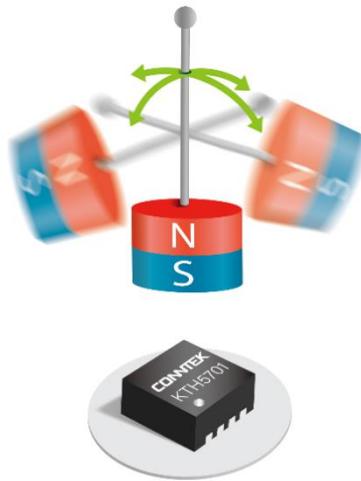


3D摇杆应用笔记



关于本文档

范围和目的

本文简要介绍了昆泰芯3D磁传感器在摇杆方面的应用和评估算法。

目录

1 3D磁传感器的摇杆应用	3
1.1 3D摇杆测量原理	3
1.2 算法说明.....	5
1.3 低功耗实现方法	7

1 3D磁传感器的摇杆应用

KTH5701是一款低功耗数字输出的3D霍尔芯片，能够测量来自X，Y，Z三个方向的磁场。将磁铁安装在摇杆上的合理位置后，当用户移动摇杆时，摇杆上安装的磁铁也随着一起运动。因此芯片可以根据采样回三轴磁场信息，通过软件建模还原出摇杆的运动信息。用户可选择SPI或I2C与芯片进行数据交互。

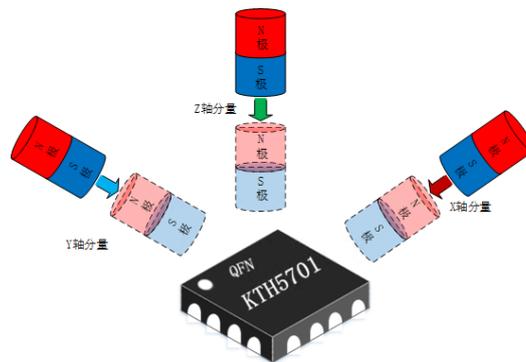
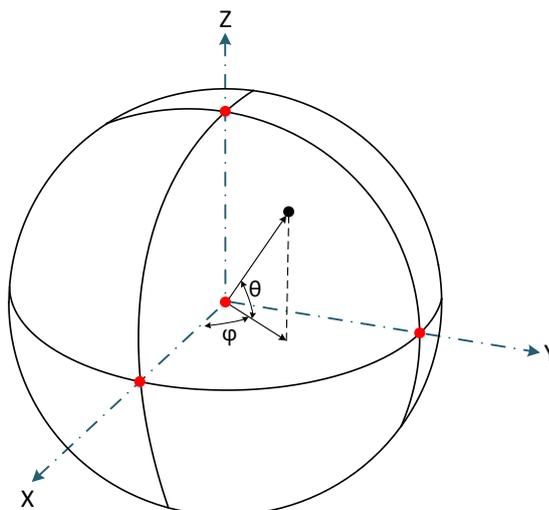


图1 三轴磁场测量敏感方向

1.1 3D摇杆测量原理

如果需要芯片检测到的磁场来反馈摇杆的运动轨迹，那么首先应该做到，芯片检测到的磁感应强度与摇杆的位置一一对应，那么我们就需要有参数来实现这个一一对应的关系。由磁仿真结果可以得知当摇杆晃动的时候，芯片检测到的磁感应强度（磁场矢量）与X轴，Y轴，Z轴会有一个固定的夹角。如图所示。



以芯片上的磁感应区域（Die）为原点建立坐标系。磁感应强度会与坐标轴Z轴形成一个夹角 θ ，该磁场矢量在XY平面上的投影与X轴也会形成一个夹角 φ 。由这两个夹角就可以确定的磁矢量的方向，再由 B 来确定该磁矢量的大小。所以 θ φ B 这三个参数可以确定摇杆的空间位置。则可以得到：

$$X_B = B \cdot \cos \theta \cdot \cos \varphi$$

$$Y_B = B \cdot \cos \theta \cdot \sin \varphi$$

那么怎么样可以得到 θ φ B 这三个参数呢？

由以下公式计算。

$$\varphi = \arctan \frac{B_y}{B_x} = \text{ATAN2}(B_x, B_y)$$

$$\theta = \arctan \frac{B_{xy}}{B_z} = \arctan \frac{\sqrt{B_x^2 + B_y^2}}{B_z} = \text{ATAN2}(B_{xy}, B_z)$$

$$B = \sqrt{B_x^2 + B_y^2 + B_z^2}$$

上述参数说明：

B_x ：芯片输出的X轴磁感应强度检测值。

B_y ：芯片输出的Y轴磁感应强度检测值。

B_z ：芯片输出的Z轴磁感应强度检测值。

X_B ：由三轴磁场值计算所得的坐标值。

Y_B ：由三轴磁场值计算所得的坐标值。

X_{output} ：用户所需的坐标值。

Y_{output} ：用户所需的坐标值。

B ：当前坐标下芯片上检测到的总磁场。

B_{xy} ： B 在坐标系下XY平面上的投影的磁场强度。

1.2 算法说明（使用极角与方位角来标识摇杆运动）

示例程序给出了3D摇杆算法，用户只需要按照以下示例说明，就能迅速完成相关应用的开发与示例程序的移植。

Step 1.寄存器的配置说明

为了实现本应用的目标功能，首先需要对寄存器进行正确配置。

如下图所示

```
void KTH57XXRegInitial(void)
{
    KTH57XXWriteRegister (0x1430, 28);
    KTH57XXWriteRegister (0x0001, 29);
    KTH57XXWriteRegister (0x8000, 30);
}
```

寄存器30一定要配置为0x8000，其他配置方式不满足本示例程序的需求。

寄存器28与寄存器29，用户可以根据自己的需求进行调节，详见KTH5701datasheet

Step 2.芯片开启测量

```
/*第二步：开启睡眠唤醒模式*/
KTH57XXWakeupSleep(0X0f);
```

本实例程序选择了睡眠唤醒模式，即对芯片发送一次睡眠唤醒模式命令后，芯片会周期性的对三轴磁感应强度进行测量，不需要多次发送测量命令。但需要多次发送读取测量的命令，才能将持续测量的磁感应强度读回主控，即为如下图所示的函数。

```
KTH57XXDataRead(0X0f);
```

Step 3.处理芯片读回的测量数据

进入 1.2 中的读取函数我们可以看到，读取测量的函数，将读回的数据全部存储进 DataReadFrame 这个数组中。

```
uint8_t KTH57XXDataRead(uint8_t axis)
{
    uint8_t i;
    uint8_t com;
    uint8_t writeLen;
    uint8_t readLen = 1; //根据axis来计算一共需要读回多少测量数据, 存放在counter中

    com = DATA_READ | axis;
    writeLen = 1;

    for(i=0; i<4; i++)
    {
        readLen = readLen + ( axis & 0x01 ) * 2;
        axis = axis >> 1;
    }

    HAL_I2C_Master_Transmit(&hi2c1, IIC_ADDR, &com, writeLen, 0xff);
    HAL_I2C_Master_Receive(&hi2c1, IIC_ADDR, DataReadFrame, readLen, 0xff);

    return readLen;
}
```

所以接下来根据 KTH5701 的通信协议对 DataReadFrame 数组中的内容进行处理。

```
temp = (DataReadFrame[1] << 8) + DataReadFrame[2]; //temperature
angleAZ = ((DataReadFrame[3] << 8) + DataReadFrame[4]) / COEFANGLE / COEFRADIAN; //azimuth
magIntensXY = ((DataReadFrame[5] << 8) + DataReadFrame[6]) * COEFCORDIC / SENSXY; //magnetic flux density in the XY plane
magZ = ((DataReadFrame[7] << 8) + DataReadFrame[8] - 32768) / SENSZ; // Z-axis magnetic flux density
```

(DataReadFrame[1] << 8) + DataReadFrame[2] : 代表芯片测量到的温度

(DataReadFrame[3] << 8) + DataReadFrame[4] : 代表芯片测量到的 XY 平面角度,

即可以代表上文摇杆原理说明中的 φ

(DataReadFrame[5] << 8) + DataReadFrame[6] : 代表芯片测量到的 XY 平面磁感应强度, 即可以代表上文摇杆原理说明中的 B_{xy} 。

(DataReadFrame[7] << 8) + DataReadFrame[8] : 代表芯片测量到的 Z 轴磁感应强度, 即可以代表上文摇杆原理说明中的 B_z 。

Step 4.坐标计算

将以上从芯片直接读回的数据, 转换为统一单位的磁感应强度值 (单位: 高斯)。

```
temp = (DataReadFrame[1] << 8) + DataReadFrame[2]; //temperature
angleAZ = ((DataReadFrame[3] << 8) + DataReadFrame[4]) / COEFANGLE / COEFRADIAN; //azimuth
magIntensXY = ((DataReadFrame[5] << 8) + DataReadFrame[6]) * COEFCORDIC / SENSXY; //magnetic flux density in the XY plane
magZ = ((DataReadFrame[7] << 8) + DataReadFrame[8] - 32768) / SENSZ; // Z-axis magnetic flux density
```

magIntensXY, magZ。两个变量即为使用 sensitivity 处理后的数据。

angleAZ 为根据 KTH5701 通信协议处理后的弧度值。

在本单片机的库函数中, sin cos 函数使用的是弧度值而不是角度值, 所以将 angleAZ 处理为弧度值, 希望用户能够根据所用主控的库函数进行调整。

如果用户所使用的库函数中正余弦曲线输入变量, 被定义为角度值。

则替换为以下计算方式。

```
angleAZ = (( DataReadFrame[3] << 8) + DataReadFrame[4]) / COEFANGLE; //azimuth
```

Step 5. 计算摇杆所处位置的总磁感应强度与极角。

```
magAmount = sqrt( pow ( magIntensXY, 2) + pow ( magZ , 2 ) );  
anglePolar = atan2( magZ , magIntensXY); //polar
```

其中 magAmount 为摇杆原理说明中的 B ， anglePolar 为摇杆原理说明中的 θ

Step 6. 计算当前摇杆坐标

根据 B ， φ ， θ 计算得到当前摇杆的 XY 平面投影值。

```
Xcode = magAmount * cos( anglePolar ) * cos( angleAZ ) * SENSXY;  
Ycode = magAmount * cos( anglePolar ) * sin( angleAZ ) * SENSXY;
```

我们推荐直接使用 B ， φ ， θ 来标识当前摇杆的位置，但大多数用户希望得到 XY 平面的投影坐标，因此在 1.6 中对 XY 平面坐标投影计算。

1.3 算法说明（使用磁场比率来标识摇杆的运动）

1.2 中的算法涉及正余弦函数与浮点数计算。以下给出一种更加节约资源的计算方式，同样能够标识标识摇杆的运动动作。

重复 1.2 中的 step 1. 2. 3. 4. 计算过程后

Step 5. 直接通过芯片输出的值计算平面坐标投影值。

```
Xcode = magIntensXY * cos( angleAZ ) / magZ;  
Ycode = magIntensXY * sin( angleAZ ) / magZ;
```

1.4 低功耗应用及系统唤醒实现方法

许多用户对系统处于不工作状态时，对本芯片及系统有低功耗的需求。KTH5701 芯片集成有一种测量模式叫做唤醒睡眠模式，即在这种模式下，芯片能够检测到外部磁感应强度的变化，并且在检测到的变化量达到设定阈值时，会拉高芯片的 INT 脚，并且功耗低于持续感应模式。

KTH5701 的功耗与滤波参数的配置，使用时主控采用的测量模式，测量时的待机时长等等参数有关，以下对如何在芯片的使用过程中，调节功耗进行说明。

滤波参数设置的越大，芯片输出的测量数据就越精确，功耗越高。

与滤波参数有关的寄存器有：寄存器 28 中的 tempOsr, magnOsr, digCtrl

待机时长设置的越大，芯片的功耗越低，但测量频率也随之降低。

待机时长的设置寄存器为：寄存器 29 中的 measTime

1.4.1 三种测量模式功耗模式说明

在相同的测量频率下，不同模式的测量功耗如下所示。芯片配置为：

位名	temp0sr	magn0sr	digCtrl	measTime
配置值（十六进制）	1	1	1	1

测量模式	采样频率	功耗
持续感应		0.314mA
唤醒睡眠	50Hz	0.256mA
单次测量		0.255mA

如上表所示，在相同的采样频率下，单次测量模式下所产生的功耗最低。以下对三种测量模式的区别做出说明。

如将芯片设置为相同的采样频率，比如50Hz，即两次测量的时间间隔为20ms，并且输出测量数据。但在某些滤波深度寄存器的设定下，芯片完成一次测量的时间并不需要20ms，那么完成一次测量后这个时候芯片处于不测量的状态。以上三种模式功耗有所不同，就是这个不测量状态下，芯片的功耗会有所不同。

这个不进行测量的状态，如果芯片被设置为持续感应模式或者唤醒睡眠模式，我们称之为待机时间，对应的功耗为KTH57系列数据手册中写的待机功耗（ $I_{DD,STBY}$ 或 $I_{DD,WAKE_STBY}$ ）。即持续感应模式下的待机功耗为 $61.7 \mu A$ （ $I_{DD,STBY}$ ），唤醒睡眠模式下的待机功耗为 $2.4 \mu A$ （ $I_{DD,WAKE_STBY}$ ）。芯片处于待机状态时仍然保留了一部分功能，比如唤醒睡眠模式下，如果磁场阈值达到设定值，会对芯片的INT脚进行拉起等等。而单次测量模式下，芯片进行一次测量完毕后，就进入空闲状态，这个状态和芯片上电，读取完OTP内容后的稳定状态是一致的，此时对应的功耗为KTH57系列数据手册中写的空闲状态功耗（ $I_{DD,IDLE}$ ） $1.4 \mu A$ 。

所以在50HZ测量频率的设定下，即芯片需要在20ms内完成一次测量的设定下，芯片处于正在测量时的功耗，三种模式都相同。而芯片处于不测量的状态时，单次测量模式的功耗 < 唤醒睡眠模式的功耗 < 持续感应模式的功耗。

1.4.2 三种模式的应用方式推荐

由1.3.1可知，持续感应和唤醒睡眠两种模式功耗更高，但是使用这两种模式时，芯片只需要发送一次使芯片开始测量的命令后，芯片就会按照设定的频率自主对磁感应强度进行测量，后续主控只需要设定读取测量命令的频率即可。

单次测量状态下，芯片的功耗会更低，但每完成一次测量之后就进入空闲，不会自主的按照一定的频率进行测量，也就是说主控要同时控制发送单次测量模式命令和读取测量命令的频率。

唤醒睡眠模式有一个特殊功能，所谓的唤醒睡眠并不是指芯片会睡眠，而是芯片可以通过检测外界磁场的变化来产生一个脉冲，从而唤醒外部系统。比如在某些应用场景下，系统和摇杆都不需要被使用，则可以让系统进入休眠状态，并设置一个外部中断口检测外部来的脉冲。当摇杆再次被使用时，芯片周围的磁感应强度一定会发生变化，当这个变化量满足芯片寄存器中设定的阈值时，则芯片的INT脚会产生一个高电平，即可以用这个高电平来唤醒休眠的系统。

假设以下应用场景，用户使用了一段时间摇杆后，把产品放置一旁，经过一段时间后再次使用摇杆。那么可以在用户使用摇杆时，采用单次测量模式，根据自己的需要修改测量频率，摇杆长时间不动后，将芯片设置为唤醒睡眠模式，并且将系统休眠，这样会大大降低产品处于静置状态时的功耗。然后用户再次移动摇杆，芯片检测到外部磁场发生变化，INT脚产生高电平信号唤醒系统，将芯片退出唤醒睡眠模式，进入单次测量。然后重复上述过程。

需要注意的是，芯片处于持续感应模式或唤醒睡眠模式时，如果需要对芯片寄存器进行修改，或进入其他模式，都需要先发送空闲模式命令，否则无法修改。

1.4.3 各类滤波参数对功耗的影响

滤波参数设置的越大，芯片输出的测量数据就越精确，功耗越高。

与滤波参数有关的寄存器有：寄存器 28 中的 tempOsr, magnOsr, digCtrl 降低芯片的测量频率同时也是一种减少功耗的有效办法。

下面以单次测量模式为例，给出不同测量频率，或不同滤波参数下的功耗比例。

测量模式	magnOsr	tempOsr	digCtrl	采样频率	平均功耗
单次测量	1	1	2	50HZ	0.36mA

Application Note	低功耗、高精度3D Hall传感器	
2		0.62mA
3		1.14mA
1		0.72mA
2	100HZ	1.23mA
3		2.28mA
1		1.08mA
2	150HZ	1.85mA
3		3.42mA
1		1.44mA
2	200HZ	2.47mA
3		-
1		1.79mA
2	250HZ	3.09mA
3		-
1		2.15mA
2	300HZ	3.711mA
3		-
1		2.51mA
2	350HZ	-
3		-
1		2.86mA
2	400HZ	-
3		-
1		3.23mA
2	450HZ	-
3		-
1		3.58mA
2	500HZ	-
3		-

注：- 代表当前设定下，无法满足所需的频率要求。

由以上表格可知，测量频率越快，或者滤波深度设置越大，芯片的功耗越高。