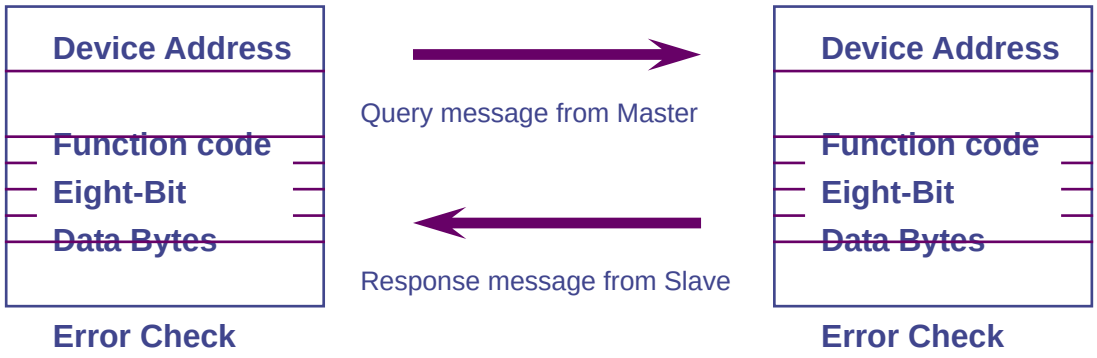
第二章：Modbus 通信协议说明

Modbus 通信协议基本上是遵循 Master and Slave 的通信步骤，有一方扮演 Master 角色采取主动询问方式，送出 Query Message 给 Slave 方，然后由 Slave 方依据接到的 Query Message 内容准备 Response Message 回传给 Master。即使目前硬件通信已经可以达到双方互相主动通信的能力，但是于 Modbus 通信协议的规定，必须一方为 Master，另一方为 Slave 不能互换角色。一般使用上，监控系统(HMI)都为 Master，PLC、电表、仪表等都为 Slave，HMI 系统一直 Polling Slave 的各种 relay and register 最新数值，然后做显示及各种逻辑计算及控制调整等处理。

1 共享的通信协议

1.1 Query and Response Cycle



图(2-1)：Master / Slave and Query / Response Cycle

Device Address：表示该设备的编号，于同一个串行式网络上此为唯一的号码。于 TCP/IP 上可以使用 IP Address 区分之，所以该 Device Address 保留此字段可以使用或不使用。

Function Code：表示要求 Slave 处理各种不同资料或程序的 Command，以不同的 Function Number 来区分之。

Eight-Bit Data Bytes：依据 Function Code 而有不同的详细资料定义，Slave 设备依据此两字段资料，做各种处理。

Error Check：当通信传送资料时，因考虑信号可能会受外界干扰，所以必须加上 Error Check Code，使得 message 接收方可以就接到的资料再计算一次 Code，如果正确则做正常处理，不正确则不做处理。于串行式通信规定有 CRC and LRC 等两种方式。于 TCP/IP 通信，因为通信 Error Check 已经被 TCP/IP 的阶层处理掉，所以于 Modbus/TCP 通信协议上不用此字段。

1.2 基本资料格式（Data Format）说明

以上 Query and Response Message 基本格式如下图所示。所有资料格式最大长度为 256 字符。

- RTU 格式：

开始间隔	Device Address	Function Code	Data	CRC check	结束间隔
T1-T2-T3-T4	8 Bits	8 Bits	Number of 8 Bits	16 Bits	T1-T2-T3-T4

图(2-2)：RTU data format

- T1-T2-T3-T4：RTU 规定每次 Query 或 Response Message 的结束，是以未再接到下一个字符间隔时间来判断。其规定为 3.5 字符的通信时间，举例来说：通信速率为 9600 bps、每个字符含 8 bits 再加上 1 start bit 及 1 stop bit 后，一个字符为 10 bits。计算 3.5 字符的通信时间为 (3.5 * 10) / 9600 = 0.00365 秒。于通信协议的文件上以 T1-T2-T3-T4 来表示此数值。
- 所有传送的字符都是照原值 8 Bits 传送，不做任何处理加工。
- Data：Number of 8 Bits 是表示每个 Function Code 有不同数目的详细资料规定。
- 通信资料的 Error Check 采用 CRC 计算方式，于「串行式通信协议」小节内说明。

- ASCII 格式：

起始字符	Device Address	Function Code	Data	LRC check	结束字符
:	2 字符	2 字符	数个字符	2 字符	2 字符 <CR> <LF>

图(2-3)：ASCII data format

- 起始字符及结束字符：因为所传送资料都是为 ASCII 码，以十六进制表示，也即是一定为 0123456789ABCDEF 等 16 个 ASCII 码。所以用特殊的字符规定开始或结束。
- 由 Device Address 至 Data 等资料，都是将 8 bits 原始值转换为两码的十六进制 ASCII 码，所以其实际传送的字符数约为 RTU 格式的两倍。
- Data：数个字符是表示每个 Function Code 有不同数目的详细资料规定。
- 通信资料的 Error Check 采用 LRC 计算方式，于「串行式通信协议」小节内说明。

- TCP/IP 格式：

起始字符组	Device Address	Function Code	Data	Error check	结束规定
6 个起始字符	8 Bits	8 Bits	Number of 8 Bits	不使用	不使用

图(2-4)：TCP/IP data format

- 起始字符组：于前面再多加 6 个字符，以定义一些 TCP/IP 的需要系数。说明如下：
  Byte 0：本次通信 Message 的编号以 2 bytes 整数（Byte 0、1）表示，此 byte 为上字符，一般是由 Master 编号之，以区分每次 Message。如果是 Slave 则将 Master 传来的 Query Message 照转至 Response Message。
  Byte 1：本次通信 Message 的编号下字符。
  Byte 2：通信协议识别号码以 2 bytes 整数（Byte 2、3）表示，此 byte 为上字符，于此处为零。
  Byte 3：通信协议识别号码下字符，于此处为零。
  Byte 4：Message 长度以 2 bytes 整数（Byte 4、5）表示，此 byte 为上字符（由 Device Address 至 Data 为止），因为长度不能超过 256 位，所以此位永远为零。
  Byte 5：Message 长度下字符（由 Device Address 至 Data 为止）。
- 由 Device Address 至 Data 内容同 RTU 格式。
- Modbus 规定 IP Port No. 为 502。

- 举例说明三种格式：

Function Code-3 读取 Output Register 数值为例。Device address：6。Start Address：40123（Modbus 规定 Output Register 由 40001 开始）。通信协议内则将 40001 去除，以 122 表示也就是十六进制 0x007A、读取点数：3。

Query Message	通信内容 十六进制	ASCII Code	RTU 8-bits field 二进制	TCP 8-bits field 二进制
TCP Byte-0				0000 0000
TCP Byte-1				0000 0001
TCP Byte-2				0000 0000
TCP Byte-3				0000 0000
TCP Byte-4				0000 0000
TCP Byte-5				0000 0110
ASCII 起始字符		:		
Device Address	06	0 6	0000 0110	0000 0110

Function Code	03	0 3	0000 0011	0000 0011
Start Address (Hi byte)	00	0 0	0000 0000	0000 0000
Start Address (Lo byte)	7A	7 A	0111 1010	0111 1010
No. of registers (Hi byte)	00	0 0	0000 0000	0000 0000
No. of register (Lo byte)	03	0 3	0000 0011	0000 0011
Error Check Byte-0				
Error Check Byte-1				
结束字符		<CR><LF>		

图(2-5)：Example of Query Message

回传的 3 点 register 数值为 789、12345、-567 也就是十六进制 0x0315、0x3039、0xFDC9 等

Response Message	通信内容 十六进制	ASCII Code	RTU 8-bits field 二进制	TCP 8-bits field 二进制
TCP Byte-0				0000 0000
TCP Byte-1				0000 0001
TCP Byte-2				0000 0000
TCP Byte-3				0000 0000
TCP Byte-4				0000 0000
TCP Byte-5				0000 1001
ASCII 起始字符		:		
Device Address	06	0 6	0000 0110	0000 0110
Function Code	03	0 3	0000 0011	0000 0011
Byte count	06	0 6	0000 0110	0000 0110
Data-1 (Hi byte)	03	0 3	0000 0011	0000 0011
Data-1 (Lo byte)	15	1 5	0001 0101	0001 0101
Data-2 (Hi byte)	30	3 0	0011 0000	0011 0000
Data-2 (Lo byte)	39	3 9	0011 1001	0011 1001
Data-3 (Hi byte)	FD	F D	1111 1101	1111 1101
Data-3 (Lo byte)	C9	C 9	1100 1001	1100 1001
Error Check Byte-0				
Error Check Byte-1				
结束字符		<CR><LF>		

图(2-6)：Example of Response Message

1.3 Function Code 说明

Function Code 有二十几种，但是一般使用上都以 1、2、3、4、5、6、15、16 等八种最为常用，以及另外特殊使用的 20、21 两种，此为 General Reference Register，绝大部份的 Modbus 设备并不会提供此 Register。于 PLC 上主要的控制数据有下列四种型式。此八种 Function Code 就是处理这些控制资料，详细说明如下各点：

控制数据四种型式：

- DI：Digital Input，以一个 bit 表示 On/Off，用来记录控制信号的状态输入，例如：开关，接触点，马达运转，超限 switch…等等。于 PLC 上被称为 Input relay、input coil 等。
- DO：Digital Output，以一个 bit 表示 On/Off，用来输出控制信号，以激活或停止马达，警铃，灯光…等等。于 PLC 上被称为 Output relay、Output coil 等。
- AI：Analog Input，以 16 bits integer 表示一个数值，用来记录控制信号的数值输入，例如：温度、流量、料量、速度、转速、文件板开度、液位、重量…等等。于 PLC 上被称为 Input register。
- AO：Analog Output，以 16 bits integer 表示一个数值，用来输出控制信号的数值，例如：温度、流量、速度、转速、文件板开度、饲料量…等等设定值。于 PLC 上被称为 Output register、Holding register。

Modbus Function Code	说明
01	Read Coil Status (output relay)
02	Read Input Status (input relay)
03	Read Holding Registers (output register)
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
07	Read Exception Status
08	Diagnostics
09	Program 484
10	Poll 484
11	Fetch Comm. Event Ctr.
12	Fetch Comm. Event Log
13	Program Controller
14	Poll Controller
15	Force Multiple Coils
16	Preset Multiple Registers
17	Report Slave ID
18	Program 884/M84
19	Reset Comm. Link
20	Read General Reference
21	Write General Reference
22	Mask Write 4x Register
23	Read/Write 4x Register
24	Read FIFO Queue
43	Read Device Identication
65 to 72	开放给一般使用者定义
100 to 110	开放给一般使用者定义

图(2-7)：Modbus Function Code 一览表

以下说明常用 10 种 Function Code：每种 Function Code 以举例说明其中间 Message 的格式至于前面的 TCP Byte，ASCII 起始字符及后面的 Error Check、结束字符等都是一样规定，参考前一节内容即可。

Function Code-1：读取 DI 资料，于 Modbus 规定 Relay Address 由 00001 开始。但是通信协议内取后面四位数，且由零起算，例如：于文件上 Relay Address 为 00345，其通信协议内转换的 Address 为 344。

由 Device Address 17 读取 Relay Address 20 – 32 的 DI 资料，通信协议内 Start Address 为 19，读取点数 13。

Query Message	通信内容 十六进制
Device Address	11
Function Code	01
Start Address (Hi byte)	00
Start Address (Lo byte)	13
No. of points (Hi byte)	00
No. of points (Lo byte)	0D

回传资料以一个 Byte 即 8 bits 为一组，每一个 Bit 表示一点 Relay On/Off 状态。例如下表
Data(Relay 27 – 20)的状态为 ON-ON-OFF-ON-OFF-OFF-ON-ON。
Data(Relay 32 – 28)的状态为 OFF-ON-OFF-ON-ON-ON。（前面 3 bit 不算，因为只到 Relay 32 为止）
其个别 Relay 状态，举例： Relay 27 为 ON、Relay 31 为 ON、Relay 23 为 OFF。

Response Message	通信内容 十六进制
Device Address	11
Function Code	01
Byte count	02
Data (Relay 27 – 20)	D3
Data (Relay 32 – 28)	17

图(2-8)：Example of Function Code - 1 Message

Function Code-2：读取 DO 资料，于 Modbus 规定 Relay Address 由 10001 开始。但是通信协议内取后面四位数，且由零起算，例如：于文件上 Relay Address 为 10678，其通信协议内转换的 Address 为 677。

由 Device Address 23 读取 Relay Address 10102 – 10134 的 D0 资料，通信协议内 Start Address 为 101，读取点数 33。

Query Message	通信内容 十六进制
Device Address	17
Function Code	02
Start Address (Hi byte)	00
Start Address (Lo byte)	65
No. of points (Hi byte)	00
No. of points (Lo byte)	21

回传资料以一个 Byte 即 8 bits 为一组，每一个 Bit 表示一点 Relay On/Off 状态。例如下表
Data(Relay 109 – 102)的状态为 ON-OFF-ON-OFF-ON-OFF-ON-OFF。
Data(Relay 117 – 110)的状态为 OFF-ON-OFF-OFF-OFF-ON-OFF-ON。
Data(Relay 125 – 118)的状态为 OFF-OFF-ON-OFF-OFF-ON-ON-ON。
Data(Relay 133 – 126)的状态为 ON-OFF-OFF-OFF-OFF-OFF-ON-ON。
Data(Relay 134 – 134)的状态为 ON。（前面 7 bit 不算，因为只到 Relay 134 为止）

Response Message	通信内容 十六进制
Device Address	17
Function Code	02
Byte count	05
Data (Relay 109 – 102)	AA
Data (Relay 117 – 110)	45
Data (Relay 125 – 118)	27
Data (Relay 133 – 126)	83
Data (Relay 134 – 134)	01

图(2-9)：Example of Function Code - 2 Message

Function Code-3：读取 AO 资料，于 Modbus 规定 Register Address 由 40001 开始。但是通信协议内取后面四位数，且由零起算，例如：于文件上 Register Address 为 44321，其通信协议内转换的 Address 为 4320。

由 Device Address 41 读取 Register Address 40765 – 40770 的 A0 资料，通信协议内 Start Address 为 764，读取点数 6。

Query Message	通信内容 十六进制
Device Address	29
Function Code	03
Start Address (Hi byte)	02
Start Address (Lo byte)	FC
No. of registers (Hi byte)	00
No. of registers (Lo byte)	06

回传资料以两个 Bytes 表示 16 bits 整数值。例如下表

Register 40765 整数值：99

Register 40766 整数值：12336

Register 40767 整数值：－1417

Register 40768 整数值：789

Register 40769 整数值：767

Register 40770 整数值：1

Response Message	通信内容 十六进制
Device Address	29
Function Code	03
Byte count	0C
Data-1 (Hi byte)	00
Data-1 (Lo byte)	63
Data-2 (Hi byte)	30
Data-2 (Lo byte)	30
Data-3 (Hi byte)	FA
Data-3 (Lo byte)	77
Data-4 (Hi byte)	03
Data-4 (Lo byte)	15
Data-5 (Hi byte)	02
Data-5 (Lo byte)	FF
Data-6 (Hi byte)	00
Data-6 (Lo byte)	01

图(2-10)：Example of Function Code - 3 Message

Function Code-4：读取 AI 资料，于 Modbus 规定 Register Address 由 30001 开始。但是通信协议内取后面四位数，且由零起算，例如：于文件上 Relay Address 为 30988，其通信协议内转换的 Address 为 987。

由 Device Address 30 读取 Register Address 30123 – 30127 的 AI 资料，通信协议内 Start Address 为 122，读取点数 5。

Query Message	通信内容 十六进制
Device Address	1E
Function Code	04
Start Address (Hi byte)	00
Start Address (Lo byte)	7A
No. of registers (Hi byte)	00
No. of registers (Lo byte)	05

回传资料以两个 Bytes 表示 16 bits 整数值。例如下表
Register 30123 整数值：2581
Register 30124 整数值：57
Register 30125 整数值：969
Register 30126 整数值：－24544
Register 30127 整数值：170

Response Message	通信内容 十六进制
Device Address	1E
Function Code	04
Byte count	0A
Data-1 (Hi byte)	0A
Data-1 (Lo byte)	15
Data-2 (Hi byte)	00
Data-2 (Lo byte)	39
Data-3 (Hi byte)	03
Data-3 (Lo byte)	C9
Data-4 (Hi byte)	A0
Data-4 (Lo byte)	20
Data-5 (Hi byte)	00
Data-5 (Lo byte)	AA

图(2-11)：Example of Function Code - 4 Message

Function Code-5：写入 DO 一点资料，Address 规定与 Function Code-2 一样。

由 Device Address 10 写入 Relay Address 10012 的 D0 资料，通信协议内 Start Address 为 11。如果设定为 ON 于 Force Data 设定十六进制 0xFF00，如果设定为 OFF 于 Force Data 设定十六进制 0x0000。

Query Message	通信内容 十六进制
Device Address	0A
Function Code	05
Start Address (Hi byte)	00
Start Address (Lo byte)	0B
Force Data (Hi byte)	FF
Force Data (Lo byte)	00

以 Query Message 作为 Response Message 传回。

Response Message	通信内容 十六进制
Device Address	0A
Function Code	05
Start Address (Hi byte)	00
Start Address (Lo byte)	0B
Force Data (Hi byte)	FF
Force Data (Lo byte)	00

图(2-12)：Example of Function Code - 5 Message

Function Code-6：写入 AO 一点资料，Address 规定与 Function Code-3 一样。

由 Device Address 13 写入 Register Address 40112 的 A0 资料，通信协议内 Start Address 为 111。设定 16 bits 整数值为 999 即是十六进制 0x03E7。

Query Message	通信内容 十六进制
Device Address	0D
Function Code	06
Start Address (Hi byte)	00
Start Address (Lo byte)	6F
Preset Data (Hi byte)	03
Preset Data (Lo byte)	E7

以 Query Message 作为 Response Message 传回。

Response Message	通信内容 十六进制
Device Address	0D
Function Code	06
Start Address (Hi byte)	00
Start Address (Lo byte)	6F
Preset Data (Hi byte)	03
Preset Data (Lo byte)	E7

图(2-13)：Example of Function Code - 6 Message

Function Code-15：写入 DO 多点资料，Address 规定与 Function Code-2 一样。

由 Device Address 17 写入 Relay Address 10011 – 10022 的 D0 资料，通信协议内 Start Address 为 10，写入点数 12。所设定状态如下：

第一个 Force Data byte

Bit	0	1	0	1	0	1	0	1
Relay	18	17	16	15	14	13	12	11

第二个 Force Data byte

Bit	0	0	0	0	0	0	1	1
Relay	-	-	-	-	22	21	20	19

Query Message	通信内容 十六进制
Device Address	11
Function Code	0F
Start Address (Hi byte)	00
Start Address (Lo byte)	0A
No. of relay (Hi byte)	00
No. of relay (Lo byte)	0C
Byte Count	02
Force Data (Relay 18 – 11)	55
Force Data (Relay 22 – 19)	03

以 Query Message 前面 6 bytes 作为 Response Message 回传。

Response Message	通信内容 十六进制
Device Address	11
Function Code	0F
Start Address (Hi byte)	00
Start Address (Lo byte)	12
No. of relay (Hi byte)	00
No. of relay (Lo byte)	03

图(2-14)：Example of Function Code - 15 Message

Function Code-16：写入 AO 多点资料，Address 规定与 Function Code-3 一样。

由 Device Address 39 写入 Register Address 40310 – 40312 的 A0 资料，通信协议内 Start Address 为 309，写入点数 3。写入数值如下所示：每个 register 数值为 16 bits 整数值。
Register 40310 设定值：784
Register 40311 设定值：12706
Register 40312 设定值：－16183

Query Message	通信内容 十六进制
Device Address	27
Function Code	10
Start Address (Hi byte)	01
Start Address (Lo byte)	35
No. of registers (Hi byte)	00
No. of register (Lo byte)	03
Byte count	06
Data-1 (Hi byte)	03
Data-1 (Lo byte)	10
Data-2 (Hi byte)	31
Data-2 (Lo byte)	A2
Data-3 (Hi byte)	C0
Data-3 (Lo byte)	C9

以 Query Message 前面 6 bytes 作为 Response Message 回传。

Response Message	通信内容 十六进制
Device Address	27
Function Code	10
Start Address (Hi byte)	01
Start Address (Lo byte)	35
No. of registers (Hi byte)	00
No. of register (Lo byte)	03

图(2-15)：Example of Function Code - 16 Message

Function Code-20：读取 General Reference Register，此为 Extended Memory File 。此 Register 可分成 10 个 File 编号为 File No. 1 - 10，每个 File 开头 Address 都为 600000。每个 Register 都可以读取或写入。注意此 Register Address 使用者定义时 600000 起算，Protocol 内部 offset 由 0 起算。与 Holding Regsiter（使用者定义 40001 起算，Protocol 内部 offset 由 0 起算）稍有不同。本 Function Code 可以分成多数个 Group 读取不同 Address 的资料。

由 Device Address 17 读取下列两个 Group 的 Register 值
Group 1 由 File 4、开始 Address offset 为 1、读取 2 点 Register。
Group 2 由 File 3、开始 Address offset 为 9、读取 2 点 Register。

Query Message	通信内容 十六进制
Device Address	11
Function Code	14
Byte Count	0E
Group-1 Reference Type	06
Group-1 File No. (Hi byte)	00
Group-1 File No. (Lo byte)	04
Group-1 Start Addr. (Hi byte)	00
Group-1 Start Addr.(Lo byte)	01
Group-1 Registers Count (Hi byte)	00
Group-1 Registers Count (Lo byte)	02
Group-2 Reference Type	06
Group-2 File No. (Hi byte)	00
Group-2 File No. (Lo byte)	03
Group-2 Start Addr. (Hi byte)	00
Group-2 Start Addr.(Lo byte)	09
Group-2 Registers Count (Hi byte)	00
Group-2 Registers Count (Lo byte)	02

Response Message 如下。

Response Message	通信内容 十六进制
Device Address	11
Function Code	14
Byte Count	0C
Group-1 Byte Count	05
Group-1 Reference Type	06
Group-1 Data-1 (Hi byte)	0D
Group-1 Data-1 (Lo byte)	FE
Group-1 Data-2 (Hi byte)	00
Group-1 Data-2 (Lo byte)	20
Group-2 Byte Count	05
Group-2 Reference Type	06
Group-2 Data-1 (Hi byte)	33
Group-2 Data-1 (Lo byte)	CD
Group-2 Data-2 (Hi byte)	00
Group-2 Data-2 (Lo byte)	40

图(2-16)：Example of Function Code - 20 Message

Function Code-21：写入 General Reference Register，Address 规定与 Function Code-20 一样。

由 Device Address 17 写入下列一个 Group 的 Register 值
Group 1 由 File 4、开始 Address offset 为 7、写入 3 点 Register。
Register 600007 设定值：1711
Register 600008 设定值：1214
Register 600009 设定值：4109

Query Message	通信内容 十六进制
Device Address	11
Function Code	15
Byte Count	0D
Group-1 Reference Type	06
Group-1 File No. (Hi byte)	00
Group-1 File No. (Lo byte)	04
Group-1 Start Address (Hi byte)	00
Group-1 Start Address (Lo byte)	07
Group-1 Registers Count (Hi byte)	00
Group-1 Registers Count (Lo byte)	03
Group-1 Data-1 (Hi byte)	06
Group-1 Data-1 (Lo byte)	AF
Group-1 Data-2 (Hi byte)	04
Group-1 Data-2 (Lo byte)	BE
Group-1 Data-3 (Hi byte)	10
Group-1 Data-3 (Lo byte)	0D

以 Query Message 全部 bytes 作为 Response Message 回传。

图(2-17)：Example of Function Code - 21 Message

1.4 Exception Code 说明

异常码（Exception Code）是用来表示，当有任何通信资料异常时，由 Slave 不回传正常资料，而回传错误码号（Error Code）以提供 Master 做异常处理。其 Message 格式以下表为例说明：

由 Device Address 13 写入 Register Address 42450 的 A0 资料，通信协议内 Start Address 为 2449。设定 16 bits 整数值为 999 即是十六进制 0x03E7。

Query Message	通信内容 十六进制
Device Address	0D
Function Code	06
Start Address (Hi byte)	09
Start Address (Lo byte)	91
Preset Data (Hi byte)	03
Preset Data (Lo byte)	E7

因为该Modbus 设备，只提供最大Register Address为42048，所以超出范围，需要回传Error Code：2（ILLEGAL DATA ADDRESS）。同时将 Function Code最左边 Bit 设定为 1 表示此Function有Exception Code。

Response Message	通信内容 十六进制
Device Address	0D
Function Code	86
Error Code	02

图(2-18)：Exception Message format

由以上可知处理 Exception Code，将原有的 Function Code 的最左边 Bit 设定为 1，然后将适合的 Error code 代入。

Error Code	内容说明
1	ILLEGAL FUNCTION 此 Function Code 不能被 Slave 所处理。可能 Function Code 错误或此 Slave 设备未定义此 Function。
2	ILLEGAL DATA ADDRESS 所要求的 Address 超出范围。最常见的错误为 Start Address 加上处理的点数超出 Address 最大值。
3	ILLEGAL DATA VALUE 所传送过来的数值不合 Slave 的规定。例如：Function Code-6 为设定某一点 Coil ON 或 OFF，其 Value 只能有两个值 ON: 0xFF00、OFF: 0x0000，如果是其它值则为不合法。

4	SLAVE DEVICE FAILURE 当 Slave 处理所要求的 Function Code，发生不可预期的错误。
5	ACKNOWLEDGE 当 Slave 接到一种 Function Code 需要较长时间处理时，避免 Master 等待 Response Message 未到而产生通信逾时。所以先送此 Code，然后 Master 再以 Polling Program Function Code 要求处理结果。
6	SLAVE DEVICE BUSY Slave 正处理其它事情，目前没有时间处理所要求的 Function Code，先回传此 Error Code，然后 Master 稍后传送再一次 Query Message。
7	NEGATIVE ACKNOWLEDGE 以 Function Code-13、14 要求处理时，Slave 无法处理则回传此 Error code，然后 Slave 必须再要求 Slave 传送检查（diagnostic）后讯息。
8	MEMORY PARITY ERROR 使用于 Function Code-20、21 的 extension memory 的处理，当此 memory 发生 parity 错误时，回传此 Error Code。
9	未定义。
10	GATEWAY PATH UNAVAILABLE 适用于 Gateway 的产品，表示 Gateway 无法于内部建立一个 input port 至 output port 的路径或处理程序等。最常见的错误是此 Gateway 未做使用上定义（configuration）。
11	GATEWAY TARGET DEVICE FAILED TO RESPOND 适用于 Gateway 的产品，当转送 Master Query Message 至目标 Slave 设备时，对方未有响应，可传送此 Error code 至 Slave。此种最常见的现象是该目标 Slave 设备并未于网络上。

图(2-19)：Exception Code 定义表

2 串行式的通信协议

2.1 Device Address 的规定

此 Device Address 是用来表示于 Modbus Device 的编号，于同一条串行式联机上，此编号必须是唯一的，才能区分出各别 Modbus Device。于 Modbus Message 内，规定 ASCII 格式为 2 bytes 字符，或 RTU 为 1 byte (8 bits)。其有效数值为 0 - 247，其中 1- 247 各别 Modbus Device 编号用，0 做为广播（broadcast）通信的特殊用途，会被所有 Slave Device 接受执行，此种 broadcast 用法，并不是为每一种 Modbus Device 的标准功能请注意。

实际使用上常常会发生的无法通信的现象，必须确认 Device Address 是否有设定或重复设定的操作，此为最常发现的错误点。

2.2 RTU 及 ASCII 两种资料格式

前一个章节已经举例说明两种资料的基本格式，于此再说明一些重要原则。列于下表

RTU	ASCII
Message 内容照原始值传送	每一个原始值字符都转换为 2 个十六进制 ASCII Code，所传送的数目为 RTU 的两倍。
以未收到下一个字符的时间间隔，来区分出每段 Message，于通信应用程序的设计上较复杂。通信协议定义是未接到 3.5 字符的时间，表示 Message 结束。但是实际上每种 Device 硬件执行效率不同，完全依照协议设计，有时还会通信断线现象。最好的设计是比协议时间稍长，以保证收到完整 Message。	以特殊的起始字符及结束字符，区分出每段 Message，于通信应用程序的设计较为简单。
字符硬件信号 1 start bit 8 data bits, least significant bit send first 1 bit for parity check 1 stop bit if parity is used; 2 bits if no parity	字符硬件信号 1 start bit 7 data bits, least significant bit send first 1 bit for parity check 1 stop bit if parity is used; 2 bits if no parity

图(2-20)：RTU and ASCII format 比较表

2.3 CRC 及 LRC 两种 check

两种 Message 防止内容错误的检查机制，是当双方要将 Message 传给对方时，必须先做每个资料字符的特定计算方式（CRC 或 LRC），将结果放置于 Message 后面。然后对方接到此 Message 时，也照同样公式计算一次，如果内容相同，则可以正常处理。如果内容不同，则可能是传送过程中，受到外部信号干扰，使得资料不对，例如某一个 Bit 原为 1 变成 0，则必须放弃此 Message 不做处理。

- CRC check：RTU 格式所使用。CRC 全名为 Cyclical Redundancy Check 是一个 16 bits 的数值，计算步骤如下。
  1. 首先将一个十六进制 0xFFFF 存入一个 unsigned short 变量内。称做 CRC Register。
  2. 将第一个 Message 8 bits byte 与 CRC Register low byte 做 Exclusive OR，将结果带入 CRC Register low byte。
  3. 将 CRC Register 往右移 1 bit（LSB 位置），于最左 bit 补 0（MSB 位置）。
  4. 取入 LSB 位置 bit，如果为 0 再进入第 3 步骤。如果为 1 将 CRC Register 与十六进制数 0xA001 Exclusive OR。
  5. 重复第 3、4 步骤，直到此 byte 的 8 bits 都处理完成。
  6. 然后每一个 Message 8 bits byte 都做同样处理，得到一个 16 bits 数值。将此数值以 Low byte and High byte 顺序置于 Message 最后面。

7. 注意Low byte and High byte顺序，此刚好与Message内 High byte and Low byte 顺序相反。

Query Message	通信内容 十六进制	RTU 二进制
Device Address	0D	0000 1101
Function Code	06	0000 0110
Start Address (Hi byte)	09	0000 1001
Start Address (Lo byte)	91	1001 0001
Preset Data (Hi byte)	03	0000 0011
Preset Data (Lo byte)	E7	1110 0111
CRC Check (Lo Byte)		1001 1011
CRC Check (Hi Byte)		1100 1101

图(2-21)：CRC 计算例

- LRC check ：ASCII 格式所使用。LRC 全名为 Longitudinal Redundancy Check。计算原理是将全部 Message 不包含起始字符「：」及结束字符<CR><LF>，由第一个至最后一个字符照顺序相加（注意：此为原始值的字符，也就是先将两个十六进制码转换回原始字符），相加之间如果产生 8 bit 以上的溢位（carries）舍弃溢位，得到一个 8 bits 位值，最后将相加结果取 1 的补码，得到一个最后值转换成两个十六进制 ASCII 码置于 Message 后面。

Query Message	通信内容 十六进制	ASCII Code
起始字符		:
Device Address	0D	0 D
Function Code	06	0 6
Start Address (Hi byte)	09	0 9
Start Address (Lo byte)	91	9 1
Preset Data (Hi byte)	03	0 3
Preset Data (Lo byte)	E7	E 7
LRC Check 字符 – 1		6
LRC Check 字符 – 2		9
结束字符 – 1		<CR> 十六进制 0x0D
结束字符 – 2		<LF> 十六进制 0x0A

图(2-22)：LRC 计算例

以上两种计算子程序会列于以下章节的实际程序例内。

3 TCP/IP 的通信协议

Modbus/TCP 主要格式已于「1.2.资料基本格式」说明清楚。此处再提醒读者的是 IP Address 与 Modbus Address 的区分，Slave 设备必须具有给外部多个 Master 联机的能力。

3.1 IP Address 与 Modbus Address 的区分

于同一 Ethernet 网络系统上 IP Address 必须是唯一的，可以说如同 Modbus Address 的唯一编号来区分串行式联机上的各别设备。其主要差别在于 IP Address 是由操作系统面所管理，Modbus Address 则是由通信协议内所制定的。于 Modbus/TCP 通信协议的定义，其 Modbus Address 字段还是被保留，但是可以不用。因此市面上 Modbus/TCP 的设备，此字段值有些永远设定为零，有些还是沿用，需要特别注意，否则可能无法联机。

3.2 Modbus Slave 设备需具备被多个 Master 联机架构

于串行式通信上只有一个Modbus Master设备，然后下面连上多台Modbus Slave设备。也就是说，每次通信一定由Modbus Master主动送出Query Message，然后某一台Slave响应Response Message，一次只有一个通信再进行。但是于Ethernet网络上，则可能有多个Modbus Master同时会联机至同一台Modbus Slave上，此时Slave系统就要有接受多个联机架构的功能。所以一般具备Modbus/TCP Slave设备，都会列出最多允许几个Master联机的规格。