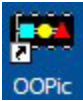


OOPic Tutorial 1

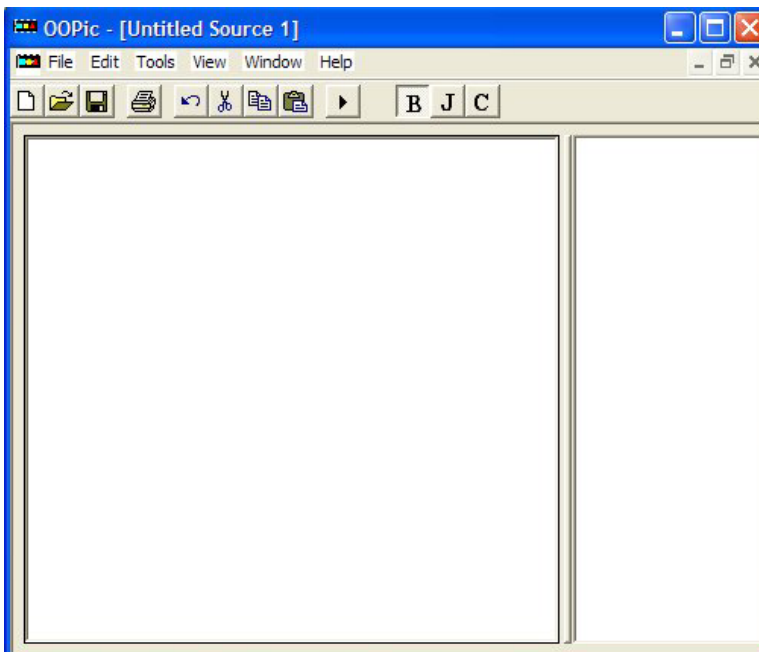
In this tutorial I would like to explain how to program your NewCDBot without using the CDBot Programmer Compiler (CPC). Instead you will learn write code using the OOPic Multi-Language Compiler, which is readily available as a free download from the OOPic website: <http://www.oopic.com/dloadp.html>. You may need to download and install the Port95nt.exe program. Now that you have the software installed we are ready to begin the tutorial.

I have chosen to illustrate all programming examples using a Basic syntax. However, you can write your code using C or Java since the OOPic supports C, Basic, or Java programming syntax. If you ever run into trouble figuring out programming syntax just visit the following OOPic web page: <http://www.oopic.com/langlist.htm> and it will provide you with a list of the OOPic's language syntax.

We are finally ready to open up the OOPic Multi-Language Compiler and start writing some code. To open up the OOPic compiler simply browse your start menu or desktop for the OOPic icon shown below.



Once the program is up and running you should get a screen similar to the one shown below.

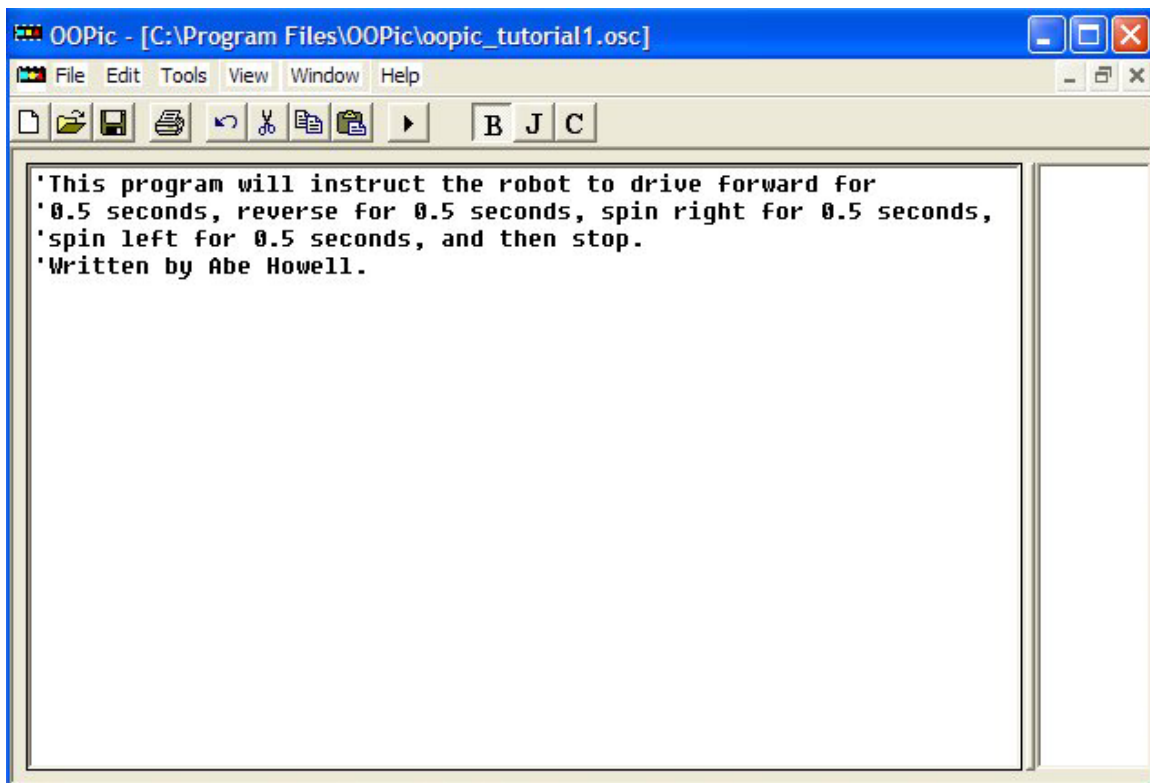


The first step in writing a new program is to decide on a name and then use the Save As menu command to save our program with the new name. For this program I have chosen to name it oopic_tutorial1, but you can name your program whatever you want.

I always like to add a comment at the top of my program so that everyone knows who wrote it and what it is supposed to do. For this program we will be writing code to drive our robot forward for 0.5 seconds, reverse for 0.5 seconds, spin right for 0.5 seconds, spin left for 0.5 seconds, and then stop. To enter a comment you need to start the line of code with an apostrophe as shown below.

```
'This program will instruct the robot to drive forward for  
'0.5 seconds, reverse for 0.5 seconds, spin right for 0.5 seconds,  
'spin left for 0.5 seconds, and then stop.  
'Written by Abe Howell.
```

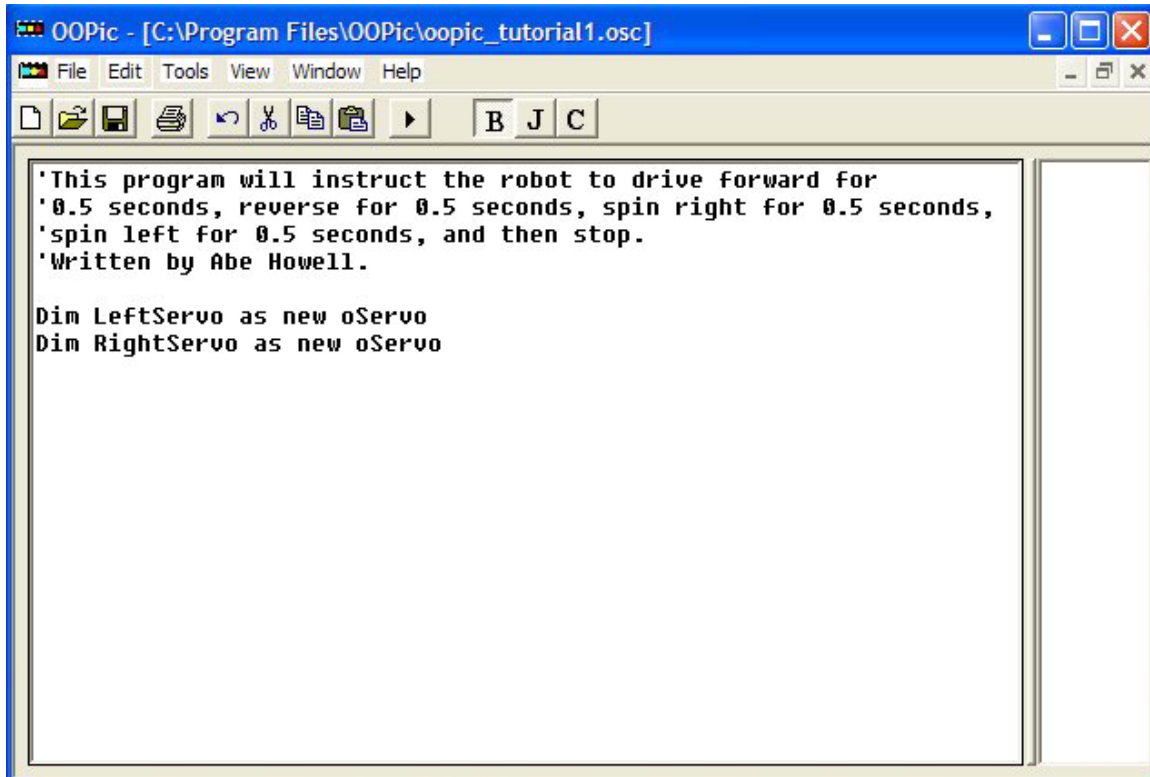
I would now like to enter the above comment into our program. Once done it should appear as shown below.



After entering a comment the next step is to declare all the objects that we intend to use in our program. For this program we will only need to declare the oServo object, which will allow us to gain control of the two modified hobby servos and thereby command the robot to move. We will need one oServo object for the left servo and one for the right servo. To declare an object you simply use the Dim command as shown below.

Dim LeftServo as new oServo
Dim RightServo as new oServo

You have to supply a unique name to the declared object. I have chosen LeftServo and RightServo as the two unique names for our oServo objects. You can now type the above declarations into your program.



The screenshot shows a window titled "OOPic - [C:\Program Files\OOPic\loopic_tutorial1.osc]". The menu bar includes File, Edit, Tools, View, Window, and Help. The toolbar contains icons for file operations and editing, along with buttons for Bold (B), Justify (J), and Center (C). The main text area contains the following code:

```
'This program will instruct the robot to drive forward for  
'0.5 seconds, reverse for 0.5 seconds, spin right for 0.5 seconds,  
'spin left for 0.5 seconds, and then stop.  
'Written by Abe Howell.  
  
Dim LeftServo as new oServo  
Dim RightServo as new oServo
```

After you have declared all your needed objects you must then configure and initialize them. I like to do this using an Initialize subroutine. Since we are programming in Basic we use subroutines. To declare the start and end of a subroutine named Initialize you simply type in the following:

```
Sub Initialize( )  
End Sub
```

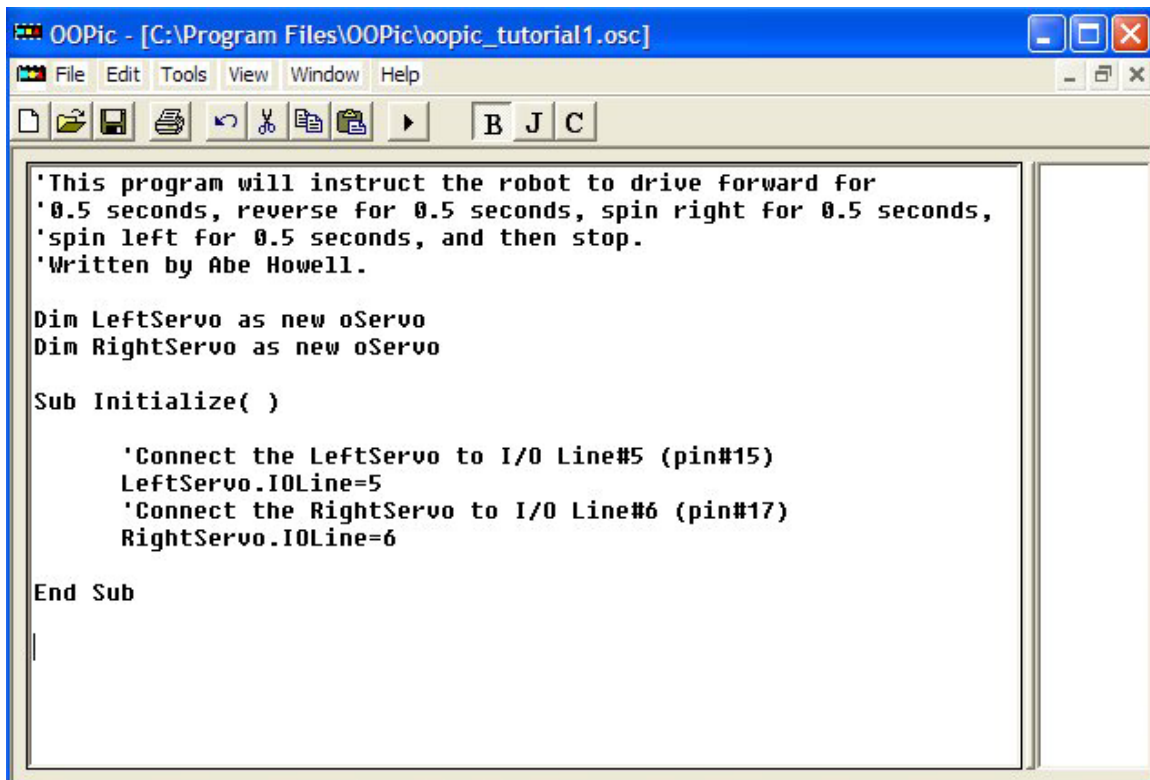
In our Initialize subroutine we need to configure the oServo objects. More specifically we need to tell the OOPic where the left and right servos are connected on its I/O connector. We need to tell it that the left servo is connected to I/O line#5 (pin#15) and the right servo is connected to I/O line#6 (pin#15). To accomplish this we will need to set the IOLine property of each oServo object. The appropriate code for our Initialize subroutine is shown below. I also like to add comments to explain what each line of code is doing.

Sub Initialize()

```
'Connect the LeftServo to I/O Line#5 (pin#15)
LeftServo.IOLine=5
'Connect the RightServo to I/O Line#6 (pin#17)
RightServo.IOLine=6
```

End Sub

You can go ahead and type the above code into your program so that it appears as shown below.



The screenshot shows a window titled "OOPic - [C:\Program Files\OOPic\loopic_tutorial1.osc]". The window has a menu bar with "File", "Edit", "Tools", "View", "Window", and "Help". Below the menu bar is a toolbar with icons for file operations and editing. The main text area contains the following code:

```
'This program will instruct the robot to drive forward for
'0.5 seconds, reverse for 0.5 seconds, spin right for 0.5 seconds,
'spin left for 0.5 seconds, and then stop.
'Written by Abe Howell.

Dim LeftServo as new oServo
Dim RightServo as new oServo

Sub Initialize( )

    'Connect the LeftServo to I/O Line#5 (pin#15)
    LeftServo.IOLine=5
    'Connect the RightServo to I/O Line#6 (pin#17)
    RightServo.IOLine=6

End Sub
```

Every OOPic program must contain a Main subroutine, this is the subroutine that is executed when you turn on your OOPic. We will add the code to make our robot move in the Main subroutine. Before we do this it is necessary for us to fully understand how to control a modified hobby servo. A modified hobby servo is just a hobby servo that has been modified to rotate continuously. An unmodified servo is only capable of approximately 180 degrees of rotation, which would not be too useful if used as a drive motor for a robot. After a servo has been modified it will have what is called a stop value. This stop value is the value required to make the servo stop rotating. For our modified servos this value should be 53. This means that if we send this stop value to our two modified servos they both should stop. We can also use this stop value to make our robot's servos spin in whatever direction we want. To do this we simply add or subtract from the stop value and then send this new value to our servo. To send a value to our

servos we simply set the appropriate oServo object equal to the value we want to send. Suppose we want to command our left servo to stop. We simply enter the following into our program in the Main subroutine.

```
LeftServo=53  
LeftServo.Operate=cvTrue
```

The first line commands the left servo to the stop value and the second line is needed to actually turn on the left servo. Suppose we wanted to command both the left and right servos in such a way that they would make our robot go forward. We would need the left servo to rotate clockwise and the right servo to rotate counter clockwise. To accomplish this we need to subtract from the stop value for the left servo and add to the stop value for the right servo. I will illustrate this in the code below.

```
LeftServo=43  
RightServo=63  
LeftServo.Operate=cvTrue  
RightServo.Operate=cvTrue
```

Now suppose we want the robot to go in reverse, we need to rotate the left servo counter clockwise and the right servo clockwise. The code shown below will accomplish this.

```
LeftServo=63  
RightServo=43  
LeftServo.Operate=cvTrue  
RightServo.Operate=cvTrue
```

To make the robot spin right we would need to rotate the left servo counter clockwise and the right servo counter clockwise as shown in the code below.

```
LeftServo=63  
RightServo=63  
LeftServo.Operate=cvTrue  
RightServo.Operate=cvTrue
```

To make the robot spin left we would need to rotate the left servo clockwise and the right servo clockwise as shown in the code below.

```
LeftServo=43  
RightServo=43  
LeftServo.Operate=cvTrue  
RightServo.Operate=cvTrue
```

We could simply use all the above snippets of code to make our robot drive forward for 0.5 seconds, reverse for 0.5 seconds, spin right for 0.5 seconds, spin left for 0.5 seconds, and then stop, but I would like to create several subroutines for handling these motion

commands. The final program is listed below. You can see where I added the motion subroutines and then called them from within the Main subroutine. The code will only execute once as listed below. If you want it to repeat over and over again simply add a Do Loop around the calls to the motion subroutines in the Main Sub.

'This program will instruct the robot to drive forward for
'0.5 seconds, reverse for 0.5 seconds, spin right for 0.5 seconds,
'spin left for 0.5 seconds, and then stop.
'Written by Abe Howell.

Dim LeftServo as new oServo
Dim RightServo as new oServo

Sub Initialize()

 'Connect the LeftServo to I/O Line#5 (pin#15)
 LeftServo.IOLine=5
 'Connect the RightServo to I/O Line#6 (pin#17)
 RightServo.IOLine=6

End Sub

Sub Main()

 'Call the Initialize subroutine
 Initialize

 'Call FWD subroutine
 FWD
 'Wait 0.5 seconds
 OOPic.Delay=50

 'Call BACK subroutine
 BACK
 'Wait 0.5 seconds
 OOPic.Delay=50

 'Call SPINRIGHT subroutine
 SPINRIGHT
 'Wait 0.5 seconds
 OOPic.Delay=50

 'Call SPINLEFT subroutine
 SPINLEFT
 'Wait 0.5 seconds
 OOPic.Delay=50

 'Call STOP subroutine
 STOP
 'Wait 0.5 seconds
 OOPic.Delay=50

End Sub

```
Sub FWD( )  
    LeftServo=43  
    RightServo=63  
    LeftServo.Operate=cvTrue  
    RightServo.Operate=cvTrue
```

```
End Sub
```

```
Sub BACK( )  
    LeftServo=63  
    RightServo=43  
    LeftServo.Operate=cvTrue  
    RightServo.Operate=cvTrue
```

```
End Sub
```

```
Sub SpinLeft( )  
    LeftServo=43  
    RightServo=43  
    LeftServo.Operate=cvTrue  
    RightServo.Operate=cvTrue
```

```
End Sub
```

```
Sub SpinRight( )  
    LeftServo=63  
    RightServo=63  
    LeftServo.Operate=cvTrue  
    RightServo.Operate=cvTrue
```

```
End Sub
```

```
Sub STOP( )  
    LeftServo.Operate=cvFalse  
    RightServo.Operate=cvFalse
```

```
End Sub
```
