



# **MT84G**

## **Serial Commands Manual**



---

## Table of Content

<b>1. Introduction .....</b>	<b>3</b>
1.1 Manual Description .....	3
1.2 Product Requirements.....	3
<b>2. Command Format .....</b>	<b>3</b>
2.1 Read Register.....	4
2.2 Write Register .....	6
2.3 Write Flash .....	8
2.4 Reset Register to Default.....	9
<b>4. Command List.....</b>	<b>10</b>
<b>Appendix.....</b>	<b>32</b>
Appendix A - Code ID .....	32
<b>Version History.....</b>	<b>33</b>

## 1. Introduction

### 1.1 Manual Description

This user's manual describes serial command format and full list of commands that can be configured over serial communication.

### 1.2 Product Requirements

Model	Firmware Version	Interface
MT84G	HS:V0.00; F:V1.1.33; 2021.09.15 or later version	UART
		USB VCP

## 2. Command Format

User can configure MT84G by sending serial commands from the host. Please make sure the communication protocols of MT84G match that of the host.

MT84G's communication protocols are:

Parameter	Default
Baud Rate	9600
Parity	None
Data Bits	8
Stop Bits	1
Hardware Flow Control	None

## 2.1 Read Register

User can read up to 256 bytes of register.

Command format:

**Send: [Head1] [Type] [Length] [Address] [Data] [CRC]**

[Head1] = 0x7E 0x00 (2 bytes)

[Type] = 0x07 (1 byte)

[Length] = 0x01 (1 byte)

[Address] = 0x0000~0x00FF (2 bytes), the starting address of register to read

[Data] = 0x00~0xFF (1 byte), the number of bytes of register to read; 0x00 means 256 bytes.

[CRC] = CRC\_CCITT checksum (2 bytes)

Calculation range: [Type] [Length] [Address] [Data]. The calculation method is CRC\_CCITT, characteristic polynomial:  $X^{16}+X^{12}+X^5+1$ . That is, the polynomial coefficient is 0x1021, and the initial value is all 0. For single byte, the highest bit is calculated first, and it is output directly without inversion. Reference code of C language is as follows:

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i !=0; i /=2)
        {
            crc *= 2;
            if((crc&0x10000) !=0) //After the last CRC is multiplied by 2, if the first digit is 1,
            then divide by 0x11021
            crc ^= 0x11021;
            if((*ptr&i) !=0) //If this bit is 1, then CRC = CRC of the previous bit + this
            bit/CRC_CCITT
            crc ^= 0x1021;
        }
        ptr++;
    }
    return crc;
}
```

\*Note: If CRC is not required, simply replace CRC\_CCITT checksum by 0xAB 0xCD.

**Respond: [Head2] [Type] [Length] [Data] [CRC]**

(1) Read success

[Head2] = 0x02 0x00 (2 bytes)

[Type] = 0x00 (1 byte), read success

[Length] = the number of bytes of data; 0x00 means 256 bytes

[Data] = 0x00~0xFF (1 byte), register status

[CRC] = CRC\_CCITT checksum

(2) CRC verification failure

No response

(3) Unknown command

No response

Example:

Read the first register from address 0x00FF

(1) Read success, with 0x3E as response

Send: 0x7E 0x00 0x07 0x01 0x00 0xFF 0x01 0x02 0xBE

Respond: 0x02 0x00 0x00 0x01 0x3E 0xE4 0xAC

(2) CRC verification failure

Send: 0x7E 0x00 0x08 0x01 0x00 0xFF 0x3E 0x11 0x22

Respond: None

(3) When the delay in between two bytes of command is over 400ms, it is treated as unknown command

Send: 0x7E 0x00 0x07 0x01 0x00 0xFF 0x01

Respond: None

## 2.2 Write Register

User can write up to 256 bytes of register.

The register status will be reset after power-off. To store register status, please perform Write Flash.

Command format:

**Send: [Head1] [Type] [Length] [Address] [Data] [CRC]**

[Head1] = 0x7E 0x00 (2 bytes)

[Type] = 0x08 (1 byte)

[Length] = 0x00~0xFF (1 byte), the number of byte in [Data] field, and the number of consecutive writes; 0x00 means to 256 bytes.

[Address] = 0x0000~0xFFFF (2 bytes), the starting address of register to write

[Data] = 0x00~0xFF (1~256 bytes), the data to write register with. To write multiple registers, make sure to write from low bit to high bit.

[CRC] = CRC\_CCITT checksum (2 bytes)

Calculation range: [Type] [Length] [Address] [Data]. The calculation method is CRC\_CCITT, characteristic polynomial:  $X^{16}+X^{12}+X^5+1$ . That is, the polynomial coefficient is 0x1021, and the initial value is all 0. For single byte, the highest bit is calculated first, and it is output directly without inversion. Reference code of C language is as follows:

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i !=0; i /=2)
        {
            crc *= 2;
            if((crc&0x10000) !=0) //After the last CRC is multiplied by 2, if the first digit is 1,
            then divide by 0x11021
            crc ^= 0x11021;
            if((*ptr&i) !=0) //If this bit is 1, then CRC = CRC of the previous bit + this
            bit/CRC_CCITT
            crc ^= 0x1021;
        }
        ptr++;
    }
}
```

```
}  
return crc;  
}
```

\*Note: If CRC is not required, simply replace CRC\_CCITT checksum by 0xAB  
0xCD.

**Respond: [Head2] [Type] [Length] [Data] [CRC]**

(1) Write success

[Head2] = 0x02 0x00 (2 bytes)

[Type] = 0x00 (1 byte), write success

[Length] = 0x01 (1 byte)

[Data] = 0x00 (1 byte)

[CRC] = 0x33 0x31 (2 bytes), CRC\_CCITT checksum

(2) CRC verification failure

No response

(3) Unknown command

No response

Example:

Write 0x3E to the register of address 0x00FF

(1) Write success

Send: 0x7E 0x00 0x08 0x01 0x00 0xFF 0x3E 0xA0 0xFB

Respond: 0x02 0x00 0x00 0x01 0x00 0x33 0x31

(2) CRC verification failure

Send: 0x7E 0x00 0x08 0x01 0x00 0xFF 0x3E 0x11 0x22

Respond: None

(3) When the command length is insufficient, or the delay after sending 0x7E

0x00 is over 400ms, it is treated as unknown command

Send: 0x7E 0x00 0x08 0x01 0x00 0xFF 0x3E

Respond: None

## 2.3 Write Flash

To store all register status in flash memory, perform Write Flash.

\*Note: It is unable to store single register status. All register status will be stored when performing Write Flash.

Command format:

**Send: [Head1] [Type] [Length] [Address] [Data] [CRC]**

[Head1] = 0x7E 0x00 (2 bytes)

[Type] = 0x09 (1 byte)

[Length] = 0x01 (1 byte)

[Address] = 0x0000 (2 bytes)

[Data] = 0x00 (1 byte)

[CRC] = 0xDE 0xC8 (2 bytes), CRC\_CCITT checksum

**Respond: [Head2] [Type] [Length] [Data] [CRC]**

(1) Store success

[Head2] = 0x02 0x00 (2 bytes)

[Type] = 0x00 (1 byte), write success

[Length] = 0x01 (1 byte)

[Data] = 0x00 (1 byte)

[CRC] = 0x33 0x31 (2 bytes), CRC\_CCITT checksum

(2) CRC verification failure

No response

(3) Unknown command

No response



## 2.4 Reset Register to Default

To reset all register status to default in flash memory, perform Reset Register to Default.

Command format:

**Send: [Head1] [Type] [Length] [Address] [Data] [CRC]**

[Head1] = 0x7E 0x00 (2 bytes)

[Type] = 0x09 (1 byte)

[Length] = 0x01 (1 byte)

[Address] = 0x0000 (2 bytes)

[Data] = 0xFF (1 byte)

[CRC] = CRC\_CCITT checksum

**Respond: [Head2] [Type] [Length] [Data] [CRC]**

(1) Reset success

[Head2] = 0x02 0x00 (2 bytes)

[Type] = 0x00 (1 byte), write success

[Length] = 0x01 (1 byte)

[Data] = 0x00 (1 byte)

[CRC] = 0x33 0x31 (2 bytes), CRC\_CCITT checksum

(2) CRC verification failure

No response

(3) Unknown command

No response

## 4. Command List

This chapter contains full list of address and register.

Address	0x0000
Register	Function
Bit7	0: Good Read LED – Off 1: Good Read LED – On
Bit6	0: Mute 1: Unmute
Bit5-4	00: Aimer – Always Off 01: Aimer – Flash 10: Aimer – Always On 11: Aimer – On
Bit3-2	00: Illumination – Always Off 01: Illumination – Normal 10/11: Illumination – Always On
Bit1-0	00: Trigger Mode 01: Serial Trigger Mode 10: Continuous Mode 11: Auto-sensing Mode
Address	0x0001
Register	Function
Bit7	Serial Trigger Mode 0: Trigger Command Response – Enable 1: Trigger Command Response – Disable
Bit6	Reserved
Bit5	Trigger Mode 0: Trigger Condition – Level 1: Trigger Condition – Pulse
Bit4	Continuous Mode 0: Trigger Control – On 1: Trigger Control – Off
Bit3-2	00: Beep Frequency – Low 01: Beep Frequency – Medium

	10: Beep Frequency – High
Bit1-0	Reserved
Address	0x0002
Register	Function
Bit7-1	Reserved
Bit0	Serial Trigger Mode – Trigger Command 1: Trigger 0: Not Trigger
Address	0x0003
Register	Function
Bit7-2	HID cycle inquiry Cycle = (Reg0x0003[7:2]+1) ms
Bit1	0: Configuration Barcode Beep – On 1: Configuration Barcode Beep – Off
Bit0	0: Disable Config Barcode Data Output 1: Enable Config Barcode Data Output
Address	0x0004
Register	Function
Bit7-0	Image Stabilization Timeout 0x00~0xFF: 0ms~25500ms
Address	0x0005
Register	Function
Bit7-0	Scan Interval 0x00~0xFF: 0ms~25500ms
Address	0x0006
Register	Function
Bit7-0	LED Timeout 0x00~0xFF: 0ms~25500ms (0x00 = unlimited)
Address	0x0007
Register	Function
Bit7	0: Sleep Mode – Disable 1: Sleep Mode – Enable

Bit6-0	Sleep Timeout (High Bit14-8) Unit = 100ms
Address	0x0008
Register	Function
Bit7-0	Sleep Timeout (Low Bit7-0) Unit = 100ms
Address	0x0009
Register	Function
Bit7-2	Delay Before HID Release Delay = (Reg0x0009[7:2]) ms
Bit1-0	Image Rotation 00: Image Rotation – Off 01: Image Rotation – On 10/11: Reserved
Address	0x000A
Register	Function
Bit7	Block Chinese characters 0: unblock Chinese characters 1: block Chinese characters
Bit6-5	Virtual Keyboard 00: Disable Ctrl Code 01: Ctrl Mode 10: Alt Mode
Bit4	0: Numeric Keypad – Number Keys Off 1: Numeric Keypad – Number Keys On
Bit3	0: Numeric Keypad –Symbol Keys Off 1: Numeric Keypad – Symbol Keys On
Bit2-1	Reserved
Bit0	HID Prefix (Shift+Ctrl+r) 0: Disable HID Prefix 1: Enable HID Prefix
Address	0x000B
Register	Function
Bit7-0	Good Read Beep Duration

	0x00~0xFF: 0ms~255ms
Address	0x000C
Register	Function
Bit7-2	Delay After HID Release Delay = (Reg0x000C[7:2]) ms
Bit1	0: CapsLock - Off 1: CapsLock - On
Bit0	Reserved
Address	0x000D
Register	Function
Bit7	Reserved
Bit6	Virtual Keyboard 0: Standard Keyboard 1: Virtual Keyboard
Bit5-4	Reserved
Bit3-2	00: Encoding Format – GBK 01: Encoding Format – UNICODE 10: Encoding Format – Raw 11: Encoding Format – UTF8
Bit1-0	Interface 00: UART 01: USB HID 10: HID & UART 11: USB VCP
Address	0x000E
Register	Function
Bit7-4	Reserved
Bit3	0: Configuration Barcode Beep – On 1: Configuration Barcode Beep – Off
Bit2	0: Good Read Beep – Off 1: Good Read Beep – On
Bit1	0: Power Up Beep – On 1: Power Up Beep – Off

Bit0	HID Emulation in Serial Interface 0: Disable 1: Enable
Address	0x000F
Register	Function
Bit7-0	Auto-sensing Sensitivity Parameter 1 0x00~0xFF: the higher the value, the lower the sensitivity
Address	0x0010
Register	Function
Bit7-0	Auto-sensing Sensitivity Parameter 2 0x00~0xFF: the higher the value, the lower the sensitivity
Address	0x0013
Register	Function
Bit7	0: Disable Identical Read Interval 1: Enable Identical Read Interval
Bit6-0	Identical Read Interval (unit = 100ms) 0x00: unlimited 0x01~0x7F: 0.1ms~12700ms
Address	0x0014
Register	Function
Bit7-0	Message Output Timeout (unit = 10ms) 0x00~0xFF: 0ms~2550ms
Address	0x0015
Register	Function
Bit7-2	Reserved
Bit1	Reserved
Bit0	HID Device Options 0: USB KBW 1: HID POS
Address	0x0016
Register	Function

Bit7-5	Reserved
Bit4	0: Disable Inverse Barcode 1: Enable Inverse Barcode
Bit3-1	Reserved
Bit0	0: Disable Enhanced Decode Capability 1: Enable Enhanced Decode Capability
Address	0x0017
Register	Function
Bit7-4	Reserved
Bit3	0: Disable QR Code Prefix (11) 1: Enable QR Code Prefix (11)
Bit2	0: Disable Code128 Prefix (11) 1: Enable Code128 Prefix (11)
Bit1	0: Enable URL Output 1: Disable URL Output
Bit0	0: Disable GS Replacement 1: Enable GS Replacement
Address	0x0018
Register	Function
Bit7-0	Set GS Replacement 0x00~0xFF: GS Replacement
Address	0x002B, 0x002A
Register	Function
Bit15	Reserved
Bit14-13	00: Parity – None 01: Parity – Odd 10: Parity – Even
Bit12-0	0x09C4: 1200bps 0x0271: 4800bps 0x0139: 9600bps 0x00D0: 14400bps 0x009C: 19200bps 0x004E: 38400bps 0x0034: 57600bps

	0x001A: 115200bps Example: 9600bps: 0x002A = 0x39, 0x002B = 0x01
Address	0x002C
Register	Function
Bit7-4	Reserved
Bit3	Reserved
Bit2-1	00: Disable All Symbologies 01: Enable All Symbologies 10/11 : Enable Default Symbologies
Bit0	Reserved
Address	0x002E
Register	Function
Bit7	0: Disable EAN-13 5-digit Supplement 1: Enable EAN-13 5-digit Supplement
Bit6	0: Disable EAN-13 2-digit Supplement 1: Enable EAN-13 2-digit Supplement
Bit5	0: Disable EAN-13 Supplement Required 1: Enable EAN-13 Supplement Required
Bit4-2	Reserved
Bit1	0: Not Send EAN-13 Check Digit 1: Send EAN-13 Check Digit
Bit0	0: Disable EAN-13 1: Enable EAN-13
Address	0x002F
Register	Function
Bit7	0: Disable EAN-8 5-digit Supplement 1: Enable EAN-8 5-digit Supplement
Bit6	0: Disable EAN-8 2-digit Supplement 1: Enable EAN-8 2-digit Supplement
Bit5	0: Disable EAN-8 Supplement Required 1: Enable EAN-8 Supplement Required
Bit4-2	Reserved
Bit1	0: Not Send EAN-8 Check Digit



	1: Send EAN-8 Check Digit
Bit0	0: Disable EAN-8 1: Enable EAN-8
Address	0x0030
Register	Function
Bit7	0: Disable UPC-A 5-digit Supplement 1: Enable UPC-A 5-digit Supplement
Bit6	0: Disable UPC-A 2-digit Supplement 1: Enable UPC-A 2-digit Supplement
Bit5	0: Disable UPC-A Supplement Required 1: Enable UPC-A Supplement Required
Bit4	0: Disable UPC-A to EAN-13 1: Enable UPC-A to EAN-13
Bit3-2	Reserved
Bit1	0: Not Send UPC-A Check Digit 1: Send UPC-A Check Digit
Bit0	0: Disable UPC-A 1: Enable UPC-A
Address	0x0031
Register	Function
Bit7	0: Disable UPC-E0 5-digit Supplement 1: Enable UPC-E0 5-digit Supplement
Bit6	0: Disable UPC-E0 2-digit Supplement 1: Enable UPC-E0 2-digit Supplement
Bit5	0: Disable UPC-E0 Supplement Required 1: Enable UPC-E0 Supplement Required
Bit4-2	Reserved
Bit1	0: Not Send UPC-E0 Check Digit 1: Send UPC-E0 Check Digit
Bit0	0: Disable UPC-E0 1: Enable UPC-E0
Address	0x0032
Register	Function
Bit7	0: Disable UPC-E1 5-digit Supplement

	1: Enable UPC-E1 5-digit Supplement
Bit6	0: Disable UPC-E1 2-digit Supplement 1: Enable UPC-E1 2-digit Supplement
Bit5	0: Disable UPC-E1 Supplement Required 1: Enable UPC-E1 Supplement Required
Bit4-2	Reserved
Bit1	0: Not Send UPC-E1 Check Digit 1: Send UPC-E1 Check Digit
Bit0	0: Disable UPC-E1 1: Enable UPC-E1
Address	0x0033
Register	Function
Bit7-1	Reserved
Bit0	0: Disable Code128 1: Enable Code128
Address	0x0034
Register	Function
Bit7-0	Code128 Min Length 0x00~0xFF: 0~255byte
Address	0x0035
Register	Function
Bit7-0	Code128 Max Length 0x00~0xFF: 0~255byte
Address	0x0036
Register	Function
Bit7	0: Not Send Code39 Stop 1: Send Code39 Stop
Bit6	0: Not Send Code39 Start 1: Send Code39 Start
Bit5	0: Disable Code32 Preamble ('A') 1: Enable Code32 Preamble ('A')
Bit4	0: Disable Code32 1: Enable Code32

Bit3	0: Disable Full ASCII Code39 1: Enable Full ASCII Code39
Bit2	0: Disable Code39 Verification 1: Enable Code39 Verification
Bit1	0: Not Send Code39 Check Digit 1: Send Code39 Check Digit
Bit0	0: Disable Code39 1: Enable Code39
Address	0x0037
Register	Function
Bit7-0	Code39 Min Length 0x00~0xFF: 0~255byte
Address	0x0038
Register	Function
Bit7-0	Code39 Max Length 0x00~0xFF: 0~255byte
Address	0x0039
Register	Function
Bit7-1	Reserved
Bit0	0: Disable Code93 1: Enable Code93
Address	0x003A
Register	Function
Bit7-0	Code93 Min Length 0x00~0xFF: 0~255byte
Address	0x003B
Register	Function
Bit7-0	Code93 Max Length 0x00~0xFF: 0~255byte
Address	0x003C
Register	Function
Bit7	0: Not Send Codabar Check Digit 1: Send Codabar Check Digit

Bit6-4	Reserved
Bit3-2	00: Codabar Verification = None 01: Codabar Verification = Mod16 10: Codabar Verification = Mod10 11: Codabar Verification = Mod10/Mod16
Bit1	0: Not Send Codabar Start/Stop 1: Send Codabar Start/Stop
Bit0	0: Disable Codabar 1: Enable Codabar
Address	0x003D
Register	Function
Bit7-0	Codabar Min Length 0x00~0xFF: 0~255byte
Address	0x003E
Register	Function
Bit7-0	Codabar Max Length 0x00~0xFF: 0~255byte
Address	0x003F
Register	Function
Bit7-6	Reserved
Bit5	0: Disable Model 1 QR Code 1: Enable Model 1 QR Code
Bit4-1	Reserved
Bit0	0: Disable QR Code 1: Enable QR Code
Address	0x0040
Register	Function
Bit7-3	Reserved
Bit2	0: Interleaved 2/5 Verification = None 1: Interleaved 2/5 Verification = Mod10
Bit1	0: Not Send Interleaved 2/5 Check Digit 1: Send Interleaved 2/5 Check Digit
Bit0	0: Disable Interleaved 2/5

	1: Enable Interleaved 2/5
Address	0x0041
Register	Function
Bit7-0	Interleaved 2/5 Min Length 0x00~0xFF: 0~255byte
Address	0x0042
Register	Function
Bit7-0	Interleaved 2/5 Max Length 0x00~0xFF: 0~255byte
Address	0x0043
Register	Function
Bit7-3	Reserved
Bit2	0: Industrial 2/5 Verification = None 1: Industrial 2/5 Verification = Mod10
Bit1	0: Not Send Industrial 2/5 Check Digit 1: Send Industrial 2/5 Check Digit
Bit0	0: Disable Industrial 2/5 1: Enable Industrial 2/5
Address	0x0044
Register	Function
Bit7-0	Industrial 2/5 Min Length 0x00~0xFF: 0~255byte
Address	0x0045
Register	Function
Bit7-0	Industrial 2/5 Max Length 0x00~0xFF: 0~255byte
Address	0x0046
Register	Function
Bit7-3	Reserved
Bit2	0: Matrix 2/5 Verification = None 1: Matrix 2/5 Verification = Mod10
Bit1	0: Not Send Matrix 2/5 Check Digit

	1: Send Matrix 2/5 Check Digit
Bit0	0: Disable Matrix 2/5 1: Enable Matrix 2/5
Address	0x0047
Register	Function
Bit7-0	Matrix 2/5 Min Length 0x00~0xFF: 0~255byte
Address	0x0048
Register	Function
Bit7-0	Matrix 2/5 Max Length 0x00~0xFF: 0~255byte
Address	0x0049
Register	Function
Bit7-3	Reserved
Bit2	0: Code11 Verification = 1bit (C or K) 1: Code11 Verification = 2bit (C+K)
Bit1	0: Not Send Code11 Check Digit 1: Send Code11 Check Digit
Bit0	0: Disable Code11 1: Enable Code11
Address	0x004A
Register	Function
Bit7-0	Code11 Min Length 0x00~0xFF: 0~255byte
Address	0x004B
Register	Function
Bit7-0	Code11 Max Length 0x00~0xFF: 0~255byte
Address	0x004C
Register	Function
Bit7-3	Reserved
Bit2	0: MSI Plessey Verification = Mod10

	1: MSI Plessey Verification = Double Mod10
Bit1	0: Not Send MSI Plessey Check Digit 1: Send MSI Plessey Digit
Bit0	0: Disable MSI Plessey 1: Enable MSI Plessey
Address	0x004D
Register	Function
Bit7-0	MSI Plessey Min Length 0x00~0xFF: 0~255byte
Address	0x004E
Register	Function
Bit7-0	MSI Plessey Max Length 0x00~0xFF: 0~255byte
Address	0x004F
Register	Function
Bit7	0: Enable GS1 Databar AI Parentheses 1: Disable GS1 Databar AI Parentheses
Bit6-1	Reserved
Bit0	0: Disable GS1 Databar 1: Enable GS1 Databar
Address	0x0050
Register	Function
Bit7	0: Enable GS1 Databar Limited AI Parentheses 1: Disable GS1 Databar Limited AI Parentheses
Bit6-1	Reserved
Bit0	0: Disable GS1 Databar Limited 1: Enable GS1 Databar Limited
Address	0x0051
Register	Function
Bit7	0: Enable GS1 Databar Expanded AI Parentheses 1: Disable GS1 Databar Expanded AI Parentheses
Bit6-1	Reserved

Bit0	0: Disable GS1 Databar Expanded 1: Enable GS1 Databar Expanded
Address	0x0052
Register	Function
Bit7-0	GS1 Databar Expanded Min Length 0x00~0xFF: 0~255byte
Address	0x0053
Register	Function
Bit7-0	GS1 Databar Expanded Max Length 0x00~0xFF: 0~255byte
Address	0x0054
Register	Function
Bit7-2	Reserved
Bit1	0: Disable Data Matrix Multicode 1: Enable Data Matrix Multicode
Bit0	0: Disable Data Matrix 1: Enable Data Matrix
Address	0x0055
Register	Function
Bit7-1	Reserved
Bit0	0: Disable PDF417 1: Enable PDF417
Address	0x0056
Register	Function
Bit7-1	Reserved
Bit0	0: Disable Han Xin 1: Enable Han Xin
Address	0x0057
Register	Function
Bit7-1	Reserved
Bit0	0: Disable MicroPDF417 1: Enable MicroPDF417



Address	0x0058
Register	Function
Bit7-1	Reserved
Bit0	0: Disable Micro QR Code 1: Enable Micro QR Code
Address	0x0059
Register	Function
Bit7-1	Reserved
Bit0	0: Disable MaxiCode 1: Enable MaxiCode
Address	0x005A
Register	Function
Bit7-1	Reserved
Bit0	0: Disable Aztec 1: Enable Aztec
Address	0x005B
Register	Function
Bit7-3	Reserved
Bit2	0: Disable Standard 2/5 Verification 1: Enable Standard 2/5 Verification
Bit1	0: Not Send Matrix 2/5 Check Digit 1: Send Standard 2/5 Check Digit
Bit0	0: Disable Standard 2/5 1: Enable Standard 2/5
Address	0x005C
Register	Function
Bit7-0	Standard 2/5 Min Length 0x00~0xFF: 0~255byte
Address	0x005D
Register	Function
Bit7-0	Standard 2/5 Max Length 0x00~0xFF: 0~255byte

Address	0x0060
Register	Function
Bit7	0: Output Data Only 1: Output Protocol
Bit6-5	00: Terminator – CR 01: Terminator – CR LF 10: Terminator – TAB 11: Terminator – None
Bit4	0: Disable RF Message 1: Enable RF Message
Bit3	0: Disable Preamble 1: Enable Preamble
Bit2	0: Disable Code ID 1: Enable Code ID
Bit1	0: Disable Postamble 1: Enable Postamble
Bit0	0: Disable Terminator 1: Enable Terminator
Address	0x0061
Register	Function
Bit7-0	00: Keyboard Layout – English (USA) 01: Keyboard Layout – Czech 02: Keyboard Layout – French 03: Keyboard Layout – German 04: Keyboard Layout – Hungarian 05: Keyboard Layout – Italian 07: Keyboard Layout – Spanish 08: Keyboard Layout – Turkish Q 09: Keyboard Layout – Turkish F 0A: Keyboard Layout – Mexican (Latin A.)
Address	0x0062
Register	Function
Bit7-4	Preamble Length 0x00~0xFF: Preamble Length
Bit3-0	Postamble Length

	0x00~0x0F: Postamble Length
Address	0x0063 – 0x0071
Register	Function
Bit7-0	Set Preamble 0x00~0xFF: Preamble data, up to 15byte
Address	0x0072 – 0x0080
Register	Function
Bit7-0	Set Postamble 0x00~0x0F: Postamble data, up to 15byte
Address	0x0081
Register	Function
Bit7-4	Reserved
Bit3-0	RF Message Length 0x00~0x0F: RF Message Length
Address	0x0082 – 0x0090
Register	Function
Bit7-0	Set RF Message 0x00~0xFF: RF Message data, up to 15byte
Address	0x0091 – 0x00AE
Register	Function
Bit7-0	Set Code ID 0x41~0x5A & 0x61~0x7A (A~Z, a~z): see Appendix A – Code ID for corresponding Code ID of each symbology
Address	0x00B0
Register	Function
Bit7-2	Reserved
Bit1-0	00: Send All 01: Send Start Only 10: Send End Only 11: Send Center Only
Address	0x00B1

Register	Function
Bit7-0	Set Start Length 0x00~0xFF: 0~255byte
Address	0x00B2
Register	Function
Bit7-0	Set End Length 0x00~0xFF: 0~255byte
Address	0x00BB
Register	Function
Bit7-3	Reserved
Bit2	0: Disable UK Plessey Verification 1: Enable UK Plessey Verification
Bit1	0: Not Send UK Plessey Check Digit 1: Send UK Plessey Check Digit
Bit0	0: Disable UK Plessey 1: Enable UK Plessey
Address	0x00BC
Register	Function
Bit7-0	UK Plessey Min Length 0x00~0xFF: 0~255byte
Address	0x00BD
Register	Function
Bit7-0	UK Plessey Max Length 0x00~0xFF: 0~255byte
Address	0x00BE
Register	Function
Bit7-3	Reserved
Bit2	0: Disable ChinaPost 2/5 Verification 1: Enable ChinaPost 2/5 Verification
Bit1	0: Not Send ChinaPost 2/5 Check Digit 1: Send ChinaPost 2/5 Check Digit
Bit0	0: Disable ChinaPost 2/5 1: Enable ChinaPost 2/5

Address	0x00BF
Register	Function
Bit7-0	ChinaPost 2/5 Min Length 0x00~0xFF: 0~255byte
Address	0x00C0
Register	Function
Bit7-0	ChinaPost 2/5 Max Length 0x00~0xFF: 0~255byte
Address	0x00C1
Register	Function
Bit7-1	Reserved
Bit0	0: Disable Code16K 1: Enable Code16K
Address	0x00C2
Register	Function
Bit7-0	Code16K Min Length 0x00~0xFF: 0~255byte
Address	0x00C3
Register	Function
Bit7-0	Code16K Max Length 0x00~0xFF: 0~255byte
Address	0x00D9 (Write only)
Register	Function
Bit7-0	0x50: Factory Default 0x55: Custom Default 0x56: Save Custom Default 0xA5: Deep Sleep Mode, can be woke up by interrupt over serial communication. (applicable for Trigger Mode only) 0x00: Write 0 to wake up MT84G. (applicable for Trigger Mode only)
Address	0x00E1 (Read only)

Register	Function
Bit7-0	Hardware Version 0x64: V1.00 0x6E: V1.10 0x78: V1.20 0x82: V1.30 0x8C: V1.40 .....
Address	0x00E2 (Read only)
Register	Function
Bit7-0	Software Version 0x64: V1.00 0x6E: V1.10 0x78: V1.20 0x82: V1.30 0x8C: V1.40 .....
Address	0x00E3 (Read only)
Register	Function
Bit7-0	Software Year (+ 2000 = Year) 0x12: 2018 0x13: 2019 .....
Address	0x00E4 (Read only)
Register	Function
Bit7-0	Software Month 0x09: Sep 0x0A: Oct 0x0B: Nov .....
Address	0x00E5 (Read only)
Register	Function
Bit7-0	Software Date 0x0A: 10 <sup>th</sup>



	0x0B: 11 <sup>th</sup> .....
--	---------------------------------

## Appendix

### Appendix A - Code ID

Symbology	Code ID
EAN-13	d
EAN-8	d
UPC-A	c
UPC-E0	c
UPC-E1	c
Code128	j
Code39	b
Code93	i
Codabar	a
Interleaved 2/5	e
Industrial 2/5	D
Matrix 2/5	v
Standard 2/5	f
ChinaPost 2/5	X
Code11	H
MSI Plessey	m
UK Plessey	n
GS1 Databar	R
GS1 Databar Limited	R
GS1 Databar Expanded	R
Code16K	X
QR Code	Q
Micro QR Code	X
Data Matrix	u
PDF417	r
MicroPDF417	R
Aztec	z
MaxiCode	x
Han Xin	h





## Version History

Rev	Date	Description	Issued
1.0	2021.10.28	Initial Release	Shaw

### **Marson Technology Co., Ltd.**

9F., 108-3, Mincyuan Rd., Sindian Dist., New Taipei City, Taiwan

TEL: 886-2-2218-1633

FAX: 886-2-2218-6638

E-mail: [info@marson.com.tw](mailto:info@marson.com.tw)

Web: [www.marson.com.tw](http://www.marson.com.tw)