

BMI085是系统级封装惯性测量单元，可提供精确的加速度和角速率测量。由于采用了系统级封装方法（两个传感器位于单个封装中），因此陀螺仪和加速度数据是以非同步方式获取的。但是，加速度计和陀螺仪之间的同步很容易实现。本文档介绍了如何在典型应用（例如增强现实或虚拟现实）中实现加速度计和陀螺仪之间的同步。

为了在BMI085上实现数据同步，需要将来自BMI085陀螺仪的数据就绪中断信号连接到BMI085加速度计的中断引脚之一（可以将其配置为输入引脚）。考虑到传感器的群时延，加速度计的内部信号处理单元使用来自陀螺仪的数据就绪信号来同步(synchronize)和内插(Interpolate)加速度计的数据。然后，加速度计部分可以将可用数据通知主机。使用此技术，可以实现同步数据并以2 kHz的ODR提供加速度计数据。

注意：数据同步是为需要高带宽的应用程序而设计的，而BMI085旨在用于此应用程序，但它也可与BMI088一起使用。然而，对于某些应用，具有非常低的带宽，并且对于加速度计和陀螺仪具有不同的带宽设置。这时，同步功能没有意义。

概念

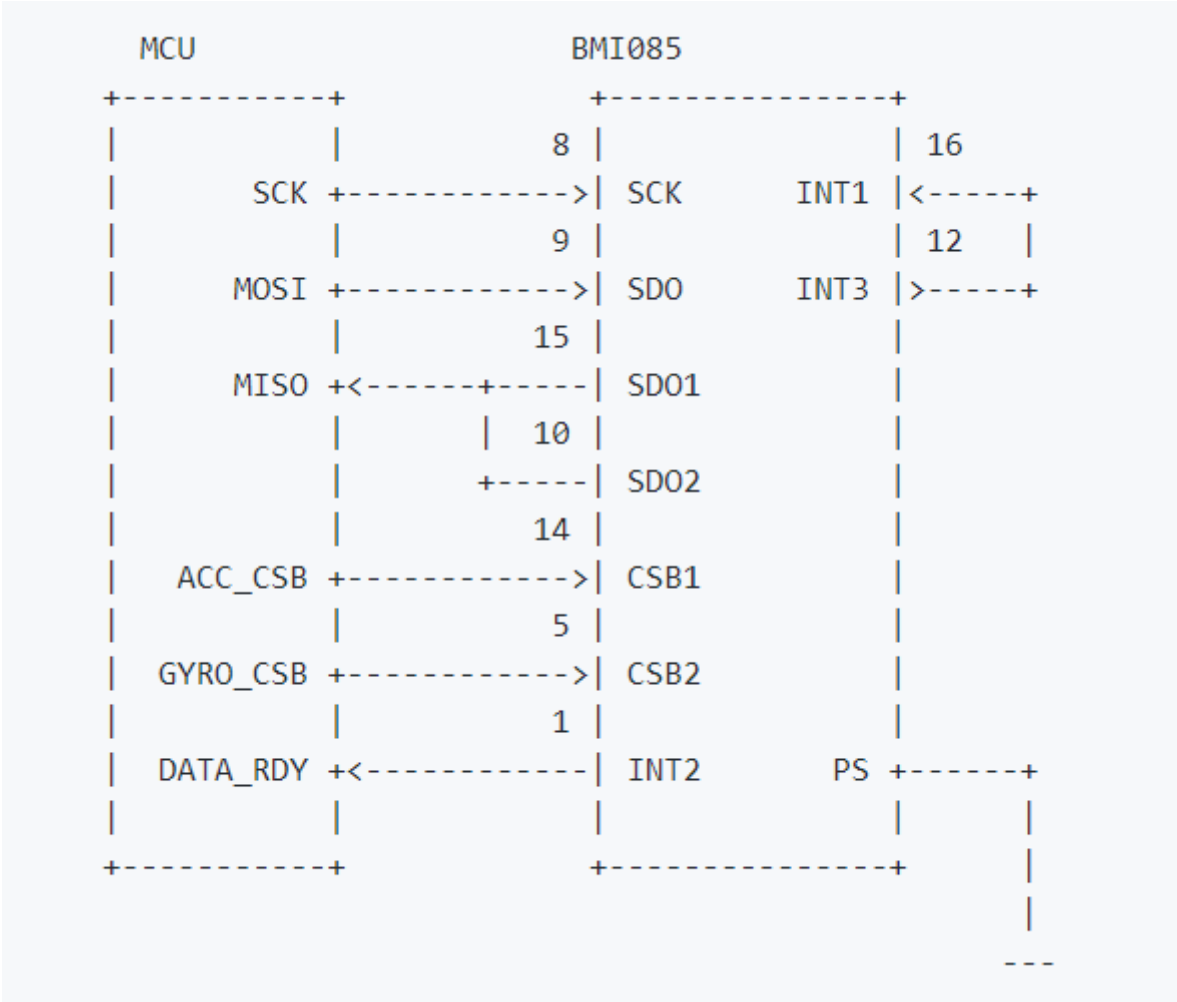
同步数据意味着陀螺仪和加速度计数据的采集是同时发生的，并且信号具有相同的传播时间（propagation time）。从运动到寄存器读出之间的时间取决于主要由信号滤波路径引起的物理传播时间。加速度计和陀螺仪数据之间到公共时间点和公共组延迟之间的同步可以使用以下各节中描述的方法来实现。

BMI085的硬件中断引脚（INT1 / INT3）用于数据同步，必须将其连接。中断引脚INT2可用于BMI085将数据就绪通知给主机。

技术实现

数据同步功能需要PCB上的传感器的物理中断引脚连接以及BMI085的特殊配置。要求和步骤如下所述。

连接图



对于对延迟至关重要的多传感器应用，建议使用SPI接口以最快的速度读取传感器数据（建议SPI时钟频率>2MHz）。

配置BMI085进行数据同步

包含bmi08x头文件

01.

#include "bmi08x.h"

复制代码

包含bmi085变量的头文件

01.

#include "bmi085.h"

复制代码

在bmi08x_defs.h中启用以下宏以使用BMI085传感器功能

01.

/** \name enable bmi085 sensor */

02.

#ifndef BMI08X_ENABLE_BMI085

03.

#define BMI08X_ENABLE_BMI085 1

04.

#endif

复制代码

要初始化BMI085进行数据同步，应创建bmi08x结构体的实例。用户需要在结构体中更新以下参数：

参数	描述
accel_id	I2C接口的设备地址（如果intf是BMI08X_I2C_INTF）
gyro_id	I2C接口的设备地址（如果intf为BMI08X_I2C_INTF）
intf	I2C或SPI
read	通过I2C / SPI接口进行读取的函数指针
write	通过I2C / SPI接口进行写入的函数指针
delay_ms	延时函数指针

通过SPI接口初始化

以下示例代码是简化代码，未添加返回值检查，以使其更易于阅读。

```
01. struct bmi08x_dev bmi08xdev = {
02.     .accel_id = 0,
03.     .gyro_id = 0,
04.     .intf = BMI08X_SPI_INTF,
05.     .read = user_spi_read,
06.     .write = user_spi_write,
07.     .delay_ms = user_delay_milli_sec
08. };
09.
10. int16_t rslt;
11.
12. /* Initialize bmi085 sensors (accel & gyro)*/
13. rslt = bmi085_init(&bmi08xdev);
14. /* Reset the accelerometer and wait for 1 ms - delay taken care inside the function */
15. rslt = bmi08a_soft_reset(&bmi08xdev);
16.
17.
18. /*! Max read/write length (maximum supported length is 32).
19. To be set by the user */
20. bmi08xdev.read_write_len = 32;
21. /*set accel power mode */
22. bmi08xdev.accel_cfg.power = BMI08X_ACCEL_PM_ACTIVE;
23. rslt = bmi08a_set_power_mode(&bmi08xdev);
24.
25. bmi08xdev.gyro_cfg.power = BMI08X_GYRO_PM_NORMAL;
26. bmi08g_set_power_mode(&bmi08xdev);
27.
28. /* API uploads the bmi08x config file onto the device and wait for 150ms
29. to enable the data synchronization - delay taken care inside the function */
30. rslt = bmi085_apply_config_file(&bmi08xdev);
31.
32. /*assign accel range setting*/
33. bmi08xdev.accel_cfg.range = BMI085_ACCEL_RANGE_4G;
34. /*assign gyro range setting*/
35. bmi08xdev.gyro_cfg.range = BMI08X_GYRO_RANGE_2000_DPS;
```

```
36.  /*! Mode (0 = off, 1 = 400Hz, 2 = 1kHz, 3 = 2kHz) */
37.  sync_cfg.mode = BMI08X_ACCEL_DATA_SYNC_MODE_2000HZ;
38.  rslt = bmi085_configure_data_synchronization(sync_cfg, &bmi08xdev);
39.
40.
41.  /*set accel interrupt pin configuration*/
42.  /*configure host data ready interrupt */
43.  int_config.accel_int_config_1.int_channel = BMI08X_INT_CHANNEL_1;
44.  int_config.accel_int_config_1.int_type = BMI08X_ACCEL_SYNC_INPUT;
45.  int_config.accel_int_config_1.int_pin_cfg.output_mode = BMI08X_INT_MODE_PUSH_PULL;
46.  int_config.accel_int_config_1.int_pin_cfg.lvl = BMI08X_INT_ACTIVE_HIGH;
47.  int_config.accel_int_config_1.int_pin_cfg.enable_int_pin = BMI08X_ENABLE;
48.
49.  /*configure Accel synchronization input interrupt pin */
50.  int_config.accel_int_config_2.int_channel = BMI08X_INT_CHANNEL_2;
51.  int_config.accel_int_config_2.int_type = BMI08X_ACCEL_SYNC_DATA_RDY_INT;
52.  int_config.accel_int_config_2.int_pin_cfg.output_mode = BMI08X_INT_MODE_PUSH_PULL;
53.  int_config.accel_int_config_2.int_pin_cfg.lvl = BMI08X_INT_ACTIVE_HIGH;
54.  int_config.accel_int_config_2.int_pin_cfg.enable_int_pin = BMI08X_ENABLE;
55.
56.  /*set gyro interrupt pin configuration*/
57.  int_config.gyro_int_config_1.int_channel = BMI08X_INT_CHANNEL_3;
58.  int_config.gyro_int_config_1.int_type = BMI08X_GYRO_DATA_RDY_INT;
59.  int_config.gyro_int_config_1.int_pin_cfg.enable_int_pin = BMI08X_ENABLE;
60.  int_config.gyro_int_config_1.int_pin_cfg.lvl = BMI08X_INT_ACTIVE_HIGH;
61.  int_config.gyro_int_config_1.int_pin_cfg.output_mode = BMI08X_INT_MODE_PUSH_PULL;
62.
63.  int_config.gyro_int_config_2.int_channel = BMI08X_INT_CHANNEL_4;
64.  int_config.gyro_int_config_2.int_type = BMI08X_GYRO_DATA_RDY_INT;
65.  int_config.gyro_int_config_2.int_pin_cfg.enable_int_pin = BMI08X_DISABLE;
66.  int_config.gyro_int_config_2.int_pin_cfg.lvl = BMI08X_INT_ACTIVE_HIGH;
67.  int_config.gyro_int_config_2.int_pin_cfg.output_mode = BMI08X_INT_MODE_PUSH_PULL;
68.
69.  /* Enable synchronization interrupt pin */
70.  rslt = bmi085_set_data_sync_int_config(&int_config, &bmi08xdev);
```

[复制代码](#)

读取原始加速度数据

```
01.  /* Declare an instance of the sensor data structure for accel */
02.  static struct bmi08x_sensor_data accel_bmi085;
03.
04.  rslt = bmi08a_get_data(&accel_bmi085, &bmi08xdev);
```

[复制代码](#)

读取同步数据

```
01.  /* Declare an instance of the sensor data structure for accel */
02.  static struct bmi08x_sensor_data accel_bmi085;
```

```
03.  /* Declare an instance of the sensor data structure for gyro */
04.  static struct bmi08x_sensor_data gyro_bmi085;
05.
06.  rslt = bmi085_get_synchronized_data(&accel_bmi085,&gyro_bmi085, &bmi08xdev);
```

复制代码