

MPU-6050 六軸感測器實驗

Rev.0.00
Sep 14, 2015

Introduction

我們介紹當下最流行的一款六軸感測器（三軸加速度計+三軸陀螺儀）：MPU-6050，該感測器廣泛用於四軸飛行器設計，具有非常廣泛的應用範圍。本章我們將使用 RX63T 來驅動 MPU-6050，讀取其感測資料。

Target Device

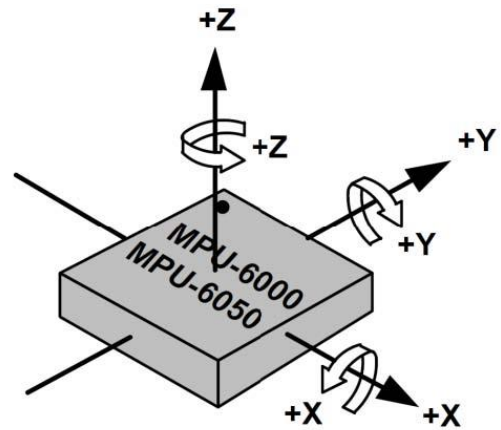
QB-RX63TH-21x

Contents

| | |
|--|-----------|
| 1. MPU-6050 簡介..... | 3 |
| 2. 範例程式目錄..... | 4 |
| 2.1 Workspace Structure..... | 4 |
| 2.1.1 RX63T_MPU6050 目錄..... | 4 |
| 2.1.2 BSP 目錄..... | 4 |
| 3. QB-RX63TH-21x 與周邊結線方式..... | 5 |
| 3.1 MPU-6050 (GY-521) - I2C Bus Interface (RIIC) | 5 |
| 3.2 Serial Port (UART to USB) - Serial Communications Interface (SCIc) | 5 |
| 3.3 結線示意圖 | 6 |
| 4. MPU6050 初始化的暫存器設定 | 7 |
| 4.1 主要暫存器 | 7 |
| 4.2 暫存器介紹與設置 | 7 |
| 4.2.1 PWR_MGMT_1..... | 7 |
| 4.2.2 SMPLRI_DIV..... | 8 |
| 4.2.3 CONFIG | 8 |
| 4.2.4 GYRO_CONFIG..... | 8 |
| 4.2.5 ACCEL_CONFIG | 8 |
| 5. 程式內容與流程 | 9 |
| 5.1 Main loop..... | 9 |
| 5.2 MPU-6050 initialize | 11 |
| 6. 操作流程..... | 12 |
| 6.1 啟動範例程式..... | 12 |
| 6.2 程式運行與結果 | 12 |

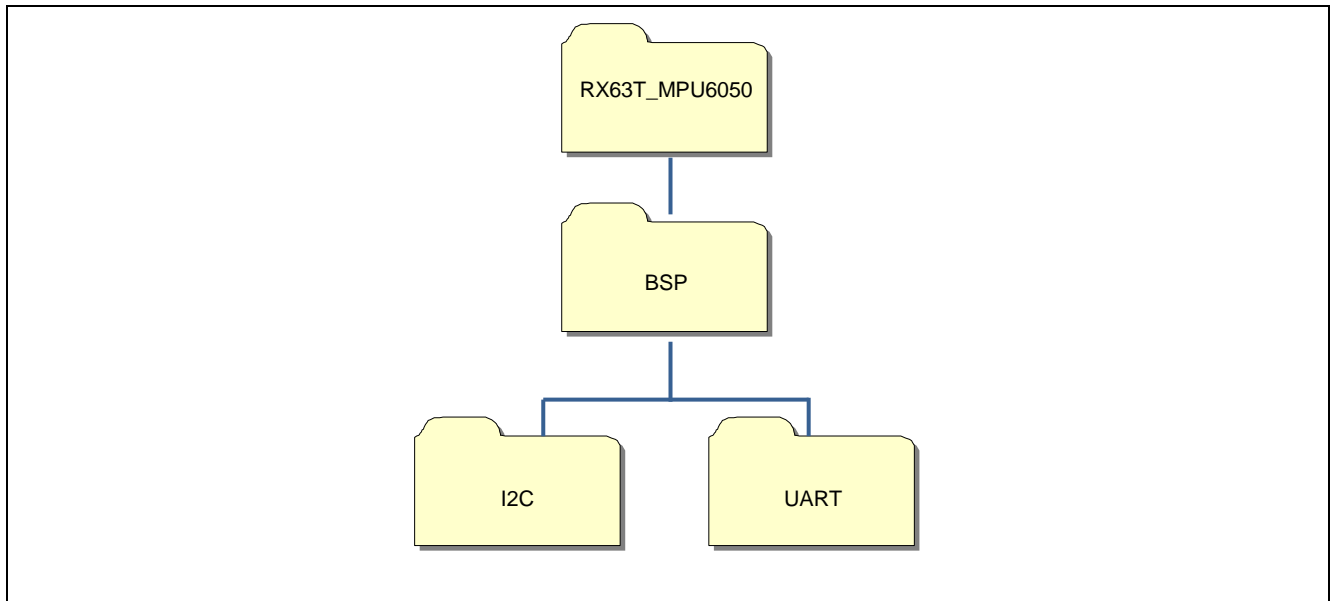
1. MPU-6050 簡介

16 位元三軸加速度器和三軸陀螺儀。
支援 400KHz IIC 匯流排通訊方式。
加速度和角速度測量範圍可程式化設計。
加速度的測量範圍 $\pm 2g$ 、 $\pm 4g$ 、 $\pm 8g$ 和 $\pm 16g$ 。
角速度測量範圍為 ± 250 、 ± 500 、 ± 1000 和 $\pm 2000^\circ/\text{sec}$ 。
內建數位運動處理器(DMP)可減少 MCU 融合演算資料的負荷。
輸出值為 16 位元數值的補數。
內建溫度感測器。
從機位址 0xD0



2. 範例程式目錄

2.1 Workspace Structure



RX63T_MPU6050 Workspace file structure

2.1.1 RX63T_MPU6050 目錄

這是專案主目錄，本目錄包含內容如下：

- 專案檔案與設定資料
- 主程式
- 硬體設定程式
- 系統定義檔: qbrx63th_if.h

2.1.2 BSP 目錄

本目錄為 MCU 周邊功能應用設定程式，內容如下：

- UART driver
- I2C driver

3. QB-RX63TH-21x 與周邊結線方式

RX63T 使用 I2C 介面 (RIIC) 連接 MPU-6050，接收 MPU-6050 回傳的感測結果，再通過序列埠 (SCIc) 上傳給電腦終端機。

3.1 MPU-6050 (GY-521) - I2C Bus Interface (RIIC)

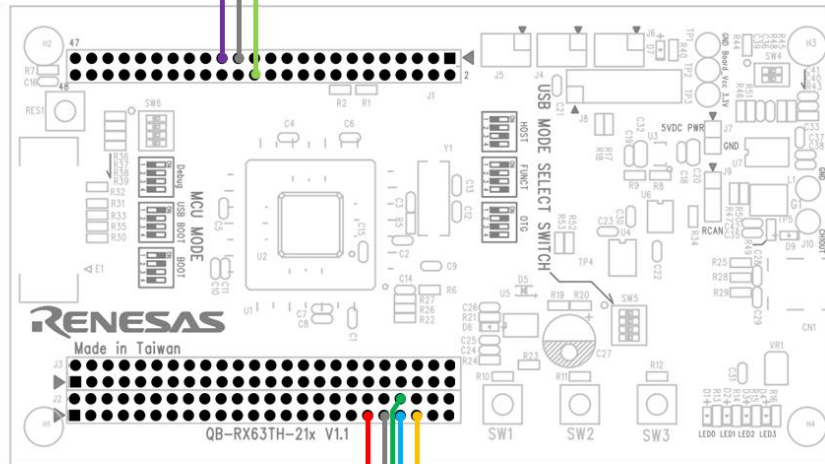
| Channel | Pin Name | Function | QB-RX63TH-21x Connector |
|---------|------------|-----------------------------|-------------------------|
| RIIC1 | P26 – SDA1 | RIIC1 serial data I/O pin | J2 Pin41 |
| | P25 – SCL1 | RIIC1 serial clock I/O pin. | J2 Pin 42 |

3.2 Serial Port (UART to USB) - Serial Communications Interface (SCIc)

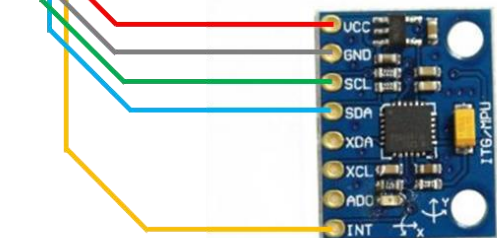
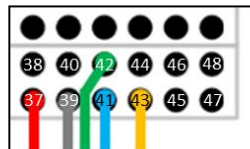
| Channel | Pin Name | Function | QB-RX63TH-21x Connector |
|---------|------------|---------------------------|-------------------------|
| SCI1 | PD5 – RXD1 | SCI1 receive data input | J1 Pin 26 |
| | PD3 – TXD1 | SCI1 transmit data output | J1 Pin 29 |

3.3 結線示意圖

| J1 Pin No. | MCU I/O Port & Func. |
|------------|----------------------|
| 26 | PD5-RXD1 |
| 27 | GND |
| 29 | PD3-TXD1 |



| J2 Pin No. | MCU I/O Port & Func. |
|------------|----------------------|
| 37 | Vcc – 5V |
| 39 | GND |
| 41 | P26 – SDA1 |
| 42 | P25 – SCK1 |
| 43 | P24 – IRQ4 |



4. MPU6050 初始化的暫存器設定

1. 啟動感測器；
2. 設置陀螺儀採樣頻率；
3. 設置低通濾波器頻率；
4. 設置陀螺儀自檢測和測量範圍；
5. 設置加速度自檢測、測量範圍；
6. 讀取感測值；

4.1 主要暫存器

| 名稱 | 地址 | 功能 |
|--------------|------|-----------------|
| PWR_MGMT_1 | 0x6B | 啟動感測器 |
| SMPLR_DIV | 0x19 | 設置陀螺儀採樣頻率 |
| CONFIG | 0x1A | 設置低通濾波器頻率 |
| GYRO_CONFIG | 0x1B | 設置陀螺儀自檢測和測量範圍 |
| ACCEL_CONFIG | 0x1C | 設置加速度自檢測和測量範圍 |
| ACCEL_XOUT_H | 0x3B | 存放 X 軸加速度的高 8 位 |
| ACCEL_XOUT_L | 0x3C | 存放 X 軸加速度的低 8 位 |
| ACCEL_YOUT_H | 0x3D | 存放 Y 軸加速度的高 8 位 |
| ACCEL_YOUT_L | 0x3E | 存放 Y 軸加速度的低 8 位 |
| ACCEL_ZOUT_H | 0x3F | 存放 Z 軸加速度的高 8 位 |
| ACCEL_ZOUT_L | 0x40 | 存放 Z 軸加速度的低 8 位 |
| TEMP_OUT_H | 0x41 | 存放溫度高 8 位元 |
| TEMP_OUT_L | 0x42 | 存放溫度低 8 位元 |
| GYRO_XOUT_H | 0x43 | 存放 X 軸角速度的高 8 位 |
| GYRO_XOUT_L | 0x44 | 存放 X 軸角速度的低 8 位 |
| GYRO_YOUT_H | 0x45 | 存放 Y 軸角速度的高 8 位 |
| GYRO_YOUT_L | 0x46 | 存放 Y 軸角速度的低 8 位 |
| GYRO_ZOUT_H | 0x47 | 存放 Z 軸角速度的高 8 位 |
| GYRO_ZOUT_L | 0x48 | 存放 Z 軸角速度的低 8 位 |

4.2 暫存器介紹與設置

4.2.1 PWR_MGMT_1

PWR_MGMT_1 寫入 0x00 完成感測器的啟動。

4.2.2 SMPLRI_DIV

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|----------------|--------------------|-----------------|------|------|------|------|------|------|------|
| 19 | 25 | SMPLRT_DIV[7:0] | | | | | | | |

$$\text{Sample Rate} = \text{Gyroscope Output Rate} / (1 + \text{SMPLRT_DIV})$$

當低通濾波器不致能 (即 DLPF_CFG = 0 or 7) 時，陀螺儀輸出頻率為 8kHz。

當低通濾波器致能時，陀螺儀輸出頻率為 1kHz (參考暫存器 0x1A)。

*加速度計的輸出速度為 1KHz

4.2.3 CONFIG

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|----------------|--------------------|------|------|-------------------|------|------|---------------|------|------|
| 1A | 26 | - | - | EXT_SYNC_SET[2:0] | | | DLPF_CFG[2:0] | | |

設定低通濾波截止頻率

| DLPF_CFG | Accelerometer (F _s = 1kHz) | | Gyroscope | | |
|----------|--|---------------|-------------------|---------------|----------------------|
| | Bandwidth (Hz) | Delay (ms) | Bandwidth (Hz) | Delay (ms) | F _s (kHz) |
| 0 | 260 | 0 | 256 | 0.98 | 8 |
| 1 | 184 | 2.0 | 188 | 1.9 | 1 |
| 2 | 94 | 3.0 | 98 | 2.8 | 1 |
| 3 | 44 | 4.9 | 42 | 4.8 | 1 |
| 4 | 21 | 8.5 | 20 | 8.3 | 1 |
| 5 | 10 | 13.8 | 10 | 13.4 | 1 |
| 6 | 5 | 19.0 | 5 | 18.6 | 1 |
| 7 | RESERVED | | RESERVED | | 8 |

4.2.4 GYRO_CONFIG

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|----------------|--------------------|-------|-------|-------|-------------|------|------|------|------|
| 1B | 27 | XG_ST | YG_ST | ZG_ST | FS_SEL[1:0] | | - | - | - |

XG_ST、YG_ST、ZG_ST 為 0 時陀螺儀在所在軸上不進行自檢測，為 1 時進行自檢測。

陀螺儀測量範圍

FS_SEL[0 : 1]=0 ±250°/s
 FS_SEL[0 : 1]=1 ±500°/s
 FS_SEL[0 : 1]=2 ±1000°/s
 FS_SEL[0 : 1]=3 ±2000°/s

4.2.5 ACCEL_CONFIG

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|----------------|--------------------|-------|-------|-------|--------------|------|------|------|------|
| 1C | 28 | XA_ST | YA_ST | ZA_ST | AFS_SEL[1:0] | | - | | |

XA_ST、YA_ST、ZA_ST 為 0 時加速度計在所在軸上不進行自檢測，為 1 時進行自檢測

加速度計測量範圍

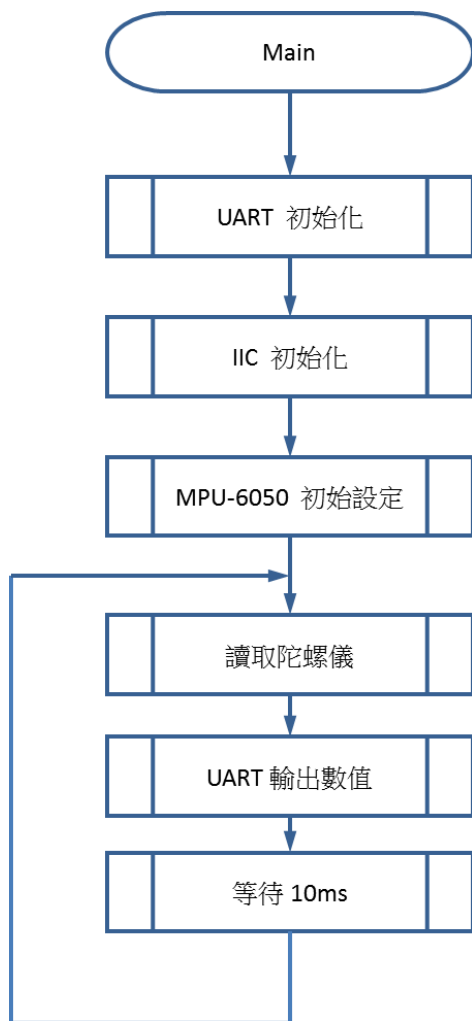
FS_SEL[0 : 1]=0 ±2g
 FS_SEL[0 : 1]=1 ±4g
 FS_SEL[0 : 1]=2 ±8g
 FS_SEL[0 : 1]=3 ±16g

5. 程式內容與流程

5.1 Main loop

```
/* *****  
* Outline      : Main  
* Description  :  
***** */  
void main(void)  
{  
    // Sensor data.  
    // data[0] : X axis  
    // data[1] : Y axis  
    // data[2] : Z axis  
    short data[3];  
  
    // UART initialize. BAUD_RATE 115200, 8-bit data, no parity, 1 stop bit.  
    UART_Create();  
    // IIC initialize and IIC interrupt enable.  
    IIC_Create();  
    // MPU6050 initialize.  
    InitMPU6050();  
  
    while(1)  
    {  
        // Get Gyro data.  
        GetGyroData(data);  
        // data output.  
        printf("%d, %d, %d\r\n", data[0], data[1], data[2]);  
        I2C_Delay_ms(10);  
    }  
}
```

程式流程圖如下頁:



5.2 MPU-6050 initialize

定義 MPU-6050 暫存器 MACRO 如下

```
#define SMPLRT_DIV      0x19
#define CONFIG          0x1A
#define GYRO_CONFIG     0x1B
#define ACCEL_CONFIG    0x1C
#define ACCEL_XOUT_H    0x3B
#define ACCEL_XOUT_L    0x3C
#define ACCEL_YOUT_H    0x3D
#define ACCEL_YOUT_L    0x3E
#define ACCEL_ZOUT_H    0x3F
#define ACCEL_ZOUT_L    0x40
#define TEMP_OUT_H      0x41
#define TEMP_OUT_L      0x42
#define GYRO_XOUT_H     0x43
#define GYRO_XOUT_L     0x44
#define GYRO_YOUT_H     0x45
#define GYRO_YOUT_L     0x46
#define GYRO_ZOUT_H     0x47
#define GYRO_ZOUT_L     0x48
#define PWR_MGMT_1      0x6B
#define WHO_AM_I        0x75
#define SlaveAddress    0x68
```

MPU-6050 初始化函數.

```
void InitMPU6050(void)
{
    Single_WriteI2C(PWR_MGMT_1, 0x00);
    Single_WriteI2C(SMPLRT_DIV, 0x07);
    Single_WriteI2C(CONFIG, 0x06);
    Single_WriteI2C(GYRO_CONFIG, 0x18);
    Single_WriteI2C(ACCEL_CONFIG, 0x01);
}
```

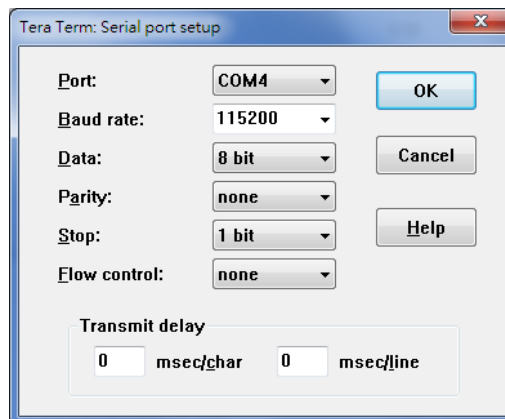
6. 操作流程

6.1 啟動範例程式

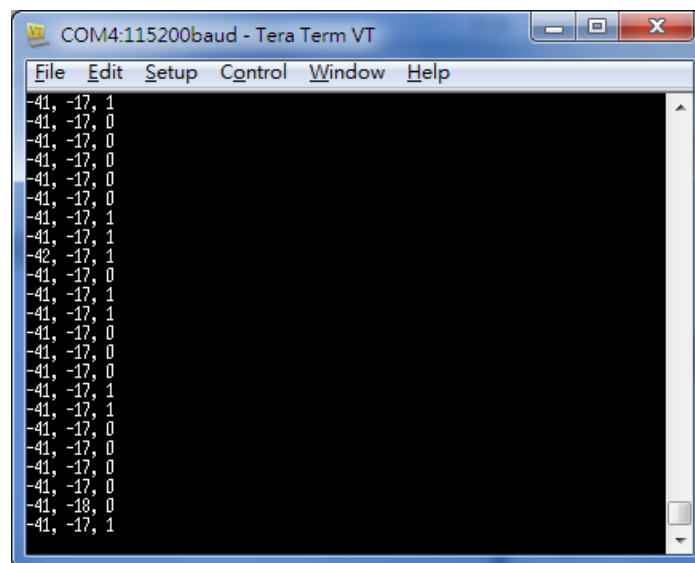
1. 將目錄 “RX63T_MPU6050” 複製至 C:\workspace .
2. 雙擊 C:\workspace\RX63T_MPU6050\RX63T_project.mtpj，啟動 CS+。

6.2 程式運行與結果

開啟終端機程式並設定通訊格式如下



運行範例程式後，終端機程式會顯示陀螺儀數值結果如下



Revision History

| Rev. | Date | Description | |
|------|---------------|-------------|---------|
| | | Page | Summary |
| 0.00 | Sep. 14, 2015 | - | . |