Use Nano130 to Control MPU9250 Sensor

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses Nano130 to control sensor MPU9250
BSP Version	Nano100B_Series_BSP_CMSIS_v3.03.000
Hardware	NuEdu-EVB-Nano130

1 Function Description

This example shows how to use Nano130 to communicate with motion sensor MPU9250 to obtain the motion track and direction of the device.

First, Nano130 initializes MPU9250 through I2C interface, setting the mode as measurement mode, sampling frequency as 125Hz, frequency width as 5Hz, acceleration measurement range ± 2G, angular velocity measurement range ± 2000 DPS.

Then, we can use I2C bus to send out instructions to obtain nine values measured by MPU9250. We can use these values to do some interesting applications. For example, use acceleration and angular velocity values to record the moving track, and use magnetic value to determine the moving direction.

2 Code Description

Initialize MPU9250 :

```
    void Init_MPU9250(void)
    {
     I2C0_Write(GYRO_ADDRESS,PWR_MGMT_1, 0x00);
     I2C0_Write(GYRO_ADDRESS,SMPLRT_DIV, 0x07);
     I2C0_Write(GYRO_ADDRESS,CONFIG, 0x06);
     I2C0_Write(GYRO_ADDRESS,GYRO_CONFIG, 0x18);
     I2C0_Write(GYRO_ADDRESS,ACCEL_CONFIG, 0x01);
    }
```

Acceleration value, angle value and magnetometer value measured by MPU9250 sensor are obtained by I2C Bus：

```
void READ_MPU9250_ACCEL(void)
{

   BUF[0]=I2C0_Read(ACCEL_ADDRESS,ACCEL_XOUT_L);
   BUF[1]=I2C0_Read(ACCEL_ADDRESS,ACCEL_XOUT_H);
   T_X=  (BUF[1]<<8)|BUF[0];
   T_X/=164;

   BUF[2]=I2C0_Read(ACCEL_ADDRESS,ACCEL_YOUT_L);
   BUF[3]=I2C0_Read(ACCEL_ADDRESS,ACCEL_YOUT_H);
   T_Y=  (BUF[3]<<8)|BUF[2];
   T_Y/=164;
   BUF[4]=I2C0_Read(ACCEL_ADDRESS,ACCEL_ZOUT_L);
   BUF[5]=I2C0_Read(ACCEL_ADDRESS,ACCEL_ZOUT_H);
   T_Z=  (BUF[5]<<8)|BUF[4];
   T_Z/=164;
}
void READ_MPU9250_GYRO(void)
{

   BUF[0]=I2C0_Read(GYRO_ADDRESS,GYRO_XOUT_L);
   BUF[1]=I2C0_Read(GYRO_ADDRESS,GYRO_XOUT_H);
   T_X=  (BUF[1]<<8)|BUF[0];
   T_X/=16.4;

   BUF[2]=I2C0_Read(GYRO_ADDRESS,GYRO_YOUT_L);
```

```
    BUF[3]=I2C0_Read(GYRO_ADDRESS,GYRO_YOUT_H);
    T_Y=  (BUF[3]<<8)|BUF[2];
    T_Y/=16.4;

    BUF[4]=I2C0_Read(GYRO_ADDRESS,GYRO_ZOUT_L);
    BUF[5]=I2C0_Read(GYRO_ADDRESS,GYRO_ZOUT_H);
    T_Z=  (BUF[5]<<8)|BUF[4];
    T_Z/=16.4;

}
void READ_MPU9250_MAG(void)
{
    I2C0_Write(GYRO_ADDRESS,0x37,0x02);//turn on Bypass Mode
    CLK_SysTickDelay(10000);
    I2C0_Write(MAG_ADDRESS,0x0A,0x01);
    CLK_SysTickDelay(10000);
    BUF[0]=I2C0_Read (MAG_ADDRESS,MAG_XOUT_L);
    BUF[1]=I2C0_Read (MAG_ADDRESS,MAG_XOUT_H);
    T_X=(BUF[1]<<8)|BUF[0];

    BUF[2]=I2C0_Read(MAG_ADDRESS,MAG_YOUT_L);
    BUF[3]=I2C0_Read(MAG_ADDRESS,MAG_YOUT_H);
    T_Y=  (BUF[3]<<8)|BUF[2];

    BUF[4]=I2C0_Read(MAG_ADDRESS,MAG_ZOUT_L);
    BUF[5]=I2C0_Read(MAG_ADDRESS,MAG_ZOUT_H);
    T_Z=  (BUF[5]<<8)|BUF[4];
}
```

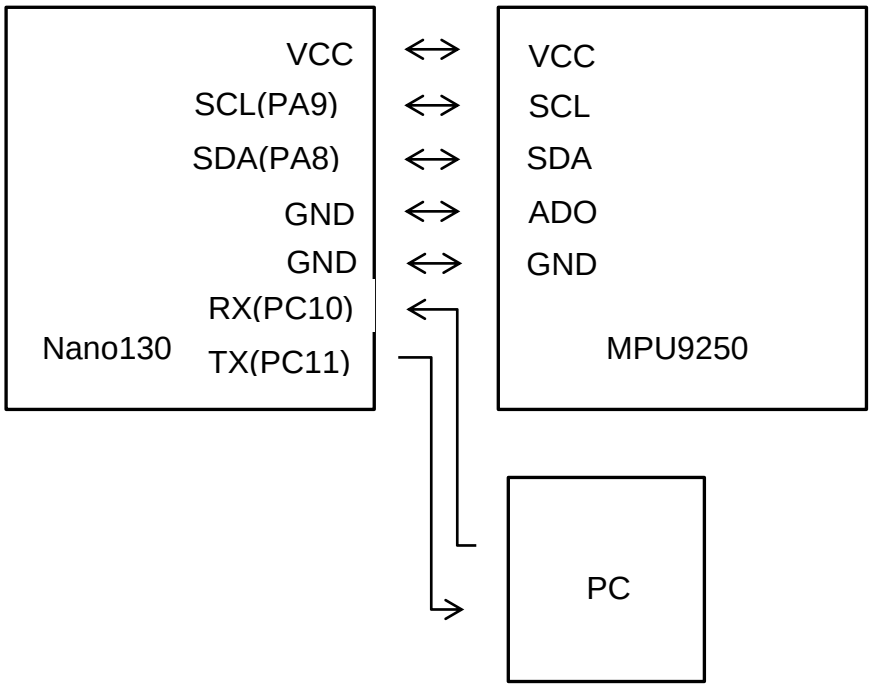
Print out the acquired value through UART 1：

```
    READ_MPU9250_ACCEL();
    printf("ACCEL:  X: %4d  Y: %4d  Z: %4d   /0.01 g\n", T_X, T_Y, T_Z);
    READ_MPU9250_GYRO();
    printf("GYRO :  X: %4d  Y: %4d  Z: %4d   /1 dps\n", T_X, T_Y, T_Z);
    READ_MPU9250_MAG();
    printf("MAG  :  X: %4d  Y: %4d  Z: %4d   /0.6 uT\n\n\n", T_X, T_Y, T_Z);
```

3 Software and Hardware Environment

- **Software Environment**
  - BSP Version
    - Nano100B_Series_BSP_CMSIS_v3.03.000
  - IDE Version
    - Keil uVersion 4.6
- **Hardware Environment**
  - Circuit Components
    - NuEdu-EVB-Nano130
    - MPU9250
  - Diagram

Nano130 transmits control commands through I2C to MPU9250 sensors and prints data by UART1.



4 Directory Information

- EC_Nano130_Control_MPU9250_Sensor_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - NuEdu — Library for NuEdu board
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click Nano130_Control_MPU9250_Sensor.uvproj.
2. Enter the compilation mode interface
   a. Compilation
   b. Download code to Flash
   c. Enter/leave debugging mode
3. Enter the debugging mode interface
   a. Execution code

6 Revision History

Date	Revision	Description
Sep.25, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.