
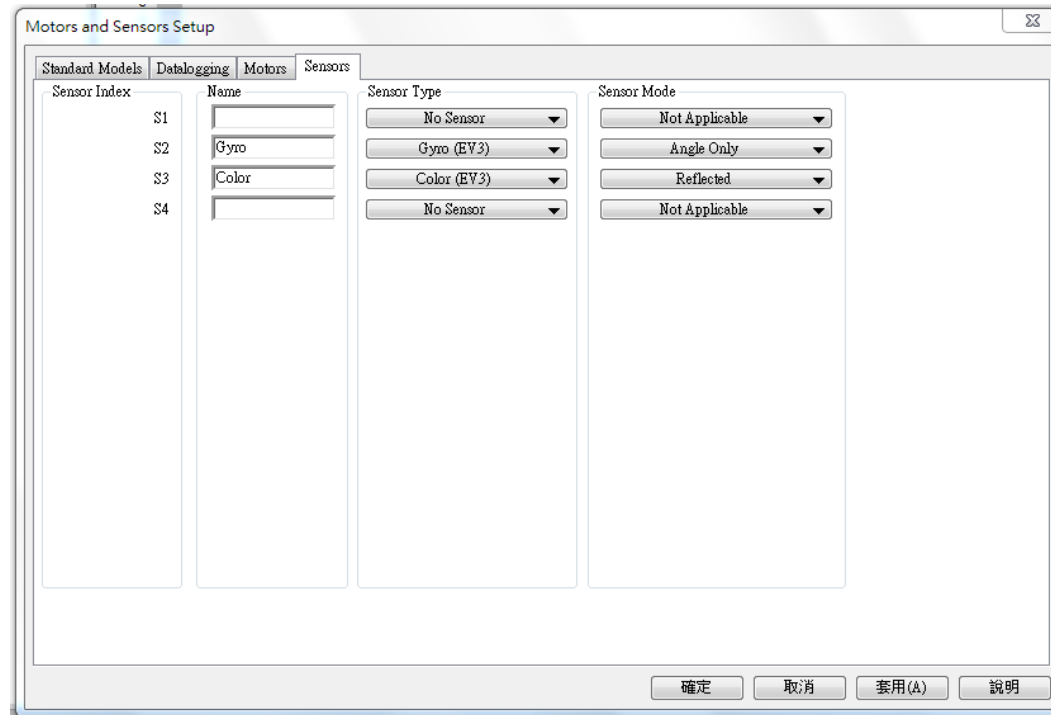


# LEGO 講義 3

1. 陀螺儀sensor範例
2. 顏色sensor範例
3. 實驗練習
4. 期末Project規定

# Sensor設定

- 首先設定輸入與輸出的接腳位置，點選  Motor and Sensor Setup 進入視窗。
- Sensor名稱設定，S2設為陀螺儀sensor，S3設為顏色sensor。



# 陀螺儀SENSOR範例

# 陀螺儀sensor範例

- 直接以內建程式測試：
  - 開機後選擇Port View > GYRO-ANG，將陀螺儀感測器接到對應的port(1~4)。
  - 旋轉陀螺儀，螢幕上會顯示對應的角度讀數。

# 陀螺儀sensor範例

- 範例程式：getGyroDegrees

//此程式會重設陀螺儀的方向，並且使機器人旋轉90度。

//將port2的陀螺儀重設為0度。

```
resetGyro(S2);
```

//一直重複執行轉向直到陀螺儀讀到的值大於90度

```
repeatUntil(getGyroDegrees(S2) > 90)
```

```
{
```

```
    //左轉
```

```
    setMotorSpeed(motorC, -50);
```

```
    setMotorSpeed(motorB, 50);
```

```
}
```

//停止馬達

```
setMotorSpeed(motorB, 0);
```

```
setMotorSpeed(motorC, 0);
```

# 陀螺儀sensor範例

- 範例程式：getGyroHeading

```
//使機器人旋轉90度
```

```
//一直重複執行轉向直到陀螺儀讀到的值大於90度(自上次重設位置)
```

```
repeatUntil(getGyroHeading(port4) > 90)
```

```
{
```

```
  //左轉
```

```
  setMotorSpeed(motor1, -50);
```

```
  setMotorSpeed(motor6, 50);
```

```
}
```

```
//停止馬達
```

```
setMotorSpeed(motor1, 0);
```

```
setMotorSpeed(motor6, 0);
```

# 陀螺儀sensor範例

- 範例程式：getGyroRate

```
//無限迴圈
```

```
repeat(forever)
```

```
{
```

```
    //如果機器人正在左轉，將之轉向右
```

```
    if(getGyroRate(S2) > 3)
```

```
    {
```

```
        //右轉
```

```
        //將馬達C的速度設為55，馬達B的速度設為45
```

```
        setMotorSpeed(motorC, 55);
```

```
        setMotorSpeed(motorB, 45);
```

```
    }
```

# 陀螺儀sensor範例

- 範例程式：getGyroRate

```
//如果機器人正在右轉，將之轉向左
else if(getGyroRate(S2) < -3)
{
    //左轉
    //將馬達C的速度設為45，馬達B的速度設為55
    setMotorSpeed(motorC, 45);
    setMotorSpeed(motorB, 55);
}

//其他狀況，直走
else
{
    //直走
    setMotorSpeed(motorC, 50);
    setMotorSpeed(motorB, 50);
}
}
```



# 顏色SENSOR範例

# 顏色sensor範例

- 直接以內建程式測試：
  - 開機後選擇Port View > COL-REFLECT，將顏色感測器接到對應的port(1~4)。
  - 在顏色偵測模式下，紅、綠、藍三個顏色的燈都會亮起來，能分辨不同顏色並傳回不同的值：  
黑(1)、藍(2)、綠(3)、黃(4)、紅(5)、白(6)、棕(7)、沒有顏色(0)
  - 注意sensor距離不可太近或太遠，否則可能讀不到值。

# 顏色sensor範例

- 範例程式：getColorHue

```
//重複直至SENSOR讀到藍色(240)
```

```
repeatUntil(getColorHue(S3) == 240)
```

```
{
```

```
    //前進
```

```
    setMotorSpeed(motorB, 50);
```

```
    setMotorSpeed(motorC, 50);
```

```
    delay(50);
```

```
}
```

```
//停止
```

```
setMotorSpeed(motorB, 0);
```

```
setMotorSpeed(motorC, 0);
```

- 讀值總共0到360，請依實際環境測試為準。



# 顏色sensor範例

- 範例程式：getColorName

```
//找物體顏色，存成字串
```

```
string colorOfObject;
```

```
repeat(forever){
```

```
    //如果Port3的顏色Sensor偵測到紅色物體，將"RED"存至字串
```

```
    if(getColorName(S3) == colorRed){
```

```
        colorOfObject = "RED";
```

```
        displayTextLine(2,"%s",colorOfObject);
```

```
        delay(50);
```

```
    }
```

```
    //如果Port3的顏色Sensor偵測到藍物體，將"BLUE"存至字串
```

```
    else if(getColorName(S3) == colorBlue){
```

```
        //Store the text Blue into the color string
```

```
        colorOfObject = "BLUE";
```

```
        displayTextLine(2,"%s",colorOfObject);
```

```
        delay(50);
```

```
    }
```

```
}
```

# 顏色sensor範例

- 範例程式：getColorRGB

```
//宣告變數
long redValue;
long greenValue;
long blueValue;
    while (true)
    {
        //讀取RGB三個channel的值
        getColorRGB(colorSensor, redValue, greenValue, blueValue);

        //將值存入DebugStreamLine
        writeDebugStreamLine("Colour detected: %d, %d, %d", redValue, greenValue,
blueValue);

        sleep(100);
    }
```

# 顏色sensor範例

- 函式：**getColorAmbient**

```
getColorAmbient(S3) ;  
//讀到port3的明暗度(red LED off)
```

- 函式：**getColorReflected**

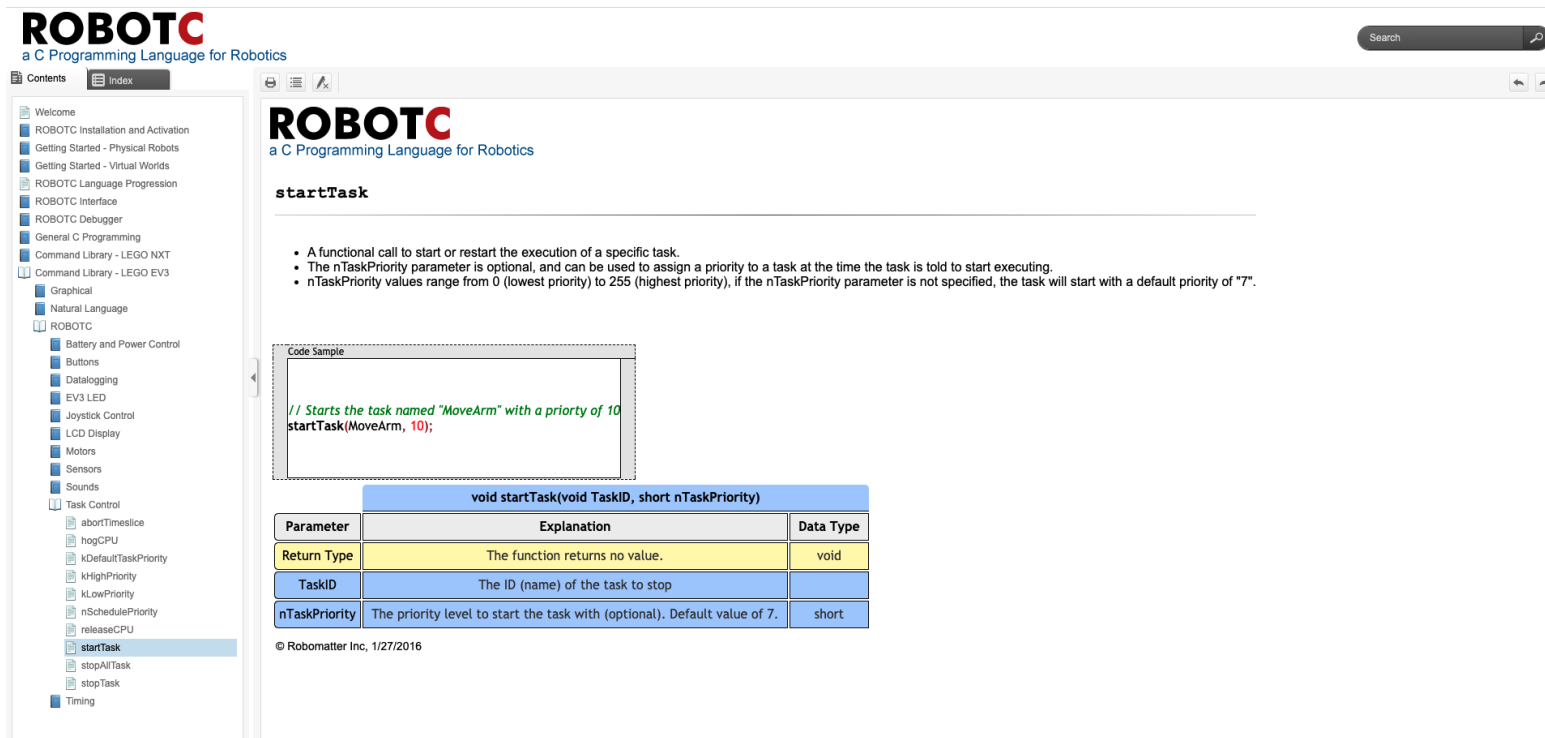
```
getColorReflected(S3) ;  
//讀到port3的明暗度(red LED on)
```

- 函式：**getColorSaturation**

```
getColorSaturation(S3) ;  
//讀到port3的飽和度
```

# !Advance – Task Control!

- <https://www.robotc.net/WebHelpMindstorms/index.htm>
- 左側欄位中選擇”Command Library – LEGO EV3”



**ROBOTC**  
a C Programming Language for Robotics

Search

Contents Index

Welcome

- ROBOTC Installation and Activation
- Getting Started - Physical Robots
- Getting Started - Virtual Worlds
- ROBOTC Language Progression
- ROBOTC Interface
- ROBOTC Debugger
- General C Programming
- Command Library - LEGO NXT
- Command Library - LEGO EV3
- Graphical
- Natural Language
- ROBOTC
  - Battery and Power Control
  - Buttons
  - Datalogging
  - EV3 LED
  - Joystick Control
  - LCD Display
  - Motors
  - Sensors
  - Sounds
  - Task Control
    - abortTimeslice
    - hogCPU
    - kDefaultTaskPriority
    - kHighPriority
    - kLowPriority
    - nSchedulePriority
    - releaseCPU
    - startTask**
    - stopAllTask
    - stopTask
    - Timing

**ROBOTC**  
a C Programming Language for Robotics

**startTask**

- A functional call to start or restart the execution of a specific task.
- The nTaskPriority parameter is optional, and can be used to assign a priority to a task at the time the task is told to start executing.
- nTaskPriority values range from 0 (lowest priority) to 255 (highest priority), if the nTaskPriority parameter is not specified, the task will start with a default priority of "7".

Code Sample

```
// Starts the task named "MoveArm" with a priority of 10
startTask(MoveArm, 10);
```

void startTask(void TaskID, short nTaskPriority)		
Parameter	Explanation	Data Type
Return Type	The function returns no value.	void
TaskID	The ID (name) of the task to stop	
nTaskPriority	The priority level to start the task with (optional). Default value of 7.	short

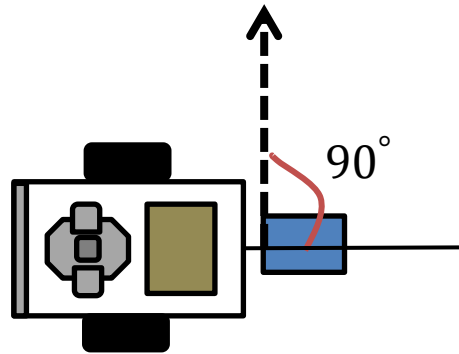
© Robomatter Inc, 1/27/2016

# 實驗練習



# 實驗練習

- 實驗一、偵測到紅色停止，綠色直走，藍色左轉90度，黃色右轉90度。



- 實驗二、使機器人沿著桌面上貼的黑線走。
- 請下禮拜一組繳交一份書面結報

# 沿黑線走概觀



50



45



40

# 期末PROJECT規定

# 期末Project規定

- 使用第3顆馬達組成一個可夾取物體的機構。
- 使用至少兩種sensor，偵測到物體存在後(物體沒有限制)，機器手臂將物體夾取，並統一將物體放在最右側。
- 期末報告時間一組15~25分鐘，需拍攝影片(將物體放在相對不同的兩種角度進行實驗)，並在報告後繳交書面資料。