SENSSET

SENSOR SEQUENCE

SSI-MU16460

6 DoF Inertial measurement unit

PRODUCTS FEATURES

- High Precision 6 DoF Mini IMU with Full Calibration
- Fully Similar as ADIS16460, Better Performance
- 3 Angular Rates and 3 Linear Acceleration Outputs
- Data Output Rate: 200 Hz
- Range: Gyro ±400 deg/s, Accelerometer ±6g
- Bias Instability: Gyro 3 deg/h, Accelerometer ±10 ug
- Wide Working Temperature: -40°C ~ +85°C

1. Technical parameters

Table 1. Gyroscope characteristics

Parameter	Value			Unit
	Min	Typ	Max	
Range	-400		+400	deg/s
Bias Instability	4	3	4	deg/h
Initial Bias Error	0.04	0.08	0.1	deg/s
Resolution		0.01		deg/s
Scale Factor Accuracy		0.1		%
Non-linearity		0.01		%FS
Random Walk	0.1	0.2	0.3	deg/√h
Bandwidth		50		Hz

Table 2. Accelerometer characteristics

Parameter	Value			Unit
	Min	Typ	Max	
Range	-6		+6	g
Bias Instability	5	10	15	ug
Initial Bias Error	3	4	5	ug
Resolution		0.5		mg
Scale Factor Accuracy		0.1		%
Non-linearity		0.01		%FS
Random Walk	0.03	0.05	0.07	m/s/√h
Bandwidth		50		Hz

Table 3. Environment Conditions

Parameter	Value	Unit
Working Temperature	-40~+85	℃
Storage Temperature	-55~+100	℃
Housing Material	Aluminium	

Table 4. Electrical Features

Parameter	Value	Unit
Input Power Supply	3.3 VDC	VDC
Power Consumption	< 0.15 W	W
Interface	SPI and UART	
Output Frequency	200	Hz

Table 5. Physical Parameter

Parameter	Value	Unit
Size	23.4*22.4*7.5	mm
Weight	<15	grams
Connector	7*2	pins

2. Pin Definition

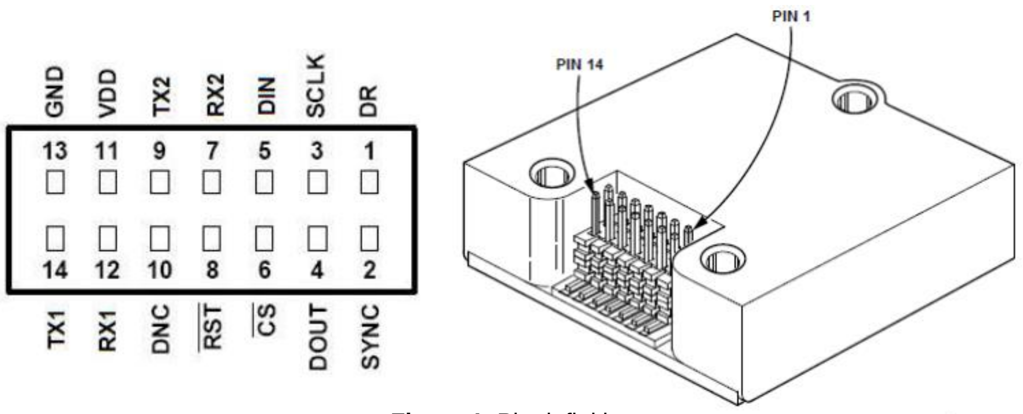


Figure 1. Pin definition

Table 6. Pin definition

Pin No.	Name	Description
1	DR	Data ready indicator, falling edge means that the data is ready
2	SYNC	External synchronization signal
3	SCLK	SPI serial clock
4	DOUT	SPI data output
5	DIN	SPI data input
6	CS	SPI chip select
7	RX2	GPS signal UART_RX (LVTTL)
8	RST	Reset
9	TX2	GPS signal UART_TX (LVTTL)
10	DNC	Do not connect
11	VDD	Power supply
12	RX1	IMU data UART_RX (LVTTL)
13	GND	Power ground
14	TX1	IMU data UART_TX (LVTTL)

3. Orientation

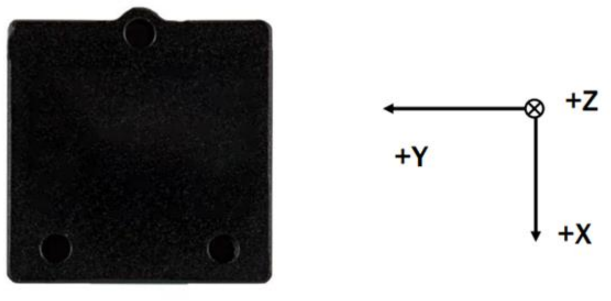


Figure 2. Orientation system

If the sensor is accelerated or rotated in the indicated directions, the corresponding channels of the device will deliver a positive acceleration or rate signal. If the sensor is at rest without any rotation and the force of gravity is acting contrary to the indicated directions, the output of the corresponding acceleration channel will be positive and the corresponding gyroscope channel will be "zero".

Example: If the sensor is at rest or at uniform motion in a gravity field according to the figure given below, the output signals are:

- 0g for X Acc Sensor and 0 °/s for the X GYR Sensor
- 0g for Y Acc Sensor and 0 °/s for the Y GYR Sensor
- -1g for Z Acc Sensor and 0 °/s for the Z GYR Sensor

4. Mechanical Dimension

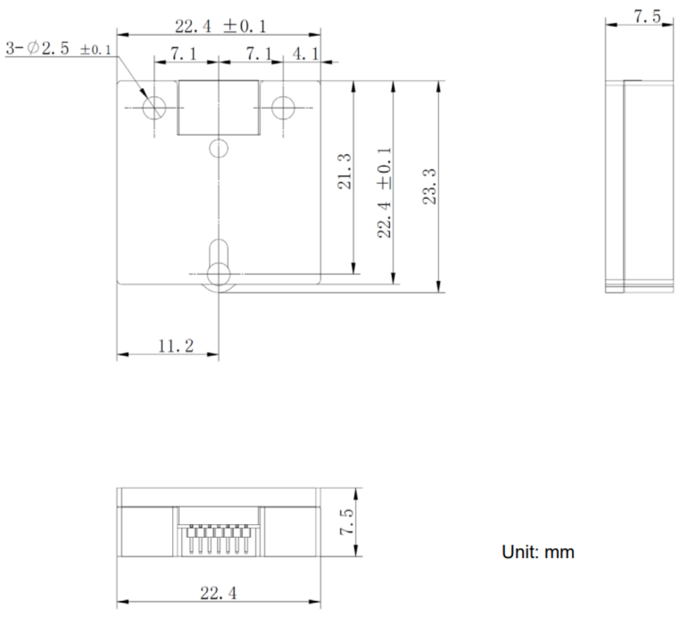
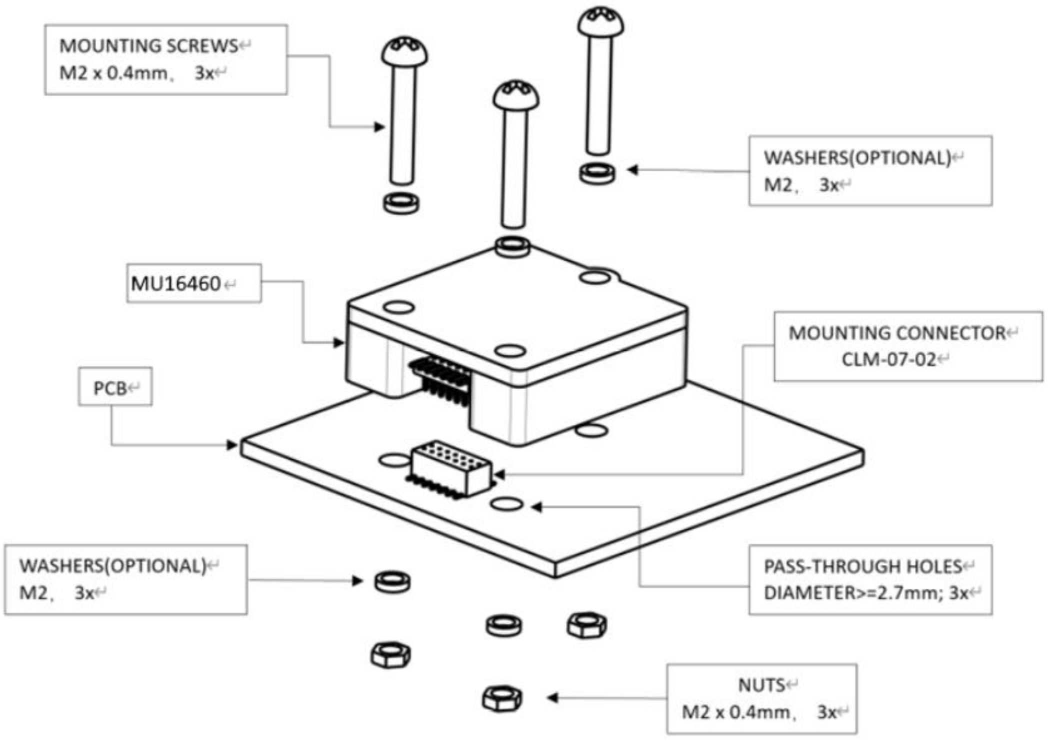


Figure 4. Mechanical Dimension

Note:
1). Note that the upper row pins of the plug-in connector corresponds to the center line of the screw hole;
2). Recommended that the the through hole's diameter (for screw hole) is 2.7mm;
3). It is recommended to use M2 screws for fixing SSI-MU16460.

5. Installation



6. Communication Protocol - UART

the electronic parameters of UART RS232 communication protocol are as follows:
- Initial bit: 1
- Data bit: 8
- Stop bit: 1
- Check bit: none each data string includes 23 bytes (Little Endian), the detailed description see as follows:

Table 7. Data string Definition

Byte No.	Name	Description
1	Synchronous byte 1	7FH (fixed value)
2	Synchronous byte 2	80H (fixed value)
3	X axis acceleration, low 8 bits	(20/65536)*N N=(int16) acceleration high 8 bits* 256+acceleration low 8 bits
4	X axis acceleration, high 8 bits	
5	Y axis acceleration, low 8 bits	(20/65536)*N N=(int16) acceleration high 8 bits* 256+acceleration low 8 bits
6	Y axis acceleration, high 8 bits	
7	Z axis acceleration, low 8 bits	(20/65536)*N N=(int16) acceleration high 8 bits* 256+acceleration low 8 bits
8	Z axis acceleration, high 8 bits	
9	X axis angular rate, low 8 bits	(1260/65536)*N N=(int16) angular rate high 8 bits* 256+angular rate low 8 bits
10	X axis angular rate, high 8 bits	
11	Y axis angular rate, low 8 bits	(1260/65536)*N N=(int16) angular rate high 8 bits* 256+angular rate low 8 bits
12	Y axis angular rate, high 8 bits	
13	Z axis angular rate, low 8 bits	(1260/65536)*N N=(int16) angular rate high 8 bits* 256+angular rate low 8 bits
14	Z axis angular rate, high 8 bits	
15	X axis roll angle, low 8bits	(360/65536)*N N= (int16) angle high 8 bits* 256+angle low 8 bits
16	X axis roll angle, high 8bits	
17	Y axis pitch angle, low 8bits	(360/65536)*N N= (int16) angle high 8 bits* 256+angle low 8 bits
18	Y axis pitch angle, high 8bits	
19	Z axis yaw angle, low 8bits	(360/65536)*N N= (int16) angle high 8 bits* 256+angle low 8 bits
20	Z axis yaw angle, high 8bits	
21	temperature sensor, low 8bits	(200/65536)*N N=(int16) temperature high 8 bits* 256+temperature low 8 bits
22	temperature sensor, low 8bits	
23	Checksum	sum of all bytes except synchronous bytes (3~22 bytes), then bit inversion

6.1 Communication Protocol – UART Data Analysis Example

```
double[] sensors = new double[9];

sensors[0] = (short)((byte)buffer[0] + ((byte)buffer[1] << 8));
sensors[1] = (short)((byte)buffer[2] + ((byte)buffer[3] << 8));
sensors[2] = (short)((byte)buffer[4] + ((byte)buffer[5] << 8));
sensors[3] = (short)((byte)buffer[6] + ((byte)buffer[7] << 8));
sensors[4] = (short)((byte)buffer[8] + ((byte)buffer[9] << 8));
sensors[5] = (short)((byte)buffer[10] + ((byte)buffer[11] << 8));
sensors[6] = (short)((byte)buffer[12] + ((byte)buffer[13] << 8));
sensors[7] = (short)((byte)buffer[14] + ((byte)buffer[15] << 8));
sensors[8] = (short)((byte)buffer[16] + ((byte)buffer[17] << 8));

// Accel x, y, z
Ax = sensors[0] * Accel_Scale / Sensor_Scale;
Ay = sensors[1] * Accel_Scale / Sensor_Scale;
Az = sensors[2] * Accel_Scale / Sensor_Scale;

// Gyro x, y, z Gx = sensors[3] * Rate_Scale / Sensor_Scale;
Gy = sensors[4] * Rate_Scale / Sensor_Scale;
Gz = sensors[5] * Rate_Scale / Sensor_Scale;

// Roll, Pitch, Yaw
Roll = sensors[6] * Angle_Scale / Sensor_Scale;
Pitch = sensors[7] * Angle_Scale / Sensor_Scale;
Yaw = sensors[8] * Angle_Scale / Sensor_Scale;
```

Remarks:
1. acceleration unit is g, angular value unit is degree/s (°/s), attitude angle degree is degree (°).
2. Accel_Scale = 20, Rate_Scale = 1260, Angle_Scale = 360, Sensor_Scale = 65536

7. Communication Protocol - SPI

SSI-MU16460 can be configured as SPI interface to communicate, it is used as slave device, and the master device should be configured as the following settings to communicate with SSI-MU16460:

• 16 bit word length and the most significant bit (MSB) transmission
• fCLK ≤ 1MHz
• CPOL = 1 (Clock Polarity) and CPHA = 1 (Clock Phase)

7.1 SPI Burst-mode Read Data

SPI master device will access to SPI bus through the following mode: Burst mode In Burst mode, SSI-MU16460 will transmit a group of continuous SPI periodic predefined data block. In Burst mode, the predefined data block that is returned with single group format by SSI-MU16460 is called data package, and it doesn't need to send multiple read commands. The lengths of these groups depends on the selected data package

Table 8. Standard Data Package SPI Burst-mode Read Mode

Data Package	Register Address	Data Length	Remarks
Standard	0x50	18 bytes	the detailed introduction refers to the following content

The standard data package includes 9 predefined registers' data. The following table indicates the data name and analysis method, these registers will be listed according to the read sequence of Burst mode read.

Table 9. Register Data

Register Name	Description	Analysis Method
status	Overrange mark bit that indicate whether the sensor is over range or not	The low 6 bits represent whether the 6 axis sensor is overrange or not, from low to high is Gx, Gy, Gz, Ax, Ay, Az; 1 means that it's overrange for its corresponding axis sensor
X_RATE	X axis gyro output	(1260/65536) * N Note: N is integal with symbol N= (int16) (angular rate high 8 bits* 256+angular rate low 8 bits)
Y_RATE	Y axis gyro output	
Z_RATE	Z axis gyro output	
X_ACCEL	X axis accelerometer output	(20/65536)*N Note: N is integal with symbol N= (int16) acceleration high 8 bits* 256+acceleration low 8 bits
Y_ACCEL	Y axis accelerometer output	
Z_ACCEL	Z axis accelerometer output	
SENSOR_TEMP	Sensor Temperature	(200/65536)*N Note: N is integal with symbol N= (int16) temperature high 8 bits* 256+temperature low 8 bits
CHECKSUM	Check Sum of all previous bytes of data	From 'RESERVED' to SENSOR_TEMP

Burst mode will be activated wen the master device requests to read data grouping (0x50) from Burst mode. It will take 9 extra SPI periods to finish the read. The following diagram indicates the Burst mode sequence. Attention: if SPI periods number is not correct, then the SPI transmission will be finished in advance or kept at Burst mode, and the following Read / Write will be not synchronized with SSI-MU16460's SPI transmission period.

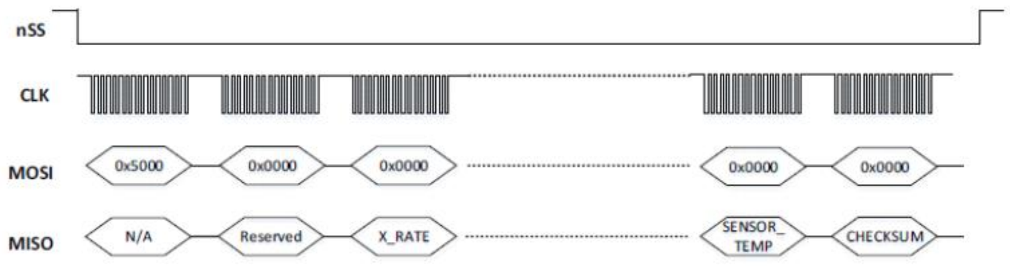


Figure 5. Burst Mode Read standard data

Operation Remarks:

- The Burst mode read of other data groups will be similar as that of the standard group. The only difference with the above method is the corresponding registers' address and the data word number is different.
- It should be careful when switching the data packages, cause the returned value is invalid during the new data package's first Burst read time. It requires one read period to fill inner Bust mode register; and the follow-up information from the data package will include the valid information.
- During Burst read period, the chip select line (nSS) can be controlled through one of the following method:
  • During whole read period, pull down nSS and keep low level. After the transmission is finished, the chip select should be set as high level.

7.2 SPI Register Read/Write Mode

In polling mode, when the SPI master sets chip selection (NSS) to low level, data transfer begins and 16-bit words consisting of register address bytes and zero bytes are timed across the MOSI line. For example, to request the unit serial number stored in register 0x58, the host sends the command 0x5800. During the next 16 clock cycles, SSI-MU16460 returns information from this address via the MISO line.

7.2.1 Single Register Read Mode

Figure 6 shows a poll mode read of a single register (accelerometer LPF cutoff frequency), consisting of two bytes, starting at register address 0x2F.

In this example, the SPI master device starts a register through MOSI that is read by a clock followed by 0x00 (0x2F00) at the address; This combination is called a read command. This is followed by 16 zeros to complete the SPI data transmission cycle. SSI-MU16460 transmits information back through MISO while the master transmits read commands over MOSI. In this transmission, the first data word sent by SSI-MU16460 (sent as a read command) consists of 16 bits of inapplicable data. The subsequent 16-bit message contains the acceleration sensor LPF cutoff frequency (the highest significant byte followed by the lowest significant byte)

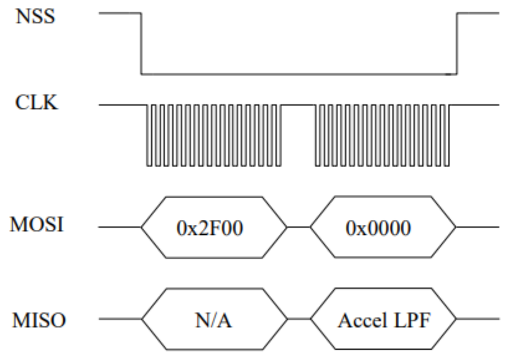


Figure 6. Read Accel LPF using read single register

Table 10. Register Data

Register Name	Address	Description	
Serial Number	0x52~0x55	Consist of 4 bytes: the highest byte first	
Product ID	0x72~0x73	Who Am I command, consist of 2 bytes ： the highest byte first. Example: SSI-MU16460 Product ID is 80 = 0x050	
FIRMWARE_VER	0x74~0x79	Firmware Version	
		0x74	Firmware Major Version string
		0x75	Firmware Minor Version string
		Firmware build date:	
		0x76-0x77	Year
		0x78	Month
		0x79	Day
		8.1.210815 8: Major version 1: Minor version 21: Year, 08: Month, 15: Day	
SPI_REG_ACCEL_FILTER	0x2F	Accelerometer Low-Pass Filter cut-off frequency	
SPI_REG_GYRO_FILTER	0x30	Gyroscope Low-Pass Filter cut-off frequency	

7.2.2 Single Register Write Mode

The SPI master device configures the SSI-MU16460 by writing to a specific register. Write one byte at a time. The specific registers and their write addresses that affect the system configuration are listed below

Table 11. Register Data

Register Name	Address	Description
SPI_REG_WRITE_UNLOCK	0x2E	Unlock sensor Low-Pass filter cut-off frequency write command
SPI_REG_ACCEL_FILTER	0x2F	Before writing new value, need writing 0x55 to Register 0x2E first
SPI_REG_GYRO_FILTER	0x30	Before writing new value, need writing 0xAA to Register 0x2E first

Table 12. Register Data

Number	Value	Description
1	LPF_UNSED	0x00 LPF is not used
2	LPF_IIR_05HZ	0x01 Cut-off frequency 5Hzs
3	LPF_IIR_10HZ	0x02 Cut-off frequency 10Hz
4	LPF_IIR_15HZ	0x03 Cut-off frequency 15Hz
5	LPF_IIR_20HZ	0x04 Cut-off frequency 20Hz
6	LPF_IIR_25HZ	0x05 Cut-off frequency 25Hz
7	LPF_IIR_30HZ	0x06 Cut-off frequency 30Hz
8	LPF_IIR_35HZ	0x07Cut-off frequency 35Hz
9	LPF_IIR_40HZ	0x08 Cut-off frequency 40Hz
10	LPF_IIR_50HZ	0x09 Cut-off frequency 50Hz
11	LPF_IIR_100HZ	0x0ACut-off frequency 100Hz
12	LPF_IIR_200HZ	0x0B Cut-off frequency 200Hz

The following example describes how to write commands to change the accelerometer sensor LPF filter cutoff frequency:

• Select a write address for the desired register, such as 0x2E to unlock the register configuration
• Change the most significant bit of the address to 1 (the write bit), such as 0x2E to 0xAE
• Create a write command by attaching a write bit/address combination to the value to be written to the register, such as 0xAE00

Figure 7 shows the command sent via SPI to modify the accelerometer LPF cutoff frequency.
Figure 8 shows the command sent via SPI to modify the gyroscope LPF cutoff frequency

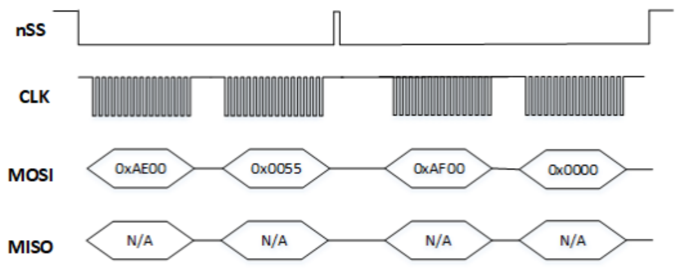


Figure 7. Write register setting accelerometer LPF cut-off frequency

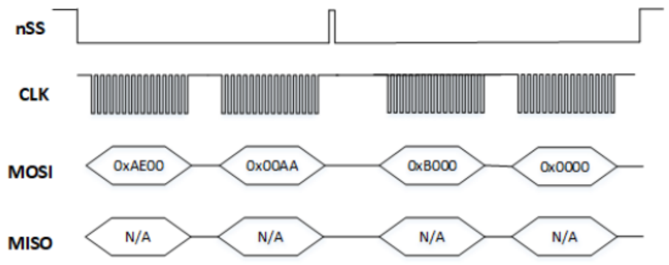


Figure 8. Write register setting gyroscope LPF cut-off frequency

7.3 SPI Time Sequence

SPI time sequency requirements please see the following table and figure. and the following operation constraints apply to SPI communication:

• CPOL = 1 (Clock Polarity) and CPHA = 1 (Clock Phase)
• 16 bit word length and the most significant bit (MSB) transmission

Table 13. SPI Time Sequence Requirement

Parameter	Description	Value	Unit
f_{CLK}	SPI clock frequency	1 (max)	MHz
t_{DELAY}	Time between successive clock cycles	20(min)	us
$t_{SU,NSS}$	nSS setup time prior to clocking data	1	us
$t_{h, NSS}$	nSS hold time following clock signal	67	ns
$t_{V, MISO}$	Time after falling edge of previous clock-edge that MISO data bit is invalid	25	ns
$t_{SU, MOSI}$	Data input setup time prior to rising edge of clock	5	ns
$t_{h,MOSI}$	Data input hold time following rising edge of clock	4	ns

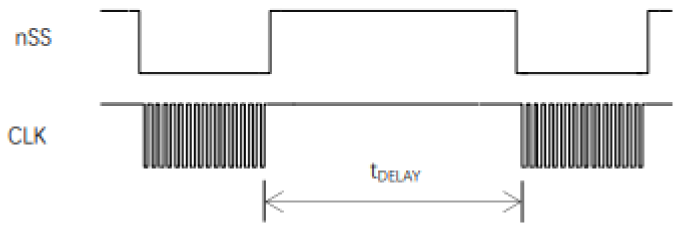


Figure 9. Relay time

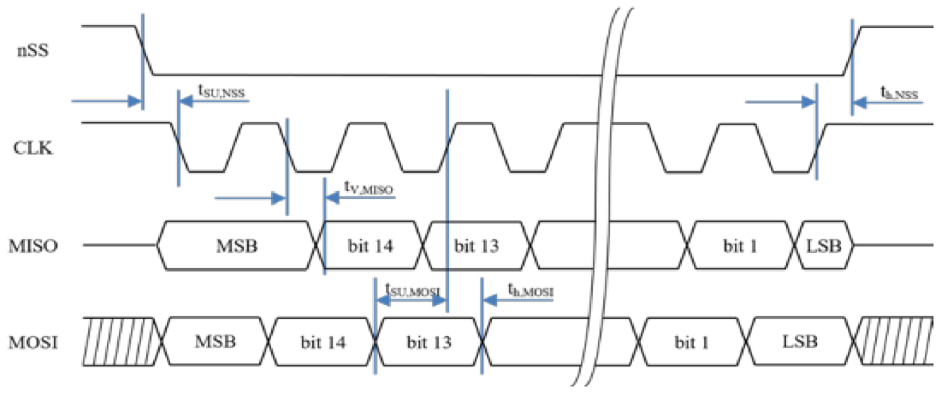


Figure 10. Time Sequence Figure

7.4 Reference Example to Read ID from SSI-MU16460:

```
#define PRODUCT_ID 0x72
uint16_t product_id_value = 0;
void ReadProductID(void)
{
HAL_GPIO_WritePin(NPCS_GPIO_Port,
NPCS_Pin, GPIO_PIN_RESET);
TxBuf[0] = PRODUCT_ID;


HAL_SPI_TransmitReceive(&hspi1, &TxBuf[0], &RxBuf[0], 4, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);

product_id_value = (RxBuf[2]<<8)|RxBuf[3];
}
```

7.5 Reference Example to Set SPI for SSI-MU16460:

```
#define SPI_REG_WRITE_UNLOCK 0x2E
#define SPI_REG_ACCEL_FILTER 0x2F
#define SPI_REG_GYRO_FILTER 0x30
void filter_gyro_set(uint8_t desired_filter)
{
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_RESET);
TxBuf[0] = SPI_REG_WRITE_UNLOCK|0x80;
TxBuf[1] = 0x00;
TxBuf[2] = 0x00;
TxBuf[3] = 0xAA;

HAL_SPI_TransmitReceive(&hspi1, &TxBuf[0], &RxBuf[0], 4, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET); HAL_Delay(1);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_RESET);
TxBuf[0] = SPI_REG_GYRO_FILTER|0x80; TxBuf[1] = 0x00;
TxBuf[2] = 0x00; TxBuf[3] = desired_filter;
HAL_SPI_TransmitReceive(&hspi1, &TxBuf[0], &RxBuf[0], 4, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);

//read back
TxBuf[0] = SPI_REG_WRITE_UNLOCK;
TxBuf[2] = SPI_REG_GYRO_FILTER;
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &TxBuf[0], &RxBuf[0], 6, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);
}
```

7.6 Reference Example to Set Filter through SPI for SSI-MU16460

```
void filter_acc_set(uint8_t desired_filter)
{
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_RESET);
TxBuf[0] = SPI_REG_WRITE_UNLOCK|0x80;
TxBuf[1] = 0x00;
TxBuf[2] = 0x00;
TxBuf[3] = 0x55;

HAL_SPI_TransmitReceive(&hspi1, &TxBuf[0], &RxBuf[0], 4, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);
HAL_Delay(1); HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_RESET);
TxBuf[0] = SPI_REG_ACCEL_FILTER|0x80;
TxBuf[1] = 0x00;
TxBuf[2] = 0x00;
TxBuf[3]  =  desired_filter;  HAL_SPI_TransmitReceive(&hspi1,  &TxBuf[0],  &RxBuf[0],  4,
200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);

//read back
TxBuf[0] = SPI_REG_WRITE_UNLOCK;
TxBuf[2] = SPI_REG_ACCEL_FILTER;
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &TxBuf[0], &RxBuf[0], 6, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);

 }
```

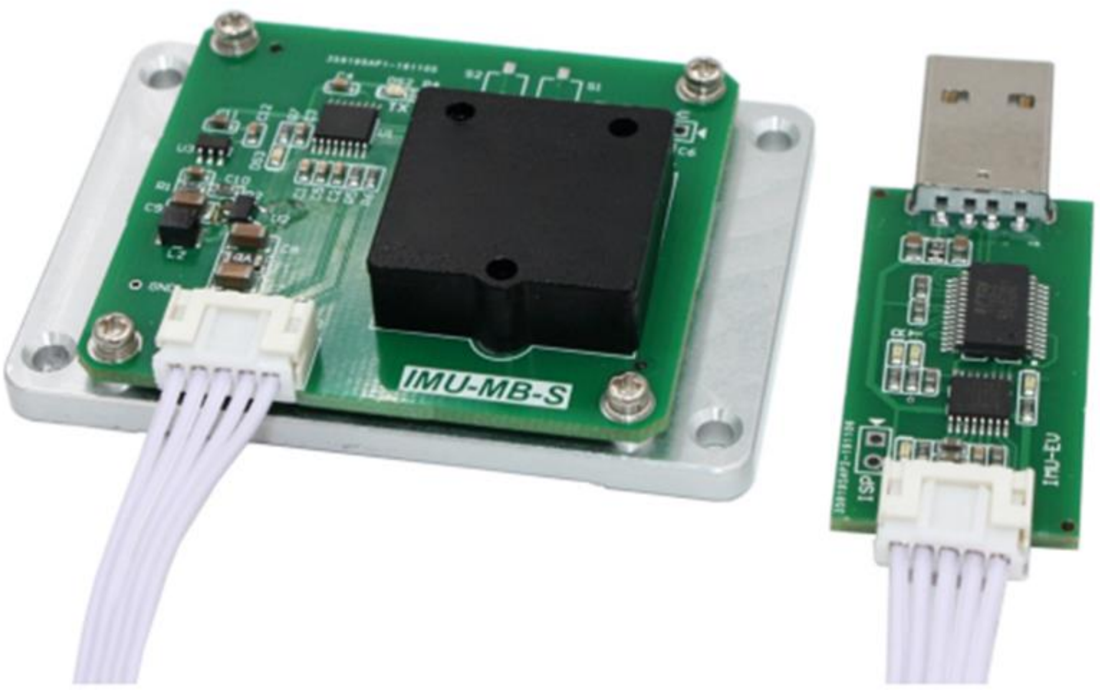
Remarks: for filter setting through SPI, it is temporary, when the sensor is power off (the settings through SPI will be invalid), it will need to do this settings again.

Serial port settings has two modes: temporary or permanent, if selecting permanent, it will be saved after reboot.

7.7 Reference Example to Read Data from SSI-MU16460

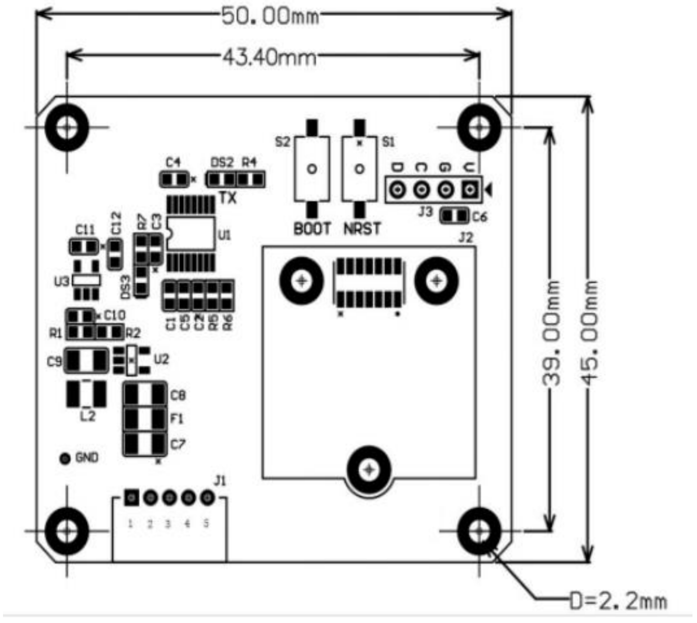
```
uint32_t spi_total_count,spi_correct_count,spi_error_count;
int16_t Buffer[7];
void BurstRead(void)
{
uint8_t TxBuf[100]={0};
uint8_t RxBuf[100]={0};
HAL_GPIO_WritePin(NPCS_GPIO_Port,
NPCS_Pin, GPIO_PIN_RESET);
TxBuf[0] = 0x50;
HAL_SPI_TransmitReceive(&hspi1, TxBuf, &RxBuf[0], BURST_LENGTH+2, 200);
HAL_GPIO_WritePin(NPCS_GPIO_Port, NPCS_Pin, GPIO_PIN_SET);
uint8_t checksum = 0;
spi_total_count++;
uint16_t spi_calc_checksum = 0;
uint16_t spi_rxd_checksum = (stdburst_buf[18]<<8)|stdburst_buf[19];
for(i=2; i<18; i++)
{
spi_calc_checksum += stdburst_buf[i];
}
if(spi_calc_checksum==spi_rxd_checksum)
{
Buffer[0] = (RxBuf[4]<<8)|RxBuf[5];//GyroX
Buffer[1] = (RxBuf[6]<<8)|RxBuf[7];//GyroY
Buffer[2] = (RxBuf[8]<<8)|RxBuf[9];//GyroZ
Buffer[3] = (RxBuf[10]<<8)|RxBuf[11];//AccelX
Buffer[4] = (RxBuf[12]<<8)|RxBuf[13];//AccelY
Buffer[5] = (RxBuf[14]<<8)|RxBuf[15];//AccelZ
Buffer[6] = (RxBuf[16]<<8)|RxBuf[17];//Temp
spi_correct_count++; }
else
{
spi_error_count++;
}
}
```

8. SSI-MU16460 Evaluation Board



SSI-MU16460 evaluation board is a PC-USB Inertial MEMS Evaluation System, it provides a Windows-compatible evaluation system for SSI-MU16460. It is composed of PCB circuit, aluminium base, and data converter cable.

9. SSI-MU16460 Evaluation Board size



10. Pin Definition of Evaluation Board

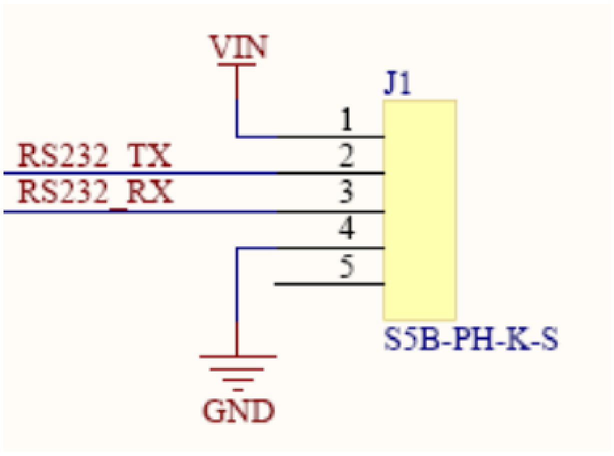


Figure 11. Pin Definition

Table 14. Pin definitions

Pins No.	Name	Description
1	VIN	5V-12V Input voltage 5~12 VDC
2	RS232_TX	RS232 TX
3	RS232_RX	RS232 RX
4	GND	Ground
5	BOOT	Inner used pin

SENSSET

SENSOR SEQUENCE

www.sensset.ru

8 (812) 309-58-32 доб. 150
info@sensset.ru

198099, г. Санкт-Петербург
ул. Калинина, дом 2, корпус 4, литера А.

Development, production and supply of high-tech sensors