

Using Sensors & Encoders

Carson Wolf

Agenda

- Sensor Basics
- Sensor Programming
 - IR Sensor
 - Color Sensor
 - Magnet Sensor
 - Encoder
- Tips
- Questions & Contact Information

Sensor Basics

Why Are Sensors Important?

- Sensor if used correctly can make your autonomous much more accurate and make driving easier during teleop.
- Sensors can be used during:
 - Autonomous
 - Use encoder and/or color sensor to improve accuracy of field movement
 - Sensor will help you find field elements and scoring elements
 - Example, Get Over It! -- IR Sensor used to find doubler baton
 - Teleop
 - Do not forget that sensors can also help you during the driver control period
 - Assist drivers with alignment, accuracy, and consistency of scoring and/or getting the scoring element.
 - Example, Get Over It! – We used 2 touch sensors to help align the robot to the baton dispenser and keep our forearm the correct distance from the dispenser.

Configuring Sensors in RobotC

The screenshot shows the 'Sensor Management' window in RobotC. The window has a tabbed interface with tabs for 'Standard Models', 'Sensor Management', 'TETRIX Controllers', 'Motors', 'Servos', and 'Sensors'. The 'Sensors' tab is selected. The window contains three columns: 'Port', 'Name', and 'Type'. There are four rows corresponding to ports S1, S2, S3, and S4. The 'Name' column has empty text boxes for each port. The 'Type' column has dropdown menus for each port, all of which are currently set to 'No Sensor'. At the bottom of the window are buttons for 'OK', 'Cancel', 'Apply', and 'Help'.

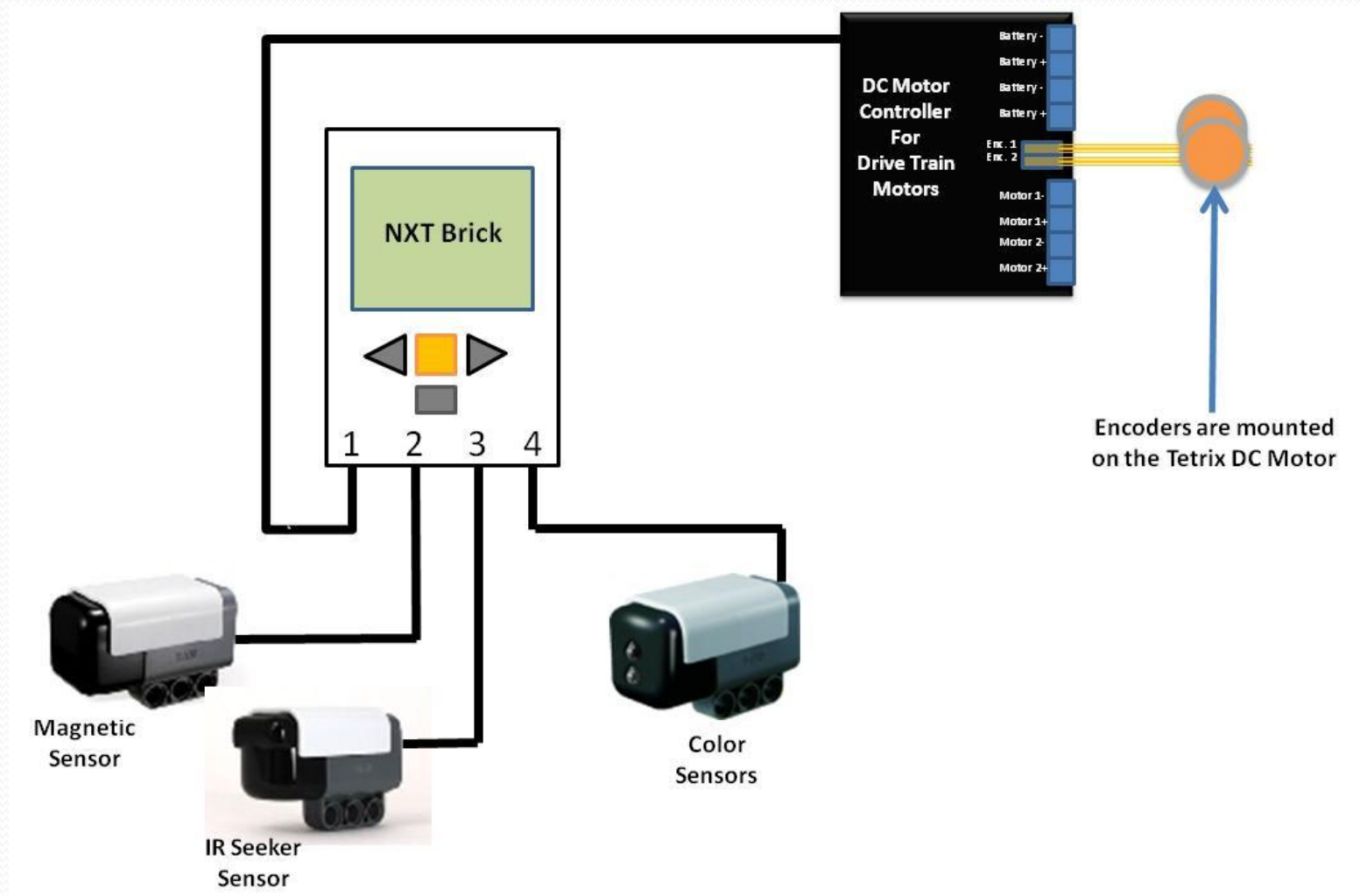
Port	Name	Type
S1		No Sensor
S2		No Sensor
S3		No Sensor
S4		No Sensor

- One Sensor port will be used by the motor/servo controllers.
- Therefore, you can only directly connect three Sensors directly to the NXT.

pragma Tips

- Sometimes RobotC will not have support for the newer sensors.
- Options for handling new sensors:
 - manually create the pragma statement
 - Select Raw Value in RobotC when configuring sensors
- Examples,
 - RobotC did not directly support the magnetic sensor during the Get Over It! challenge from last year
- What is a #pragma statement?
 - Directive to the RobotC compiler to include additional information into your code.
 - The pragma statement will tell the compiler to include functions that you use to work with the Sensor.

Connecting Sensors



Adding More Sensors

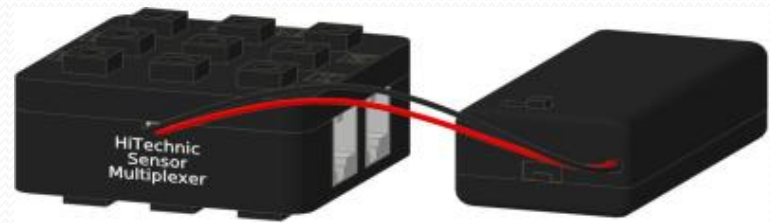
- Using more than three Sensors will require the use of multiplexer.
 - We will not be showing you how to program the multiplexer.

Touch Sensor Multiplexer



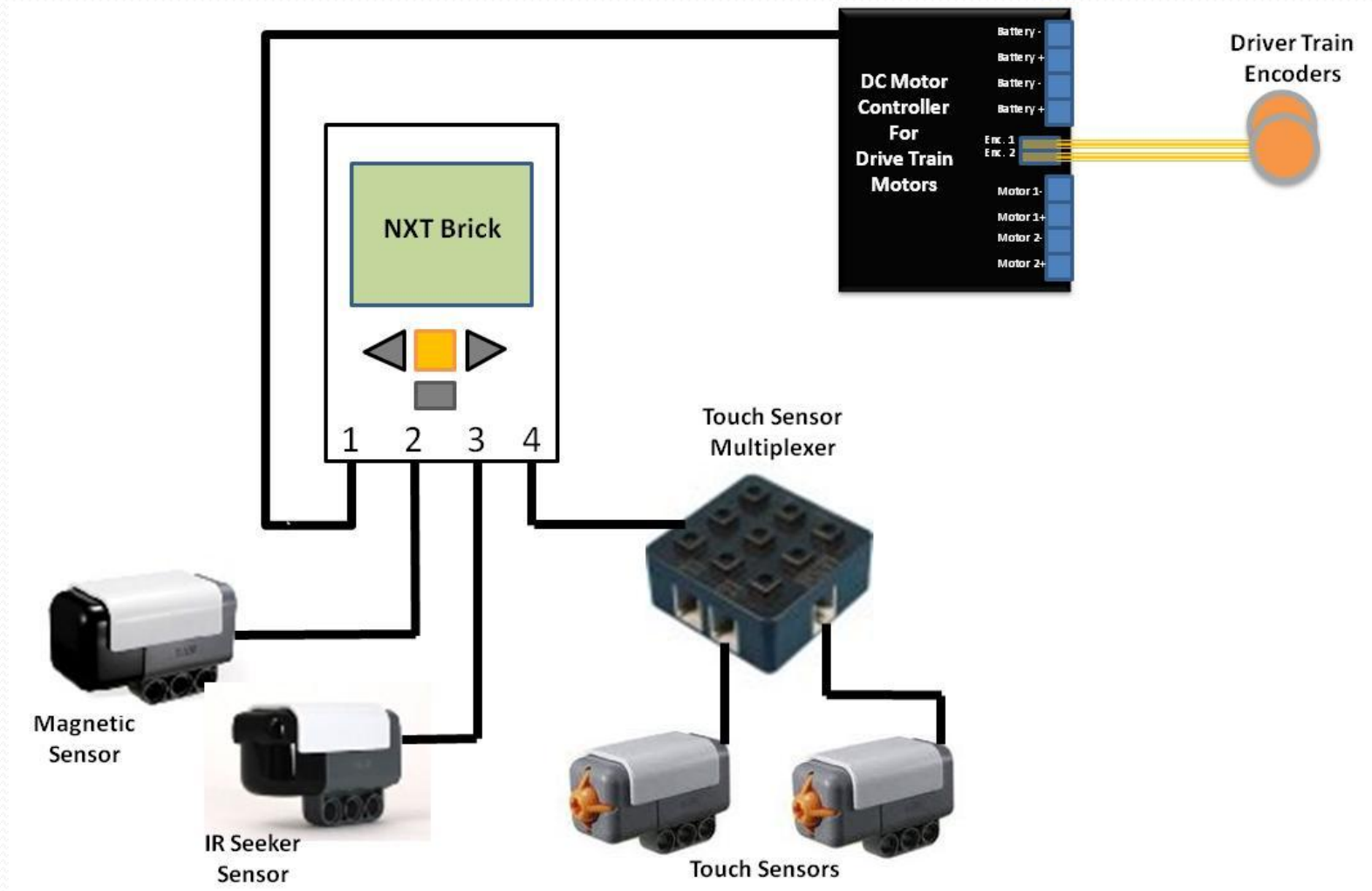
- Connect up to 4 Touch Sensors.
- Advantage: Does not require a separate power source.
- Disadvantage: Only supports the Touch Sensor.

Sensor Multiplexer



- Connect up to 4 Sensors.
- Advantage: Supports any type of sensor.
- Disadvantage: Requires a separate 9-volt power source.

More Than Three Sensor



Sensor Programming

- IR Seeker
- Magnetic
- Color
- Encoder

IR Seeker

The IR seeker has a 240 degree view divided into 9 zones to pick up an IR beacon.

- You can use most TV remote to test.
- The IR beacon has been used in the last two games.
 - High goal from Hot Shot!
 - Middle dispenser from Get Over It! to help you locate the double baton.

IR Seeker



Back of the Sensor
will have the Type
of Sensor

IR Beacon



IR Seeker

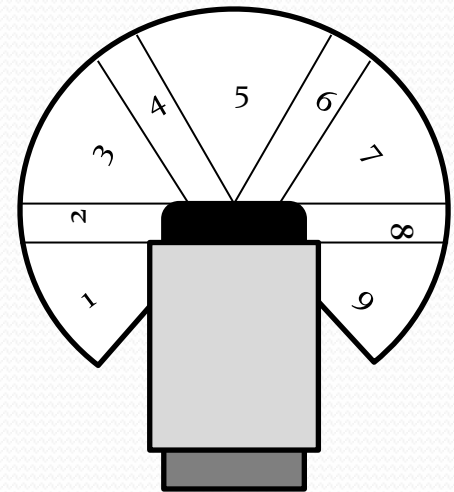
- What is an IR Seeker Sensor --- looking for an IR beacon
- IR Pragma

— Sensor port — Name of sensor

```
#pragma config(Sensor, S4, ir, sensorHiTechnicIRSeeker600)
```

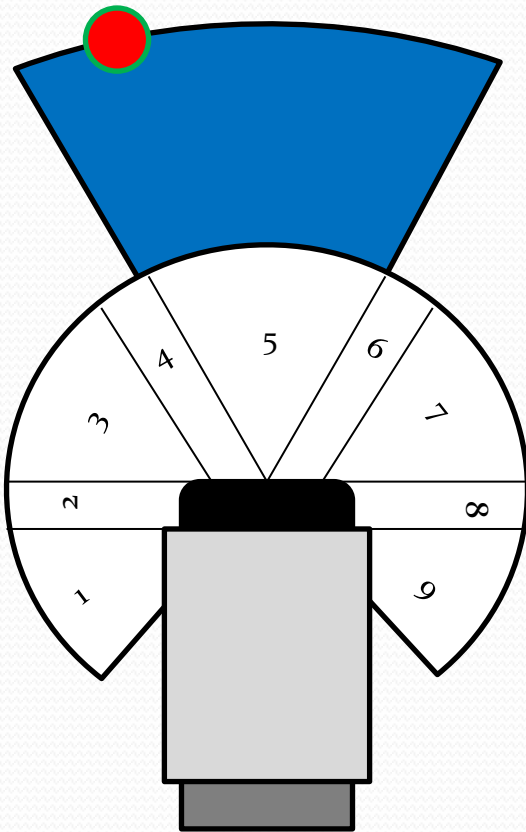
- Sensor returns an integer between 0-9
- How to get value?
 - Call the function `SensorValue`

`SensorValue[ir]` ← Name of sensor

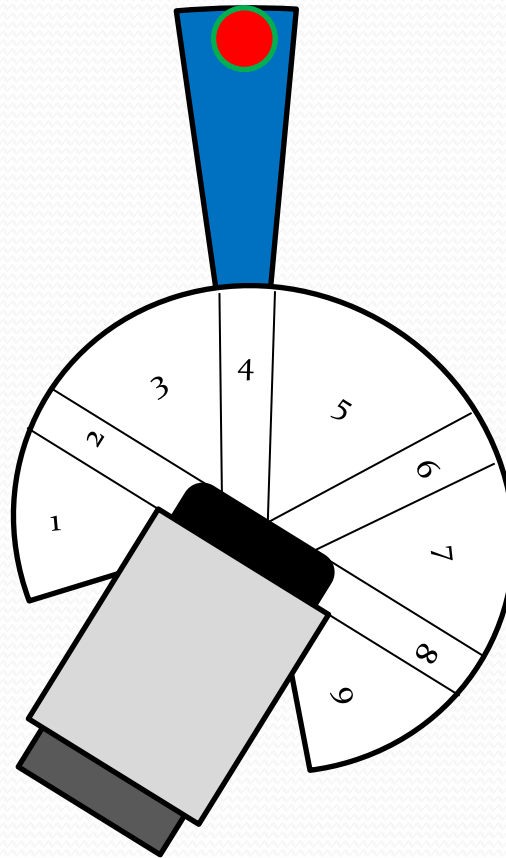


Ways to Use the IR Sensor

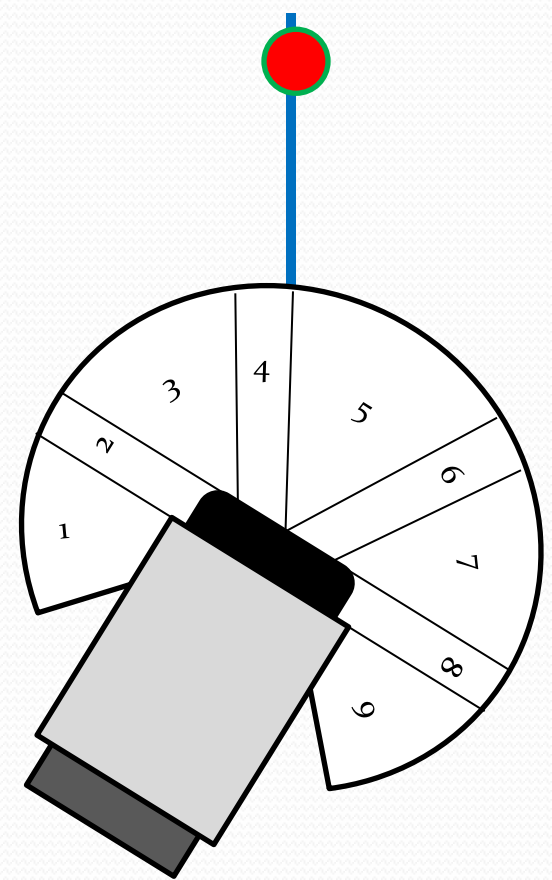
Option 1



Option 2



Option 3



IR Sensor will need to be mounted at the correct angle

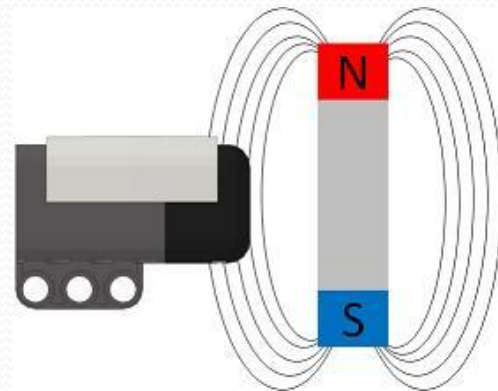
Example code

```
1  #pragma config(Sensor, S4,      irseek,          sensorHiTechnicIRSeeker600)
2
3  task main()
4  {
5      while (true)
6      {
7
8          nSeeker = SensorValue[irseek];
9
10         if(SensorValue[irseek] ==5 || SensorValue[irseek] ==0)//front of robot and no ir
11         {
12             motor[motorA]=0;//robot stops
13         }
14         else if(SensorValue[irseek] <5)//left of robot
15         {
16             motor[motorA]=-25;//robot spins left
17         }
18         else if(SensorValue[irseek] >5)//right of robot
19         {
20             motor[motorA]=25;//robot spins right
21         }
22     }
23 }
24
```

Magnet Sensor

The magnet sensor detects magnetic fields.

- A magnet was used in a game element in the Get Over It! Challenge.
- Using raw data Sensor returns an integer between 0-1024.
 - Support for the Magnetic Sensor has been added to RobotC 3.0
- When no magnet is present value will hang around 512.
 - Magnet field in your location will impact the center point.
 - Therefore, you will need to establish a basis/baseline value.
- The value will decrease from 512 the closer the sensor is to the south pole.
- The value will increase from 512 the closer the sensor is to the north pole.



Magnet Sensor

- Magnet Pragma


`#pragma config(Sensor, s2, mag, sensorNone)`

- Make sure to set a bias.

`int bias;` ← Establish variable

`bias=SensorValue[mag];` ← Set bias at start of program

`SensorValue[mag]-bias` ← Use this as the adjusted value

- Using the absolute value of the adjusted value will return a positive value regardless of the pole.

`abs(SensorValue[mag]-bias)`

Magnetic Sensor Tip

- Be careful where you mount the Magnetic Sensor on your robot.
- Do not mount the Sensor too close to your Motors.
- Electric Motors have a magnetic field.



} Too Close

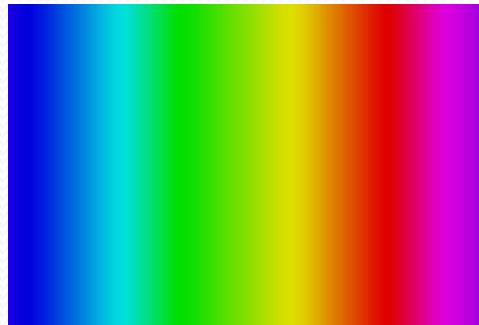
Color Sensor

What is a color sensor and how is it different from the light sensor?

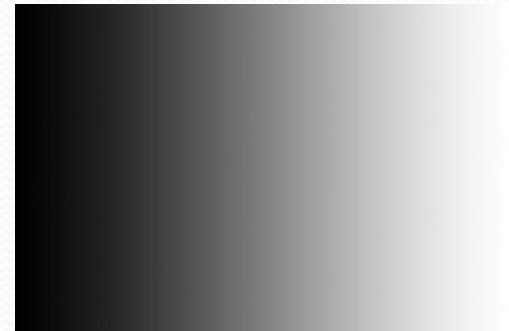
- Light sensor returns one number of light reflected off a surface
- Color sensor can return many types of values
 - Color number
 - Red
 - Blue
 - Green



Color Sensor



Light Sensor




Color Sensor

- Color Pragma

Sensor port Name of sensor

```
#pragma config(Sensor, S2, HTCS2, sensorLowSpeed)
```



- Drivers

```
#include "drivers/common.h"  
#include "drivers/HTCS2-driver.h"
```

Color number

- Function returns an integer between 0-17

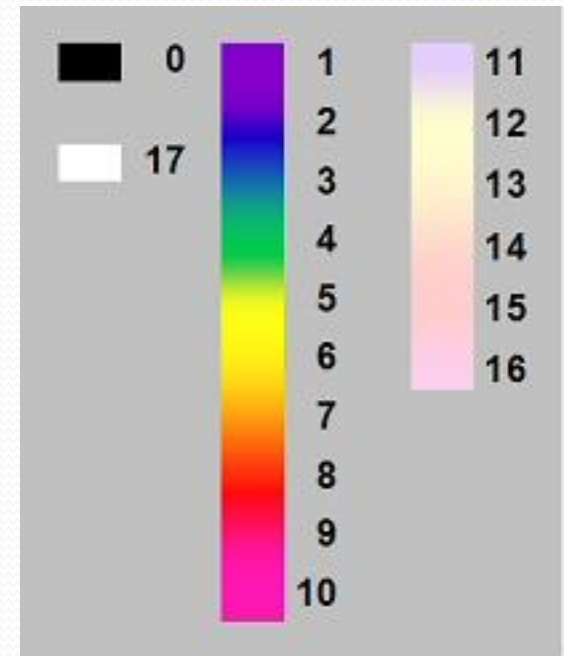
```
int color;
```

Establish variable

```
color = HTCS2readColor(HTCS2);
```

Use this function to read
color number

Name of sensor



RGB numbers

- I primarily use the RGB values; I typically do not use the color number described on the previous page.
- Function returns 3 integer between 0-255

`int green, blue, red;` ← Establish variable

Use this function to read
color number

Name of sensor

`HTCS2readRGB(HTCS2, red, green, blue)`

Motor Encoders

- Motor encoders are used to measure wheel rotations.
- Lego motors have motor encoders built in.
- The Tetrix DC Motor can have a motor encoder attached.
 - The Tetrix DC motor encoders are wired in to the HiTechnic DC Motor Controller s

Lego Motor



Tetrix DC Motor Encoder



Motor encoders

- Motor encoder are measured in ticks.
- Encoders should be initialized to zero before use.
- To get number of ticks use this for both Tetrix and Lego motors.

`nMotorEncoder[right]` Name of motor

1440 ticks per rotation



360 ticks per rotation



Example code

```
3  task main()  
4  {  
5      motor[left]=50;  
6      motor[right]=50;  
7      nMotorEncoder[left]=0;  
8      wait1Msec(50);  
9      while(nMotorEncoder[left]<1440) {}  
10     motor[left]=0;  
11     motor[right]=0;  
12 }
```

Turn on motors

Reset encoders

Wait to allow encoders to reset

Continue moving until motor has completed one full revolution.

Tips

Working with a New Sensor

- Use your spare NXT to build a simple test program with just the new sensor attached.
- Experiment with the game elements and make changes to the program.
- Display variables on the NXT screen .

```
nxtDisplayTextLine(line_number, text %d %d %d, var1, var2, var3);
```

Testing the Magnetic Sensor



Use %d to put up to 3 variables

RobotC Websites

- www.robotc.net
- www.education.rec.ri.cmu.edu/previews/robot_c_products/teaching_robot_c_tetrix_preview/
- www.robotc.net/forums

Questions?

Contact Info:

Hans Wolf

email: hans.wolf@verizon.net

Cell phone: (813) 477-4595

Carson Wolf

email: carson.wolf@verizon.net

Cell phone: (813) 352-6581