

课 程 \_\_\_\_传感与检测技术\_\_\_\_\_\_

项 目 基于CC2530实现惯导传感器MPU6050数据采集

专 业 \_\_\_\_\_\_交通运输工程\_\_\_\_\_\_\_\_\_\_

班 级 \_\_\_\_\_\_交通运输工程2301\_\_\_\_\_

学 号 \_\_\_\_\_2231801015\_\_\_\_\_\_\_\_\_

姓 名 \_\_\_\_\_\_\_\_翁泽康\_\_\_\_\_\_\_\_\_\_\_\_

日 期 \_\_\_\_\_\_2024.1.16\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| 实  验  任  务 | 基于CC2530实现惯导传感器MPU6050的数据采集 |
| 实  验  报  告  内  容 | 1.关键技术和传感器介绍：  1.1 CC2530：CC2530是由德州仪器（Texas Instruments）研发的一款低功耗无线系统芯片。该芯片遵循IEEE 802.15.4标准，并内置了强大的2.4GHz无线收发功能。其设计针对低功耗无线通信和物联网（IoT）应用，为这些领域提供了可靠的无线连接解决方案。CC2530的特性使其成为在资源受限的环境中进行有效通信的理想选择，满足了对低功耗和高性能的需求。  1.2 MPU6050：MPU6050是一种惯性测量单元（IMU），集成了三轴加速度计和三轴陀螺仪。它能够准确测量物体的加速度和角速度，提供高质量的传感器数据。通过I2C总线，MPU6050可以与主控制器进行通信，便捷地连接到各类微控制器和处理器。其高度集成和卓越的准确度使其成为广泛应用于物联网、机器人、移动设备、无人机以及虚拟现实等领域的理想选择。MPU6050还具备灵活的配置选项和中断功能，为开发者提供了更多自定义设置和响应的可能性。  1.3 I2C：I2C，全称Inter-Integrated Circuit，又被称为IIC（Inter-IC）或TWI（Two-Wire Interface），是一种串行通信协议。它是一种简单而高效的通信接口，被广泛用于连接多个芯片或模块之间的通信。 I2C协议使用两根线，一根是数据线（SDA），另一根是时钟线（SCL），使得多个设备可以在同一总线上进行通信。这种通信方式简化了硬件连接，使得不同设备之间可以方便地进行数据交换。1.4MPU6050的寄存器：  1.4.1螺旋仪配置寄存器    该寄存器我们只关心FS\_SEL[1:0]这两个位，用于设置陀螺仪的满量程范围：0，±250°/s；1，±5 00°/s；2，±1000°/s；3，±2000°/s；我们一般设置为2，即±1000°/S，  1.4.2加速度寄存器    该寄存器我们只关心AFS\_SEL[1:0]这两个位，用于设置加速度传感器的满量程范围：0，±2g；1，±4g；2，±8g；3，±16g；我们一般设置为2，即±8g，  2.实验目的  本设计中，通过采用I2C通信方式，实现了CC2530与HMC5883L的有效通信。该通信过程包括通过程序读取HMC5883L内部寄存器的值，并进行相应的运算，得到三轴的角速度和加速度信息。这些数据随后通过串口传输至电脑，通过上位机软件实现了对CC2530读取的三轴角速度和加速度的实时监测。这一设计方案通过串口通信将嵌入式系统与上位机连接，为实时监测提供了可靠的数据传输通道，有助于用户及时获取和分析传感器数据。  3.实验步骤：  3.1操作步骤：  1.在本设计中，首先通过仔细研读CC2530芯片手册，了解了HMC5883L寄存器的读取方式和通讯协议。然后，借助IAR集成开发环境，编写了相应的程序，并生成了hex文件。接着，通过烧录线将电脑与CC2530连接，使用烧录软件SmartRF Flash Programmer将编写好的程序成功烧写至CC2530芯片。  进一步，通过I2C通信协议，将MPU6050与CC2530进行连接，并利用USB转TTL线将电脑与CC2530建立通信通道。最后，通过上位机串口调试助手(XCOM)，实时观察和监测MPU6050传感器读取的三轴角速度和加速度数据。这一系列步骤实现了CC2530芯片与传感器的协同工作，通过串口通信方式将传感器数据传输至上位机，为后续数据分析和应用提供了可靠的基础。  3.2读取MPU6050数值的步骤：  3.2.1流程图：    CC2530先启动I2C,对MPU6050 的寄存器进行配置，其中将陀螺仪的量程设置为 +/- 1000°/s，加速度计的量程设置为 +/- 8g。再通过函数依次读出x轴、y轴、z轴的加速度和角速度，再通过精度将其换算成以g为单位的加速度和以°/s为单位的角速度，最后通过上位机显示三轴的角速度和加速度的数据。  3.2.2关键程序：  #include <stdio.h>  // 定义I2C通信引脚  #define SCL P1\_3  #define SDA P1\_2  // 陀螺仪配置寄存器和加速度计配置寄存器  #define MPU6050\_GYRO\_CONFIG 0x10  #define MPU6050\_ACCEL\_CONFIG 0x10  // 从MPU6050陀螺仪加速度计模块读取指定寄存器的数据  uint MPU6050\_ReadReg(uint reg) {  // 实现具体的读取寄存器的功能  // ...  }  // 从MPU6050陀螺仪加速度计模块读取X、Y、Z轴的加速度计和陀螺仪的原始数据  void MPU6050\_GetData(int\* AX, int\* AY, int\* AZ, int\* GX, int\* GY, int\* GZ) {  uint DataH, DataL;  DataH = MPU6050\_ReadReg(MPU6050\_ACCEL\_XOUT\_H);  DataL = MPU6050\_ReadReg(MPU6050\_ACCEL\_XOUT\_L);  \*AX = (DataH << 8 | DataL);  // 读取Y、Z轴的数据，类似处理  // ...  DataH = MPU6050\_ReadReg(MPU6050\_GYRO\_XOUT\_H);  DataL = MPU6050\_ReadReg(MPU6050\_GYRO\_XOUT\_L);  \*GX = (DataH << 8 | DataL);  // 读取Y、Z轴的数据，类似处理  // ...  }  // 结构体用于存储加速度计和陀螺仪的物理值  struct MPU6050\_Data {  float accel\_x;  float accel\_y;  float accel\_z;  float gyro\_x;  float gyro\_y;  float gyro\_z;  };  // 从MPU6050陀螺仪加速度计模块读取原始的加速度计和陀螺仪数据，并将其乘以比例尺转换为实际的物理值  MPU6050\_Data MPU6050\_GetDataWithScale() {  int AX, AY, AZ, GX, GY, GZ;  MPU6050\_GetData(&AX, &AY, &AZ, &GX, &GY, &GZ);  float accel\_scale = 8.0 / 32767.0; // 加速度计量程为 +/- 8g  float gyro\_scale = 1000.0 / 32767.0; // 陀螺仪量程为 +/- 1000°/s  MPU6050\_Data data;  data.accel\_x = AX \* accel\_scale;  data.accel\_y = AY \* accel\_scale;  data.accel\_z = AZ \* accel\_scale;  data.gyro\_x = GX \* gyro\_scale;  data.gyro\_y = GY \* gyro\_scale;  data.gyro\_z = GZ \* gyro\_scale;  return data;  }  // 初始化串口通信  void InitUart() {  // 实现具体的串口初始化操作  // ...  }  // 主函数，通过循环读取MPU6050模块的数据并通过串口打印输出  int main(void) {  InitUart(); // 初始化串口  while(1) {  MPU6050\_Data data = MPU6050\_GetDataWithScale();  printf("AX: %.2f g\n AY: %.2f g\n AZ: %.2f g\n", data.accel\_x, data.accel\_y, data.accel\_z);  printf("GX: %.2f °/s\n GY: %.2f °/s\n GZ: %.2f °/s\n", data.gyro\_x, data.gyro\_y, data.gyro\_z);  }  return 0;  } |
| 实  验  报  告  内  容 | 4结论：  通过在串口调试助手(XCOM)上对数据结果的观察，本实验能准确的通过CC2530芯片读取了MPU6050传感器的x轴，y轴，z轴加速度和角速度的数据。  IMG_256  通过进一步分析这些数据，我们可以检测物体的加速度变化情况，从而推测其运动方式。通过计算加速度在三个轴上的变化，我们能够了解物体的运动方向和速度。此外，通过对陀螺仪数据的处理，我们能够计算出物体的角速度，从而了解物体的旋转情况。  这样的信息对于许多应用领域都非常有用。例如，在姿态控制中，我们可以通过监测加速度和角速度来判断设备的方向和旋转情况。在运动追踪中，这些数据可以帮助我们跟踪物体的运动轨迹。在虚拟现实中，了解用户的头部姿态和运动可以实现更加沉浸式的体验。  总体而言，通过对加速度和角速度数据进行深入分析，我们可以获取关于物体运动和旋转的详细信息，为各种应用场景提供有力的支持。 |
| 教  师  评  定 |  |