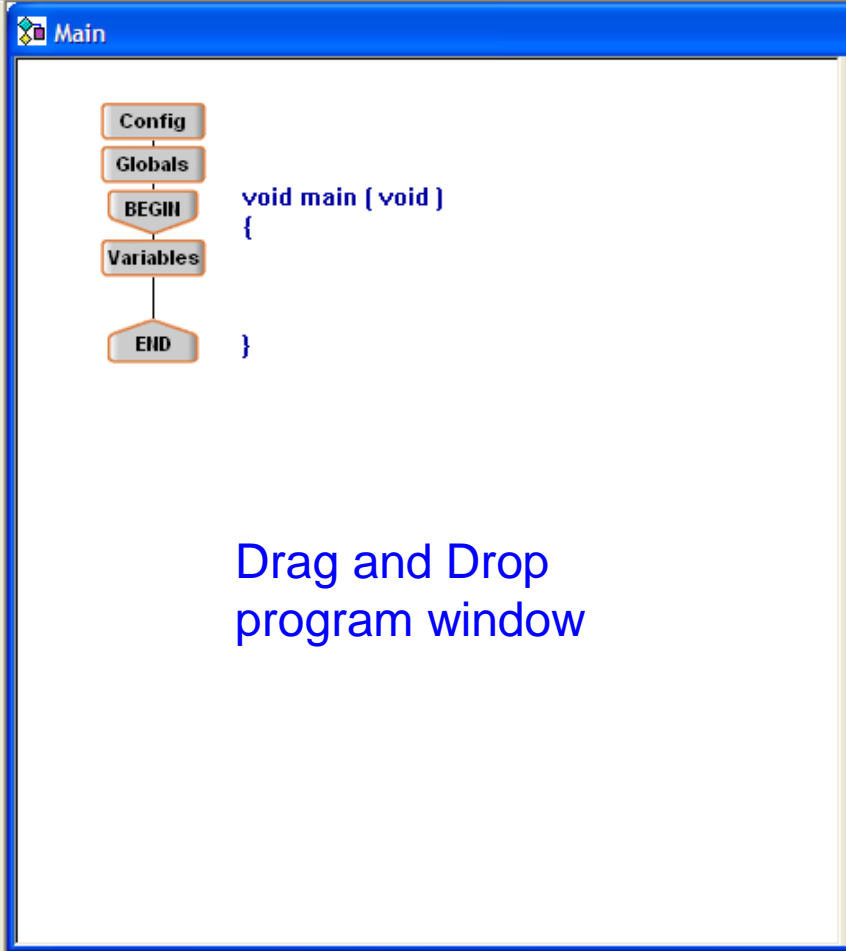
- Motor Module
- Servo Module
- Digital Output

Program Flow

- If
- Else - If
- Else
- While Loop
- For Loop
- Timer
- Wait
- Assignment
- Print To Screen
- Comment

RC Control

Function Blocks Project



```
1 #include "Main.h"
2
3 void main ( void )
4 {
5 }
```

Code window

Drag and Drop
program window

Continuous motor

void main (void)
{
}
}

Select Motor, drag and drop on the line between Begin and End

Choose motor number, as connected to the Controller

Max. speed clockwise

Motor Number: 1 (Value Range: 1..8)

Motor Direction:

Clockwise 255

Stop 127

Counter-clockwise 0

User Value (Value Range: 0..255)

Code:
SetMotor (1 , 255);

Comment:

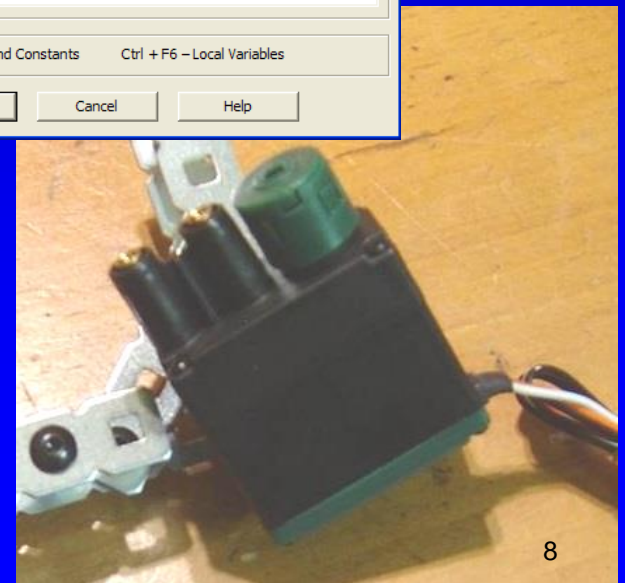
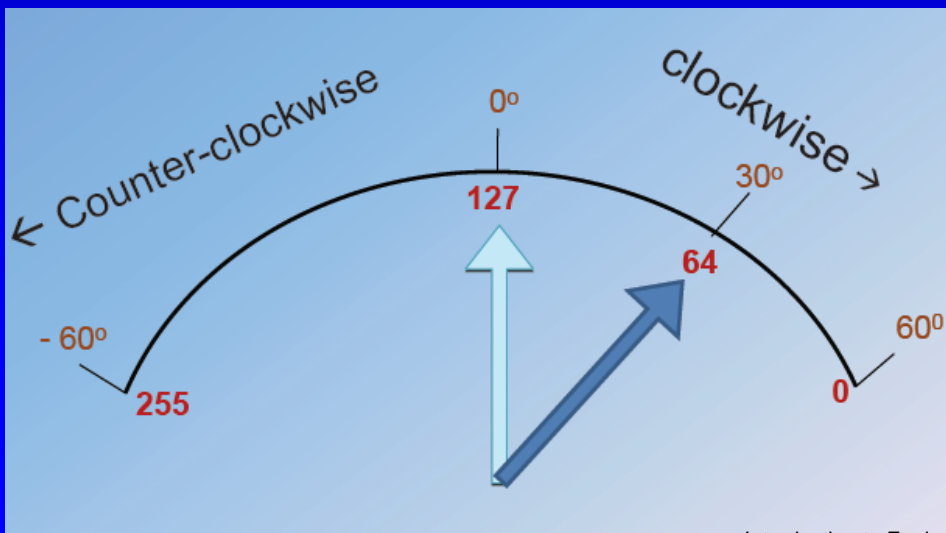
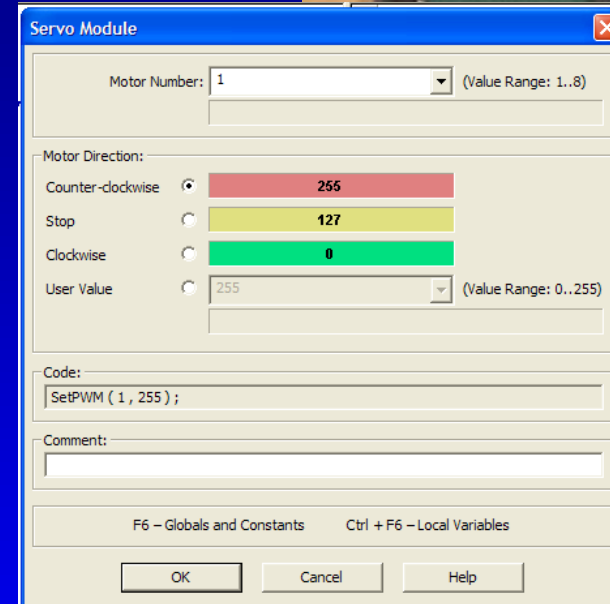
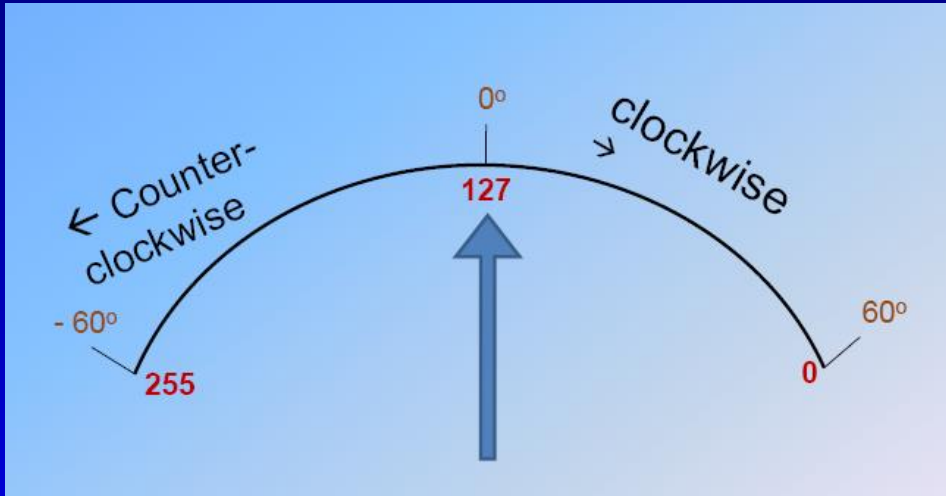
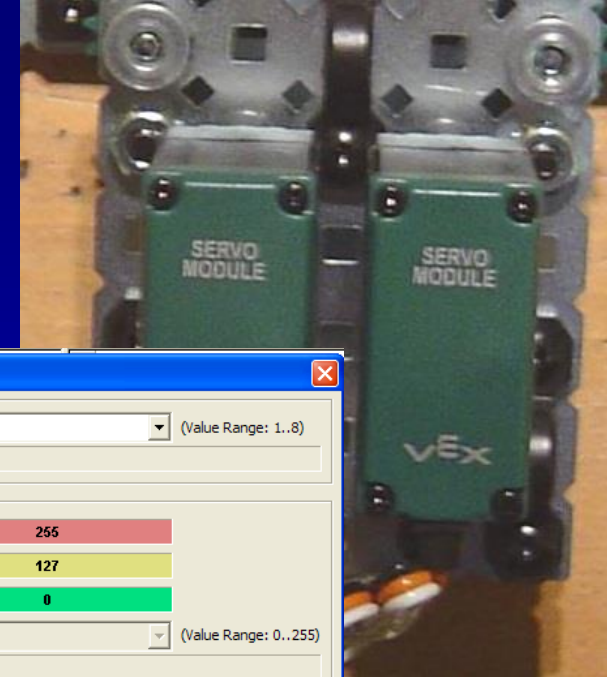
OK Cancel Help

Custom speed

Max. speed counterclockwise

Servo motor

Servos control the position of the motor shaft, angle of rotation



Robot moving forward

Main

Config

Globals

BEGIN

Variables

↑

↓

END

```

void main ( void )
{
  SetMotor [ 1 , 255 ] ;
  SetMotor [ 2 , 0 ] ;
}

```

```

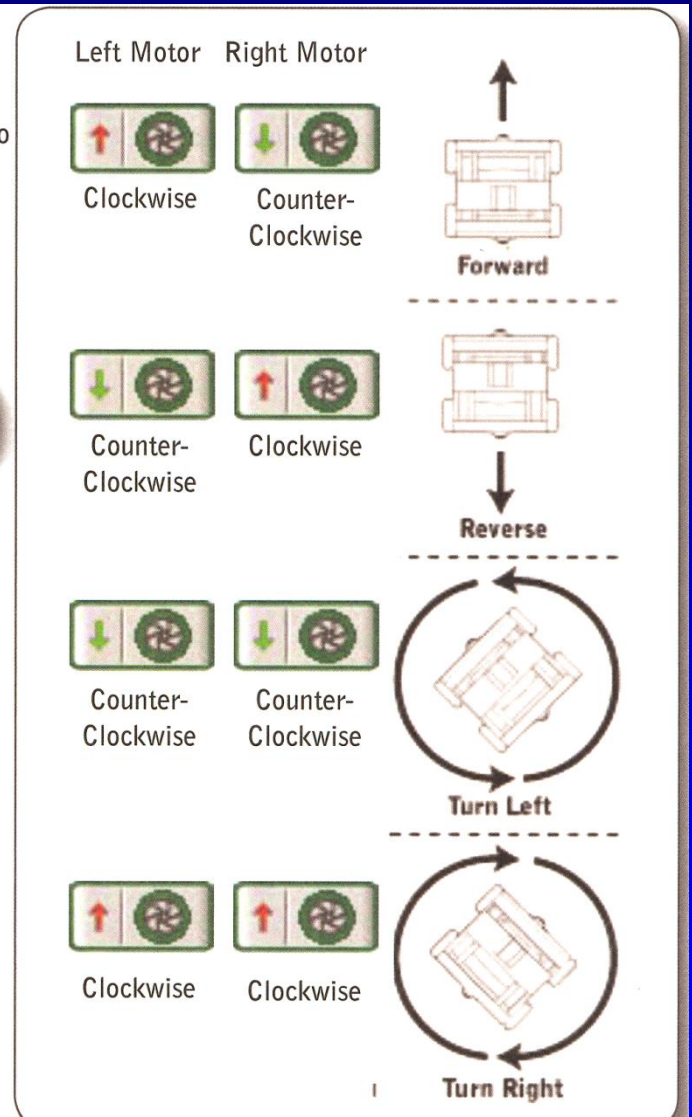
1 #include <...>
2
3 void main
4 {
5     SetM
6     SetM
7 }

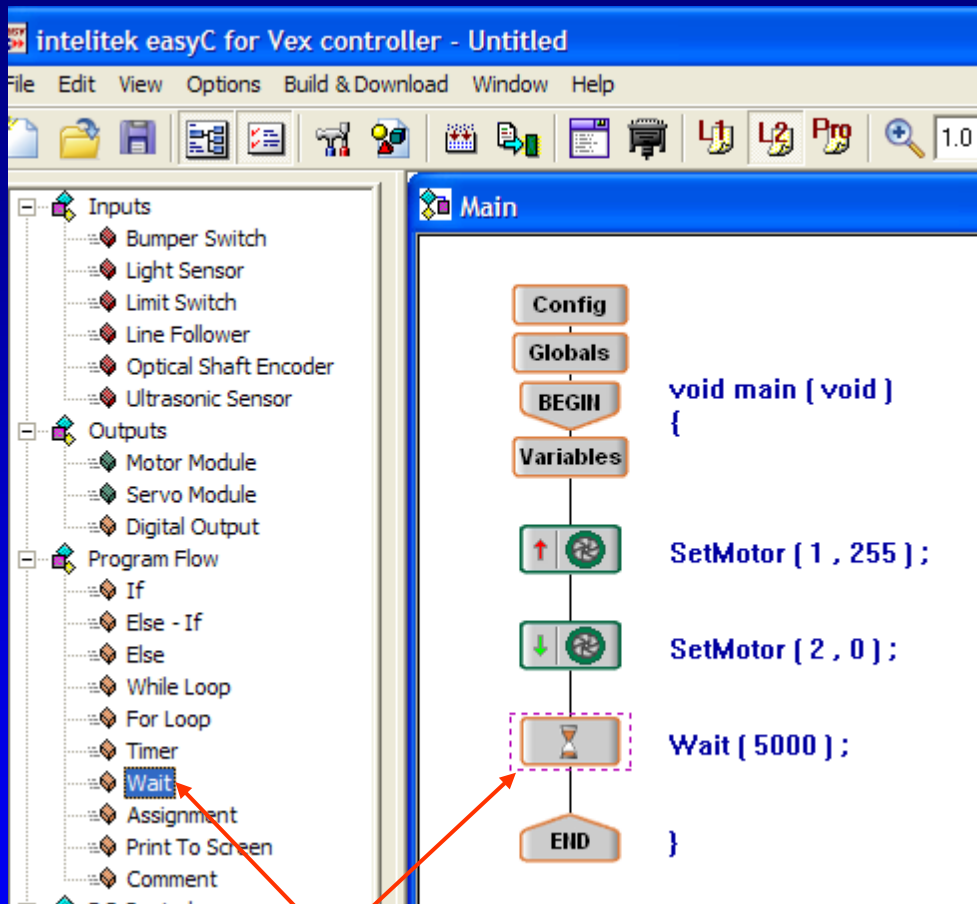
```

Motor 1 rotates at max speed clockwise

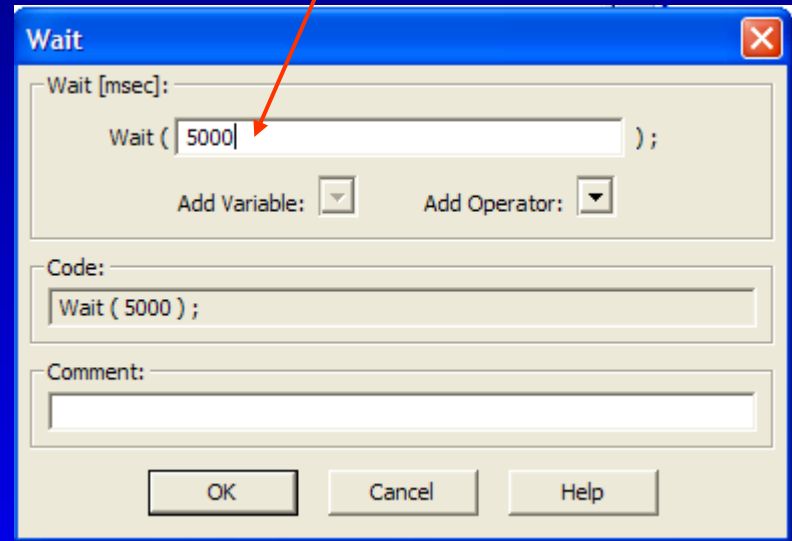
Motor 2 rotates at max speed counterclockwise

Car moves forward at high speed



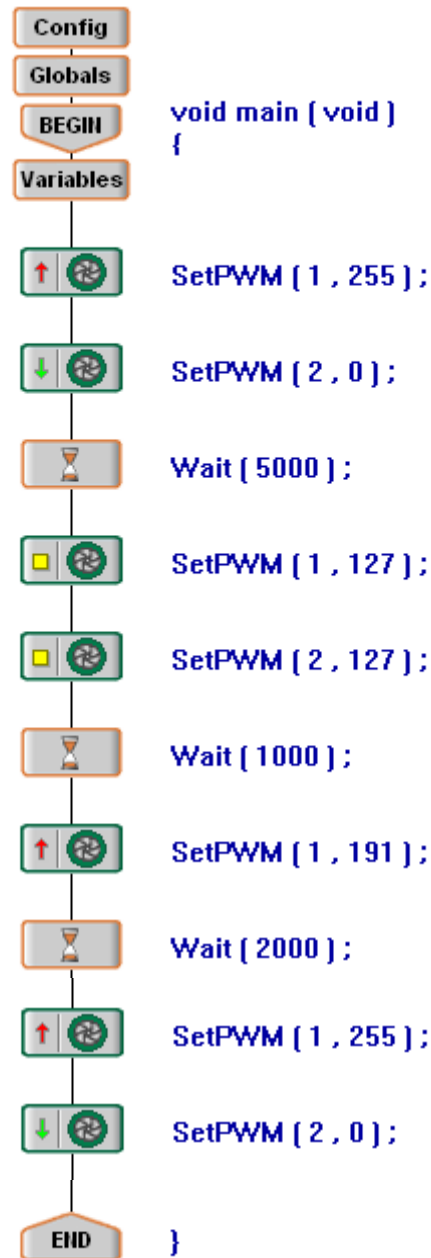
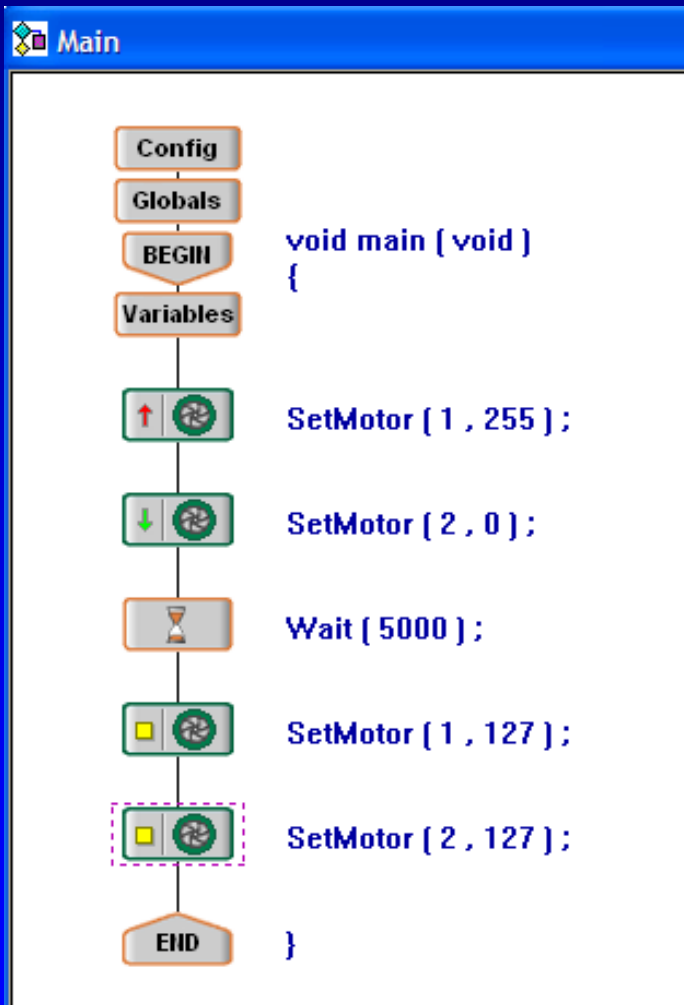


Provide time in millisecond



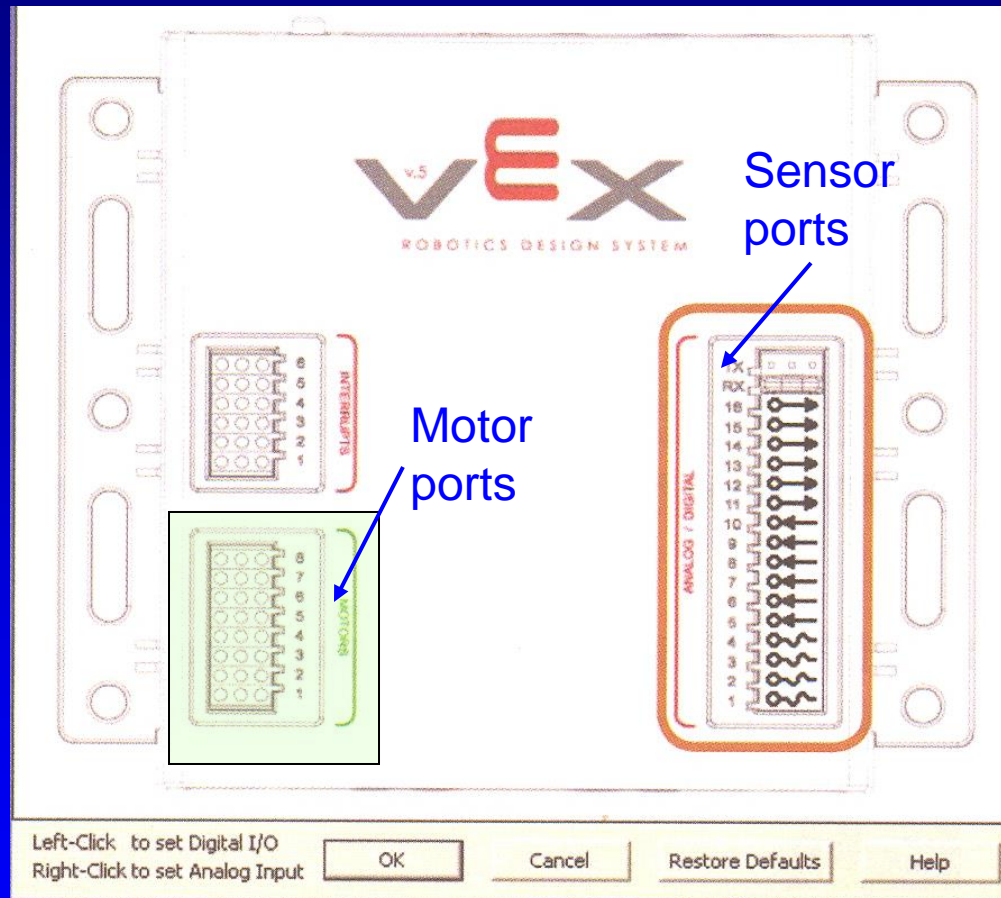
The Wait function

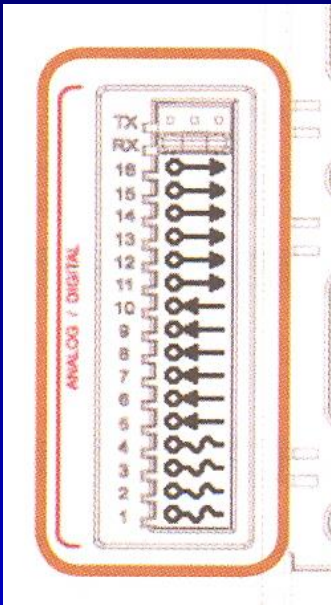
Robot goes forward at top speed for 5 seconds and then stops



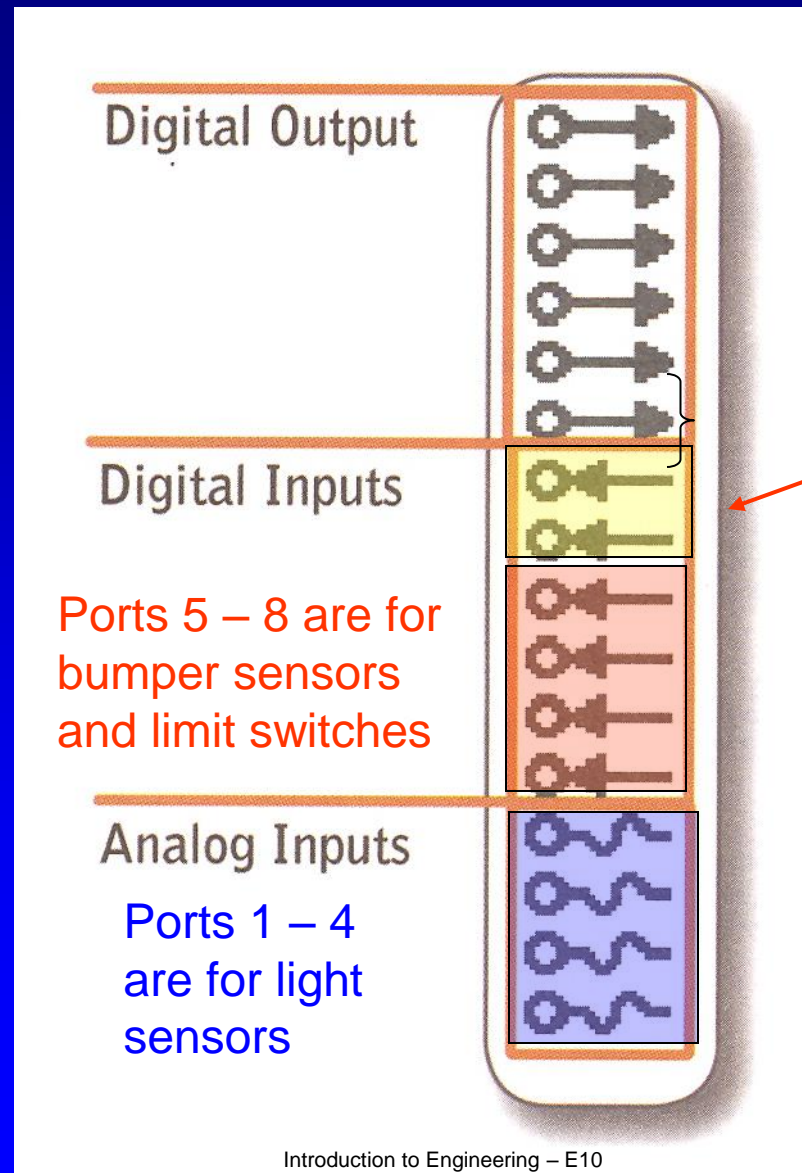
Robot goes forward at top speed for 5 seconds and then stops. It waits 1 second and then turns right for 2 seconds and goes forward.

Controller





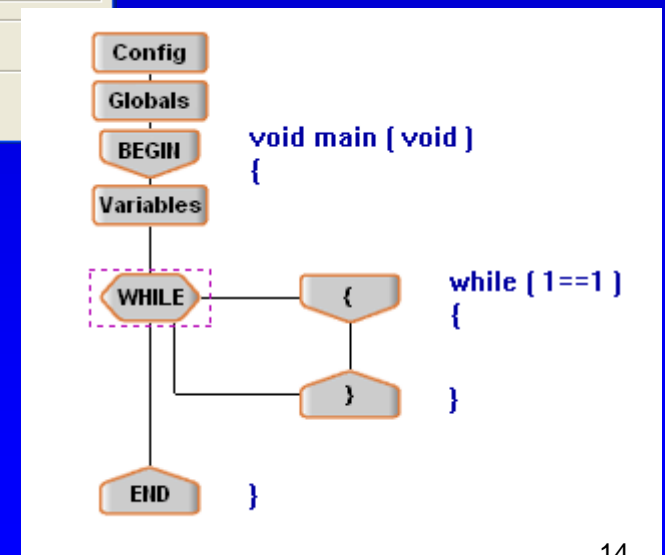
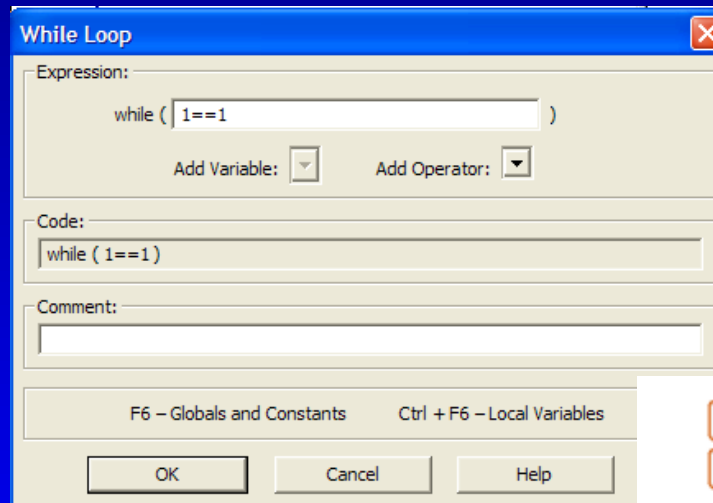
The section labeled “Analog/Digital” represents the sensor ports on the controller



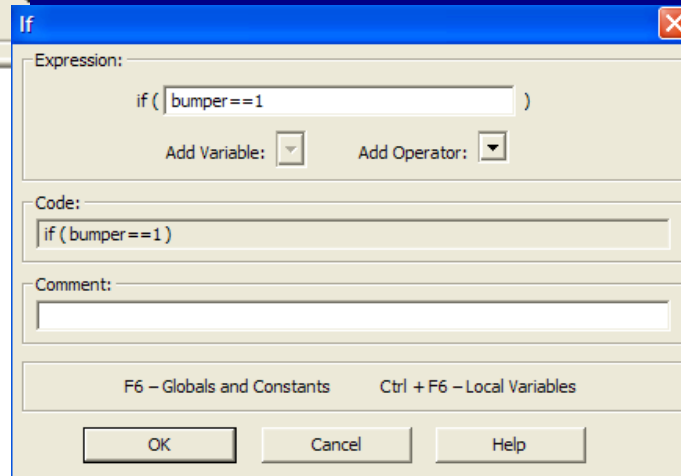
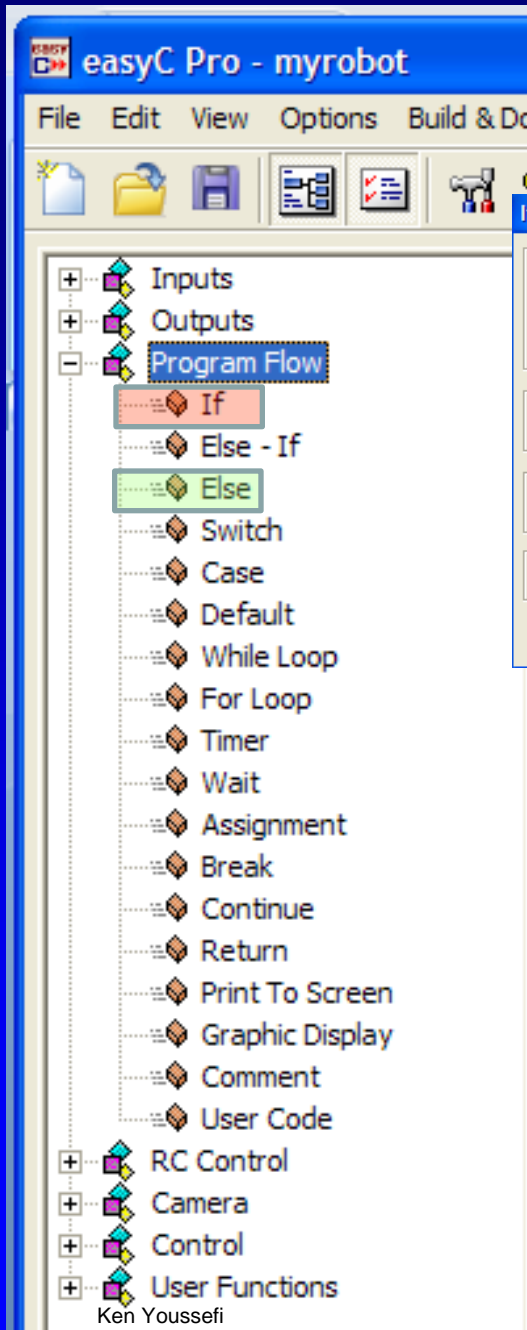
Ports 9 and 10 are preprogrammed for stops when bumper sensor is activated

While Loop

The While Loop code is executed as long as the condition is true

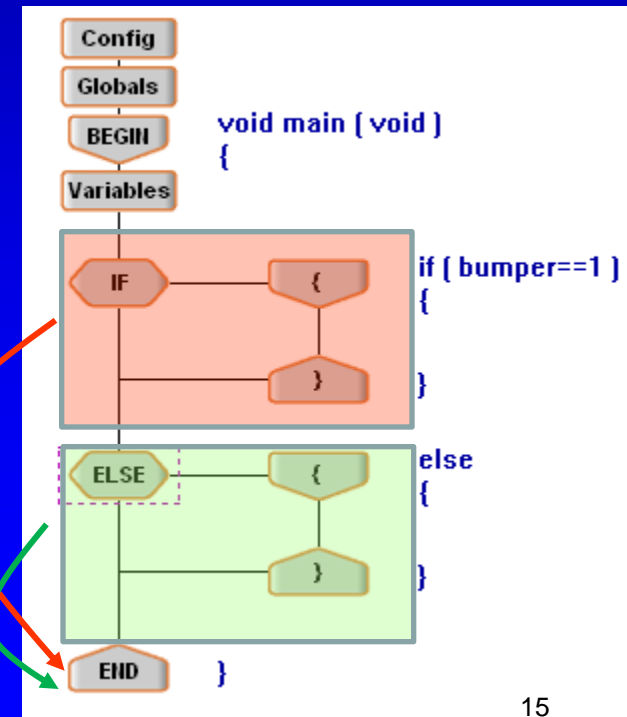


If & Else Statement



The *if* condition is evaluated (*bumper==1*). If the condition is true the code within the *if* brackets is executed.

If the condition is not true the code within the *Else* brackets is executed

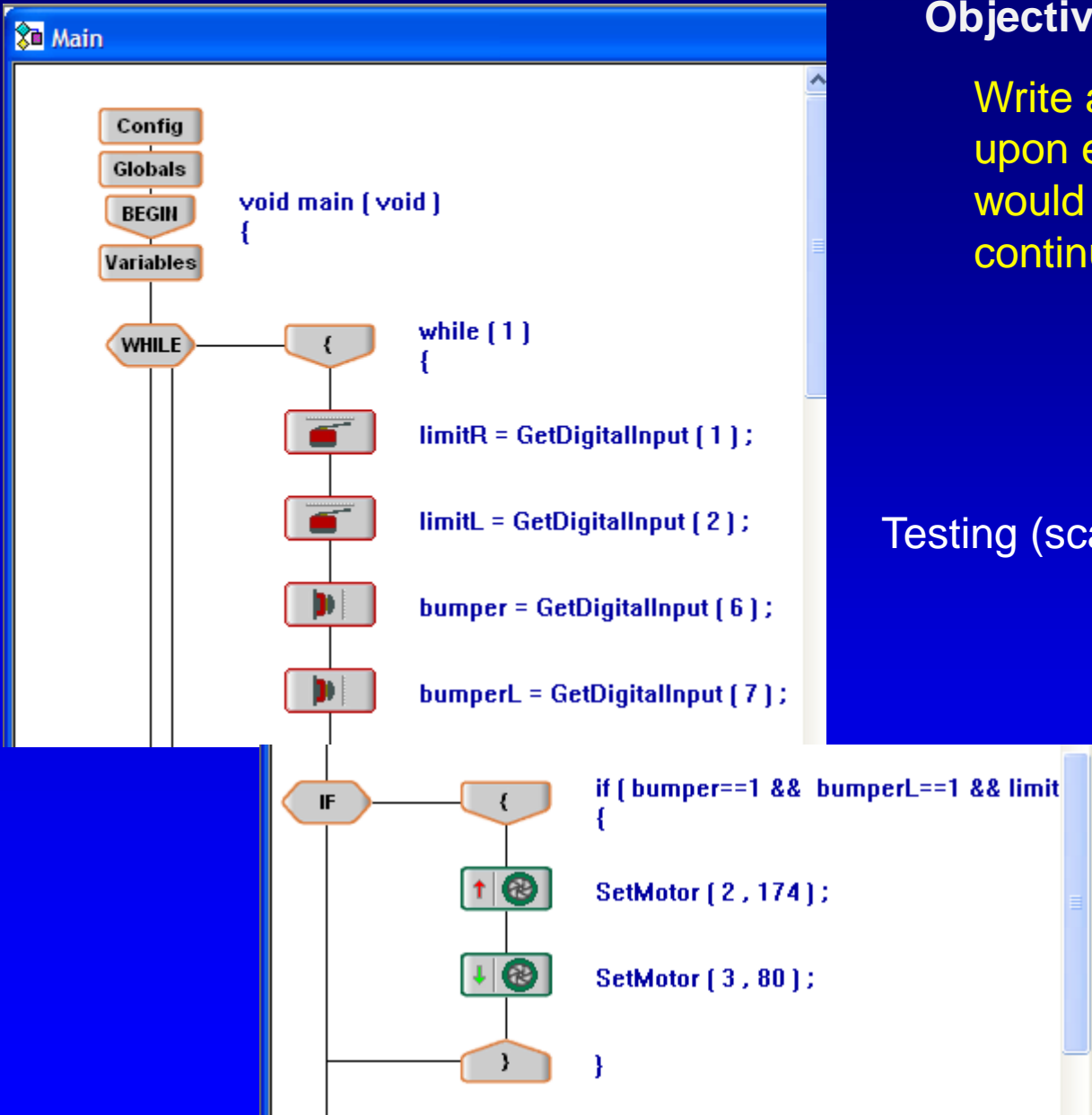


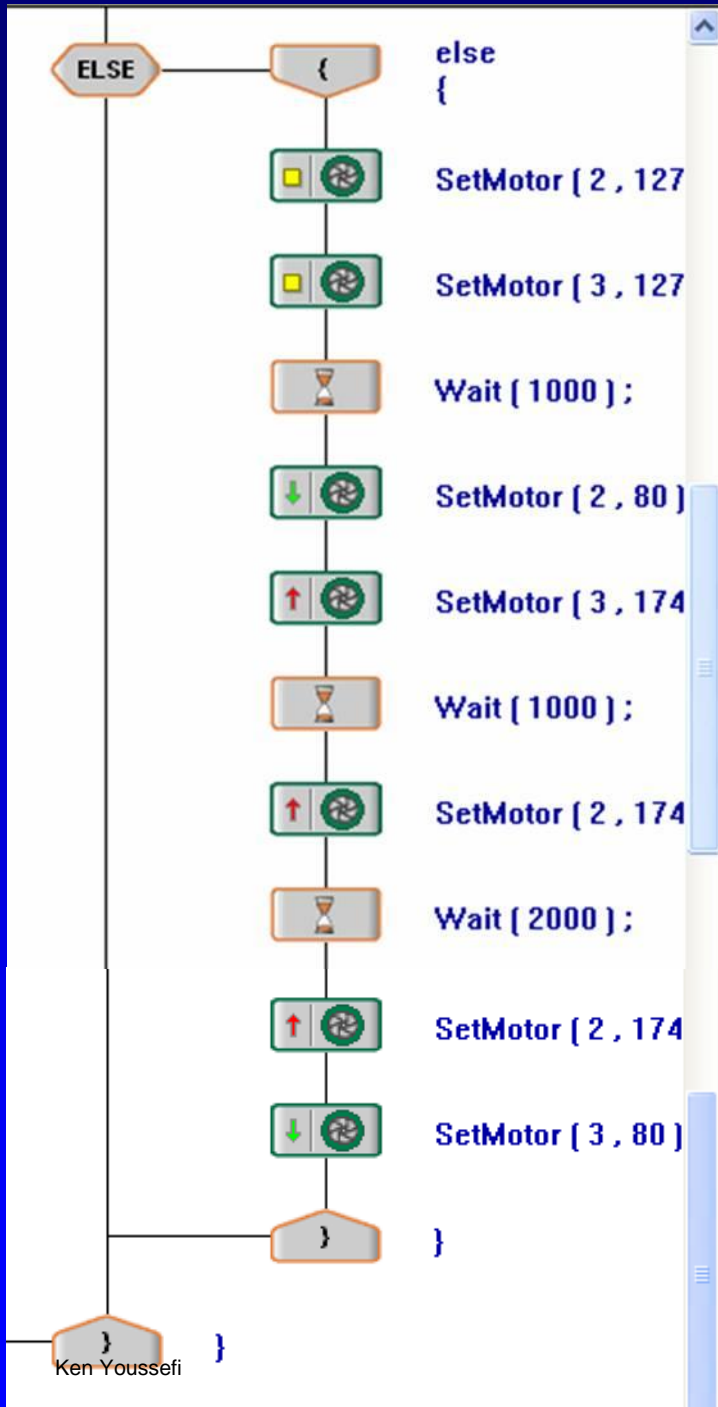
Objective

Write a program so that the robot, upon encountering any obstacle, would back up, then turn and continue.

Testing (scan) the 4 sensors

If all sensors off (no obstacle), then go forward





ELSE → one of the sensors is enabled

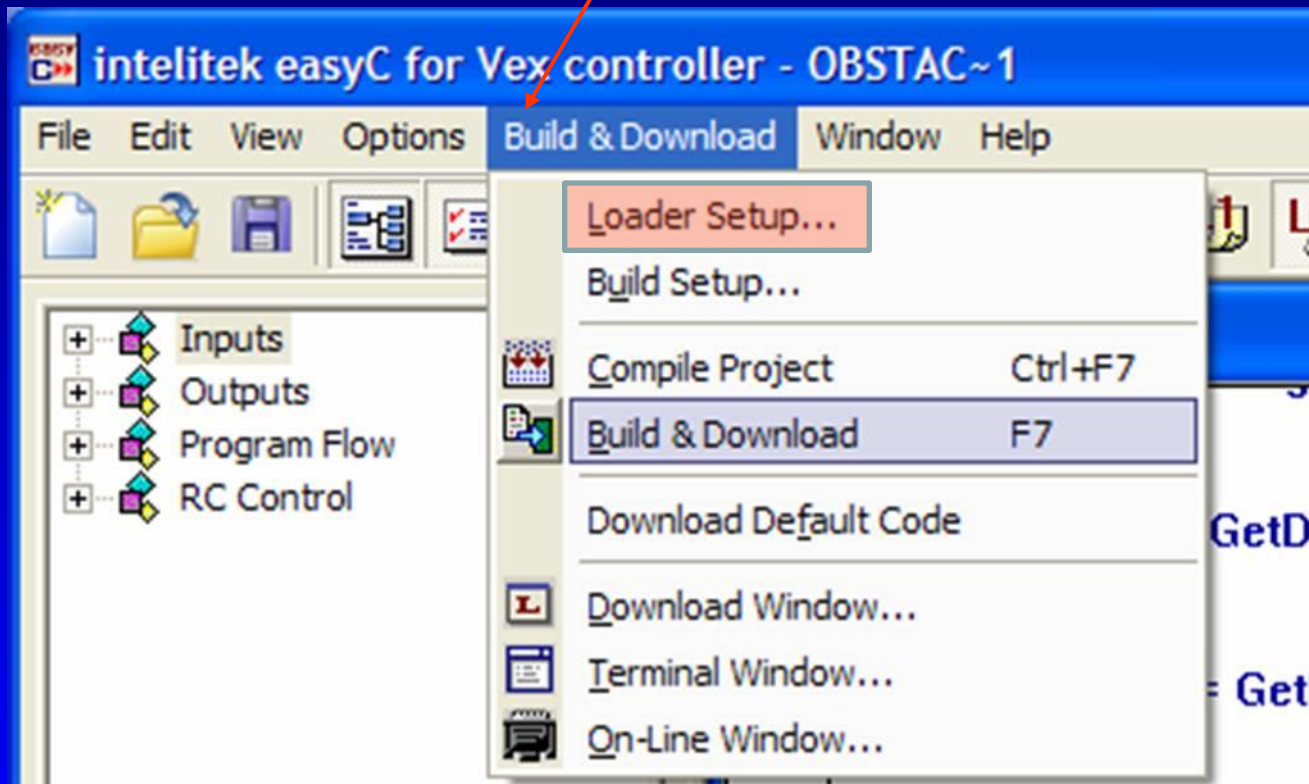
} Stop both motors for 1 second

} Back up for 1 second

} Turn for 2 seconds

} Go forward

To download the program to the controller mounted on the robot, select *Build & Download* from the pull-down menu *Build & Download*.



Connect the computer via USB to the controller for the download (orange cables), follow the screen instructions. **Make sure the COM port is correct, change the COM port using the *Loader Setup*.** Check for compilation errors and other errors on the Troubleshooting section of the manual.