

振動感測器 RS485 串列通訊埠界面，Modbus RTU Slave 軟體協議，功能碼與暫存器位址定義說明：

- 針對振動感測器連結所需的 RS485 格式設定：

Property	Value
Baudrate (bps)	115200 or 3M
Data bits	8
Stop bit	1
Parity	None
Supported Function Code	Read Holding(FC03) Read Input(FC04) Write Single(FC06)
Slave ID	0x0001

- 透過 Modbus RTU 協議對振動感測器連結所需的命令格式：

Command		Register	Comment
Supported baud rate: 3 Mbps, only.			
Raw data FIFO buffer size*4		0x02	R*2，振動資料尚未讀取的長度(一次取樣為 XYZ 各 1 筆，每次加 3)
Raw data(XYZ)*4		0x03 ~ 0x7D	R*2，3 軸振動資料
Raw data(XYZ) latest		0x83 ~ 0x85	R*2，非連續振動資料讀取；每讀取一次，會清除已暫存(舊的)於感測器內部的振動資料
Supported baud rate: 115200 bps			
Sample rate change		0x01	R*3W*1，設定取樣率，(I-type default: 7812 Hz，K-type default: 6400 Hz，此為特徵值計算最大可支援取樣率)
Temperature		0x14	R*3，溫度讀值
Stream size		0x15	R*5，寫入 Bulk transfer block size 讀取 vibration raw data
Baud Rate	High	0x17	R*3W*1，鮑率預設值：3Mbps；設定步驟：High Register & Low Register 都要設定，其鮑率值才會正確；先設定 High Register 後再設定 Low Register，完成後將感測器斷電 3 秒後重新上電，才可使用新鮑率。  例：115200bps = 0x0001C200，High Register = 0x0001，Low Register = 0xC200  例：3000000bps = 0x002DC6C0，High
	Low	0x18	

Command		Register	Comment
			Register = 0x002D, Low Register = 0xC6C0
High pass enable		0x1C	W* <sub>1</sub> , High pass enable(1: enable, 0: disable), Band Width = 3 ~ 2.5 kHz@ODR = 7812SPS
UCID		0x1B	R* <sub>3</sub> 用戶識別碼(Unique Customer Identification) 提供客戶讀取感測器的唯一碼(32bits, 每次讀取長度需為 2)。
Firmware version		0x1D	R* <sub>3</sub> 韌體版本號碼, 共 4 碼(32bits, b31~b0, 每次讀取長度需為 2) b31~b24: 感測器主版號 b23~b16: 韌體年份 b15~b8: 韌體週數 b7~b0: 次版號
Chip ID		0x80	R* <sub>2</sub> , 3 軸晶片 ID; 驗證電腦端與感測器連線是否正常(每次讀取長度需為 3)
Acceleration	RMS* <sub>3</sub>	0x1E	讀取 3 軸加速規 RMS (單位: Gravity, 9.81 m/s <sup>2</sup> ), 最大更新率: 5 Hz, RMS = Register value / 1000
		0x28	讀取 3 軸加速規 RMS (單位: Acceleration, m/s <sup>2</sup> ), 最大更新率: 5 Hz, RMS = Register value / 100
	Peak* <sub>3</sub>	0x1F	讀取 3 軸加速規 Peak (單位: Gravity, 9.8 m/s <sup>2</sup> ), 最大更新率: 5 Hz, Peak = Register value / 1000
		0x29	讀取 3 軸加速規 Peak (單位: Acceleration, m/s <sup>2</sup> ), 最大更新率: 5 Hz, Peak = Register value / 100
	Crest Factor* <sub>3</sub>	0x20	讀取 3 軸加速規 Crest Factor, 最大更新率: 5 Hz, Crest Factor = Register value / 1000
	Skewness* <sub>3</sub>	0x21	讀取 3 軸加速規 Skewness, 最大更新率: 2 ~ 5 秒, Skewness = Register vlaue / 1000
	Kurtosis* <sub>3</sub>	0x22	讀取 3 軸加速規 Kurtosis, 最大更新率: 2 ~ 5 秒, Kurtosis = Register vlaue / 1000
	Primary Frequency* <sub>3</sub>	0x3D	讀取 3 軸加速度頻率, 最大更新率: 2 ~ 5 Hz, PF = Register value / 10
Velo	RMS* <sub>3</sub>	0x32	讀取 3 軸速度 RMS (單位: mm/s), 最大更新率: 5 Hz, RMS = Register value / 100

Command		Register	Comment
	Peak* <sub>3</sub>	0x33	讀取 3 軸速度 Peak (單位：mm/s)，最大更新率：5 Hz，Peak = Register value / 100
	Crest Factor* <sub>3</sub>	0x34	讀取 3 軸速度 Crest Factor，最大更新率：5 Hz，Crest Factor = Register value / 1000
	Primary Frequency* <sub>3</sub>	0x3C	讀取 3 軸速度頻率，最大更新率：2 ~ 5 Hz，PF = Register value / 10

P.S. Endian type: Big Endian

\*1: Support only for Write Single Register(FC06)

\*2: Support only for Read Input Register(FC04)

\*3: Support only for Read Holding Register(FC03)

\*4:此為出廠預設值，若 Baud Rate: 3 Mbps 可支援 raw data 連續讀值，若為 115200 bps 僅支援特徵參數讀取功能。

\*5: Support only for Read Bulk Registers(非標準 Function Code)

● Sample rate change :

- Slave ID: 0x01
- Function Code: 0x06
- Register Address: 0x0001
- Data Bytes: 0x01
- Data: 0x0640
- CRC: 0x5ADA

Byte	Description
0x01	Slave ID
0x06	Function Code 06 (Write Single Register)
0x0001	感測器取樣率改變
0x0640	0x0640: 1600 sps 0x00: 停止轉換
0x5ADA	Cycle Redundancy Check (CRC)

1. 設定感測器的振動取樣值的開始(Data Ratio)與結束(0x00)

Hex format: 0x01, 0x06, 0x00, 0x01, 0x06, 0x40, 0xDA, 0x5A

2. 需先發送 Sample rate change 命令，觸發感測器開始轉換振動值。

讀取振動感測器最新一筆的 3 軸振動值，需經過單位換算後可得 Gravity。

```
mMB_WCOM.WriteSingleRegister(1, (ushort)FC_Regs.FC_RW_ODR, (ushort)samplerate); //設定 Sample rate
Vib_dat = mMB_WCOM.ReadInputRegisters(1, (ushort)FC_Regs.FC_RO_DataLength, 1); //取得資料緩衝區大小
Sized = (UInt16)((Vib_dat[0] >= 123) ? (123 + 1) : (Vib_dat[0] + 1));
Vib_dat = mMB_WCOM.ReadInputRegisters(1, (ushort)FC_Regs.FC_RO_DataLength, Sized);
//Vib_dat[0]: 剩餘振動資料大小
//根據Sized的大小來存取Vib_dat陣列，例如: Vib_dat[1]: X軸振動 / Vib_dat[2]: Y軸振動 / Vib_dat[2]: Z軸振動 / Vib_dat[3]: X軸振動 ... 依此類推
```

● Read Temperature :

- Slave ID: 0x01
- Function Code: 0x03
- Register Address: 0x0014
- Data Bytes: 0x01
- CRC: 0x0EC4

Byte	Description
0x01	Slave ID
0x03	Function Code 03 (Read Holding Register)
0x0014	讀取待測物表面溫度
0x01	欲讀取的資料長度
0x0EC4	Cycle Redundancy Check (CRC)

1. 讀取振動感測器上溫度數值，此數值為 ADC code 乘上 0.0078125，即為溫度讀值；因感測器內部工作產生部份熱源需進行溫度校正。

2. 校正公式：目標溫度 = 1.133 \* (ADC code \* 0.0078125) - 7.963

Hex format: 0x01, 0x03, 0x00, 0x14, 0x00, 0x01, 0xC4, 0x0E

● Raw data FIFO buffer size :

- Slave ID: 0x01
- Function Code: 0x04
- Register Address: 0x0002
- Data Bytes: 0x01
- CRC: 0x0A90

Byte	Description
0x01	Slave ID
0x04	Function Code 04 (Read Input Register)
0x02	讀取資料緩衝大小
0x01	欲讀取的資料長度
0x4B78	Cycle Redundancy Check (CRC)

1. 需先發送 Sample rate change 命令，觸發感測器開始轉換振動值。
2. 讀取振動資料緩衝大小(PC 響應速度快，數值愈小，若為 0 表示目前無振動資料可供讀取；數值愈大表示 PC 讀取速度不夠快，資料堆積在 MCU 內)。

Hex format: 0x01, 0x04, 0x00, 0x02, 0x00, 0x01, 0x90, 0x0A

● Raw data(XYZ) :

- Slave ID: 0x01
- Function Code: 0x04
- Register Address: 0x03 ~ 0x7D
- Data Length: 3 ~ 123
- CRC: 0x?

Byte	Description
0x01	Slave ID
0x06	Function Code 04 (Read Input Register)
0x03 ~ 0x7D	0x03, 0x04, 0x05(XYZ) ~ 0x7B, 0x7C, 0x7D
3 ~ 123	
0x?	Cycle Redundancy Check (CRC)

根據振動資料緩衝大小，將振動資料依序讀出(需連同 FIFO buffer size(0x02)一起讀出，做為下一次可讀取資料的大小)，需經過單位換算後可得 Gravity。

● Bulk transfer mode :

- Slave ID: 0x01
- Function Code: 0x06
- Register Address: 0x0015
- Data Bytes: 0x01
- CRC: 0x?

Byte	Description
0x01	Slave ID
0x03	Function Code 06 (Write Input Register)
0x0015	設定 Bulk transfer block size
0x01	欲讀取的 block size 大小 0: 禁能 Bulk transfer function 1 ~ 15: 可設定 block size 的大小，建議長度為 9
0x?	Cycle Redundancy Check (CRC)

1. 設定 Block transfer mode 的 block size，每個 block 的長度為 123 (3 軸 \* 41 SPS)筆資料，此長度為 41 次的取樣週期。
2. Latency 設定流程如下圖所示，在 Modbus RTU 協議下務必設定 com port latency 為 1，若為 Modbus TCP 則忽略此設定。

```
mMB_VCOM = ModbusSerialMaster.CreateRtu(serialPort1);
mMB_VCOM.Transport.Retries = 3; //don't have to do retries
mMB_VCOM.Transport.ReadTimeout = 1500;
mMB_VCOM.Transport.WriteTimeout = 1500;
mMB_VCOM.Transport.WaitToRetryMilliseconds = 100;
mMB_VCOM.Transport.LatencyTimer = 1;
LogAppendText(string.Format("FTDI Latency Timer: {0}\r\n", mMB_VCOM.Transport.LatencyTimer));
```

3. ReadBulkRegisters 的傳入參數說明如下：

- startAddress：bulk transfer，每次區塊(size: 123)傳輸大小
- Bulk size：建議值為 9 ~ 25;
- Vib\_dat[0]為感測器內部剩餘資料長度。

4. ReadBulkRegisters 的返回值解析如下：

- 若為 1 ~ n：表示陣列 Vib\_dat[1..n]內為振動資料，Vib\_dat[0]為感測器內部剩餘資料長度；Vib\_dat[1..n]依序為 XYZ 3 軸振動資料，n 的長度為 3 的倍數。
- 若為 0xFFFF：表示此次不進行 Bulk transfer 功能，表示感測器內部 raw data 的數據量未達到 1230 筆資料(此為 410 次取樣週期寬度)。

```
//Modbus transfer => True: bulk, False: normal.
max_size = (ushort)mMB_Ethernet.ReadBulkRegisters((ushort)(FC_Regs.FC_WO_StreamSize + mb_idx[idx]), this.Bulk_Size, ref Vib_dat);
if (max_size == 0xFFFF)
{
    max_size = (ushort)((Vib_dat[0] >= 123) ? (123 + 1) : (Vib_dat[0] + 1));
    Vib_dat = mMB_Ethernet.ReadInputRegisters((ushort)(FC_Regs.FC_RO_DataLength + mb_idx[idx]), max_size);
}
```

- Read Chip ID :
  - Slave ID: 0x01
  - Function Code: 0x04
  - Register Address: 0x0080
  - Data Bytes: 0x03
  - CRC: 0xE3B1

Byte	Description
0x01	Slave ID
0x04	Function Code 04 (Read Input Register)
0x0080	讀取振動感測器 Chip ID
0x03	欲讀取的資料長度
0xE3B1	Cycle Redundancy Check (CRC)

讀取振動感測器 Chip ID，此為定值

Hex format: 0x01, 0x04, 0x00, 0x80, 0x00, 0x03, 0xB1, 0xE3

- Read Raw data(XYZ) latest :
  - Slave ID: 0x01
  - Function Code: 0x04
  - Register Address: 0x0083
  - Data Bytes: 0x03
  - CRC: 0x?

Byte	Description
0x01	Slave ID
0x04	Function Code 04 (Read Input Register)
0x0083	讀取最新一筆 3 軸振動值
0x03	欲讀取的資料長度
0x?	Cycle Redundancy Check (CRC)

3. 需先發送 Sample rate change 命令，觸發感測器開始轉換振動值。
4. 讀取振動感測器最新一筆的 3 軸振動值，需經過單位換算後可得 Gravity。

- Gravity RMS / Peak / Crest Factor / Skewness / Kurtosis :

- Slave ID: 0x01
- Function Code: 0x03
- Register Address: 0x1E / 0x1F / 0x20 / 0x21 / 0x22
- Data Length: 3
- CRC: 0x?

Byte	Description
0x01	Slave ID
0x03	Function Code 03 (Read Holding Register)
0x1E ~ 0x22	讀取感測器 RMS(0x1E)/Peak(0x1F)/Crest Factor(0x20)/Skewness(0x21)/Kurtosis(0x22)
0x03	設定讀取長度為 3
0x?	Cycle Redundancy Check (CRC)

1. 使用 Read Holding 命令讀取特徵參數，需將讀取資料長度需固定為 3，一次讀取 3 軸特徵參數。如下圖所示，為讀取範例及數據換算單位計算。



2. RMS / Peak / Crest Factor 建議最大更新頻率：5 Hz

3. Skewness / Kurtosis 建議最大更新週期為 2~5 秒。

```
Vib1_dat = mMB_VCOM.ReadHoldingRegisters(1, (ushort)FC_Regs.FC_RO_XYZ_ACCRMS, 3);
Console.WriteLine("Gravity(9.8 m/s^2): {0} / {1} / {2}", ((double)Vib1_dat[0] / 1000).ToString("f3"),
    ((double)Vib1_dat[1] / 1000).ToString("f3"),
    ((double)Vib1_dat[2] / 1000).ToString("f3"));

Application.DoEvents();
Vib1_dat = mMB_VCOM.ReadHoldingRegisters(1, (ushort)FC_Regs.FC_RO_XYZ_ACCPEAK, 3);
Console.WriteLine(", Peak(9.8 m/s^2): {0} / {1} / {2}", ((double)Vib1_dat[0] / 1000).ToString("f3"),
    ((double)Vib1_dat[1] / 1000).ToString("f3"),
    ((double)Vib1_dat[2] / 1000).ToString("f3"));
```

```
FC_RO_XYZ_ACCRMS = 30, //Acceleration RMS, Value(G, 9.8 m/s^2) = Register value / 1000
FC_RO_XYZ_ACCPEAK = 31, //Acceleration Peak, Value(G, 9.8 m/s^2) = Register value / 1000
FC_RO_XYZ_ACCCREST = 32, //Acceleration Crest Factor = Register value / 1000
FC_RO_XYZ_ACCSKEWNESS = 33, //Acceleration Skewness = Register value / 1000
FC_RO_XYZ_ACCKURTOSIS = 34, //Acceleration Kurtosis = Register value / 1000

FC_RO_XYZ_ACC1RMS = 40, //Acceleration RMS, Value(m/s^2) = Register value / 100
FC_RO_XYZ_ACC1PEAK = 41, //Acceleration Peak, Value(m/s^2) = Register value / 100
```

4. 相關 4 種公式說明如下表所示：

Moment number	Name	Measure of	Formula
1	Mean	Central tendency	$\bar{X} = \frac{\sum_{i=1}^N X_i}{N}$
2	Variance	Dispersion	$\sigma^2 = \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N}$
3	Skewness	Symmetry (Positive or Negative)	$Skew = \frac{1}{N} \sum_{i=1}^N \left[ \frac{(X_i - \bar{X})}{\sigma} \right]^3$
4	Kurtosis	Shape (Tall or Flat)	$Kurt = \frac{1}{N} \sum_{i=1}^N \left[ \frac{(X_i - \bar{X})}{\sigma} \right]^4$



● Velocity RMS / Peak / Crest Factor(Periodic Read) :

- Slave ID: 0x01
- Function Code: 0x03
- Register Address: 0x32 / 0x33 / 0x34
- Data Length: 3
- CRC: 0x?

Byte	Description
0x01	Slave ID
0x03	Function Code 03 (Read Holding Register)
0x32 ~ 0x34	讀取感測器 RMS(0x32)/Peak(0x33)/Crest Factor(0x34)
0x03	設定讀取長度為 3
0x?	Cycle Redundancy Check (CRC)

1. 週期性特徵值讀取用途，建議讀取週期：0.2 秒(5 Hz)。
2. 使用 Read Holding 命令讀取特徵參數，需將讀取資料長度需固定為 3，一次讀取 3 軸特徵參數。如下圖所示，為讀取範例及數據換算單位計算。
3. RMS / Peak / Crest Factor 建議最大更新頻率：5 Hz

```
Vib1_dat = mMB_VCOM.ReadHoldingRegisters(1, (ushort)FC_Regs.FC_RO_XYZ_VELRMS, 3);
Console.WriteLine("Velocity(mm/s): {0} / {1} / {2}", ((double)Vib1_dat[0] / 100).ToString("f2"),
    ((double)Vib1_dat[1] / 100).ToString("f2"),
    ((double)Vib1_dat[2] / 100).ToString("f2"));

Vib1_dat = mMB_VCOM.ReadHoldingRegisters(1, (ushort)FC_Regs.FC_RO_XYZ_VELPEAK, 3);
Console.WriteLine("Peak(mm/s): {0} / {1} / {2}", ((double)Vib1_dat[0] / 100).ToString("f2"),
    ((double)Vib1_dat[1] / 100).ToString("f2"),
    ((double)Vib1_dat[2] / 100).ToString("f2"));

Vib1_dat = mMB_VCOM.ReadHoldingRegisters(1, (ushort)FC_Regs.FC_RO_XYZ_VELCREST, 3);
Console.WriteLine("Crest Factor: {0} / {1} / {2}", ((double)Vib1_dat[0] / 1000).ToString("f3"),
    ((double)Vib1_dat[1] / 1000).ToString("f3"),
    ((double)Vib1_dat[2] / 1000).ToString("f3"));

FC_RO_XYZ_VELRMS = 50, //Velocity RMS, Value(mm/s) = Register value / 100
FC_RO_XYZ_VELPEAK = 51, //Velocity Peak, Value(mm/s) = Register value / 100
FC_RO_XYZ_VELCREST = 52, //Velocity Crest Factor, Value = Register value / 1000
```

● Velocity Frequency :

- Slave ID: 0x01
- Function Code: 0x03
- Register Address: 0x3C
- Data Length: 3
- CRC: 0x?

Byte	Description
0x01	Slave ID
0x03	Function Code 03 (Read Holding Register)
0x3C	讀取感測器 Velocity Spectrums
0x03	設定讀取長度為 3
0x?	Cycle Redundancy Check (CRC)

1. 使用 Read Holding 命令讀取特徵參數，需將讀取資料長度需固定為 3，一次讀取 3 軸特徵參數。如下圖所示，為讀取範例及數據換算單位計算。
2. Velocity Frequency 建議更新週期為 2 ~ 5 Hz。

```
Vib1_dat = mMB_VCOM.ReadHoldingRegisters(1, (ushort)FC_Regs.FC_RO_XYZ_VELFREQ, 3);  
Console.WriteLine("FREQ: {0} / {1} / {2}", ((double)Vib1_dat[0] / 10).ToString("f1"),  
                ((double)Vib1_dat[1] / 10).ToString("f1"),  
                ((double)Vib1_dat[2] / 10).ToString("f1"));
```

```
FC_RO_XYZ_VELFREQ = 60,    //Velocity MainFreq, Value(hz) = Register value / 10
```