

Conditionals: More Fun with "If-Then" and Logic Operators







- Great job so far! There are a lot of things you already know how to do! In this tutorial and in the following ones you will get a chance to start using what you have learned. Some familiar steps will have less detailed instructions. Look at hints or at previous tutorials if you need help, or ask a friend!
- In this tutorial you will:
 - Review: How *your* program controls the SPHERES satellites
 - Practice programming with "If-Then" statements
 - Use the logic operators "==" and "!="
 - Learn about the "debug" feature







EDC Learning transforms lives,



DEBUG (



("")

2





The program inside your loop is called **once Text editor** per second by the SPHERES control system. void loop() { During each second: } tells the satellite Your Code where to go state targets uses your targets to SPHERES SPHERES Controller calculate thruster Estimator commands firing times determines where the satellite is fires thrusters at SPHERES specific times Satellite EDC Learning transforms lives, DARP/ ΛΛΡ [TOPCODER] **Aurora** 3 CASIS





- Let's create a new project to learn more about the SPHERES control system
- Open the ZR IDE
- Select "New Project"
 - Project name: Project5
 - Game: FreeMode
 - Text Editor
- Create a variable called **counter**
 - Above void init() type int counter;
 - Inside void init() type counter = 0;



[TOPCODER]

Learning transforms





DARP

CASIS



- "==" is a Logic operator that means "equals"
- Create the following "if-then" statement in your loop: If counter is equal to 10 then . . . (See example below)

```
12 void loop(){
13 //This function is called once per second. Use it to control the satellite.
14 if (counter == 10){
15
16 }
17 }
```

 Add counter++; after the "if-then" statement to increment the counter each second (see example below)

> Learning transforms

```
void loop(){
//This function is called once per second. Use it to control the satellite.
if (counter == 10){
}
counter++;
}
```

TOPCODER

Aurora





- Next add a debug statement, which prints out messages.
 - To do that we type **DEBUG((""));**
 - Then type counter is 10 into the "DEBUG" statement (within the quotation marks)
- You have created the following "if-then" statement:

• The debug message can be any text you like









- Since you have not set a position target in your program, the satellite will not move. Instead, watch for your a debug message as follows:
 - Compile, Simulate
 - Load settings: Tutorial _45
 - Before you run your simulation
 - Click the "show console" icon on the bottom of your screen
 - A grey box will appear at the bottom of your screen
 - Watch for your debug message in this box when you run the simulation
 - Your debug message for blue
 SPHERES 1 (SPH1) will appear
 after 10 seconds (see example)
 - Run simulation







EDC Learning transforms lives.

[TOPCODER]

Aurora







• You can use the debug statement to print variables as well. Use the following symbols for each data type.

Data Type	Symbol
int	%d
float	%f
string	%s

- Use this syntax: DEBUG(("text text text symbol", variable));
- For example: **DEBUG(("blahblah %d",counter))**;
- DEBUG uses the same syntax as the printf function in normal C. For more symbols to format data using DEBUG, you can refer to the following page (not created or supported by Zero Robotics):

http://www.codingunit.com/printf-format-specifiers-format-conversions-andformatted-output







- "!=" is a Logic operator that means "not equal"
- What happens if you add the following if-then statement to your loop?

If counter is not equal to 10 then....

- To do this, first:
 - We add another "if" statement right under our first one.
 - This conditional is written
 counter != 10

12	<pre>void loop() {</pre>
13	<pre>//This function is called or</pre>
14	if (counter == 10) {
15	DEBUG(("counter == 10));
16	
17	if (counter != 10) {
18	}
19	countertt:
20	
21	1







- Next we'll add another DEBUG ((" ")); statement under the new "if" statement
- This time write the message:
 DEBUG(("counter equals %d",counter));
 (don't forget the quotation marks and comma)
- Look at the program you created. What do you expect to happen when you run the simulation? Let's find out.

- •Compile, Simulate
 - Load settings: Tutorial _45
- •Before you run your simulation
 - Click the "view console" icon on the bottom of your screen

```
void loop(){
    if(counter == 10) {
        DEBUG(("counter is 10"));
    }
    if(counter != 10) {
        DEBUG(("counter equals %d",counter));
    }
    counter++;
}
```





ADD



- Remember that your program is being read every second.
- A "counter equals (number)" message was printed for each second that the counter did not equal 10 (starting from counter == 0)
- A "counter is 10" message was printed for each second that the counter was equal to 10
 - Can you find that message?
- DEBUG statements can help you check whether your program is running the way you expect it to run.

CASIS







- Congratulations!
- You are becoming a pro at conditional statements!
- You learned two more logic operators == and !=
- You wrote a program that shows your loop runs once per second
- You learned how to use DEBUG statements!

```
DEBUG(("counter == 10"));
```

DEBUG(("counter equals %d",counter));

