Vex Ultrasonic Sensor (US) Interfacing and Programming

Armen Boursalian

SJSU – Introduction to Engineering – E10

November 2012

Ultrasonic Sensors

- Generally have a transmitter and receiver.
- Transmits a high-frequency sound.
- Waits to receive echo.
- Calculates distance based on time it took to receive the echo.

$$d = (v)(t)$$

Round-Trip Distance

$$d = (v)(t)/2$$

One-Way Distance

d = distance, *v* = speed of sound, *t* = time to receive echo

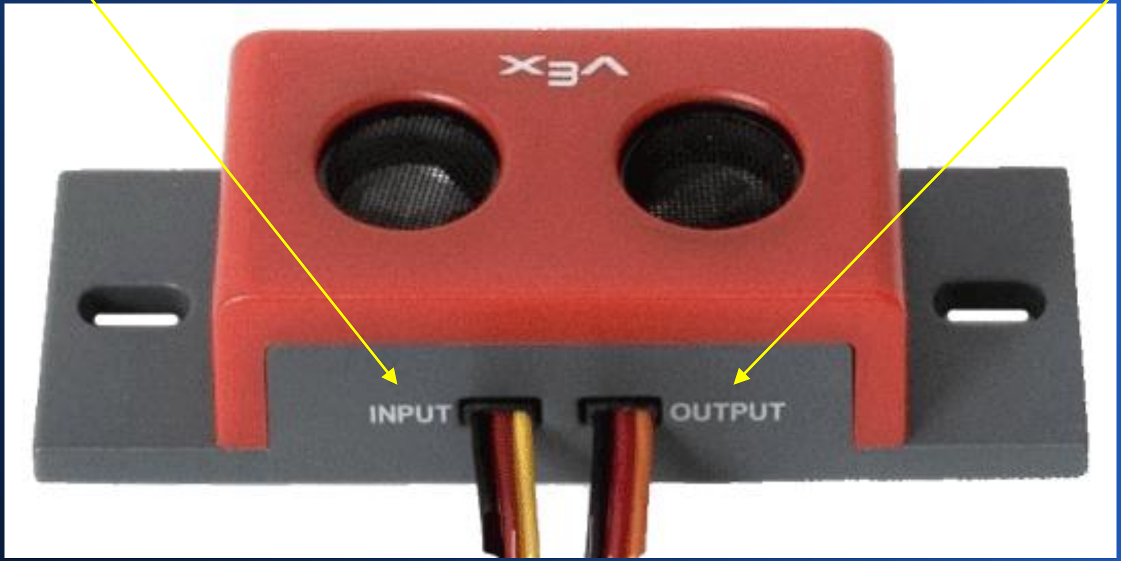
The Vex Ultrasonic Sensor

INPUT

- Sends ultrasonic wave
- Connect to Digital OUTPUT*

OUTPUT

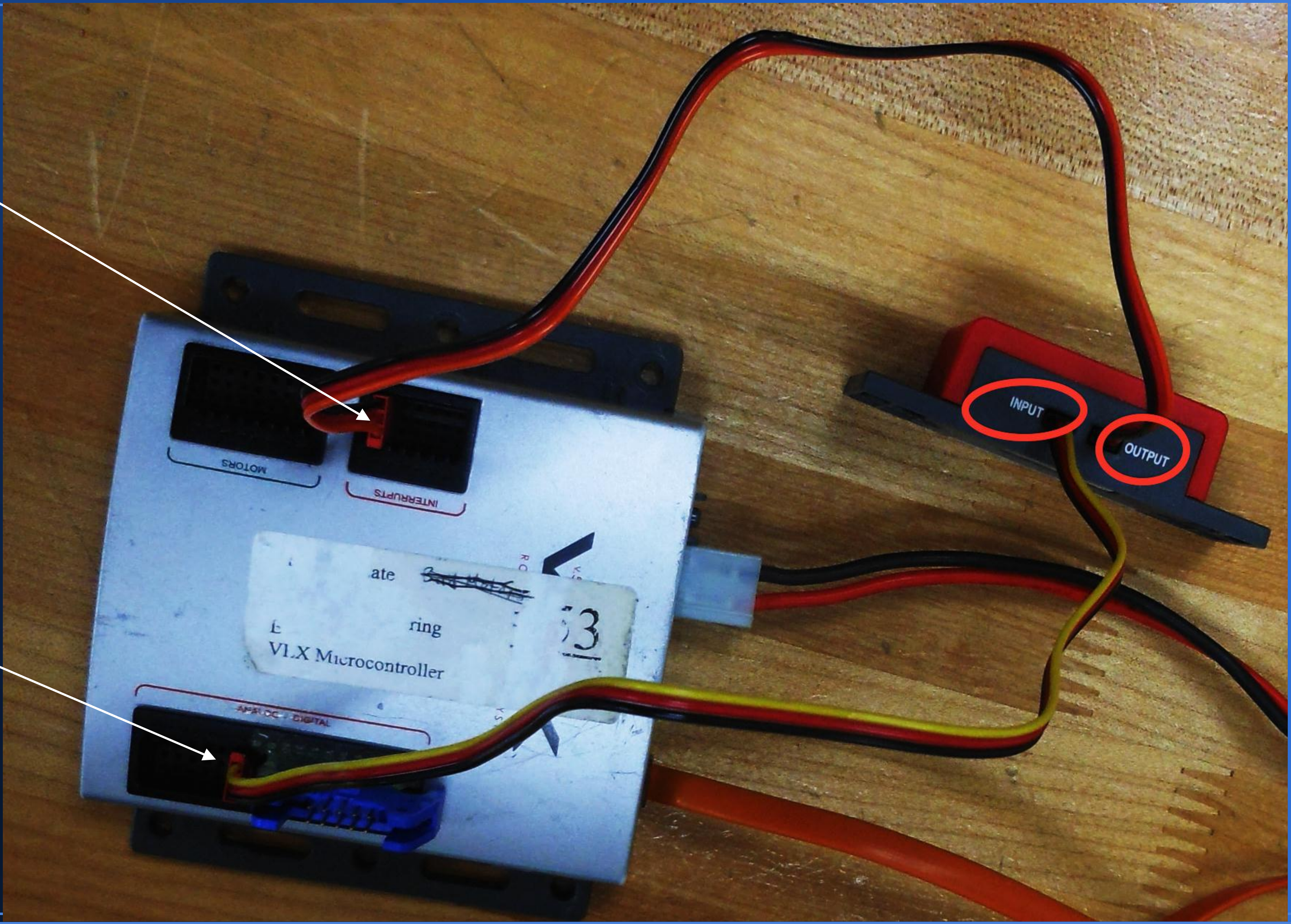
- Receives the echo
- Connect to INTERRUPT**



The labeling is a bit counter-intuitive!

*User may have to turn a Digital INPUT into Digital OUTPUT in easyC (See Page 5).
**Any INTERRUPT port may be chosen.

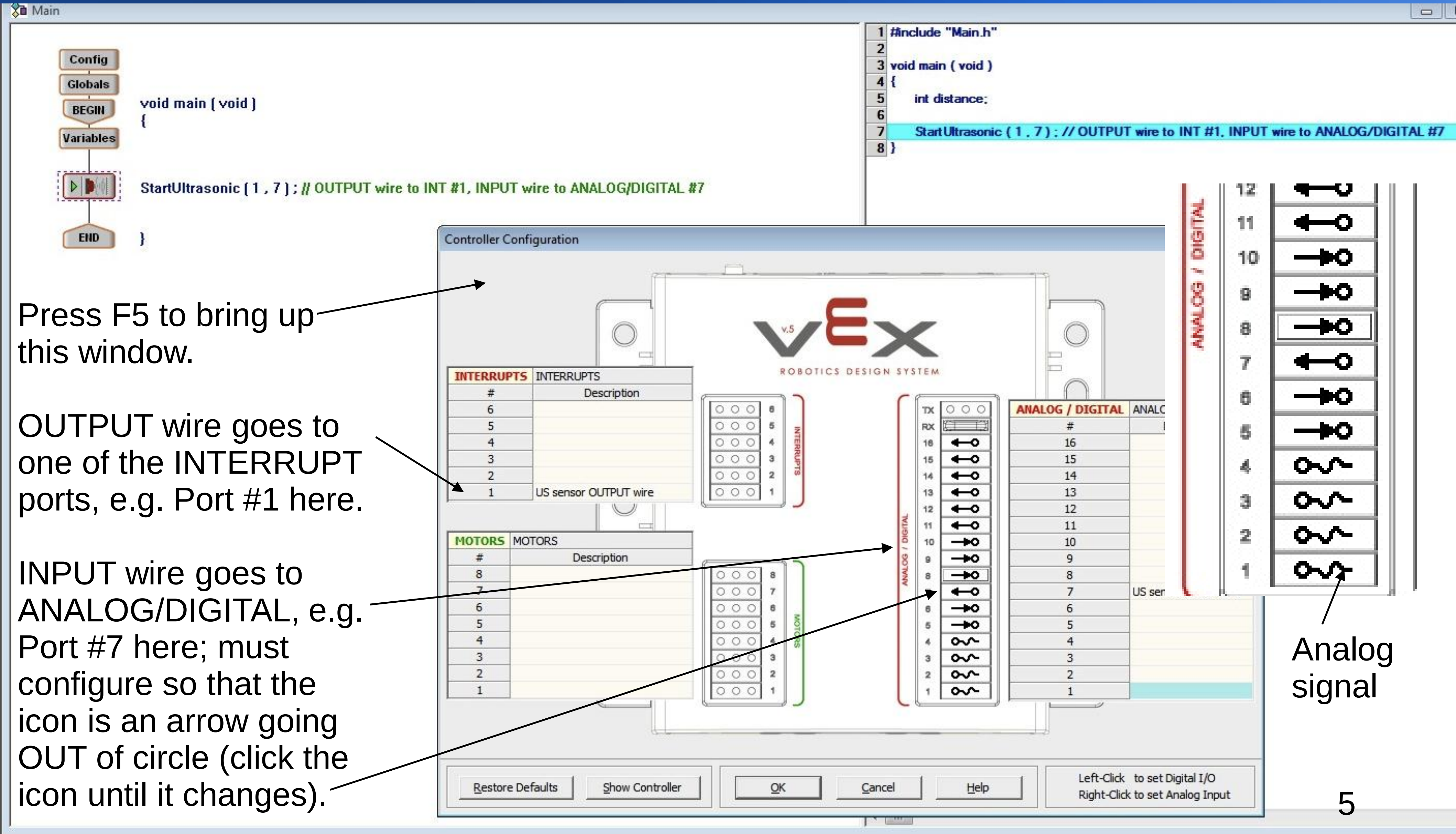
Interfacing Example



INTERRUPT
Port #1

ANALOG/DIGITAL
Port #7

Check the Vex Controller Configuration



```
void main ( void )
{
StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
}
```

```
#include "Main.h"

void main ( void )
{
     int distance;

     StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
}
```

INTERRUPTS #	INTERRUPTS Description
6	
5	
4	
3	
2	
1	US sensor OUTPUT wire

MOTORS #	MOTORS Description
8	
7	
6	
5	
4	
3	
2	
1	

ANALOG / DIGITAL #	ANALO
16	
15	
14	
13	
12	
11	
10	
9	
8	
7	US ser
6	
5	
4	
3	
2	
1	

Left-Click to set Digital I/O
Right-Click to set Analog Input

Press F5 to bring up this window.

OUTPUT wire goes to one of the INTERRUPT ports, e.g. Port #1 here.

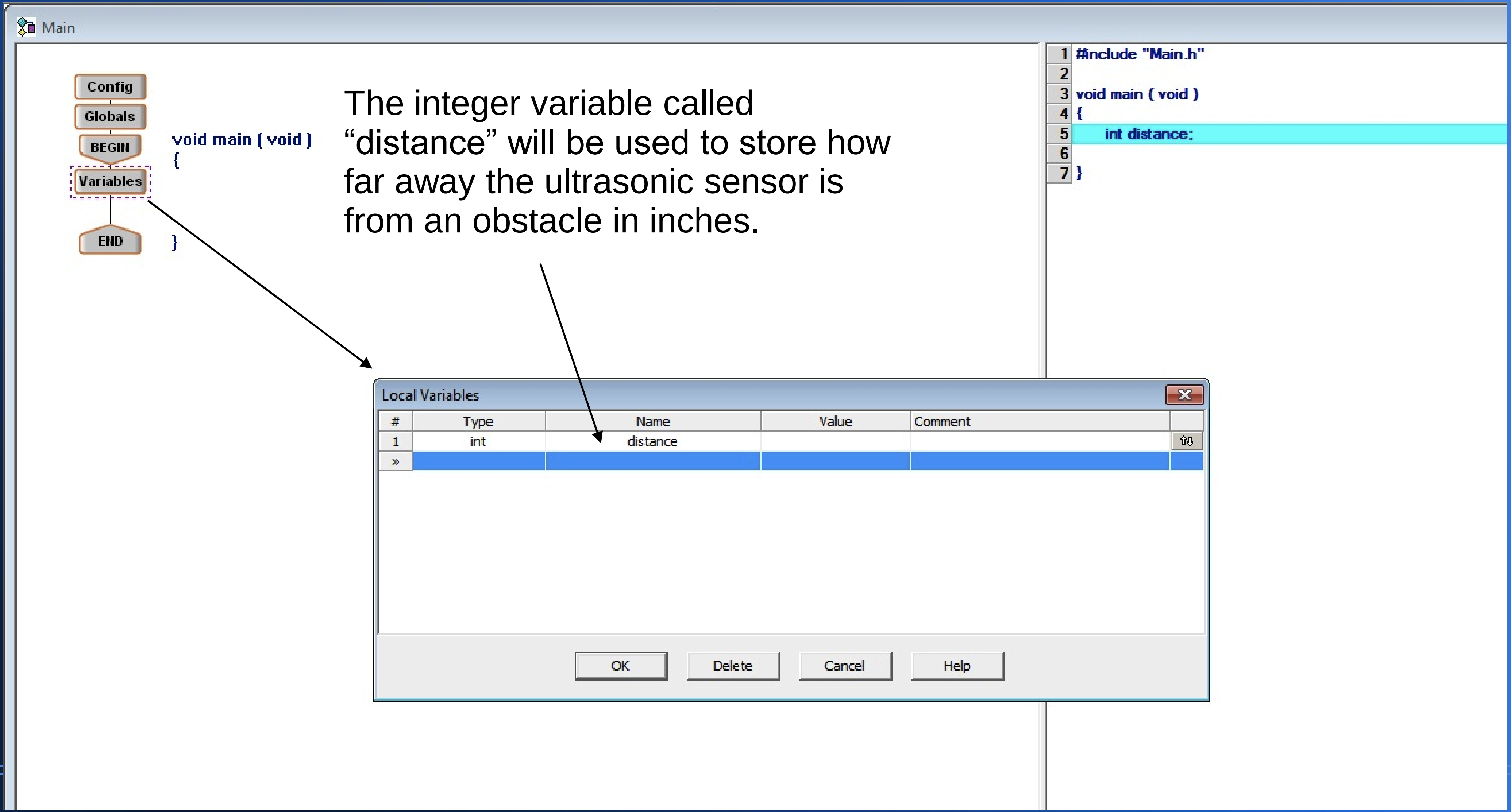
INPUT wire goes to ANALOG/DIGITAL, e.g. Port #7 here; must configure so that the icon is an arrow going OUT of circle (click the icon until it changes).

Analog signal

Programming The Ultrasonic Sensor

- 4 Steps must be taken to use the sensor:
  1) Declare a variable that will store the data received by the sensor (<u>number of inches</u>.)
  2) Call the StartUltrasonic(x,y) function (once only.)
     - x → INTERRUPT port #
     - y → Digital OUTPUT port #
  3) Include variable = GetUltrasonic(x,y) in a loop in order to poll for values (similar to using a bumper.)
     - Stores the number of inches in “variable”
  4) Use the variable to do something!

Step 1: Declare a Variable



```
void main ( void )
{

}
```

The integer variable called “distance” will be used to store how far away the ultrasonic sensor is from an obstacle in inches.

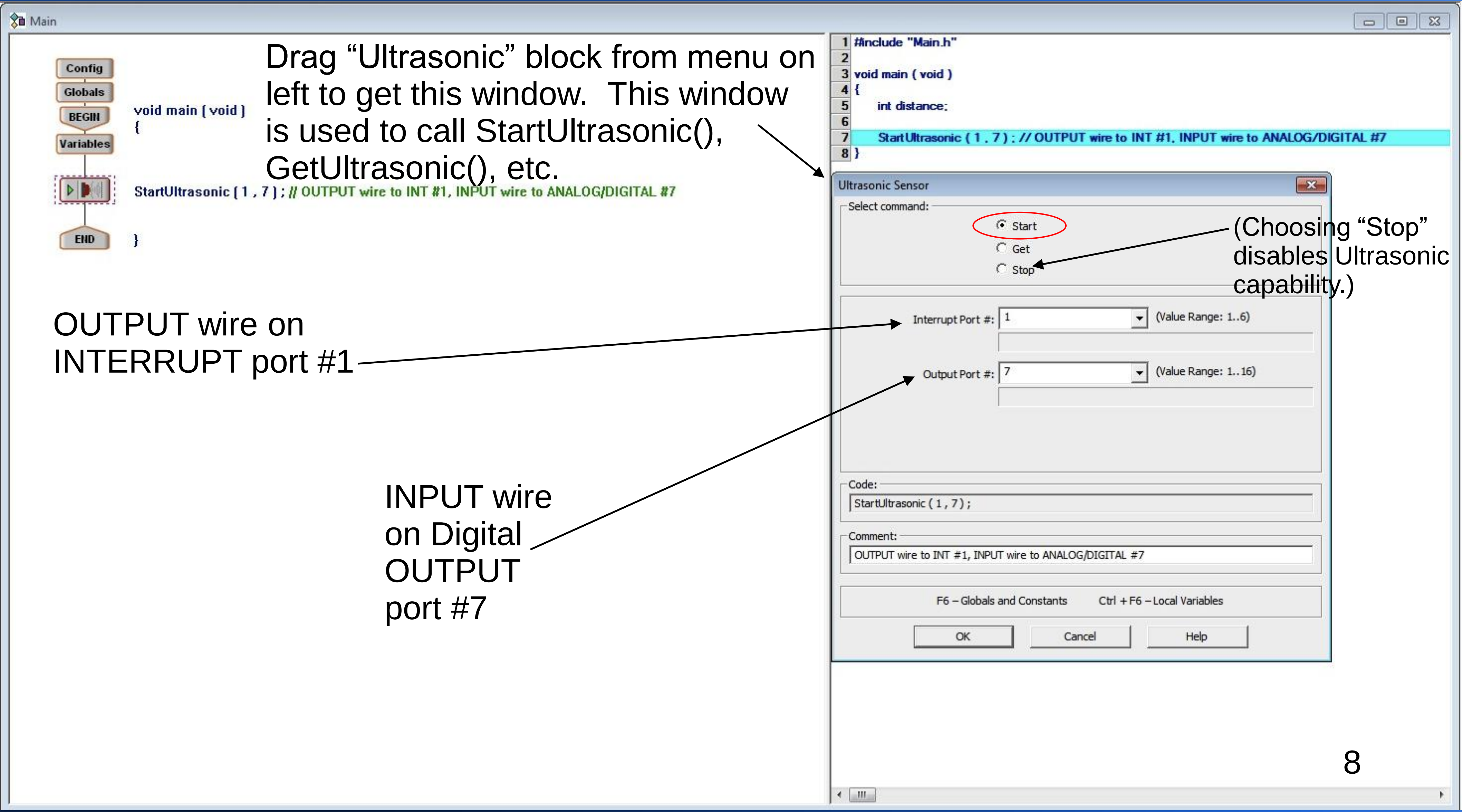
#	Type	Name	Value	Comment
1	int	distance		
»				

```
#include "Main.h"

void main ( void )
{
    int distance;

}
```

Step 2: Call the StartUltrasonic() Function



```
void main ( void )
{
StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
}
```

Drag "Ultrasonic" block from menu on left to get this window. This window is used to call StartUltrasonic(), GetUltrasonic(), etc.

```
#include "Main.h"

void main ( void )
{
    int distance;

    StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
}
```

Ultrasonic Sensor

Select command:

● Start
○ Get
○ Stop

(Choosing "Stop" disables Ultrasonic capability.)

Interrupt Port #: 1 (Value Range: 1..6)

Output Port #: 7 (Value Range: 1..16)

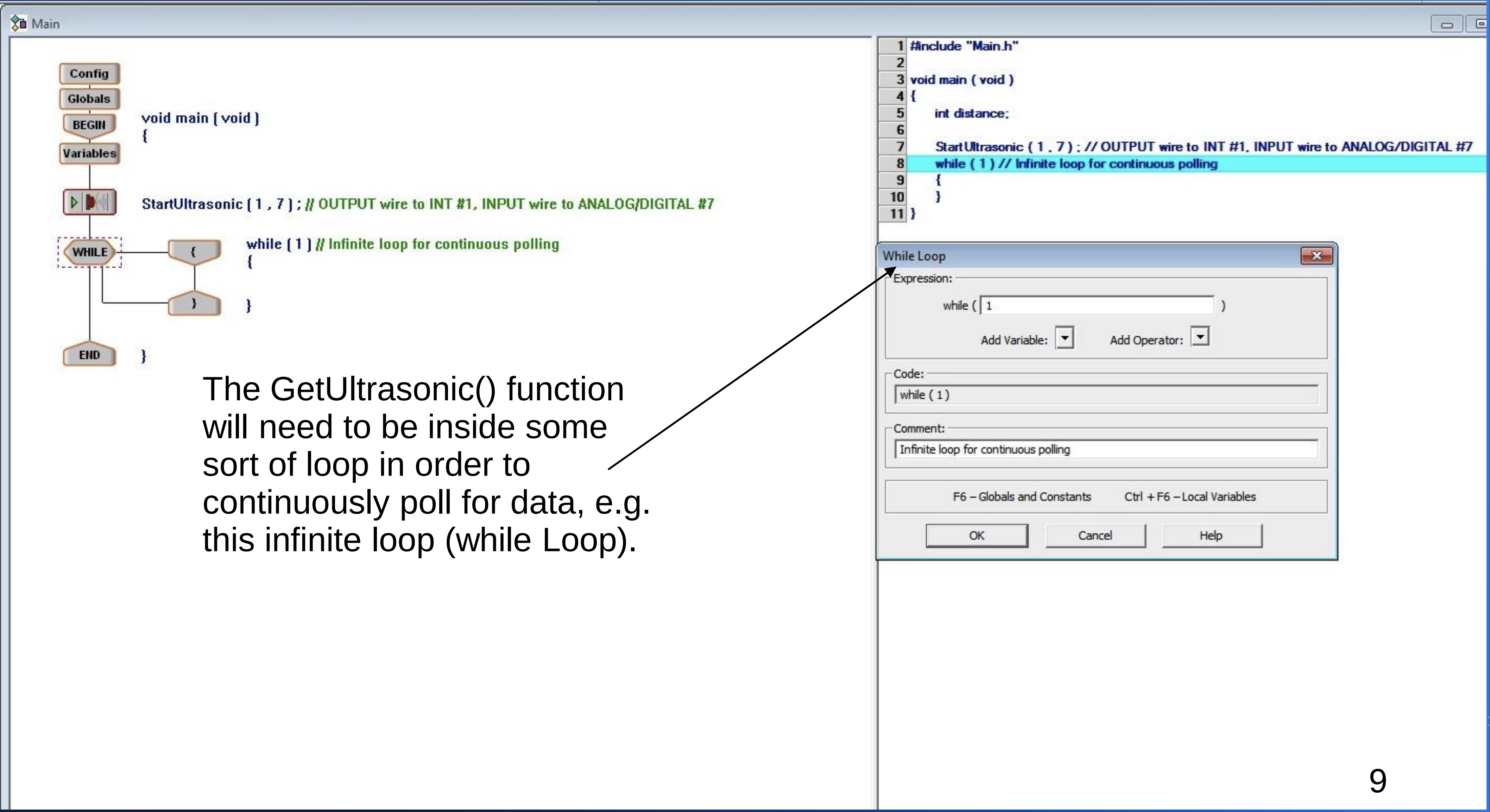
OUTPUT wire on INTERRUPT port #1

INPUT wire on Digital OUTPUT port #7

Code: StartUltrasonic (1 , 7) ;

Comment: OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7

Step 3: Polling for Data



```
void main ( void )
{

StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7

while ( 1 ) // Infinite loop for continuous polling
{

}

}
```

```
#include "Main.h"

void main ( void )
{
    int distance;

    StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
    while ( 1 ) // Infinite loop for continuous polling
    {
    }
}
```

While Loop

Expression:

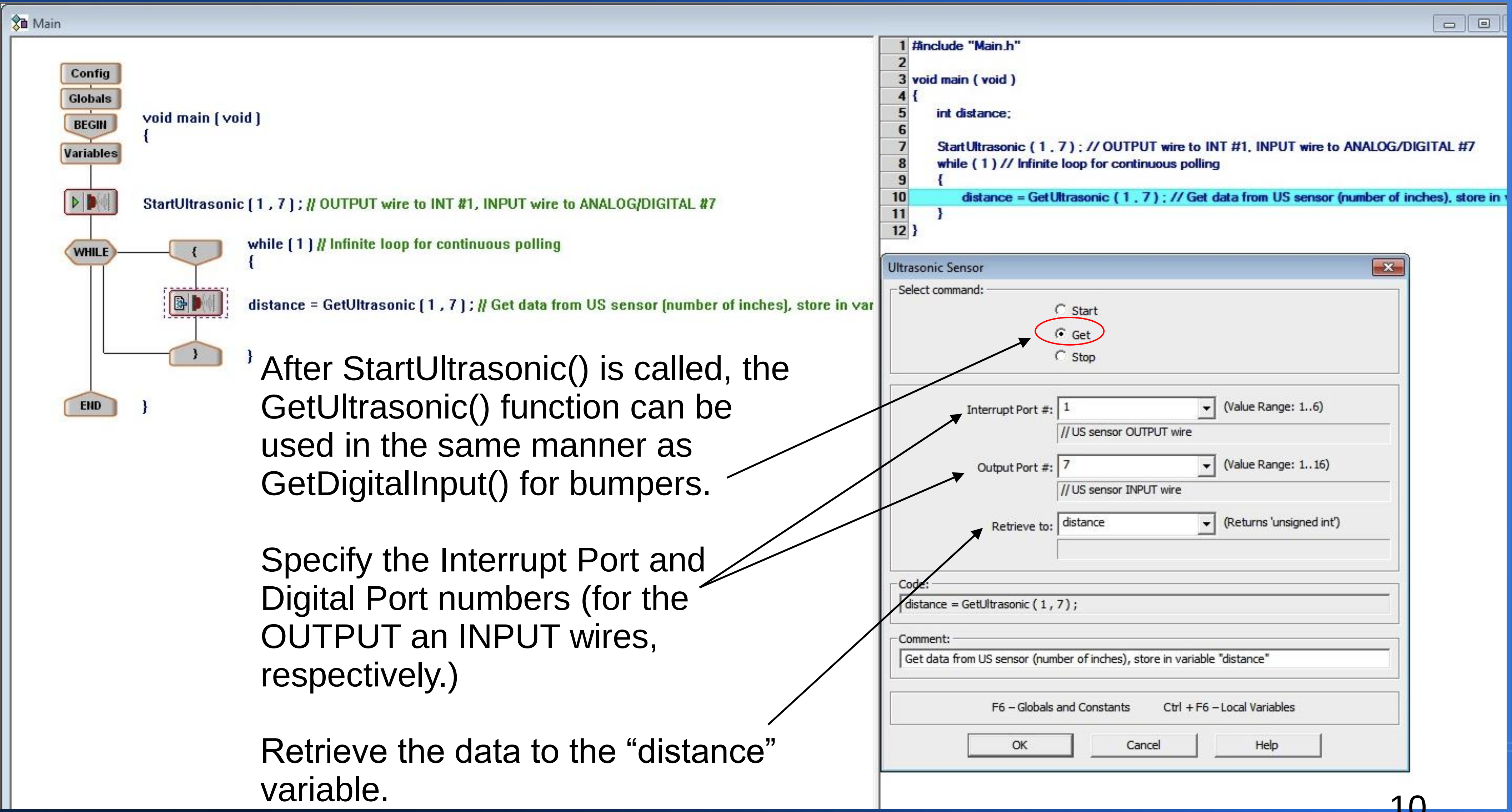
while (1)

Code: while (1)

Comment: Infinite loop for continuous polling

The GetUltrasonic() function will need to be inside some sort of loop in order to continuously poll for data, e.g. this infinite loop (while Loop).

Step 3: Polling for Data



```
void main ( void )
{

StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7

while ( 1 ) // Infinite loop for continuous polling
{

distance = GetUltrasonic ( 1 , 7 ) ; // Get data from US sensor (number of inches), store in var

}

}
```

```
#include "Main.h"

void main ( void )
{
    int distance;

    StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
    while ( 1 ) // Infinite loop for continuous polling
    {
        distance = GetUltrasonic ( 1 , 7 ) ; // Get data from US sensor (number of inches), store in
    }
}
```

Select command:
○ Start
● Get
○ Stop

Interrupt Port #: 1 (Value Range: 1..6)
// US sensor OUTPUT wire

Output Port #: 7 (Value Range: 1..16)
// US sensor INPUT wire

Retrieve to: distance (Returns 'unsigned int')

Code:
distance = GetUltrasonic (1 , 7) ;

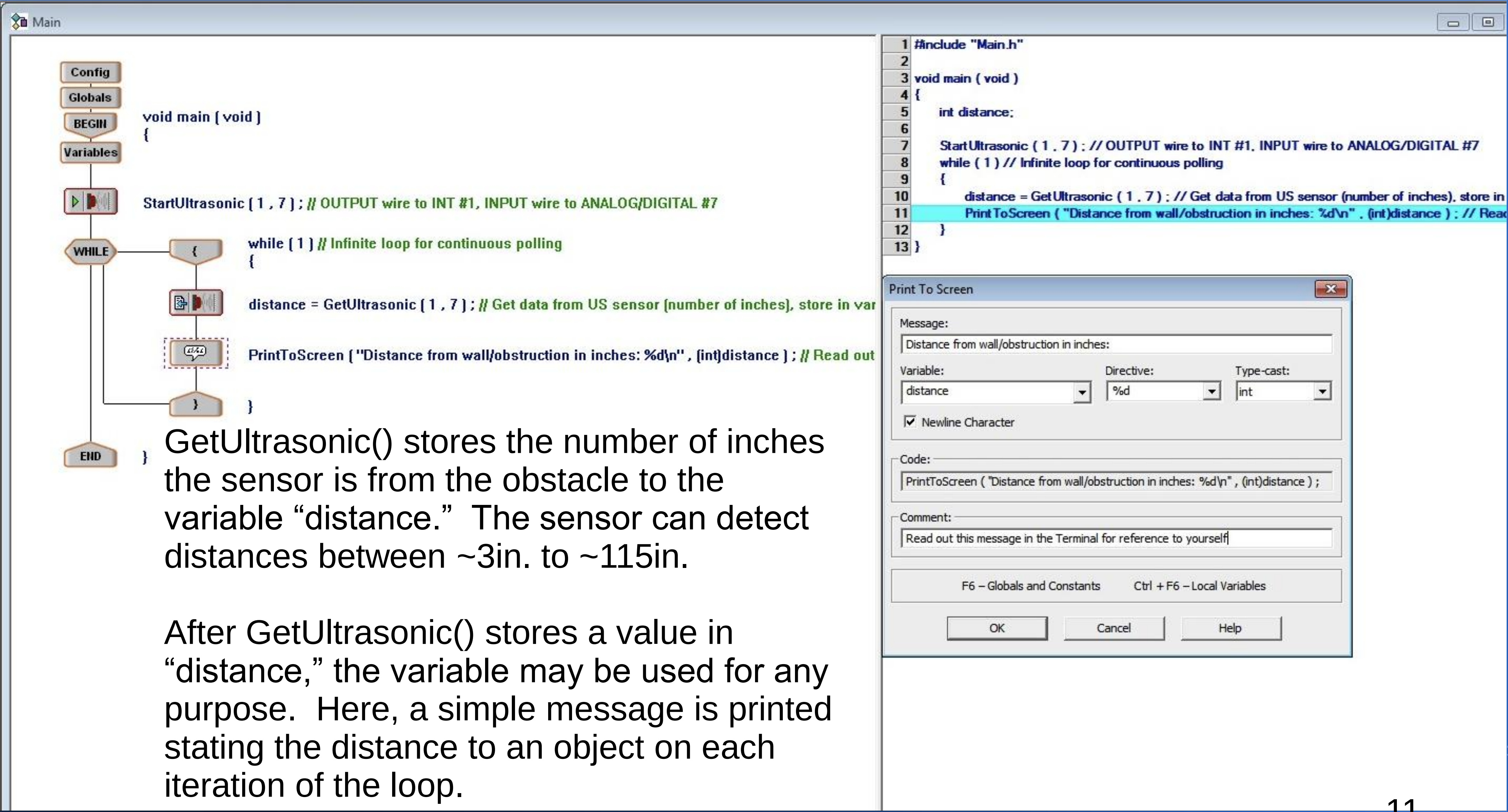
Comment:
Get data from US sensor (number of inches), store in variable "distance"

After StartUltrasonic() is called, the GetUltrasonic() function can be used in the same manner as GetDigitalInput() for bumpers.

Specify the Interrupt Port and Digital Port numbers (for the OUTPUT an INPUT wires, respectively.)

Retrieve the data to the “distance” variable.

Step 4: Use the Data



```
void main ( void )
{

StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7

      while ( 1 ) // Infinite loop for continuous polling
      {

      distance = GetUltrasonic ( 1 , 7 ) ; // Get data from US sensor (number of inches), store in var

      PrintToScreen ( "Distance from wall/obstruction in inches: %d\n" , (int)distance ) ; // Read out

      }

}
```

```
#include "Main.h"

void main ( void )
{
      int distance;

      StartUltrasonic ( 1 , 7 ) ; // OUTPUT wire to INT #1, INPUT wire to ANALOG/DIGITAL #7
      while ( 1 ) // Infinite loop for continuous polling
      {
            distance = GetUltrasonic ( 1 , 7 ) ; // Get data from US sensor (number of inches), store in
            PrintToScreen ( "Distance from wall/obstruction in inches: %d\n" , (int)distance ) ; // Read
      }
}
```

Message: Distance from wall/obstruction in inches:

Variable: distance

Directive: %d

Type-cast: int

☑ Newline Character

Code: PrintToScreen ("Distance from wall/obstruction in inches: %d\n" , (int)distance) ;

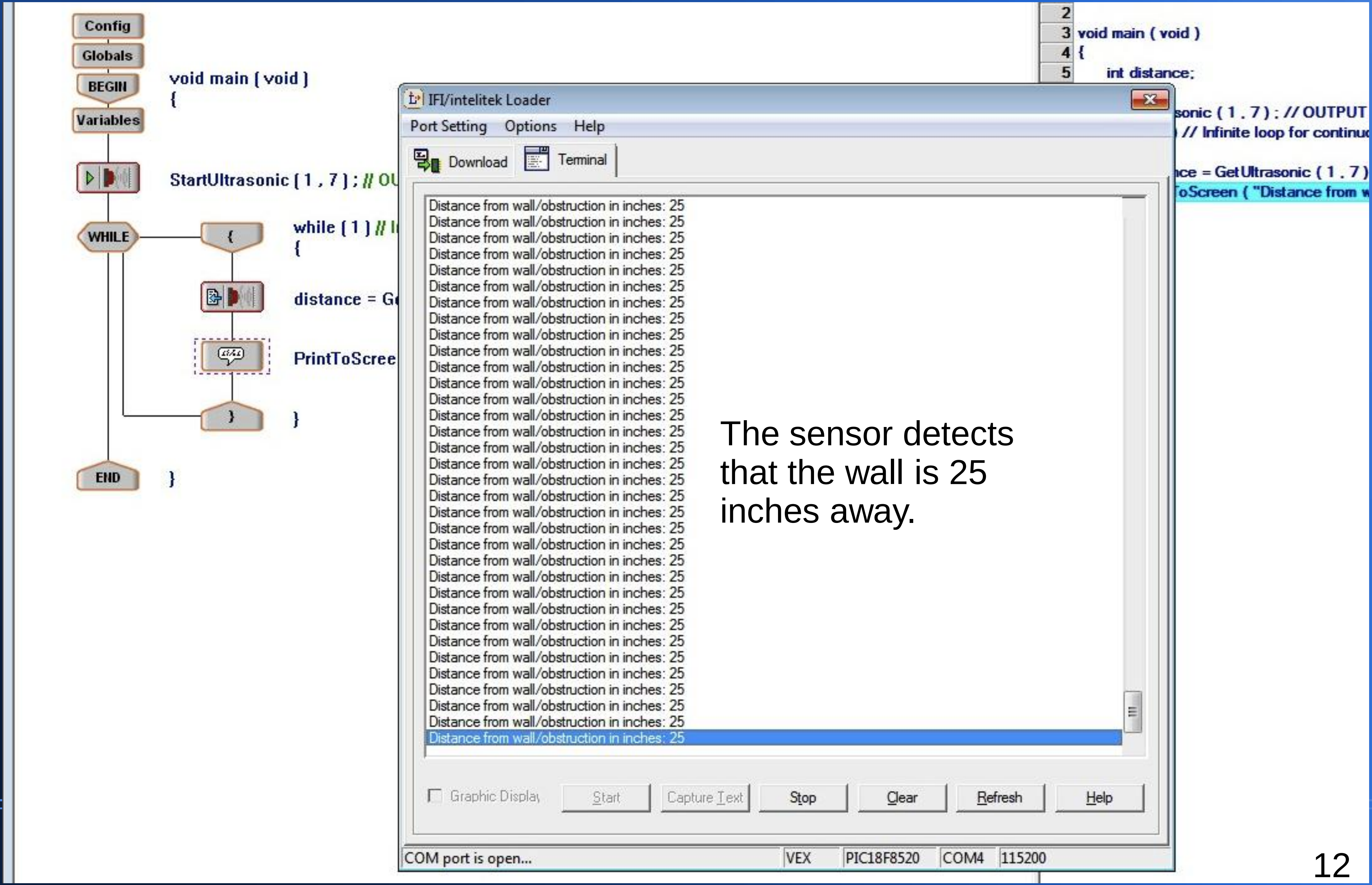
Comment: Read out this message in the Terminal for reference to yourself

F6 – Globals and Constants   Ctrl + F6 – Local Variables

GetUltrasonic() stores the number of inches the sensor is from the obstacle to the variable “distance.” The sensor can detect distances between ~3in. to ~115in.

After GetUltrasonic() stores a value in “distance,” the variable may be used for any purpose. Here, a simple message is printed stating the distance to an object on each iteration of the loop.

Step 4: Use the Data (Example)



```
void main ( void )
{
StartUltrasonic ( 1 , 7 ) ; // OU
    while ( 1 ) // I
    {
    distance = Ge
    PrintToScree
    }
}
```

```
void main ( void )
{
     int distance;
sonic ( 1 , 7 ) ; // OUTPUT
) // Infinite loop for continu
nce = GetUltrasonic ( 1 , 7 )
oScreen ( "Distance from w
```

```
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
Distance from wall/obstruction in inches: 25
```

COM port is open... | VEX | PIC18F8520 | COM4 | 115200

The sensor detects that the wall is 25 inches away.