1. Introducción

El sensor giroscópico EV3 digital mide el movimiento de rotación del robot y los cambios en su orientación.



Características del giroscopio:

1. Dos salidas separadas por eje para aplicaciones de juegos de alta velocidad y navegación de menú a baja velocidad: rango de escala completa de 2000 ° / s (juegos de alta velocidad) Rango de escala completa de 440 ° / s (apuntando) • Baja deriva de polarización sobre la temperatura • On- sensor de temperatura del chip
2. Sensor de temperatura Amplificadores integrados y filtros de paso bajo.
3. El giroscopio vibratorio MEMS de doble masa ofrece un rechazo de vibración superior en un amplio rango de frecuencias. El diseño de doble masa rechaza intrínsecamente cualquier señal causada por la aceleración lineal.
4. Tolerancia de choque de 10,000 g La tasa de salida del giroscopio no es ratiométrica al voltaje de suministro. El factor de escala se calibra en la fundición de chips y es nominalmente independiente del voltaje de suministro.

2. Cómo usar el sensor

2.1. Lee el ángulo

package ev3dev.sensors.ev3;

import ev3dev.sensors.Battery;

import lejos.hardware.port.SensorPort;

import lejos.robotics.SampleProvider;

import lejos.utility.Delay;

public class GyroSensorDemo {

//Robot Configuration

private static EV3GyroSensor gyroSensor = new EV3GyroSensor(SensorPort.S1);

//Configuration

private static int HALF\_SECOND = 500;

public static void main(String[] args) {

final SampleProvider sp = gyroSensor.getAngleMode();

int value = 0;

//Control loop

final int iteration\_threshold = 20;

for(int i = 0; i <= iteration\_threshold; i++) {

float [] sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

value = (int)sample[0];

System.out.println("Iteration: " + i);

System.out.println("Gyro angle: " + value);

Delay.msDelay(HALF\_SECOND);

}

System.out.println(Battery.getInstance().getVoltage());

}

}

2.2. Usa el modo de velocidad

package ev3dev.sensors.ev3;

import lejos.hardware.port.SensorPort;

import lejos.robotics.SampleProvider;

public class GyroSensorDemo2 {

//Robot Configuration

private static EV3GyroSensor gyroSensor = new EV3GyroSensor(SensorPort.S1);

public static void main(String[] args) {

SampleProvider sp = gyroSensor.getRateMode();

float[] sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

sp = gyroSensor.getAngleMode();

sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

sp = gyroSensor.getAngleAndRateMode();

sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

gyroSensor.reset();

}

}

2.3. Detecta que tu robot gire 90 grados

package ev3dev.sensors.ev3;

import lejos.hardware.port.SensorPort;

import lejos.robotics.SampleProvider;

import lejos.utility.Delay;

import org.slf4j.Logger;

public class GyroSensorDemo3 {

private static final Logger log = org.slf4j.LoggerFactory.getLogger(GyroSensorDemo3.class);

//Robot Configuration

private static EV3GyroSensor gyroSensor = new EV3GyroSensor(SensorPort.S1);

//Configuration

private static int HALF\_SECOND = 500;

public static void main(String[] args) {

final SampleProvider sp = gyroSensor.getAngleAndRateMode();

int value = 0;

int iterationCounter = 0;

//Control loop

while(true){

float [] sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

value = (int)sample[0];

log.info("Gyro angle/rate: {}", value);

if(value >= 90){

//Sound.getInstance().beep();

log.info("Rotated 90 degrees");

break;

}

iterationCounter++;

if(iterationCounter >= 100){

break;

}

Delay.msDelay(HALF\_SECOND);

}

}

}

## 1. Introduction

The digital EV3 Gyro Sensor measures the robot’s rotational motion and changes in its orientation.



Gyroscope Features:

1. Two separate outputs per axis for high-speed gaming applications and lower-speed menu navigation: 2000°/s full scale range (high-speed gaming) 440°/s full scale range (pointing) • Low bias drift over temperature • On-chip temperature sensor
2. Temperature sensor Integrated amplifiers & low-pass filters.
3. Dual-mass, vibratory MEMS gyroscope offers superior vibration rejection over a wide frequency range. The dual-mass design inherently rejects any signal caused by linear acceleration.
4. 10,000 g shock tolerance The Rate-Out of the gyro is not ratiometric to the supply voltage. The scale factor is calibrated at the chip foundry and is nominally independent of supply voltage.

## 2. How to use the sensor

### 2.1. Read the angle

package ev3dev.sensors.ev3;

import ev3dev.sensors.Battery;

import lejos.hardware.port.SensorPort;

import lejos.robotics.SampleProvider;

import lejos.utility.Delay;

public class GyroSensorDemo {

//Robot Configuration

private static EV3GyroSensor gyroSensor = new EV3GyroSensor(SensorPort.S1);

//Configuration

private static int HALF\_SECOND = 500;

public static void main(String[] args) {

final SampleProvider sp = gyroSensor.getAngleMode();

int value = 0;

//Control loop

final int iteration\_threshold = 20;

for(int i = 0; i <= iteration\_threshold; i++) {

float [] sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

value = (int)sample[0];

System.out.println("Iteration: " + i);

System.out.println("Gyro angle: " + value);

Delay.msDelay(HALF\_SECOND);

}

System.out.println(Battery.getInstance().getVoltage());

}

}

### 2.2. Use the rate mode

package ev3dev.sensors.ev3;

import lejos.hardware.port.SensorPort;

import lejos.robotics.SampleProvider;

public class GyroSensorDemo2 {

//Robot Configuration

private static EV3GyroSensor gyroSensor = new EV3GyroSensor(SensorPort.S1);

public static void main(String[] args) {

SampleProvider sp = gyroSensor.getRateMode();

float[] sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

sp = gyroSensor.getAngleMode();

sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

sp = gyroSensor.getAngleAndRateMode();

sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

gyroSensor.reset();

}

}

### 2.3. Detect that your robot turn 90 degrees

package ev3dev.sensors.ev3;

import lejos.hardware.port.SensorPort;

import lejos.robotics.SampleProvider;

import lejos.utility.Delay;

import org.slf4j.Logger;

public class GyroSensorDemo3 {

private static final Logger log = org.slf4j.LoggerFactory.getLogger(GyroSensorDemo3.class);

//Robot Configuration

private static EV3GyroSensor gyroSensor = new EV3GyroSensor(SensorPort.S1);

//Configuration

private static int HALF\_SECOND = 500;

public static void main(String[] args) {

final SampleProvider sp = gyroSensor.getAngleAndRateMode();

int value = 0;

int iterationCounter = 0;

//Control loop

while(true){

float [] sample = new float[sp.sampleSize()];

sp.fetchSample(sample, 0);

value = (int)sample[0];

log.info("Gyro angle/rate: {}", value);

if(value >= 90){

//Sound.getInstance().beep();

log.info("Rotated 90 degrees");

break;

}

iterationCounter++;

if(iterationCounter >= 100){

break;

}

Delay.msDelay(HALF\_SECOND);

}

}

}

