

# Robotics & Automation

## ENGT4314

### Tutorial 1

---

#### Data types:

Fundamental data types

Type	prefix	Range (Signed)
Boolean	bool	true or false
Integer	int	-32768 to +32767
Character	char	-128 to 127
Floating point	float	3.4 e-38 to 3.4 e+38
Double precision floating point	double	1.7 e-308 to 1.7 e+308
Void	void	

#### Declaring a Variable:

[signed / unsigned] <prefix> <Variable> [= <initial value>];

Eg:

```
int count = 0;           // Declare and initialize an integer.
unsigned char newCharacter; // Declare a character variable.
```

[..] optional.

#### Common Operators

+, -	Addition, Subtraction
*, /, %	Multiplication, Division, Modulus
++, --, =, +=, -=	Increment, Decrement, Assignment, Assignment by Addition/Subtraction
==, !=, ===	Equals, not equal, equality (Tests data types as well)
>, <, >=, <=	Greater than, less than, greater than or equal too, etc.

## If Statements

```
If (condition1) {  
    Statements if condition1 is true  
}else if (condition) {  
    If condition2 is true  
}else{  
    If neither are true  
}
```

Example:

```
If (var1 == var2) {  
    printf("Var1 = var2.");  
}else if (var1 < var2) {  
    printf("var1 is less than var2");  
}else{  
    printf("var1 is greater than var2");  
}
```

## Switch Statement:

```
int var;  
Switch(var) {  
  
    case 1:  
        if var = 1  
        break;  
    case 2:  
        if var = 2  
        break;  
    default:  
        Otherwise:  
  
}
```

## Loops:

### For:

```
for (initialise variable; condition; operation) {  
    statements;  
}
```

The for loop will run the statement in *initialize* and then test the *condition*, after each iteration in the loop the *operation* is executed.

Example:

```
for(i = 0; i<5; i++) {  
    printf("%d, ",i);  
}
```

Output: 0, 1, 2, 3, 4

### Do While and While loops:

A do while loop will execute the *statement* and then continue looping while *condition* is true.

```
do {  
    statements  
} while (condition);
```

A while loop will execute statements while *condition* is true

```
while (condition) {  
    statements  
}
```

## Functions:

```
<return data type> <function name>([<argument1 datatype> <argument1>, <arg2 datatype> <arg2>, etc]) {  
    statements  
}
```

Example:

```
int add(int number1, int number2) {  
    return number1 + number2;  
}  
  
printf(" 3 + 5 = %d \n", add(3,5));
```

Result:            3 + 5 = 8

## Pointers:

Pointers store the memory address of the resource, as oppose to the resource.

Define a pointer with a '\*' like this:

```
int *k;
```

Now k can hold a memory address of an integer. You can use the & operator to get the address of an variable E.g.

```
int j = 512;
```

```
k = &j;
```

Now k contains the address in memory of where j is stored. We can use \* to deference a pointer and apply operations to the data that it is pointed to E.G..

```
*k += 2;
```

```
j =?
```

We can also pass pointers to functions instead of the data itself, reducing the amount of data that is placed on the stack.





## EyeSim:

### Installing:

There are two components required:

minGW-3.1.0-1.exe

sim65win.exe

Both can be downloaded from <http://robotics.ee.uwa.edu.au/eyebot/ftp/>. Once you have downloaded both files install them using the default settings.

Note (For vista users): There is a patched batch file available that will allow minGW to run correctly on vista, if this applies to you please see the readme file in <http://robotics.ee.uwa.edu.au/eyebot/ftp/vista/>.

### Compiling to run in EyeSim:

Open the EyeSim prompt from the start menu (under EyeSim SDK).  
Navigate to the desired location of your source file.

Enter the following statement (replacing yourfile with the name of your source file).  
`gccsim yourfile.c -o yourfile.dll`

this will then compile your code into a dll to run in Eyesim.

## Creating a Sim file:

Eyesim executes sim files, thus to define your simulation environment you need to create a sim file.

A sim file typically contains the following.

```
define sim c:\eyesim\eyesim  
  
# EyeSim 6.0 Simulation Description File  
  
# world description file (either one maze one world)  
world %sim%/worlds/worlds/Soccer1998.wld  
  
# robi description file, link-library, [start pos/orient.], [graphics file]  
robi %sim%/robots/SocBot/S4.robi triangle.dll 1500 800 0
```

### Breakdown:

*define sim c:\eyesim\eyesim* - Points to the EyeSim executable if you installed EyeSim to the default directory this should be correct otherwise adjust accordingly.

*# world description file (either one maze one world)*

*world %sim%/worlds/worlds/Soccer1998.wld* – The simulation world, eg. the environment in which the robot is placed. There a number of existing worlds and maze file in the EyeSim SDK, or you could make you own (See the EyeSim help file).

*robi %sim%/robots/SocBot/S4.robi triangle.dll 1500 800 0* – This defines which robot (appearance and hardware) should be used in this case S4.robi the complied program (triangle.dll) and the x, y and  $\phi$  of the robot.

Once the sim file has been created simply double click on the sim file to open the EyeSim simulator.

## Notes:

You should include eyebot.h in you c files for EyeSim and use the Robios operations. The main function should be *int main ()*.

## EyeSim Example:

### triangle.c

```
/******  
* Triangle - Eyesim example *  
* Daniel Kingdom 5-8-08 *  
*****/  
#include "eyebot.h"  
void triangle(int height) {  
    int level;  
    int i;  
    LCDClear();  
    if (height <= 0) {  
        LCDPrintf("Invalid height.\n");  
    }else{  
        for (level = height; level > 0; level--) {  
            for(i=0; i < height - level; i++){  
                LCDPrintf("_");  
            }  
            for(i=0; i < 2*level; i++){  
                LCDPrintf("*");  
            }  
            for(i=0; i < height - level; i++){  
                LCDPrintf("_");  
            }  
            LCDPrintf("\n");  
        }  
    }  
}  
  
int main ()  
{  
    int Key = 0;  
    int level = 4;  
    triangle(level);  
    while(KEYGet() != KEY4){  
        Key = KEYGet();  
        if (Key == KEY1)  
            triangle(++level);  
        else if (Key == KEY2)  
            triangle(--level);  
    }  
}
```

## Triangle.sim

```
define sim c:\eyesim\eyesim
```

```
# EyeSim 6.0 Simulation Description File
```

```
# world description file (either one maze one world)
```

```
world %sim%/worlds/worlds/Soccer1998.wld
```

```
# robi description file, link-library, [start pos/orient.], [graphics file]
```

```
robi %sim%/robots/SocBot/S4.robi triangle.dll 1500 800 0
```