

# V5 智能传感器



# 目 录

CONTENTS

PART 01

V5惯性传感器

PART 02

V5测距仪

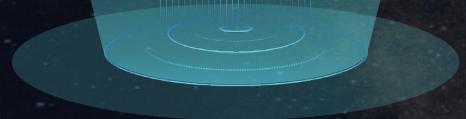
PART 03

V5光感器

PART 04

V5旋转传感器

PRAT ONE

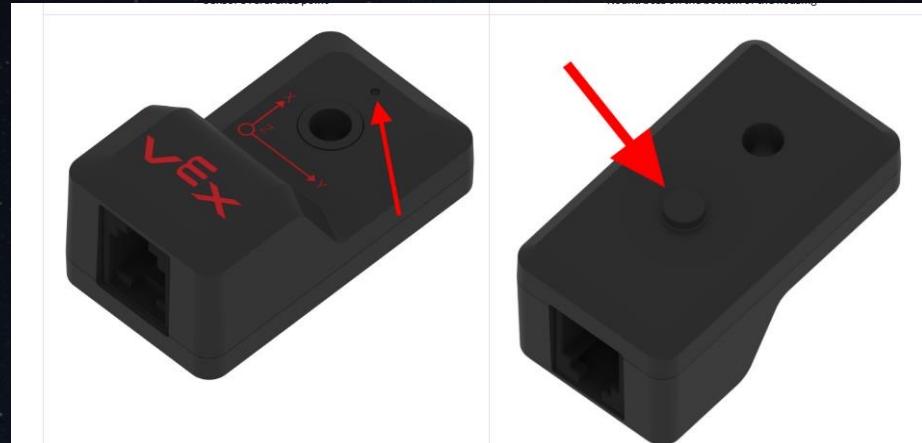


V5惯性传感器

# V5惯性传感器

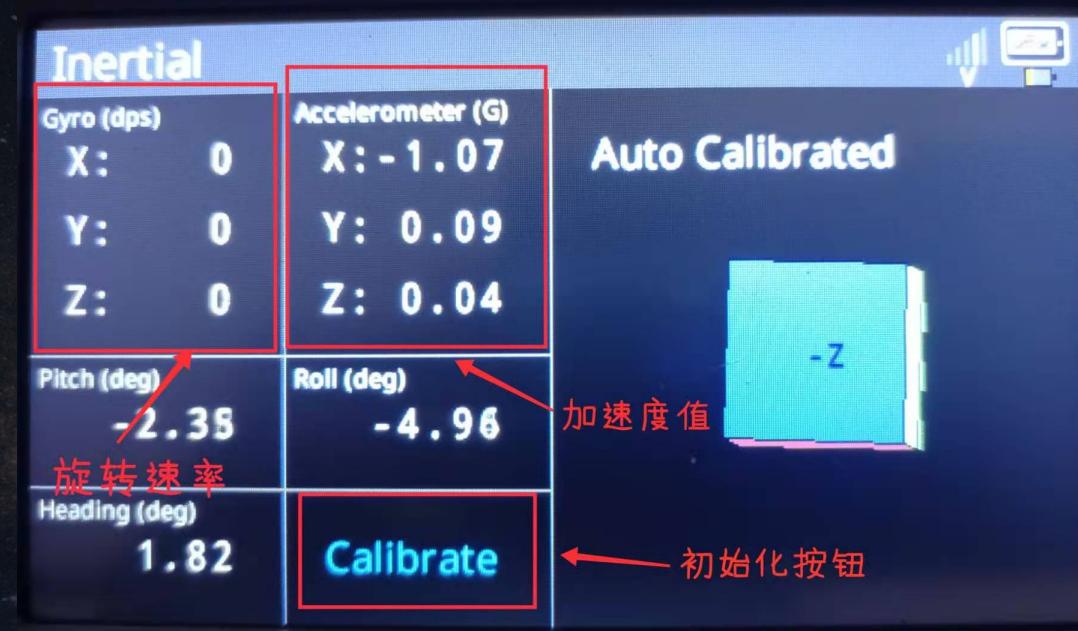
## 惯性传感器是什么？

惯性传感器是3轴（X, Y和Z）加速度计和3轴陀螺仪的组合。加速度计将检测沿任何方向的运动（加速度）变化，并且陀螺仪以电子方式保持参考位置，因此它可以相对于该参考方向测量任何方向上的旋转位置变化。



V5惯性传感器

# V5惯性传感器



惯性传感器调试界面

# V5惯性传感器

惯性传感器常用代码：

`INERTIAL.acceleration(xaxis);`

报告惯性传感器上某个轴（x、y或z）的加速度值。

`INERTIAL.calibrate();`

校准惯性传感器用于减少惯性传感器产生的漂移量

`INERTIAL.gyroRate(xaxis, dps)`

获取惯性传感器上指定轴（x、y或z）的旋转速率

`INERTIAL.heading(degrees)`

以度为单位报告惯性传感器的当前航向

`INERTIAL.orientation(pitch, degrees);`

获取惯性传感器的方向角。报告惯性传感器参数指定的单位值

`INERTIAL.rotation(degrees)`

报告惯性传感器的当前旋转角度（以度为单位）

`INERTIAL.setHeading(0, degrees);`

将惯性传感器的当前航向位置设置为设定值

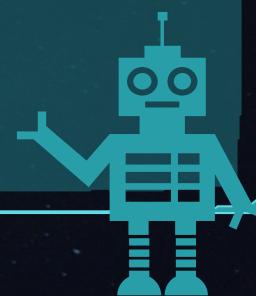
`INERTIAL.setRotation(90, degrees);`

将V5惯性传感器的旋转角度设置为提供的值

# 例 子

C l i c k   o n   t h e   I n p u t   T i t l e

编写一个程序，使用惯性传感器  
控制机器人向右旋转90度。



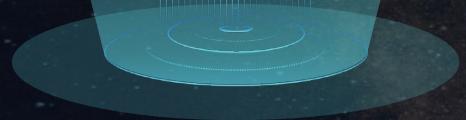
# V5惯性传感器

```
int main() {
    // Initializing Robot Configuration. DO NOT REMOVE!
    vexcodeInit();
    Inertial.calibrate();
    waitUntil(!Inertial.isCalibrating());
    Lmotor.spin(forward);
    Rmotor.spin(reverse);
    waitUntil(Inertial.heading(deg) >= 90);
    Lmotor.stop();
    Rmotor.stop();

}
```

惯性传感器程序

PRAT TWO



V5测距仪

# V5测距仪

## V5测距仪是什么？

V5测距仪可以测量到物体的距离、近似物体大小和接近速度。距离测量范围为20~2000mm。200mm以下精度约为 $\pm 15\text{mm}$ ，200mm以上精度约为5%。对象的大致大小报告为小、中或大，用于确定目标是墙还是场地要素。接近速度可以测量传感器和物体之间的接近速度。



V5测距仪

# V5测距仪



测距仪调试界面

# V5测距仪

测距仪常用代码：

DISTANCE.isObjectDetected()

报告测距仪是否在其范围内检测到物体或曲面

DISTANCE.objectDistance(inches)

报告距离测距仪最近的对象或曲面的距离

DISTANCE.objectSize()

报告检测到的对象大小的估计值

DISTANCE.objectVelocity()

以m/s为单位报告检测到的对象的速度

# 例 子

C L i c k   o n   t h e   I n p u t   T i t l e

编写一个程序，控制小车前进，  
检测小车与物体间的距离和接近速度，  
并打印在屏幕上。

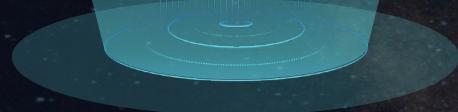


# V5测距仪

```
int main() {
    // Initializing Robot Configuration. DO NOT REMOVE!
    vexcodeInit();
    float speed;
    while(1)
    {
        speed = Distance1.objectVelocity();
        Lmotor.spin(forward);
        Rmotor.spin(forward);
        Brain.Screen.clearScreen();
        Brain.Screen.printAt(100,100,"%3fm/S",speed);
        wait(20,msec);
    }
}
```

测距仪程序

PRAT THREE



V5光感器

# V5光感器

## V5光感器是什么？

V5光感器是环境光、颜色、接近度和手势传感器的组合。颜色信息可用作RGB、色调和饱和度或灰度。当物体距离小于100mm时，颜色检测效果最好。



V5光感器

# V5光感器



光感器调试界面

# V5光感器

光感器常用代码：

OPTICAL.brightness()

报告光感器检测到的亮度

OPTICAL.color()

返回光感器检测到的颜色

OPTICAL.hue()

报告光感器检测到的色调

OPTICAL.isNearObject()

报告光感器是否在检测范围内检测到物体

OPTICAL.setLight(ledState::on);

设置光感器LED灯的状态

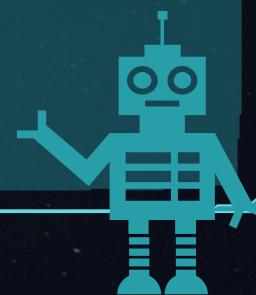
OPTICAL.setLightPower(100, percent);

设置光感器LED灯的亮度

# 例 子

C L i c k   o n   t h e   I n p u t   T i t l e

编写一个程序，将光感器检测到的亮度，色调打印在主控屏幕上。

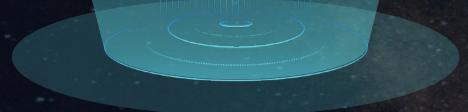


# V5光感器

```
int main() {
    // Initializing Robot Configuration. DO NOT REMOVE!
    vexcodeInit();
    float Bright,Hue;
    while(1)
    {
        Brain.Screen.clearScreen();
        Bright = Optical.brightness();
        Hue = Optical.hue();
        Brain.Screen.printAt(200,50,"% .3f",Bright);
        Brain.Screen.printAt(200,150,"% .3f",Hue);
        wait(20,msec);
    }
}
```

光感器程序

PRAT FOUR



V5 旋转传感器

# V5旋转传感器

## 旋转传感器是什么？

V5旋转传感器可以测量轴旋转位置、总旋转角度和旋转速度。旋转位置从0°到360°测量，精度为0.088。角度是绝对确定的，并且在机器人断电时不会丢失。



V5光感器

# V5旋转传感器



旋转传感器调试界面

# V5旋转传感器

旋转传感器常用代码：

ROTATION.angle()

报告旋转传感器的当前旋转角度（以度为单位）

ROTATION.position(degrees)

报告旋转传感器的当前位置

ROTATION.resetPosition();

将旋转传感器的位置重置为0

ROTATION.setPosition(0, degrees);

将旋转传感器的位置设置为指定值

ROTATION.velocity(rpm)

报告V5旋转传感器的当前速度

# 例 子

C L i c k   o n   t h e   I n p u t   T i t l e

编写一个程序，使用旋转传感器

让电机正转3圈，反转3圈然后停止。



# V5旋转传感器

```
int main() {
    // Initializing Robot Configuration. DO NOT REMOVE!
    vexcodeInit();
    Rotation.resetPosition();
    Rotation.position(degrees);
    Motor1.spin(forward);
    waitUntil(Rotation.position(degrees)>1080);
    Motor1.spin(reverse);
    Rotation.resetPosition();
    waitUntil(Rotation.position(degrees)<-1080);
    Motor1.stop();
}
```

旋转传感器程序

# 作 业

C L i c k   o n   t h e   I n p u t   T i t l e

选择一个今天学习过的传感器，分析其可以安装在机器人的什么位置，起到什么样的作用。



# THANKS

• \* \* 科 技 发 展 有 限 公 司 •



ishine