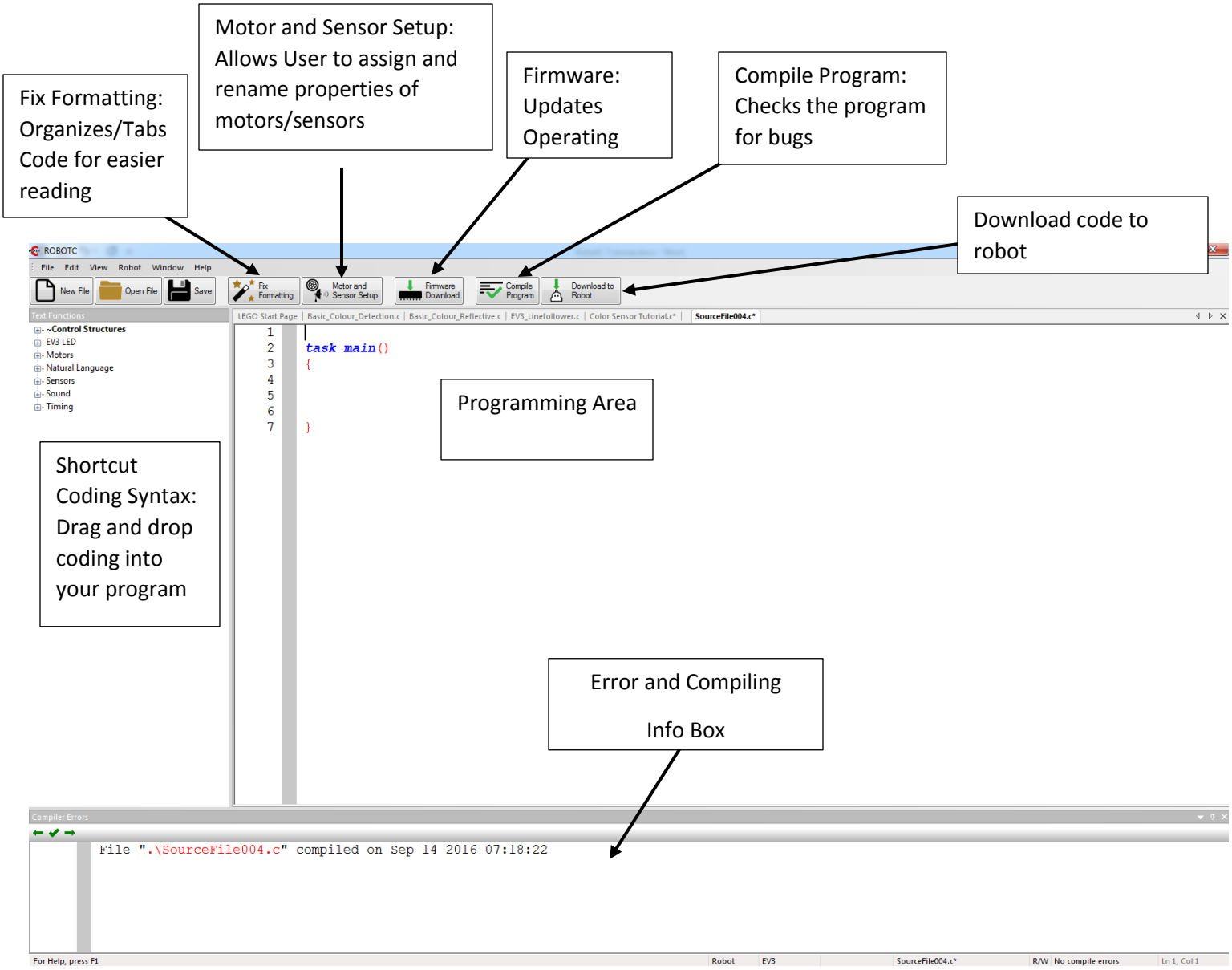
Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Robot C Basic Screen Outline

Font Height Hotkey = CTRL and +/- to increase or decrease font height



Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Firmware

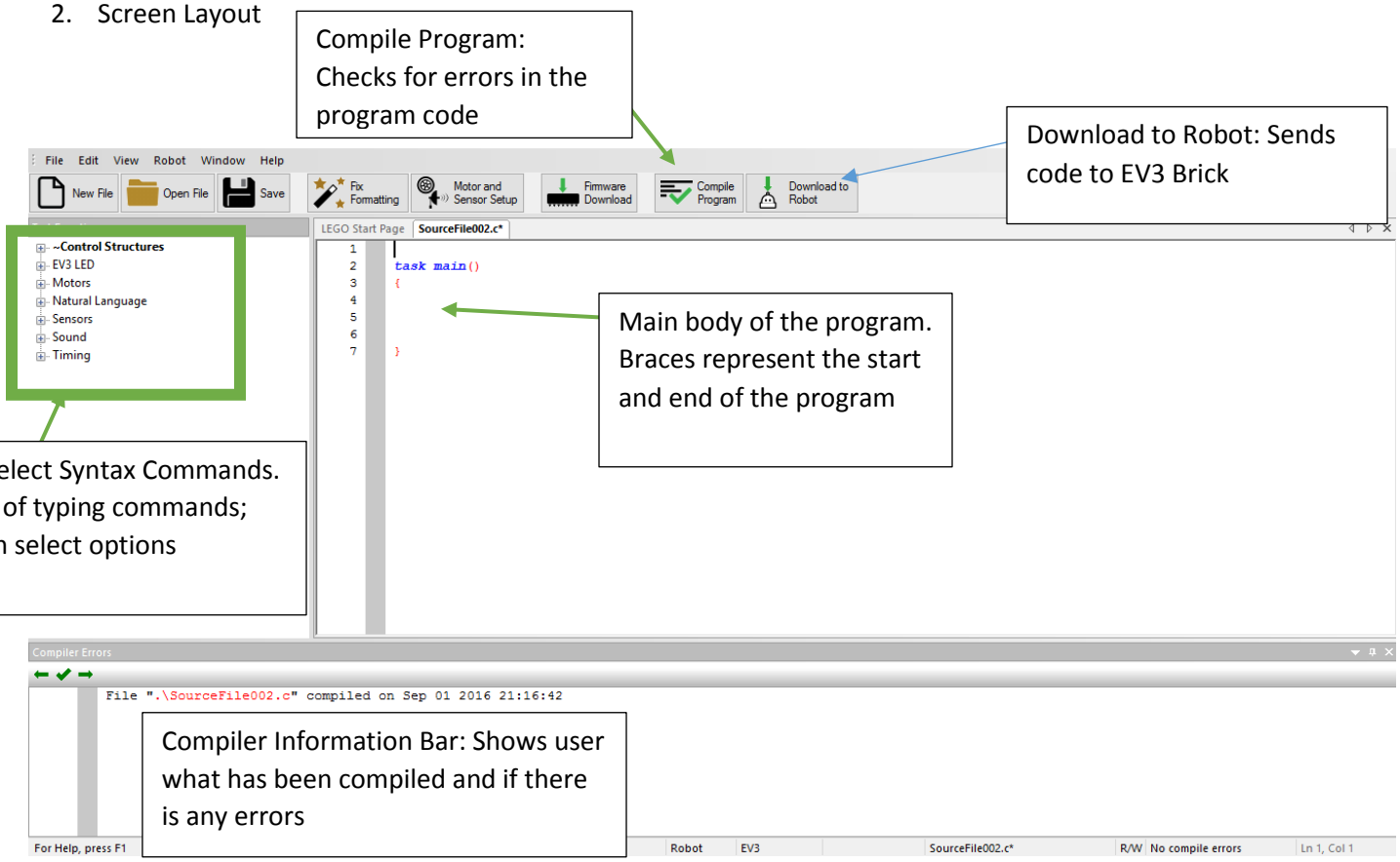
Definition: programming language that is placed as read-only memory that is able to run certain program types(I.E Firmware OS = Apple products; Firmware Android = Non-Apple Products)

1. Turn the EV3 Brick ON
2. Connect EV3 Lego Brick to the computer using the USB – Micro USB wire.
3. Adjust the firmware from function block (Labview) to structured text (Robot C)
    a. Open Robot C for Lego Mindstorm 4.x
    b. Drop Down Menu Robot > Platform Type > Lego Mindstorms > EV3
    c. Drop Down Menu Robot > Download EV3 Linux Kernel

Tutorial 1: Make It Move

A. Make It Move

1. File > New > New File
2. Screen Layout

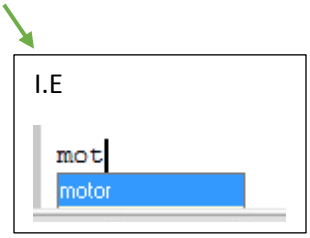


Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

3. Write the following Program
    a. May type all of the code or begin typing and drop down menus will appear to predict desired command

```
1
2   task main()
3   {
4
5   motor [motorB] = 50; //set motor speed for B Port
6   motor [motorC] = 50; //set motor speed for C Port
7   wait1Msec(3000); // set total time motors will run (1000ms = 1sec)
8
9   }
```



4. Compile the program to make sure there are no errors or bugs in the program
5. Connect the EV3 Brick to the PC using the USB cable > Select the **Download to Robot** > First time button is hit the software will ask you to save the source file and main file of the program. Navigate to the desired file location > Pop menu will appear > Click the Start button to start the program

Robot C Program 1: Make it Move

Write a Program that does the following. Move now faster than power setting of 25

1. Mark a start point with tape
2. Go forward as provided in the original program for 9 inches
3. Turn Right 90 degrees (Mark spot on the floor w/ tape)
4. Pause 2.5 seconds
5. Go forward 1.5 feet
6. Turn Left 45 Degrees (Mark spot on the floor w/ tape)
7. Pause for PI seconds (3.14 seconds)
8. Go Forward 2 feet
9. Pause 2.68 seconds
10. Move the robot in reverse back to the start point (Do not need to pause at the turns). Robot should come close to the original marked position on the floor.

Show program to Teacher.

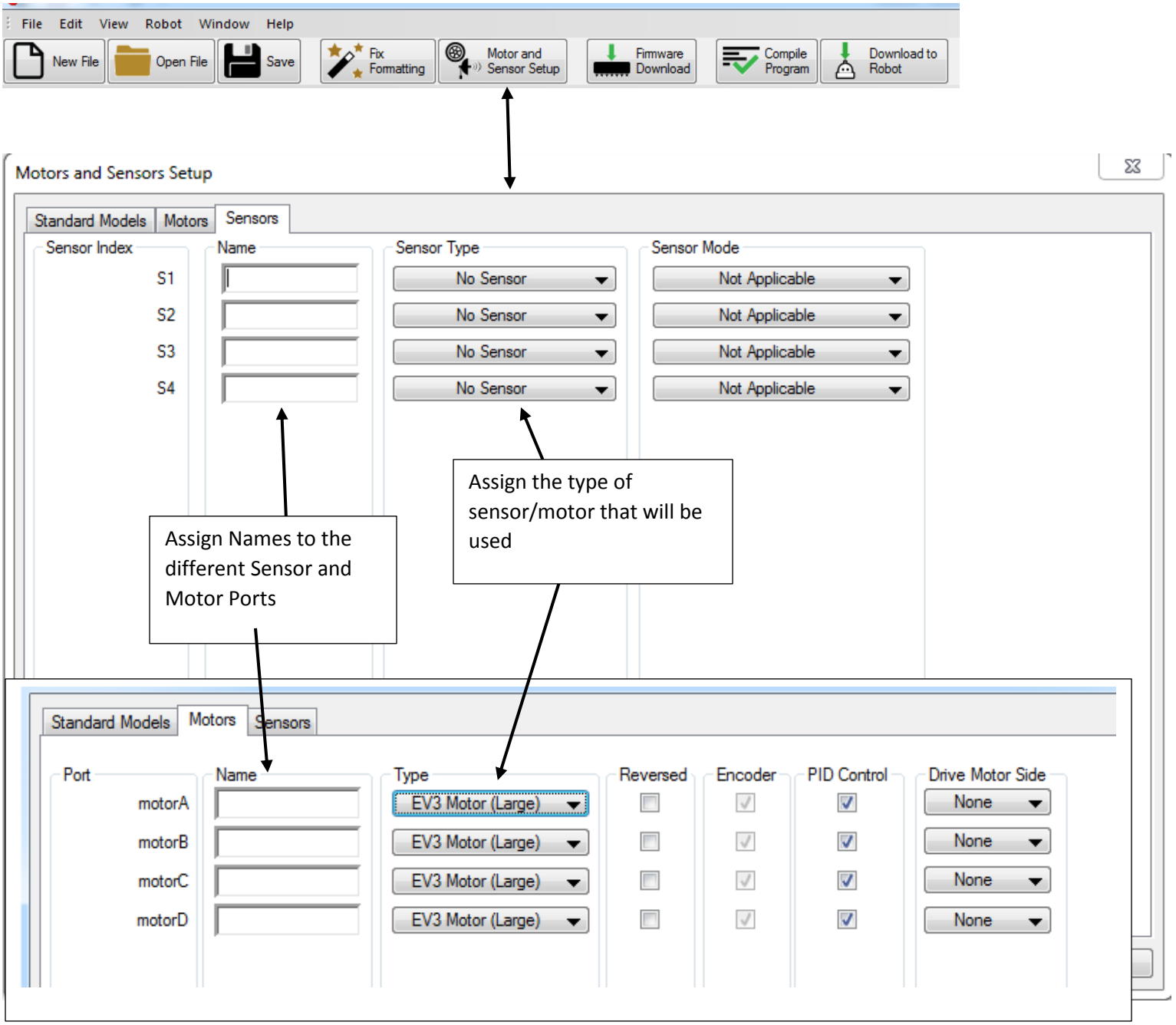
Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Sensors

When using sensors or motors it is helpful to rename them to variable name that stands out and is easy to type.
For Example: Default name for a Touch Sensor maybe S1. Changing the variable name to Touch instead of S1 can be helpful when trying to debug the program.

Click on the **Motor Sensor Setup Icon** at the top part of the screen



Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Coding: When coding with a sensor the syntax **SensorValue [SensorName]** will be used.

Example of If Statement for Touch Sensor is touched to turn off motors:

```
If (SensorValue [Touch] == 1)
{
motor [motorB] = 0;
motor [motorC] = 0;
}
```

Sensors Values

Port View on the EV3 Brick can be used to see what values are be returned for the sensor or the following code can be typed in the program to display data on the brick

Example is for Color Sensor (Code is in Bold; all other type are comments about the code)

```
// Write the amount of reflected light to the screen
// This is a value between 0 and 100, where 0 means no reflected
 // light and 100 means all light is being reflected
displayBigTextLine(4, "Reflected: %d", SensorValue[Colour]);
Sleep (20); // Wait 20 ms to get 50 readings per second
```

Touch: 0 = Not Pressed

1= Pressed

Ultrasonic: Units cm distance

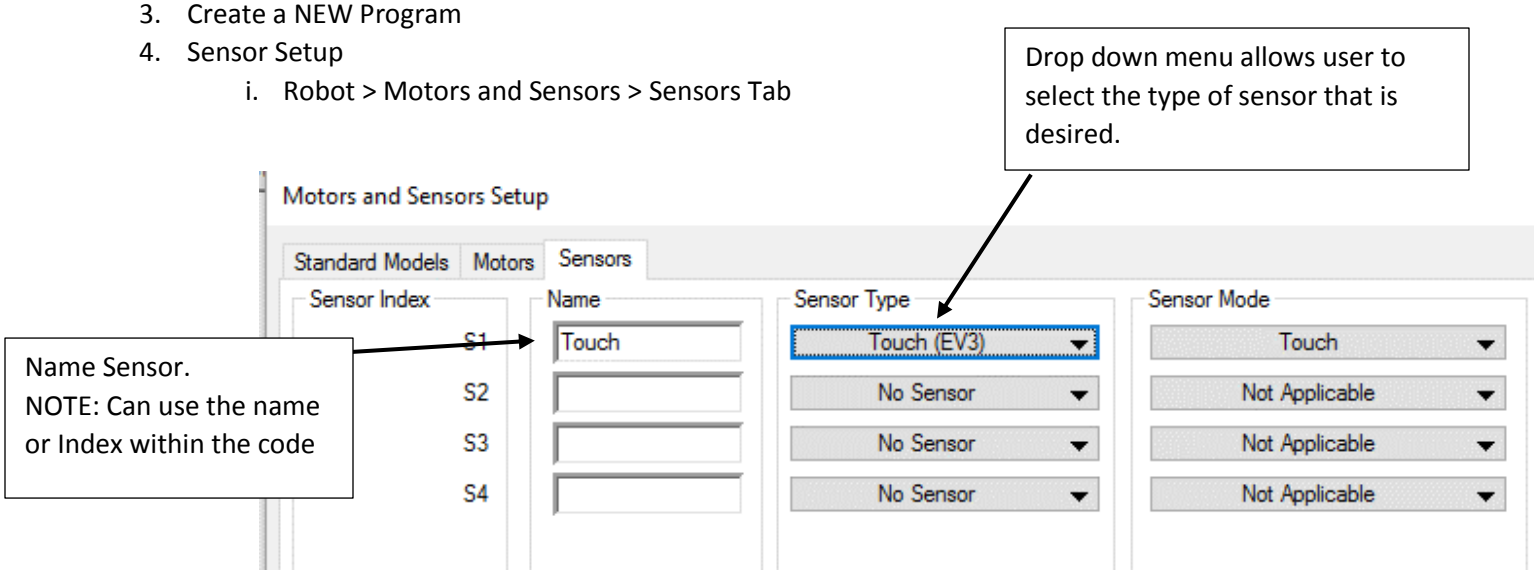
Color: Solid Colors or Light Reflected Percentage

Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Tutorial 2: Touch Sensor

1. Mount the Touch sensor to the front of your robot
2. When using the Touch Sensor there is two values 0 = not pressed and 1 = pressed. Using an If or While statements the programmer can make the robot do conduct different actions based on the inputted data.
3. Create a NEW Program
4. Sensor Setup
    i. Robot > Motors and Sensors > Sensors Tab



    ii. Apply > Ok

5. Write the following program

6.
```
1  #pragma config(Sensor, S4,     Touch,          sensorEV3_Touch)
2  //*!!Code automatically generated by 'ROBOTC' configuration wizard
3
4  task main()
5  {
6    While (Touch == 0)
7    {
8      motor [motorB, 25]; // Shorthand for of motor [motorB] = 25;
9      motor [motorC, 25]; // Shorthand for of motor [motorC] = 25;
10     if (Sensorvalue [Touch] == 1)
11     {
12       motor [motorB, 0]; // Shorthand for of motor [motorB] = 0;
13       motor [motorC, 0]; // Shorthand for of motor [motorC] = 0;
14     }
15   }
16 }
```

7. Upload the program and test the touch sensor with a barrier

Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Robot C Program 2: Touch Sensor

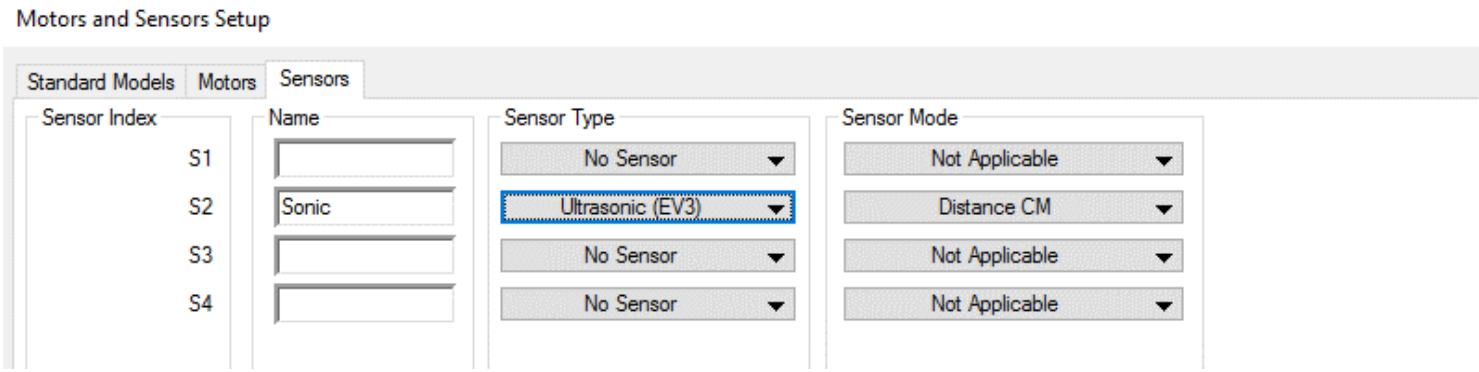
Write a program that will do the following

1. Touch sensor starts the robot going forward (Speed not greater than 25)
2. Touch sensor pressed a second time will stop the robot

Show Teacher upon completion

Tutorial 3: Ultrasonic Sensor

1. Setup: Set the Ultrasonic sensor in the Motor and Sensor Setup to Sonic . Be sure you place the name in the same port. See below for example. Ultra Sensor is set in Port 2.



2. Write the following program.
NOTE: the ultrasonic sensor works with default units of cm

```
1  #pragma config(Sensor, S2,     Sonic,          sensorEV3_Ultrasonic)
2  //*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//
3
4  task main()
5  {
6
7    while (true)
8    {
9      if (SensorValue [Sonic] >= 5)
10     {
11       motor (motorB) = 20;
12       motor (motorC) = 20;
13     }
14
15     else
16     {
17       motor (motorB) = 0;
18       motor (motorC) = 0;
19     }
20   }
21 }
```

While (true) loop is a way to run the program continuously without having a counter or end. In essence it is a forever loop

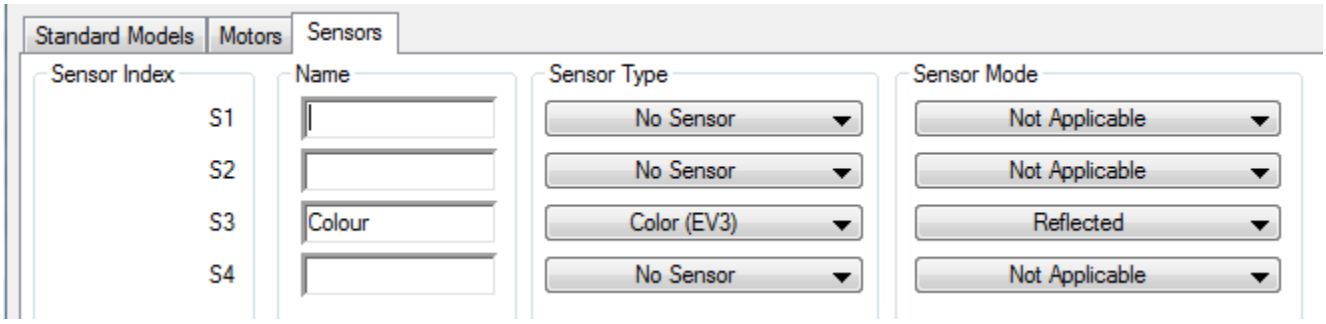
3. Upload and test the program

Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Tutorial 4: Color Sensor

1. Setup: Set the color sensor in the Motor and Sensor Setup to Colour (Note the word Color cannot be used because of a conflict with the sensor name and comparison code). Be sure you place the name in the same port. See below for example. Color Sensor is set in Port 3.



Type the following program.

```
#pragma config(Sensor, S3,     Colour,         sensorEV3_Color)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

task main()
{
  while (true)//While (true) makes the software run continously until the off button is pressed on the robot)
  {
    // Write the amount of reflected light to the screen
    // This is a value between 0 and 100, where 0 means no reflected
    // light and 100 means all light is being reflected
    displayBigTextLine(4, "Reflected: %d", SensorValue[Colour]);

    // Wait 20 ms to get 50 readings per second
    sleep(20);

    if (SensorValue [Colour]  >= 40) //NOTE: Comparison Value for SensorValue [Colour]
                                     //      is based on the ligth reflected percentage
    {
      motor [motorB] = 0;
      motor [motorC] = 0;
      wait1Msec (10);
    }

    else
    {
      motor [motorB] = 15;
      motor [motorC] = 15;
    }
  }
}
```

Program should do the following

When a color sensor light reflection is greater than 40 then the motors will stop. If light reflection is less than 40 than the motors will move. Note: Depending on the surface the comparison value may have to be adjusted.

2. Modify the program accordingly to work on your surface.

Edit/Modify the program for the surface that you are working on so the robot will not fall of the table.

Robot C Programming Tutorial

By: Matthew Jourden
Engineering Program
Brighton High School

Robot C Program 3: Capstone

Design an attachment to your robot that will pick up the predefined object. BE sure to set the type of motor in the Motor and Sensors Setup menu. NOTE: Must use sensors for all movements; NO predefined distances/timing with the motors.

1. Design a gripper that will grab and move a desired object
2. Touch Sensor to Start the Robot
3. Color Sensor
    a. Stop at 1st black line
    b. Pause for 1.4 seconds
    c. Turn Right
    d. Stop at Edge of table
    e. Pause for 1.4 seconds
    f. Back up 2 wheel rotations
    g. Turn Left
    h. Drive to the object
4. Sense Object to be picked up (May use any type of sensor)
5. Drive to the drop off area.
6. Release object
7. Reverse robot without touching or knocking the object over
8. Stop
9. Return to the Start position

Show teacher when completed