RFM-003

User Manual

2005/06/01

Ver. 1.00C

Sunion Electronic Corporation

11F, 123-7, Shine De Rd., San Chung City, Taipei 241, Taiwan, R.O.C.

TEL : +886-2-8512-1456 FAX : +886-2-8512-1457

http://www.sunion.com.tw/

— Table of Contents —

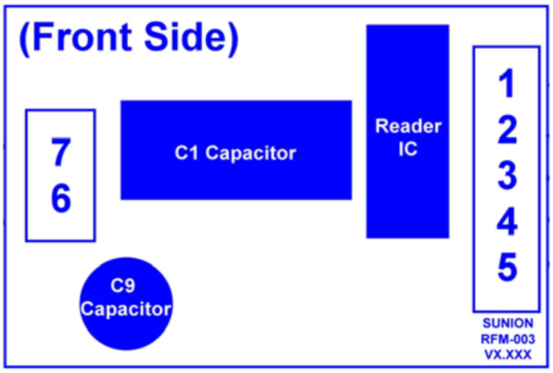
1. Introduction of RFM-003 module

1-1 Type of application available

All 125KHz with ISO standard Read Only applications are available.
(For Read/Write application or other non-ISO standard applications please feel free to contact your Sunion personal.)

1-2 RFM-003 Pin assignments

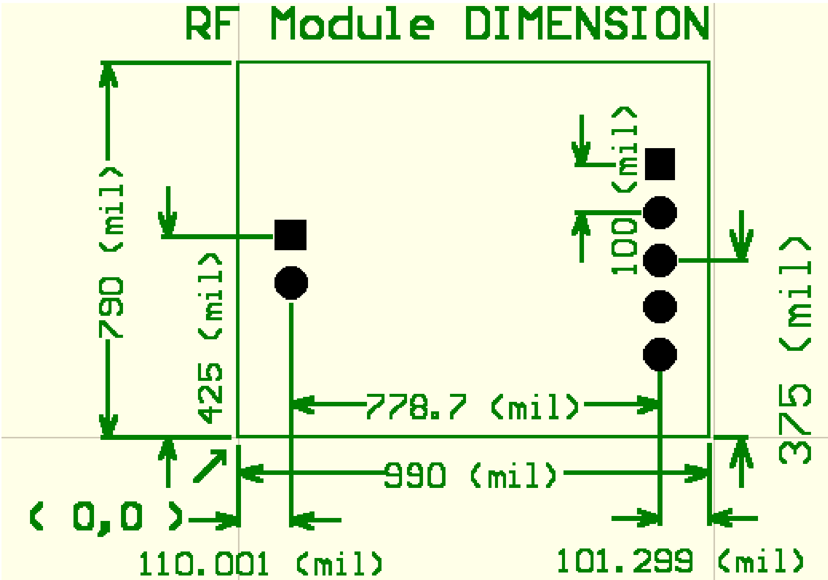
RFM-003 Pin position and assignments:



Pin No.	Name	Function
1	**SCLK**	**Serial clock input**
2	**DATA**	**Serial data In/Out**
3	**GND**	**GND**
4	**GND**	**GND**
5	**VDD**	**VDD**
6	**ANT2**	**Antenna out 2**
7	**ANT1**	**Antenna out 1**

1-3 Dimension specification

L=25mm W=20mm H=11mm



2. Electrical characteristics

2.1 Operation Specification

Temperature：Tamb = - 40℃ to + 85℃
Supply Voltage：Vdd = 4.1V to 5.5V

PARAMETERS AND CONDITIONS	SYMBOL	NOTE	MIN.	TYP.	MAX.	UNIT
Power Supply						
Supply Voltage	V_{DD}		**4.1**	**5**	**5.5**	**V**
Supply current power down mode	$I_{DDsleep}$			**1**	**5**	**uA**
Supply current excluding antenna current	I_{DDon}			**5**	**10**	**mA**
Logic Signals						
Input logic high	V_{IH}		$0.7V_{DD}$			**V**
Input logic low	V_{IL}				$0.3V_{DD}$	**V**
Output logic high	V_{OH}		$0.9V_{DD}$			**V**
Output logic low	V_{OL}				$0.1V_{DD}$	**V**
Input leakage current			**-1**		**1**	**uA**

PLL	SYMBOL	NOTE	MIN.	TYP	MAX.	UNIT
Antenna capture frequency range	F_{ANT_C}		**100**		**150**	**KHz**
Antenna locking frequency range	F_{ANT_L}		**100**	**125**	**150**	**KHz**

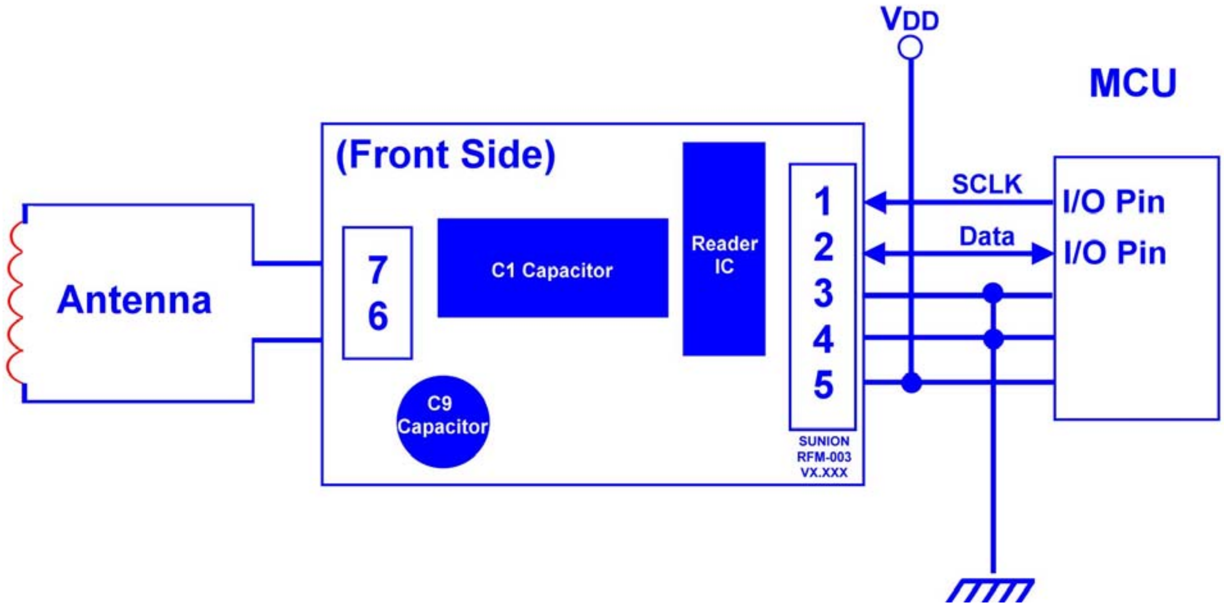
NAME	SYMBOL	NOTE	MIN.	TYP	MAX.	UNIT
Current through ANT1 and ANT2 pins. Continuous wave	I_{ANT}				**180**	mA_p
Current through ANT1 and ANT2 pins. Duty cycle 20% t_{on}<400ms	I_{ANT}				**400**	mA_p

Antenna driver	SYMBOL	NOTE	MIN.	TYP	MAX.	UNIT
Diagnostic ANT driver threshold high	V_H			$0.5V_{DD}$		**V**
Diagnostic ANT driver threshold low	V_L			$0.5V_{DD}$		**V**

2.2 Antenna Specification

Antenna inductance = 430uH ～ 460uH
Standard Antenna = 14 * 10.5 cm (Inductance = 425uH)

2.3 Example of Micro-Control Unit (MCU) connection



3. Reading Control Procedure

3-1 Reset Module Timing:

First, set the “CLK” to High, then “DTAT” to high; Wait for a while (At least 200ns) then set the “CLK” to low, and also, set the “DATA” back to low (The minimum timing for ts is 50ns).

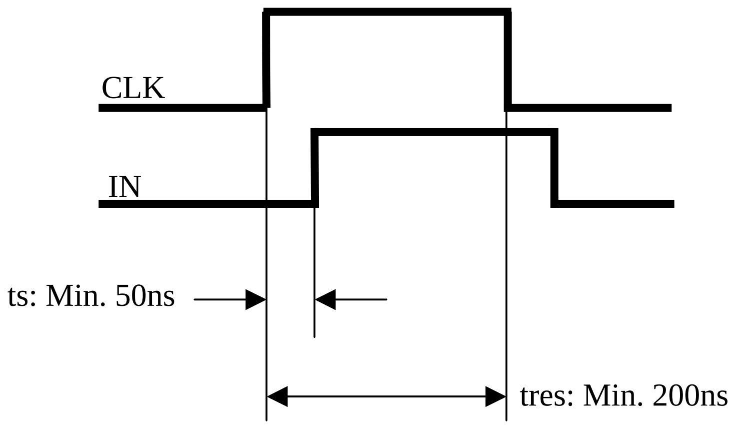


Fig 1.Module Reset

3-2 Module Timing of Entering Data:

First, set the “DATA” to high, then set “CLK” to High; Wait for a while (At least 200ns) then set the “CLK” to low, and also, set the “DATA” back to low (The minimum timing for ts is 50ns).

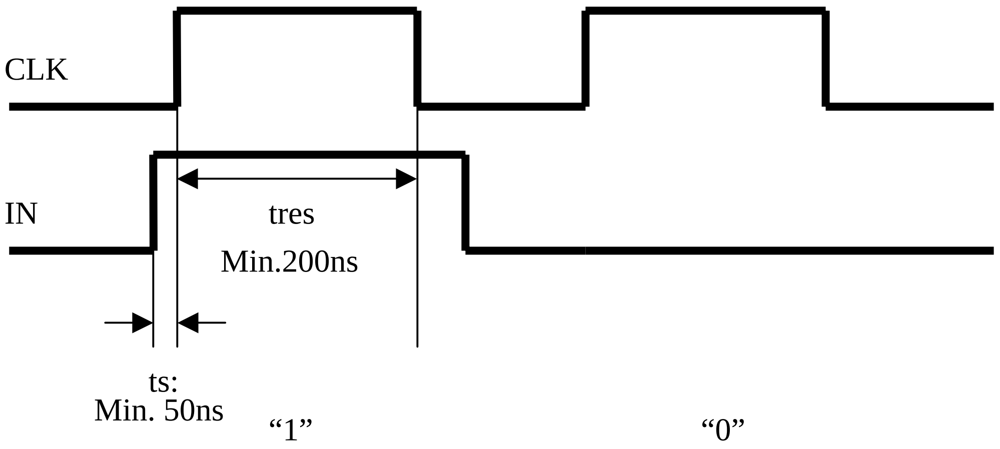
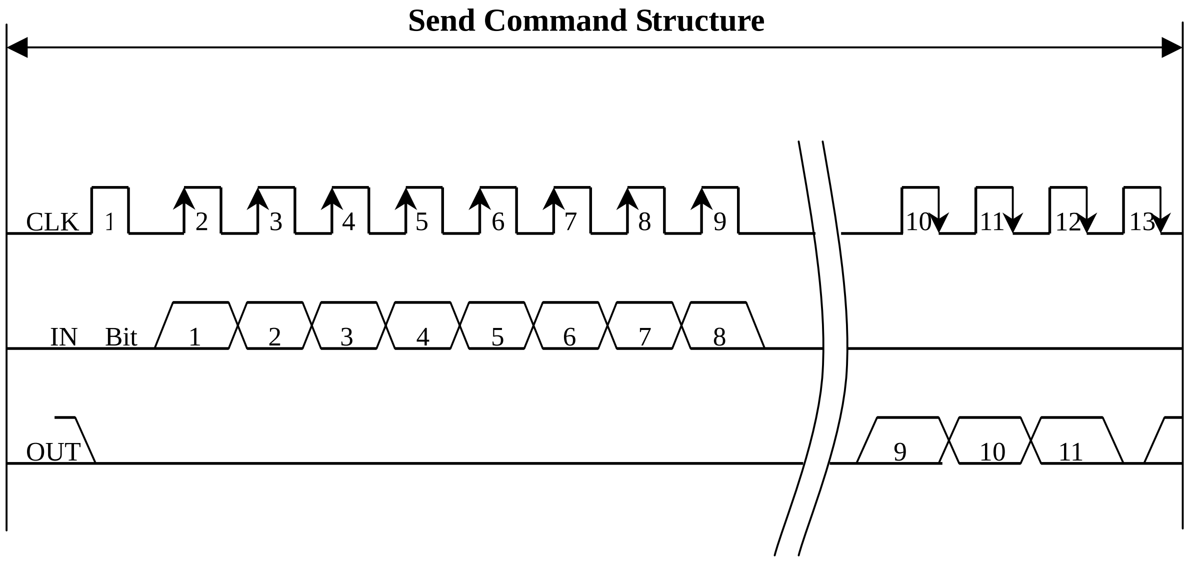
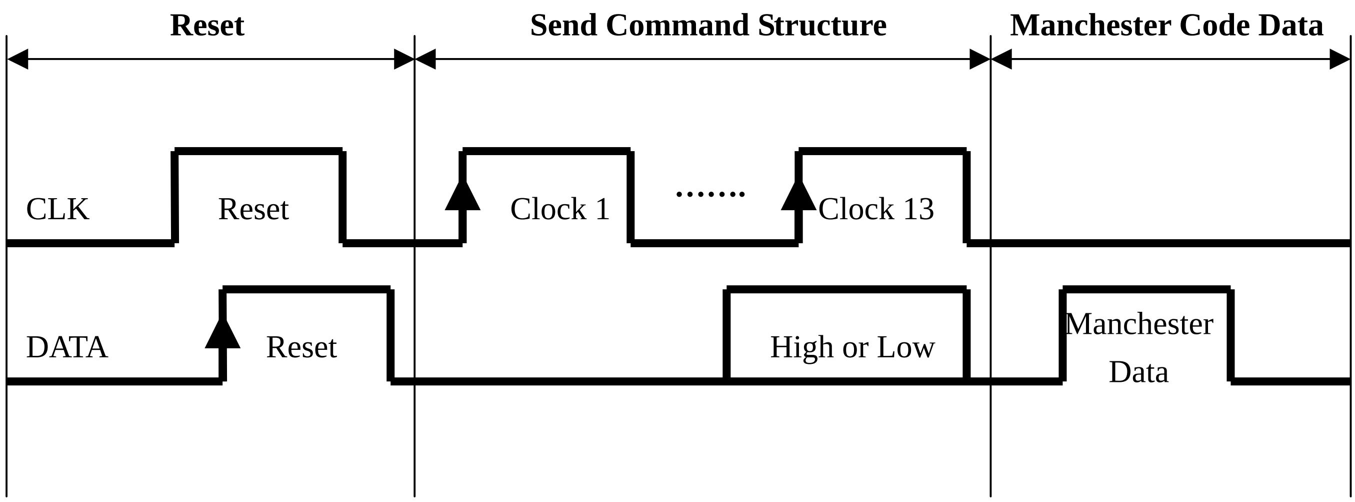


Fig. 2. Module timing of entering data

3-3 Transmit Command and receive Transponder Data:

First, send the "Reset" command then send the "Command Structure Data", IN/OUT will receive the Transponder Data.





3.Command Program; Command transmit clock as above

Activate RF to read the card:

	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
ON	0	1	0	0	1	0	0	0
OFF	0	0	0	0	1	0	0	0

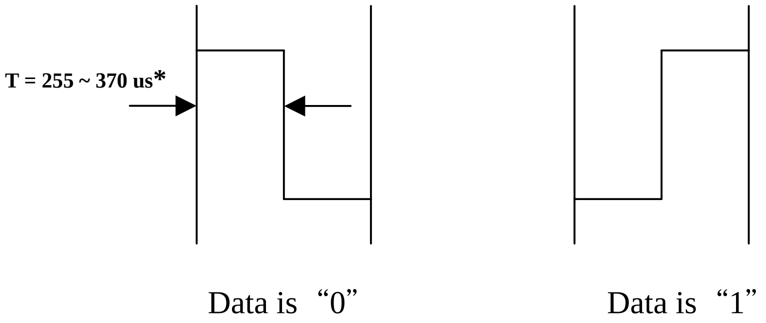
Respond from Module:

	Status	IF = “0”	IF = “1”	Correct Status
Bit9	Antenna Status	Correct Loading	Short Circuit	0
Bit10	Entering Status	Correct Signal	No-Enter-Signal	0
Bit11	PLL Status	Locked	Not Locked	0

Note: After transmit the No.9 CLK (CLK1~CLK9) and 8-bit command, send the other 4 CLK (CLK10~CLK13), wait for the module to feedback the status of that 3-bit (bit9~bit11).

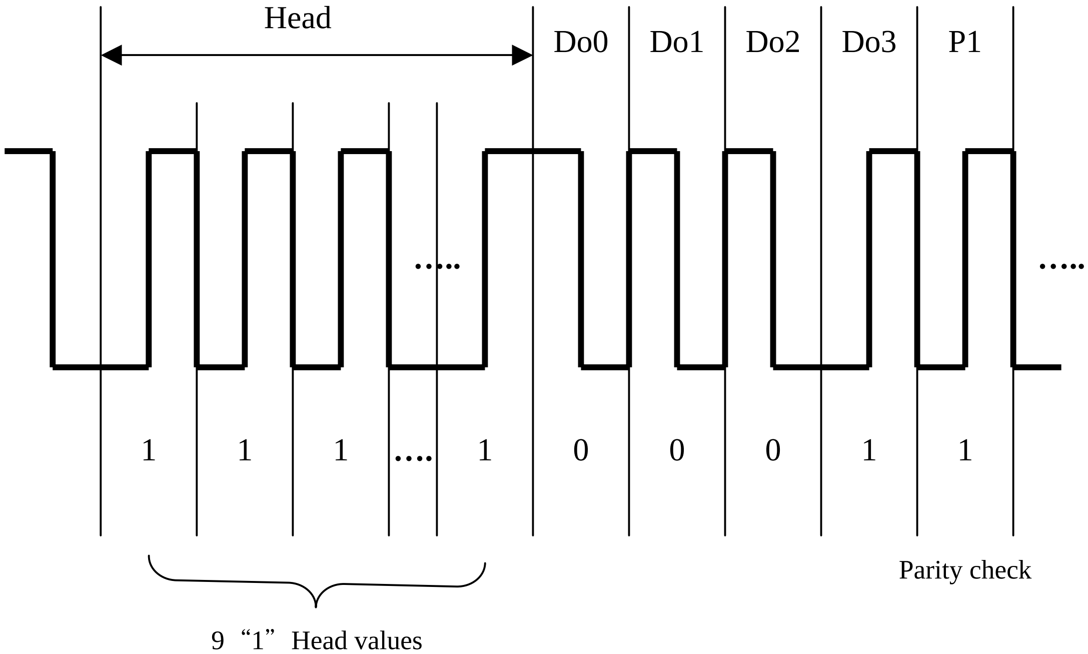
4. The data format of the Transponder

4-1 Description of Manchester Code:



- **"T" is a reference range, in practical, this figure will vary in depends on different card manufacturers or even with same manufacturer but different production lot. But in general, the "T" should fall in the range as stated above.**

4-2 Data format example of EM Manchester Code:



4-3 Data saving format inside the memory

1 1 1 1 1 1 1 1 1			
D00	D01	D02	P0
	D03		P1
D04	D05	D06	P2
	D07		P3
D08	D09	D10	P4
	D11		P5
D12	D13	D14	P6
	D15		P7
D16	D17	D18	P8
	D19		P9
D20	D21	D22	
	D23		
D24	D25	D26	
	D27		
D28	D29	D30	
	D31		
D32	D33	D34	
	D35		
D36	D37	D38	
	D39		
PC0	PC1	PC2	0
	PC3		

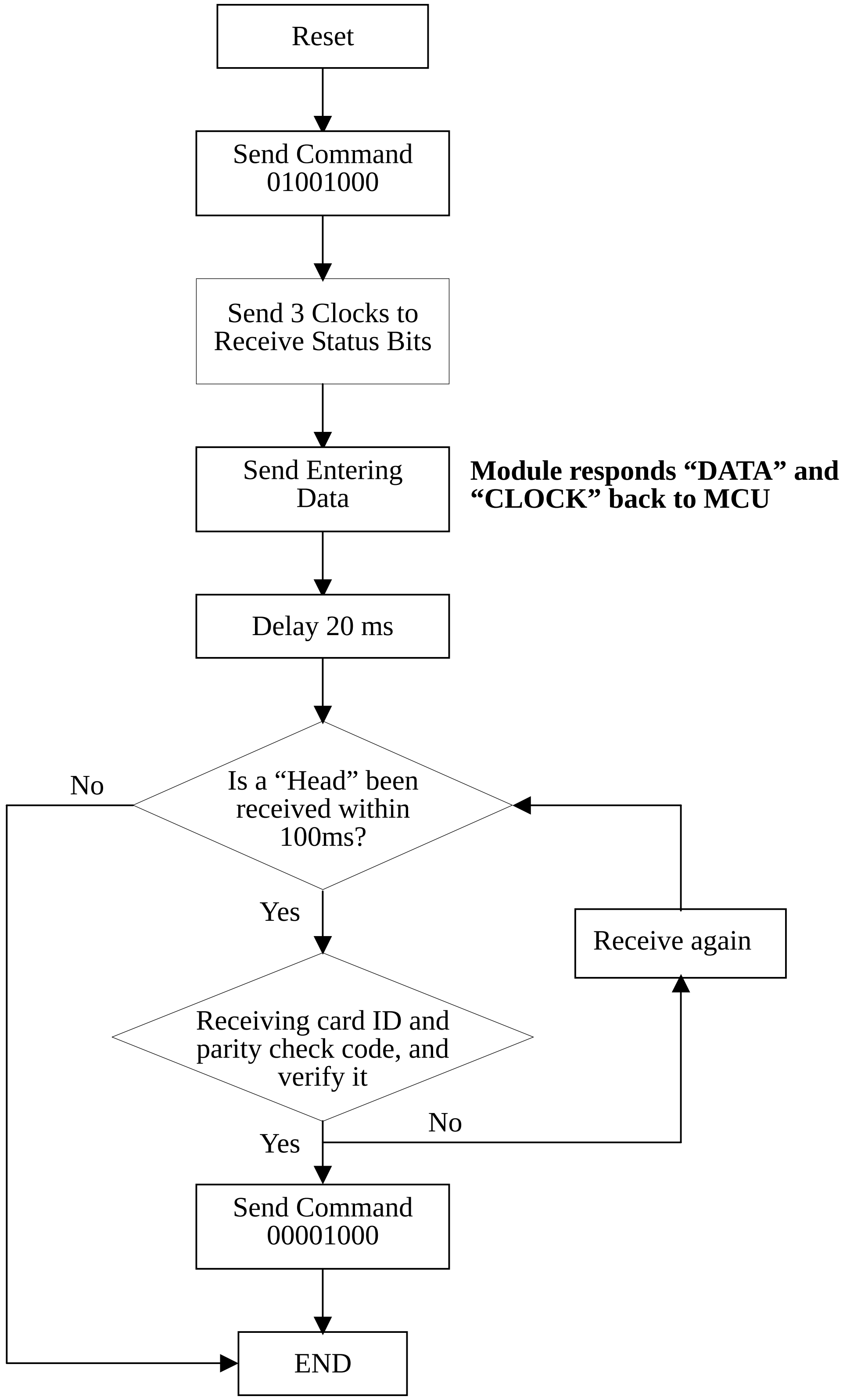
9 Start bit

Bits column parity check.

Column parity check

Total 64 bites, 9 bites for start value,
10 bits for row parity check, 40 bits data,
4bites column parity check.

5. Reading procedure flow chart



6. Software Example

```
;==========================================================================
; RFM003.ASM; Software example for MCU 8051. Frequency=11.0592Mhz
; T0CountL =  is a "Down Counter", one count down is about 0.4ms per count, then next is generated by
              "T0 Interrupt".
;==========================================================================
Data_Pin        Bit     P3.3
Clock_Pin       Bit     P3.2
;==========================================================================
RFIDInit:
                clr     Data_Pin
                setb    Clock_Pin
                ret
;==========================================================================
RFIDRead:
                push    DR2
                mov     A,#01001000B
                lcall   RFID_Command            ; Start transmit
                setb    Data_Pin
                mov     A,T0CountL
                add     A,#205                  ; (255-205) x 0.4 ms = 20 ms.
                cjne    A,T0CountL,$            ; delay 20 ms.
                mov     R2,T0CountL
                inc     R2                      ; (255-1) x 0.4 ms = 101 ms.
rfid_read1:
                mov     A,R2
                cjne    A,T0CountL,rfid_read2
                clr     C
                sjmp    rfid_read6
rfid_read2:
                lcall   RFID_RHead
                jnb     F0,rfid_read1
                mov     R4,#8                   ; read 8 bits data = 1.
rfid_read3:     call    RFID_RBit
                jnb     F0,rfid_read1
                jnc     rfid_read1              ; jmp if bit = 0.
```

```
            djnz    R4,rfid_read3

            mov     R4,#5                   ; read 5 bytes data.
            mov     R1,#RFIDBuf
rfid_read4: lcall   RFID_RByte
            jnb     F0,rfid_read1
            inc     R1
            djnz    R4,rfid_read4

            mov     R4,#5                   ; if card ID= FFFFFFFFFF then ignore.
            mov     R1,#RFIDBuf
rfid_read5: cjne    @R1,#0ffH,rfid_read6
            inc     R1
            djnz    R4,rfid_read5
            sjmp    rfid_read1
            setb    C
rfid_read6:
            mov     F0,C
            clr     Data_Pin
            setb    Clock_Pin
            mov     A,#00001000B            ; Close transmit
            lcall   RFID_Command
            clr     Data_Pin
            setb    Clock_Pin
            pop     DR2
            mov     C,F0                    ; Success, return C = 1.
            Ret
;======================================================================
; serial interface command.
;======================================================================
RFID_Command:
            setb    Clock_Pin               ; Reset.
            setb    Data_Pin
            clr     Clock_Pin
            clr     Data_Pin

            setb    Clock_Pin               ; send clock 1, stop send data..
            nop
            clr     Clock_Pin
```

```
                mov     R7,#8                   ; send clock 2 ~ clock 9.
rfid_comd1:
                rlc     A
                mov     Data_Pin,C
                setb    Clock_Pin
                nop
                clr     Clock_Pin
                djnz    R7,rfid_comd1

                mov     R7,#4                   ; read clock 10 ~ clock 13.
                setb    Data_Pin
rfid_comd2:     setb    Clock_Pin
                nop
                clr     Clock_Pin
                djnz    R7,rfid_comd2
                ret
;================================================================================
RFID_RHead:
                lcall   Check_DataLow
                jnb     F0,RFID_Error
                lcall   Check_DataHigh
                jnc     RFID_Error              ; jmp High Too short or Time Out.
                ret
;--------------------------------------------------------------------------------
RFID_RByte:
                mov     B,#2
rfid_rbyte1:    mov     R5,#4
                mov     R6,#0                   ; R6 = parity.
rfid_rbyte2:    lcall   RFID_RBit
                jnb     F0,RFID_Error
                mov     A,@R1
                rlc     A
                mov     @R1,A
                djnz    R5,rfid_rbyte2
                lcall   RFID_RBit
                jnb     F0,RFID_Error
                mov     A,R6
                jb      ACC.0,RFID_Error
                djnz    B,rfid_rbyte1
```

```
                sjmp    RFID_Succc
;===============================================================================
RFID_RBit:
                jnb     Data_Pin,rfid_rbit1
                lcall   Check_DataHigh
                jnb     F0,RFID_Error
                lcall   Check_DataLow
                jnc     RFID_Error
                clr     C                               ; set data = 0.
                sjmp    RFID_Succ
rfid_rbit1:
                lcall   Check_DataLow
                jnb     F0,RFID_Error
                lcall   Check_DataHigh
                jnc     RFID_Error
                inc     R6                              ; set data = 1, parity++.
RFID_Succ:      setb    F0
                ret
RFID_Error:     clr     F0
                ret
;===============================================================================
; 130us < Data High or Data Low < 370 us
;===============================================================================
Check_DataHigh:
                mov     R7,#57                          ; check high 370 us.
                mov     A,R2
check_high1:    cjne    A,T0CountL,check_high2
                clr     C                               ; Time Out, return C = 0. F0 = 0.
                clr     F0
                ret
check_high2:
                jnb     Data_Pin,check_high3
                djnz    R7,check_high1
                setb    C                               ; if > 370 us, return C = 1, F0 = 0.
                clr     F0
                ret
check_high3:
                cjne    R7,#35,check_high4              ; if < 130us, return C = 0, F0 = 1.
check_high4:    setb    F0                              ; Success, return C = 1, F0 = 1.
```

```
                ret
;--------------------------------------------------------------------------------
Check_DataLow:
                mov     R7,#57                  ; Check Low 370 us.
                mov     A,R2
check_low1:     cjne    A,T0CountL,check_low2
                clr     C                       ; Time Out, return C = 0. F0 = 0.
                clr     F0
                ret
check_low2:
                jb      Data_Pin,check_low3
                djnz    R7,check_low1
                setb    C                       ; if > 370 us, return C = 1, F0 = 0.
                clr     F0
                ret
check_low3:
                cjne    R7,#35,check_low4       ; if < 130us, return C = 0, F0 = 1.
check_low4:     setb    F0                      ; Success, return C = 1, F0 = 1.
                ret
```

1 125 KHz Antenna Manufacturing instructions

1. Measure or examine the size and shape of the mechanism you want to put the antenna to decide of how big and what will be the shape of your antenna*.
2. Use the size and shape you measured for antenna to build up the tooling. You can use any kind of materials for that antenna tooling except metal; the only thing you have to worry about is would it be able to sustain the force when you wind the wire around? Also, after finished, the antenna should be able to pull easily out of that tooling.
3. Choose enamel-insulated wire with appropriate diameter, general speaking a wire with 0.5mm diameter should be ok; Also, to optimize the inductance and Q value, you should use thicker wire for bigger antenna (it also will decrease the amount of circles); the thinner wire for smaller antenna (the amount of circles, in the other hand, will increased); We suggest you make more circles on first trying, it would be very helpful when performing adjustment hereafter.
4. Winding the wires on that tooling circle by circle then use inductance meter to measure the inductance value, the right value is around 425 mH**. (Reduce circles if the value greater then 425 mH)
5. Use tape or other suitable things to fasten the antenna you have just made to prevent any possible distortion when adjusting or pulling out of the tooling.
6. Connect antenna to reader for testing; First, connect an oscilloscope's probe to a circuit with bigger inductance (like the relay circuit) then approaching it to antenna (fig.1), now, you should be able to see a wave pattern shown on your oscilloscope like fig.2.
7. Hold that position of both probe and circuit on it then try to slowly decrease or increase the amount of circles and also keep an eye on the scope, stop when reach the maximum amplitude. ***
8. Test the reader's reading range with transponder (tag) then repeat step 7th for fine tuning until you get the maximum reader range.
9. Pull the antenna out from tooling, there we are~~!

***. According to our experience, the antenna's shape will have a great influence in reading; generally, with the same inductance the square type will have better range in compare with circle type and full square type is better then rectangle.**

****. We suggests using the method of circle by circle to wind the wire instead of crisscross, because the effect is almost the same but more easily to adjust with circle by circle.**

*****. Sometimes the maximum amplitude doesn't mean the best range, it is because even we have the best transmitting, but however, the sensitive of receiving is decreased (we are using same antenna for both transmit and receive, that why!), therefore, we need to repeat step 8th to make sure we are in the best configuration.**

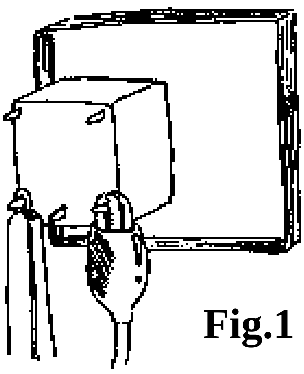


Fig.1

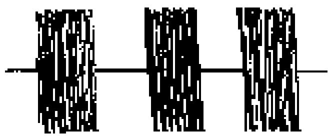


Fig.2