A Tutorial

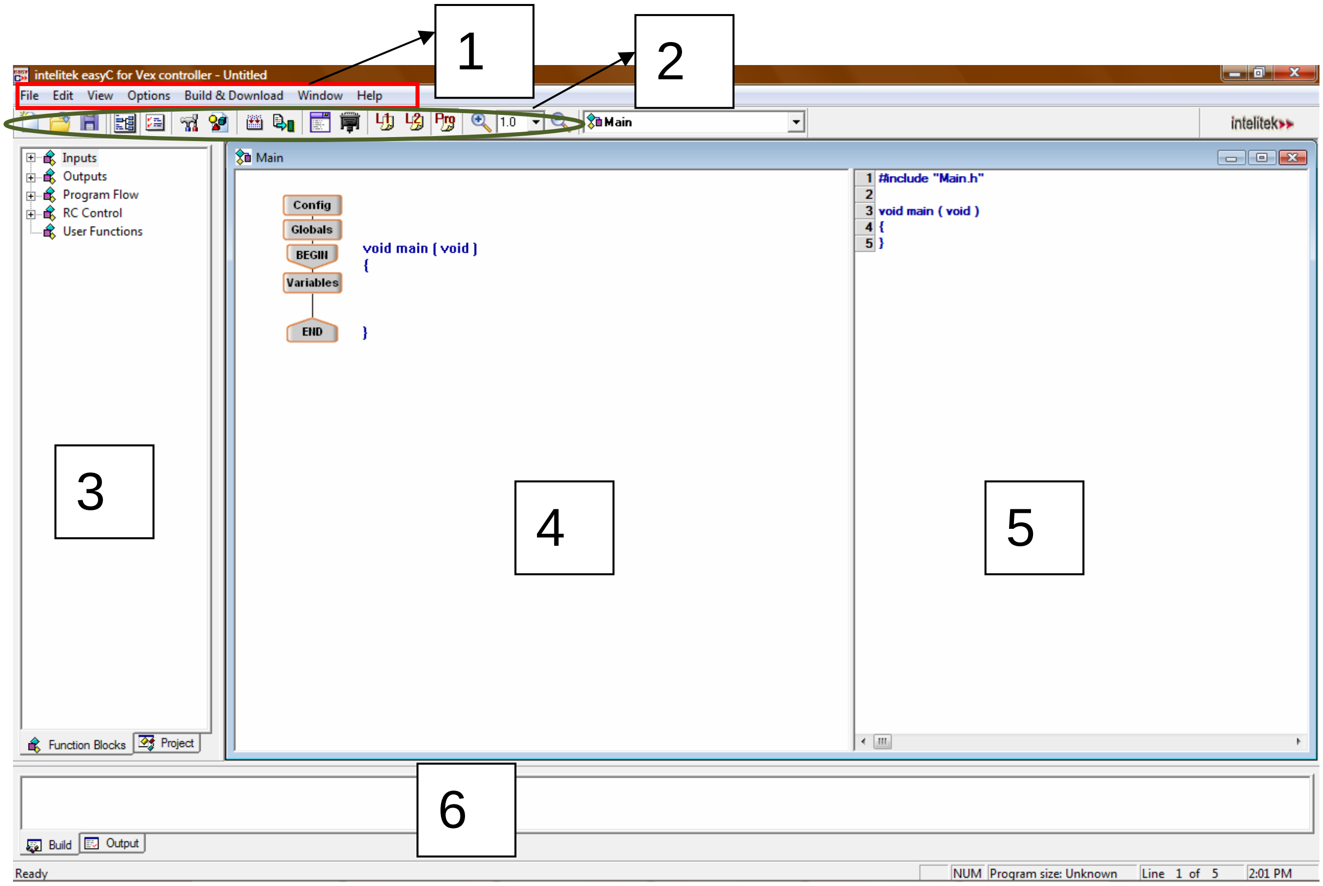
On

Understanding

The Environment of the

EasyC software

The following is the main window that you see when you open the Easy C software.



```
void main ( void )
{

}
```

```
#include "Main.h"

void main ( void )
{
}
```

1. **Menu Bar**
2. **Tool Bar**
3. **Function Blocks Window**

   In this window you have various function blocks that help you build the program.
4. **Programming Window**

   In this window you build your program.
5. **C Programming Window**

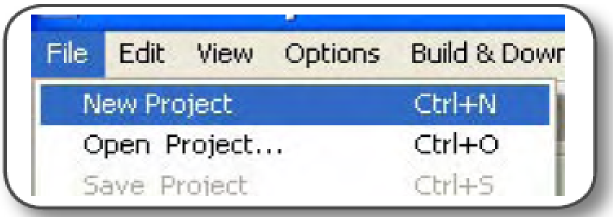
   As you build your program in the programming window, an automatic equivalent C code is generated in this window.
6. **Status & Output Window**

   When you compile your program this window shows the errors in your program.

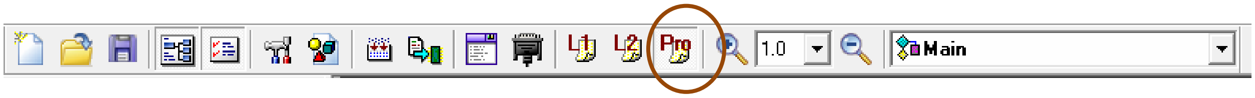
Programming Sequence

Let us write a simple program to understand the environment of the EasyC software. The following program makes the robot to move forward.

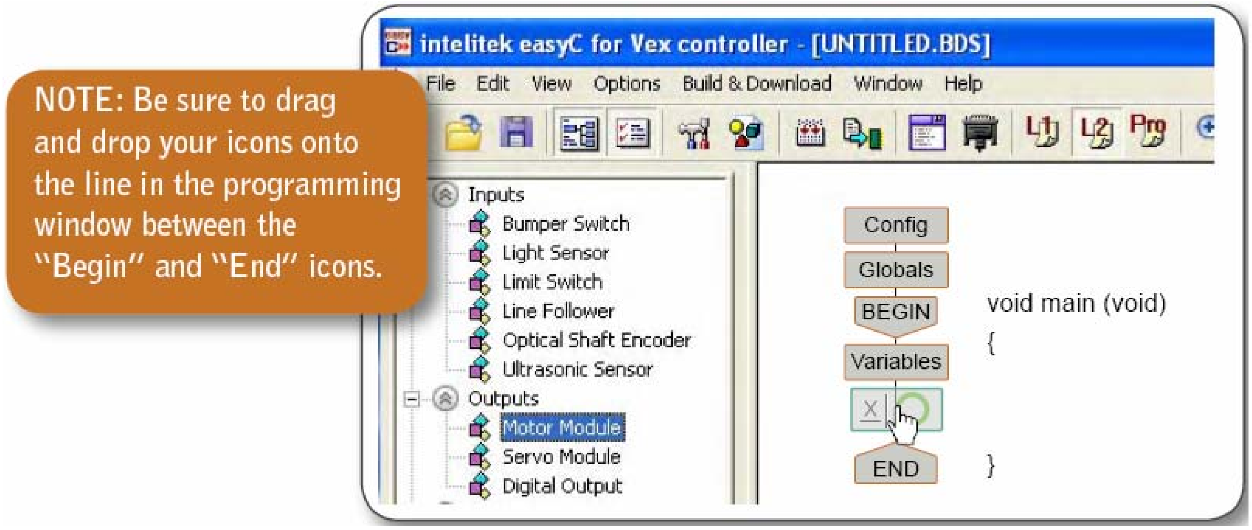
1. Launch easyC and select "New Project" from the "File" menu



2. Select "Pro" from the tool bar.

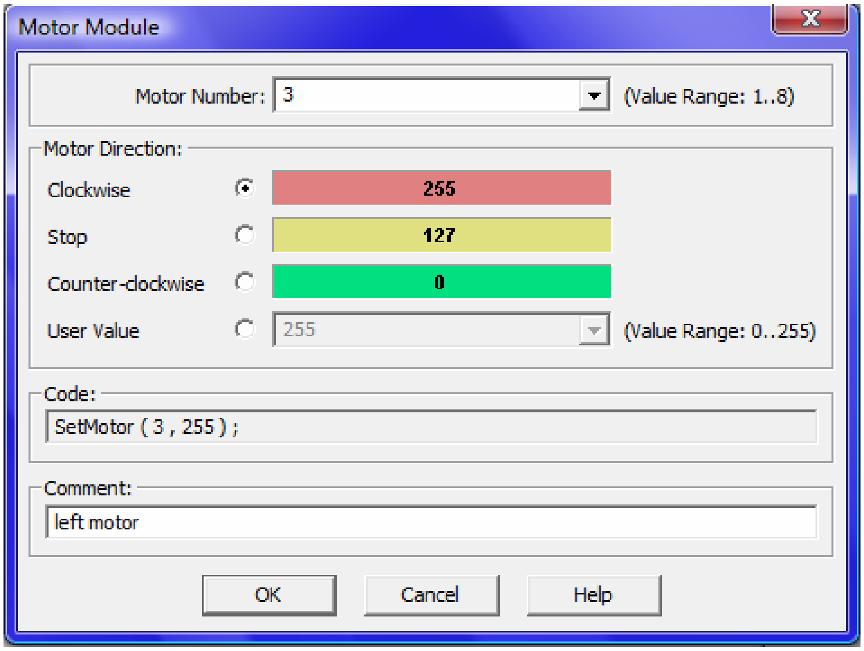


3. In the Function Blocks Window, under the "outputs" heading, find the "Motor Module" block. Left click and drag the "Motor Module" block into the programming window, between the BEGIN and END blocks. Release the block there.


NOTE: Be sure to drag and drop your icons onto the line in the programming window between the "Begin" and "End" icons.

```
void main (void)
{

}
```

4. The motor configuration window will appear as shown below.



Motor Number: 3 (Value Range: 1..8)

Motor Direction:

Clockwise	●	255
Stop	○	127
Counter-clockwise	○	0
User Value	○	255 (Value Range: 0..255)

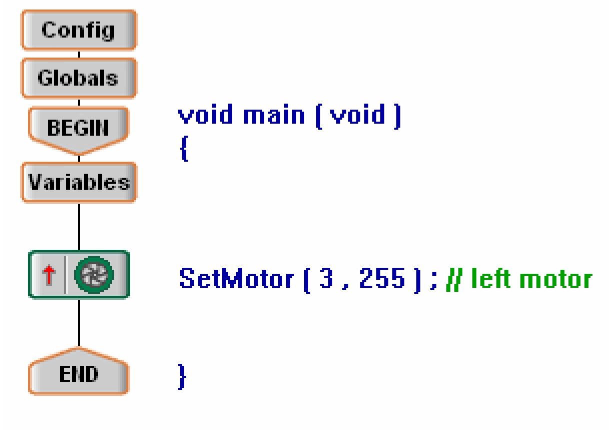
Code:

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

Comment: left motor

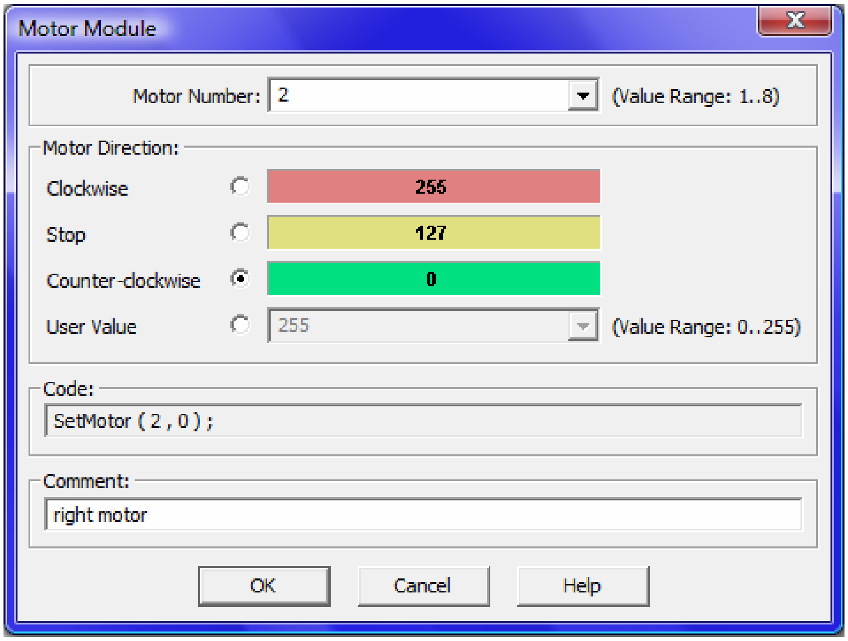
"Motor Number" is the motor port number to which the motor is connected to. Here left motor is assumed to be connected to port 3. So 3 is selected.
"Motor Direction" sets the direction of rotation of the motor. You can select the default values provided or specify a value by selecting the "user value" option and entering the value in the box. Here "clockwise" is selected.
"Code" is automatically generated as you select different options.
"Comment" is an optional field where you can enter any comment that helps you understand the code easily.
NOTE: *For detailed explanation you can click "Help". By Help you can even find information about all other function blocks.*

5. After selecting the above options click "ok".
6. Your program now looks like this.



```
void main ( void )
{

SetMotor ( 3 , 255 ) ; // left motor

}
```

7. In the similar way drag and drop one more motor module to configure the right motor. Assume right motor is connected to motor port 2 and it is set to counter-clockwise direction.



Motor Number: 2 (Value Range: 1..8)

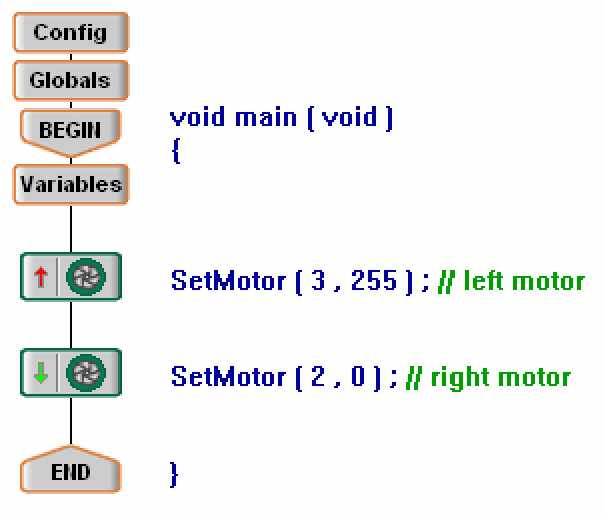
Motor Direction:

Clockwise	○	255
Stop	○	127
Counter-clockwise	●	0
User Value	○	255 (Value Range: 0..255)

Code: SetMotor (2 , 0) ;

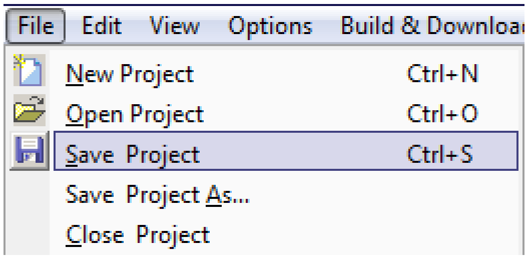
Comment: right motor

8. Your final code looks as shown below

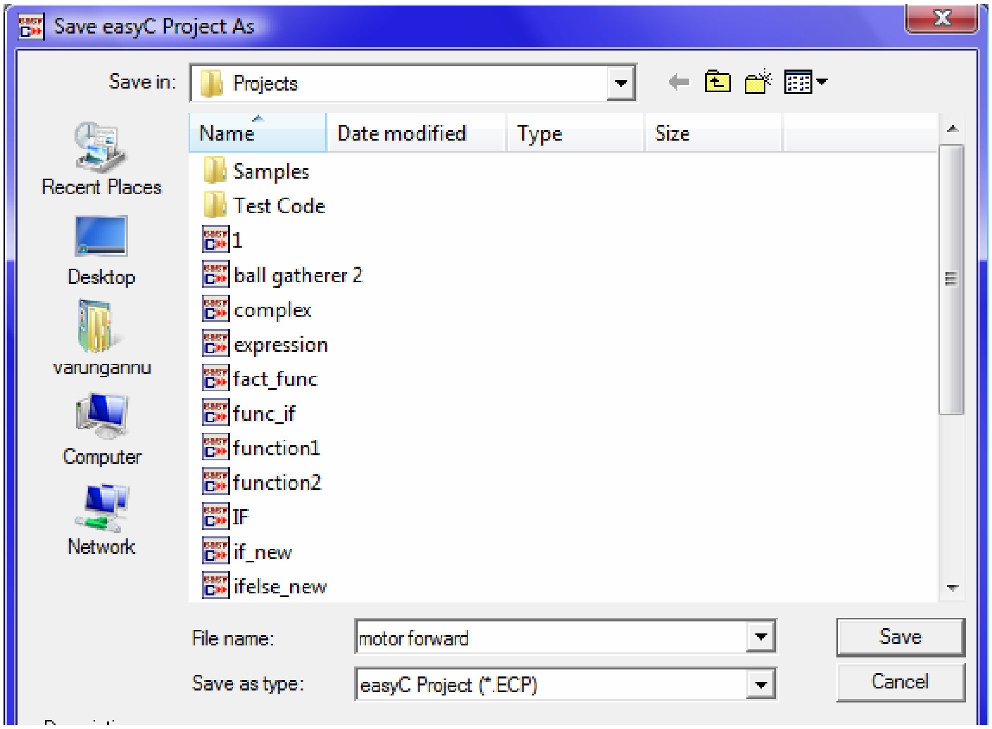


```
void main ( void )
{

SetMotor ( 3 , 255 ) ; // left motor

SetMotor ( 2 , 0 ) ; // right motor

}
```

9. Now save your project. To save your project select “Save project” from the “File” menu.



10. “Save Project As” window appears as shown below. Type a name for your project under “File name” and click “save”.



File name: motor forward

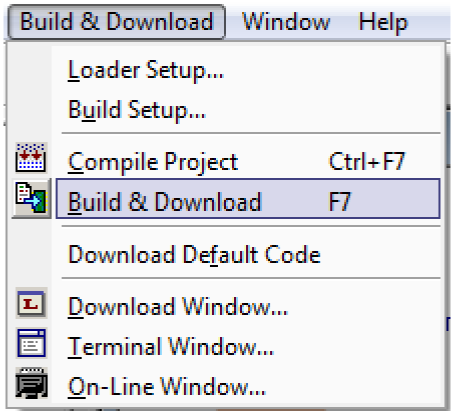
Save as type: easyC Project (*.ECP)

11. It is always a good practice to save your project frequently. This will allow you to avoid losing work in case something goes wrong.

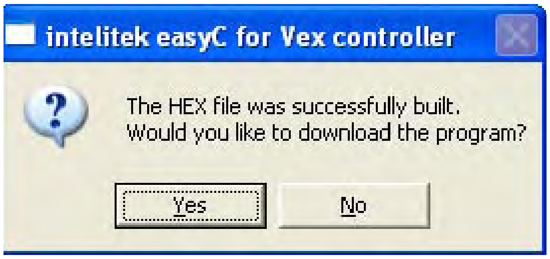
Downloading the code onto the VEX Controller

You have written the program to make the robot move forward. Not let us see how to download this code into the VEX controller.

1. Make sure your programming hardware is set up properly and the robot is on. Then select the "Build & Download" option from the "Build & Download" menu.

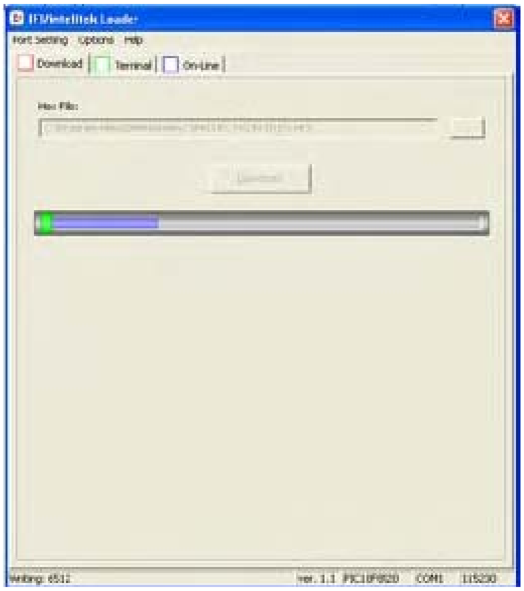


2. A confirmation screen will appear as shown below. Click "yes".



The HEX file was successfully built.
Would you like to download the program?

3. EasyC will replace old code with the new code on the robot.



Writing: 6512 ver. 1.1 PIC18F8520 COM1 115200

4. If you receive any error message during compile and download refer to the following "Troubleshooting" section. If there are no errors, your robot will start executing the code.

Troubleshooting

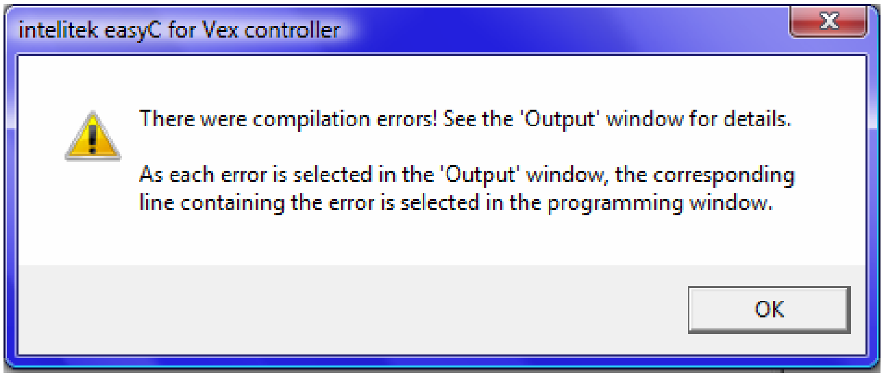
There are mainly two types of errors that you encounter.

1. Compilation errors
2. Download errors

Compilation errors:

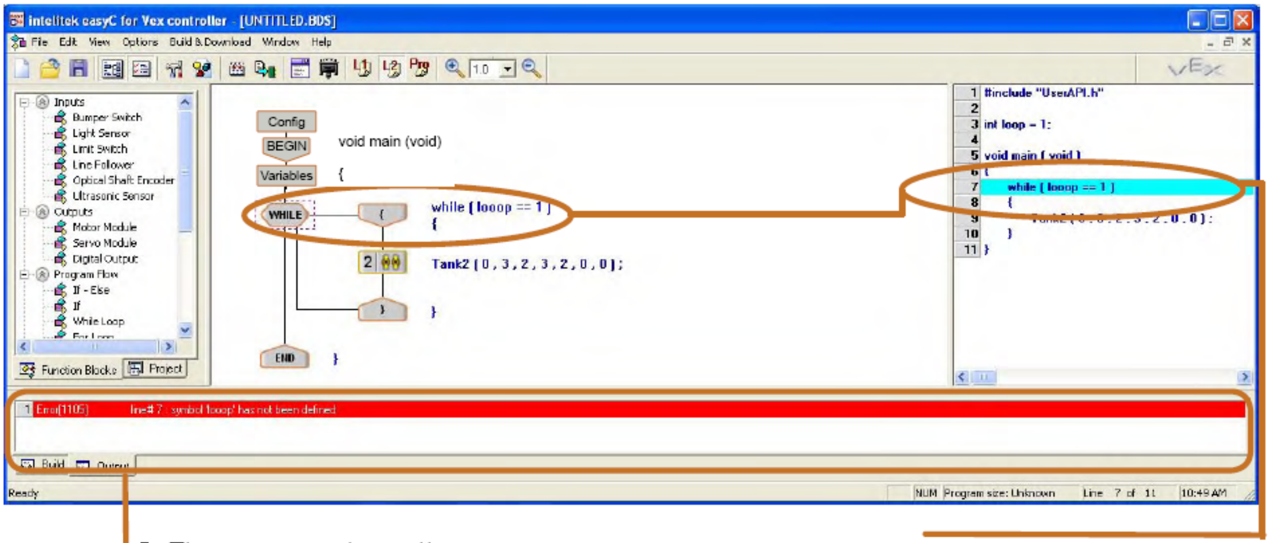
Compilation errors are errors in your code that prevent the VEX Controller from understanding the program.

If you click on the "Build and Download" button and you have compilation errors in your code, you will receive the following error message.



There were compilation errors! See the 'Output' window for details.

As each error is selected in the 'Output' window, the corresponding line containing the error is selected in the programming window.



```
void main (void)
{
    while ( looop == 1 )
    {
        Tank2 ( 0 , 3 , 2 , 3 , 2 , 0 , 0 );
    }
}
```

```
1  #include "UserAPI.h"
2
3  int loop = 1;
4
5  void main ( void )
6  {
7       while ( looop == 1 )
8       {
9            Tank2 ( 0 , 3 , 2 , 3 , 2 , 0 , 0 );
10      }
11 }
```

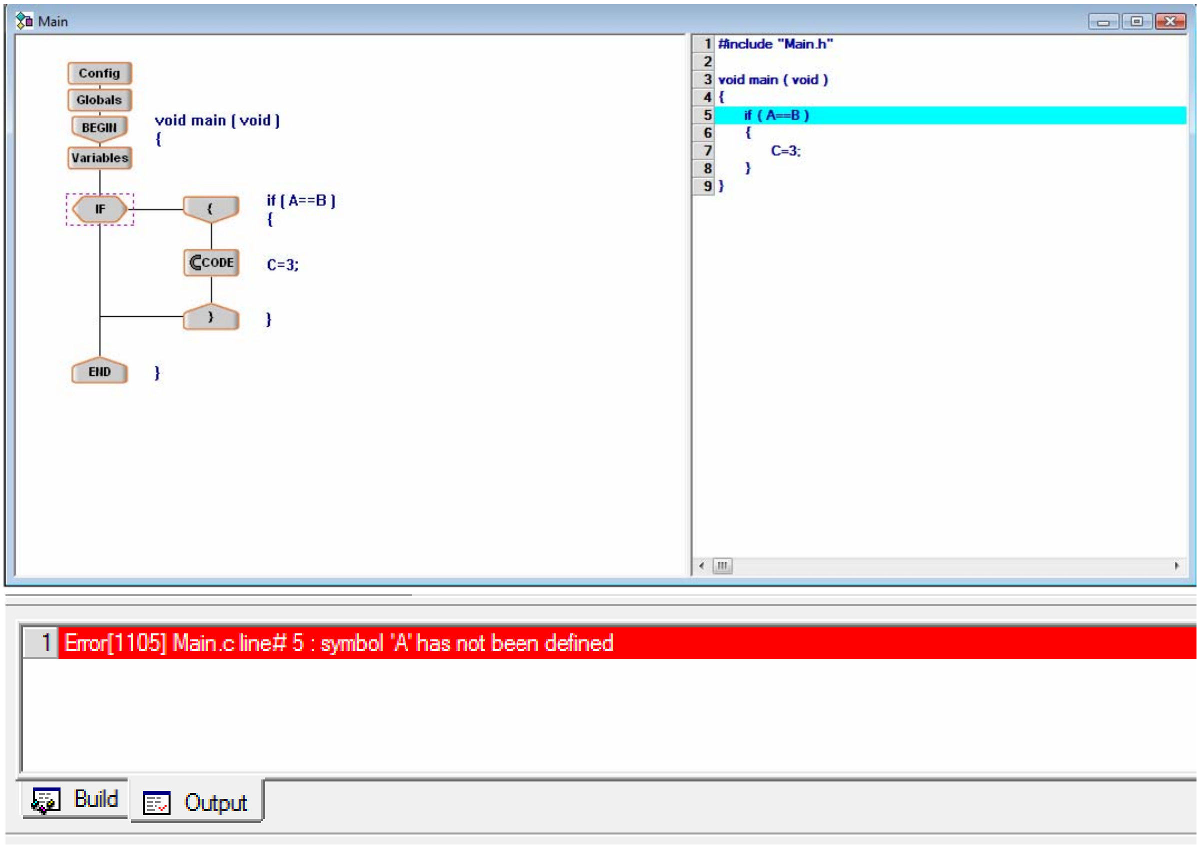
```
1 Error[1105]        line# 7 : symbol 'looop' has not been defined
```

1. The output window will display what type of error has occurred and the line number indicating where the error is located.

2. If you click on the error in the "Output Box", the line that the error is on will be highlighted in the "C Programming Window". The function block will also be selected

Compilation errors will be displayed in the output window. Some of the common compilation errors are as follows

1. **"Symbol has not been defined":** This error occurs when you misspell a variable name of if you try to use a variable that has not been defined in the "Globals window" or "variable window" at the top of the program.

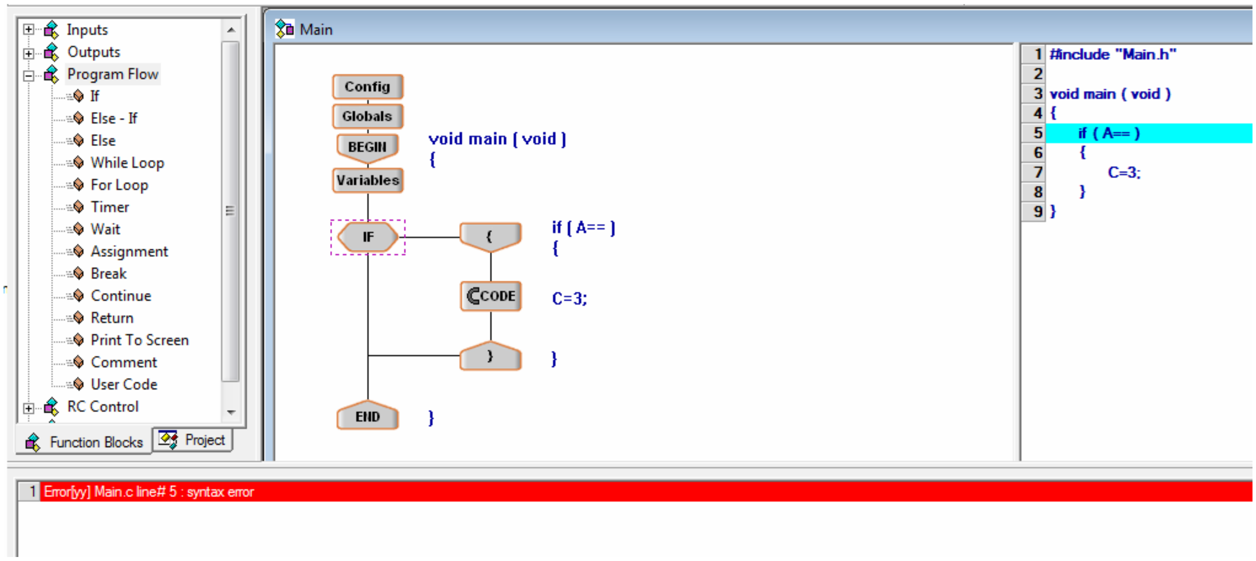


```
void main ( void )
{
    if ( A==B )
    {
        C=3;
    }
}
```

```
1 #include "Main.h"
2
3 void main ( void )
4 {
5     if ( A==B )
6     {
7         C=3;
8     }
9 }
```

```
1 Error[1105] Main.c line# 5 : symbol 'A' has not been defined
```

2. **"Syntax error":** This error occurs frequently when you do not complete the condition in an if statement or while statement. For example, in "if (loop==)", the "loop==" condition is not a complete statement.



```
void main ( void )
{
    if ( A== )
    {
        C=3;
    }
}
```

```
1 #include "Main.h"
2
3 void main ( void )
4 {
5     if ( A== )
6     {
7         C=3;
8     }
9 }
```

```
1 Error[yy] Main.c line# 5 : syntax error
```

Download Errors

Download errors occur when you are attempting to download code to your robot. These errors can occur because your hardware is not set up properly or because your software is not configured properly with your hardware.

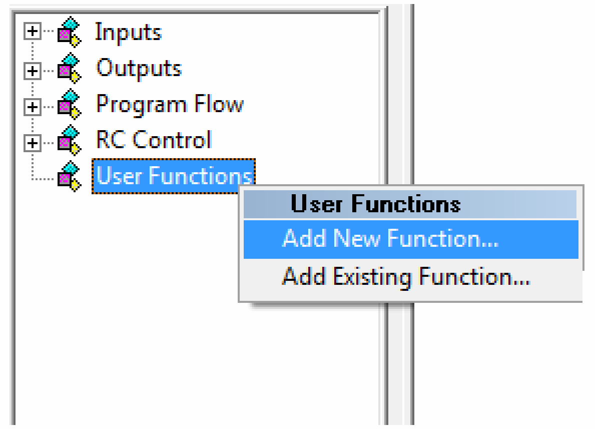
Some of the things to consider for avoiding these errors are

1. Make sure your robot is on.
2. Check your battery levels.
3. Be sure all cables are plugged in.

Functions

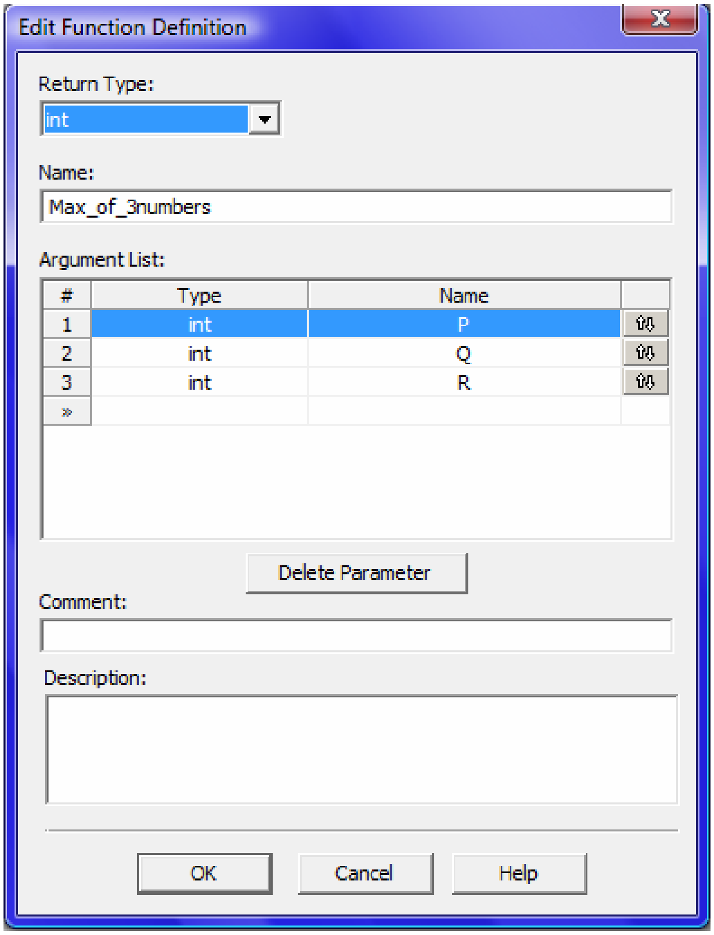
Now let us see how to define a function and use that function in another program. We write a program to find the greatest of three numbers with passing values.

1. In the function blocks window right click on "User Functions" and then select "Add New Function.."



2. A "Function definition window" will appear as shown below.

- Select Return type as "int".
- Enter a name for the function. Here it is "MAX_of_3numbers".
- Then define the variables and their type to be used in this function. Here we defined three variables P, Q, R which are of the type "int".
- Then enter any comments which are optional.
- Then click "ok"



Edit Function Definition

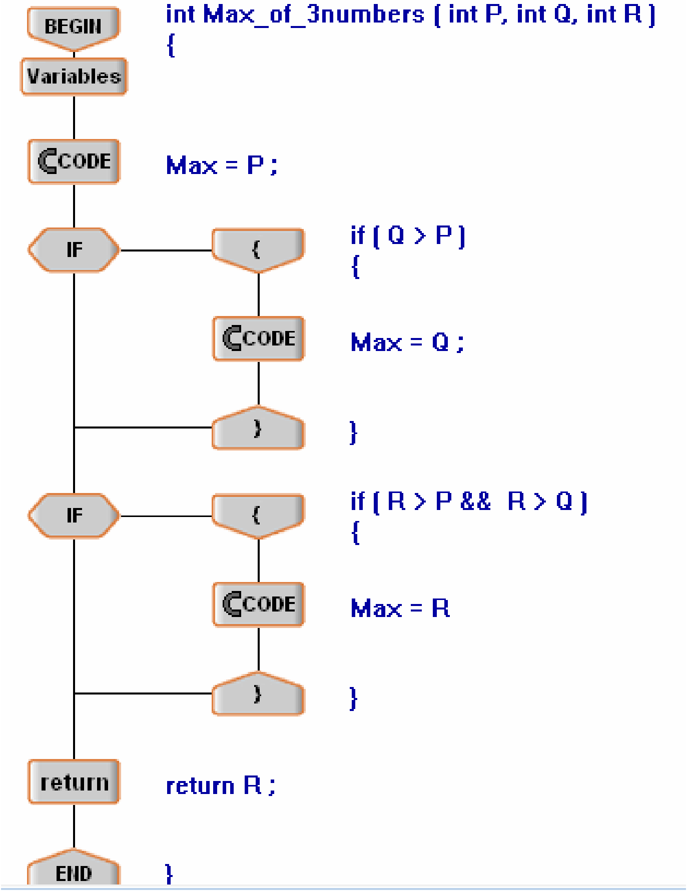
Return Type: int

Name: Max_of_3numbers

Argument List:

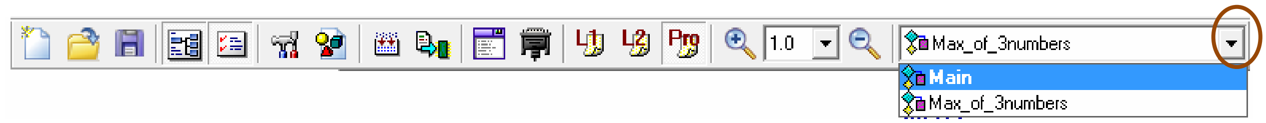
#	Type	Name
1	int	P
2	int	Q
3	int	R
»		

3. Then build the code for the function by dragging and dropping required function blocks as we discussed previously. The following shows the completely built function.

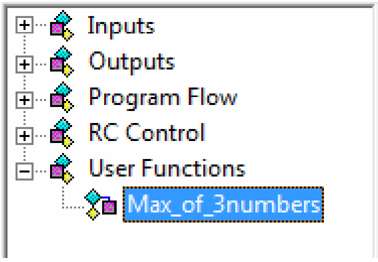


```
int Max_of_3numbers ( int P, int Q, int R )
{

Max = P ;

          if ( Q > P )
          {

          Max = Q ;

          }

          if ( R > P &&  R > Q )
          {

          Max = R

          }

return R ;

}
```

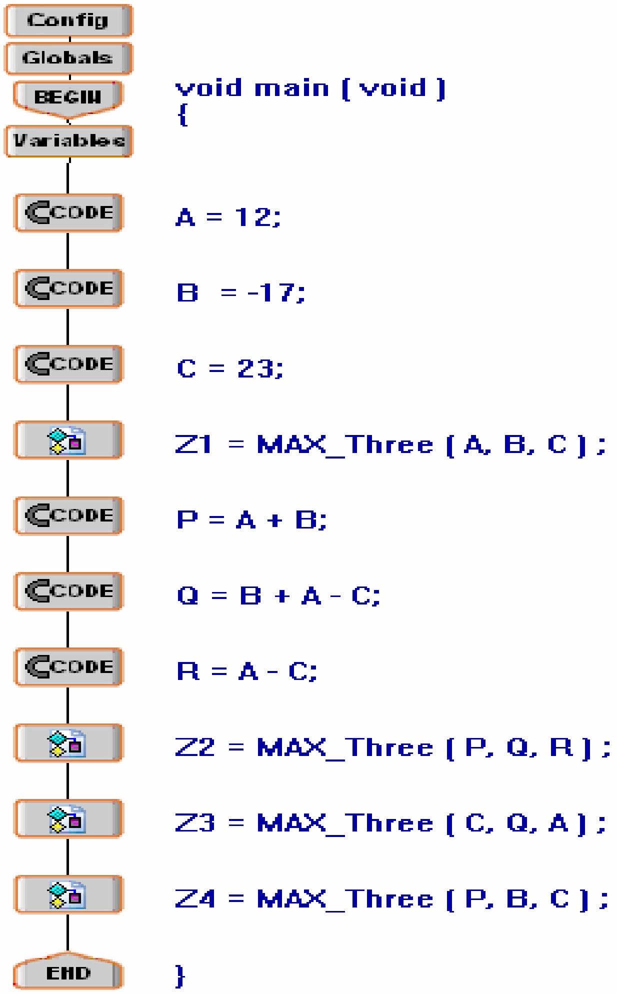
4. Save this function. In order to use this function in our program first go to the main. In order to navigate to the main function, on the tool bar click on the scroll down list (arrow mark) and then select the main.



5. You can also navigate between the main function and the defined function by the above procedure.
6. When you go to the main function, under the “User Functions” block you will see your defined function.



7. In order to use this function in main program just drag and drop “Max_of_3numbers” into your main program.
8. The final main program is shown below



```
void main ( void )
{

A = 12;

B  = -17;

C = 23;

Z1 = MAX_Three ( A, B, C ) ;

P = A + B;

Q = B + A - C;

R = A - C;

Z2 = MAX_Three ( P, Q, R ) ;

Z3 = MAX_Three ( C, Q, A ) ;

Z4 = MAX_Three ( P, B, C ) ;

}
```

9. Finally save your program and troubleshot if any errors are there.