MC10P11B User Manual V1.1

8 Bit MCU designed by SinoMCU

2013/4/28

上海晟矽微电子股份有限公司

Shanghai SinoMCU Microelectronics Co., Ltd.

Contents

1 Introduction 3
1.1 Product Features 3
1.2 Block Diagram 3
1.3 Pin Assignment 4
1.4 Pin Description 4
2 CPU 4
2.1 Instruction Set 4
2.2 Address Space 5
2.3 Program Memory – ROM 5
2.4 User Data Memory – RAM 5
2.5 Control Registers 5
3 System Clock 5
4 T-shape Keyboard Scanning Circuit.... 5
4.1 Principle of Scanning 6
4.2 Explanation about Combination of Pressed Keys.... 7
4.3 Relative Registers 11
5 GPIO 12
6 IROUT Port 12
7 Interrupt 13
7.1 General Description 13
7.2 Keyboard Interrupt 13
8 System Operation Modes 14
8.1 STOP Mode 14
8.2 WAIT Mode 14
9 Electrical Specification 15
9.1 Absolute Rating 15
9.2 DC Electrical Characteristics.... 15
9.3 AC Electrical Characteristics.... 15
10 Typical Application Schematics.... 16
11 Dimension of Package 18
12 Appendix 19
12.1 Internal RC Frequency 19
12.2 Idle Current 20
13 Revision History 21

MC10P11B User Manual V1.1

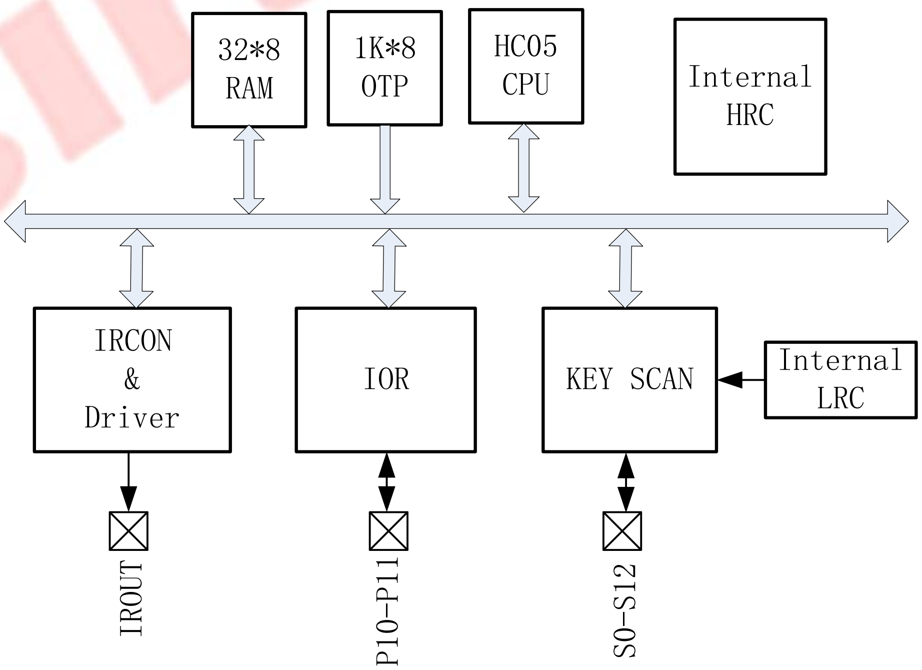
1 Introduction

MC10P11B is a high-performance, 8-bit Microcontroller. It has internal high-accuracy RC oscillator circuit, T-shape keyboard scanning circuit and infrared emission diode driving circuit. It provides perfect solution for the remote control of TV, DVD, STB etc.

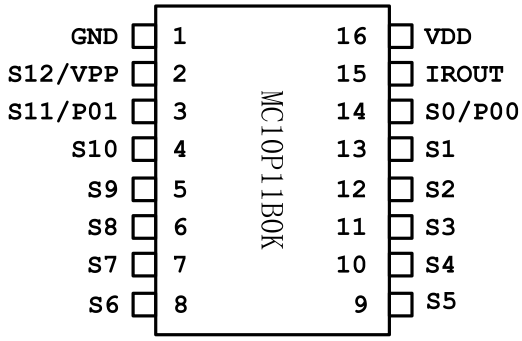
1.1 Product Features

- 8-bit CPU with CISC structure
- 1K*8 bits OTP ROM
- 32 bytes RAM (including stack)
- Internal high-accuracy 4MHz RC oscillator (Frequency deviation less than 1%; CONDITION: 3.0V, 25 ℃)
- T-shape keyboard scanning circuit, which can support 91 keys
- Infrared remote control code output IROUT (open-drain structure), which can drive infrared emission diode directly
- 2 GPIOs
- Low power dissipation (idle current less than 3uA@3V)
- Data stored in RAM can be maintained (CONDITION: supply voltage is higher than 1.1V)
- Serial programming interface circuit
- Protecting program memory data
- Operating voltage range: 1.8～5.5V
- Package type: SOP16, SOP8

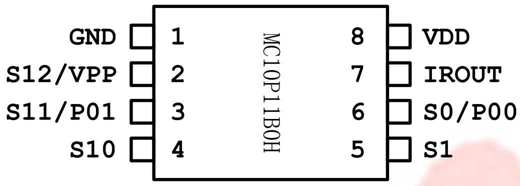
1.2 Block Diagram



1.3 Pin Assignment



SOP16



SOP8

1.4 Pin Description

Name	Direction	Function Description
GND	P	Ground
S12	I/O	Key scanning port
VPP		Programming high voltage input
S11	I/O	Key scanning port
P01		GPIO
S9~S1	I/O	Key scanning port
S0	I/O	Key scanning port
P00		GPIO
IROUT	O	IR output
VDD	P	Source

2 CPU

2.1 Instruction Set

MC10P11B uses HC05 compatible instruction set. For detail information about instruction set, please refer to the datasheet “HC05 Instruction Set” provided by SinoMCU.

Note: Instruct "MUL" is not available.

2.2 Address Space

$0000-$0002：Control Register
$0003-$00DF：Reserved
$00E0-$00FF：RAM (including Stack)
$0100-$1BFF：Reserved
$1C00-$1FFF：OTP ROM

2.3 Program Memory - ROM

Program memory of MC10P11B, which is used to store instructions, is an OTP ROM with size of 1K bytes. The highest address ($1FF0～$1FFF) area of program memory is reset/interrupt vector area.

2.4 User Data Memory - RAM

User data memory of MC10P11B has 32 bytes, which are shared with stack. For more information about stack, please refer to the datasheet "HC05 Instruction Set".

2.5 Control Registers

All the registers of MC10P11B are listed below. Detail functions of these registers are described in the following contents.

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
$00	KEY	R/W	K7	K6	K5	K4	K3	K2	K1	K0	uuuu uuuu
$01	MCR	R/W	KBIE	KBIF	OUTC	K12	K11	K10	K9	K8	001u uuuu
$02	IOR	R/W	P01M	P01U	P01D	P01	P00M	P00U	P00D	P00	0000 0000

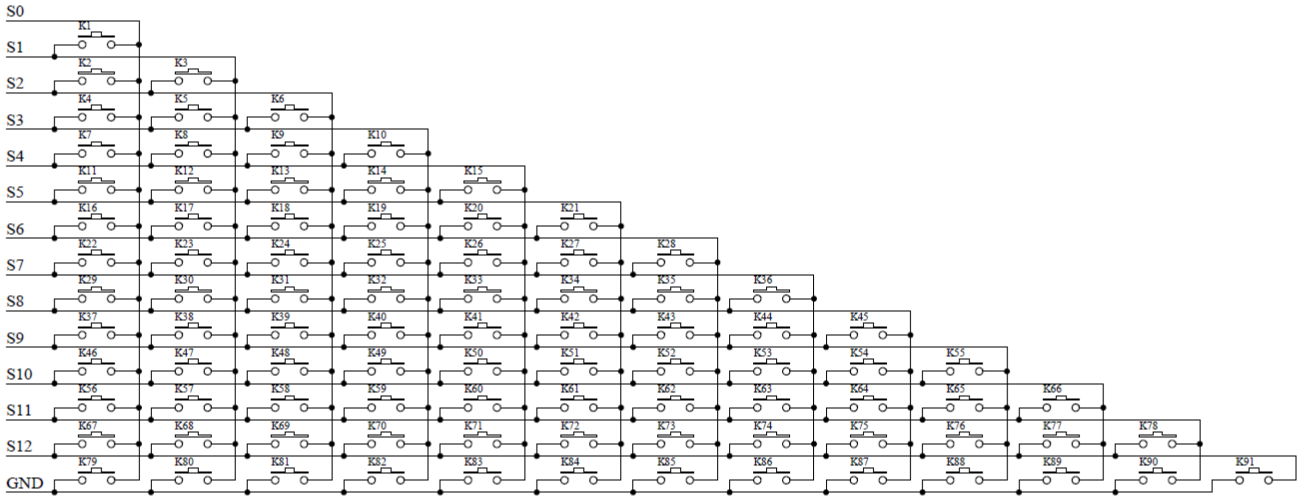
Note: "u" means the initial value is indefinite.

3 System Clock

Internal high-accuracy RC oscillator generates signal Fosc.
The primary system clock (Fsys) is 1/2 frequency division of the signal Fosc.
The frequency of MC10P11B's RC oscillator is 4MHz.

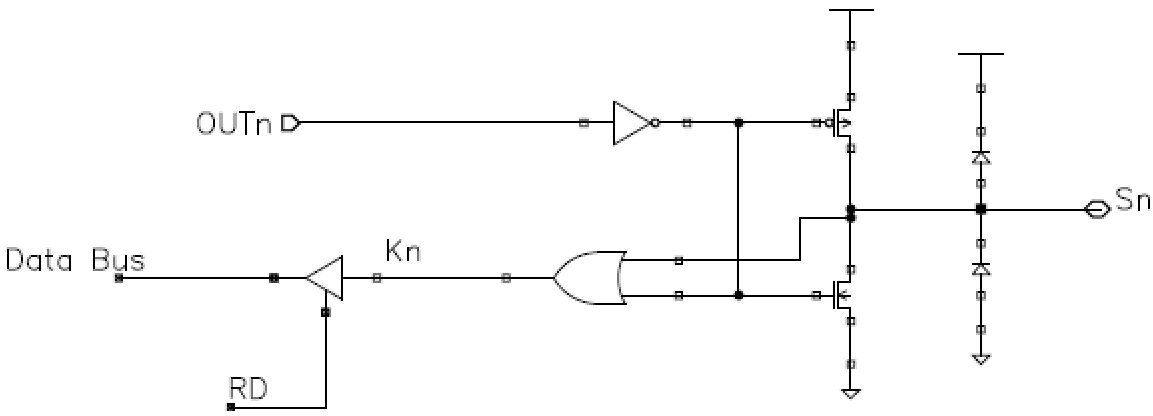
4 T-shape Keyboard Scanning Circuit

Comparing with regular mode, T-shape keyboard mode can scan more keys using the same number of pins. The pin S0～S12 and GND of MC10P11B can scan the maximum of 91 keys. The schematic is shown bellow.



4.1 Principle of Scanning

The structure of Keyboard scanning ports is shown below.



Before scanning, OUT0～OUT12 of ports S0～S12 are all maintained high level, so all the ports are pulled up.

Writing the register KEY will let OUT0 low, and make S0 enter scanning status. Writing KEY again will let OUT0 high, and make S0 exit from scanning status. Continuously, writing KEY will make S1 enter scanning status, and writing KEY again will make S1 exit from scanning status. So, repeatly writing KEY can realize scanning S0 to S12 sequentially. To scan all ports S0～S12, it is necessary to write KEY 26 times. In an entire sequence of scanning, writing KBIF in MCR will terminate and reset the process, which means a new scanning sequence will restart from S0 if writing KEY again.

In the process of scanning, you can determine which key is pressed through reading the value of K[12:0], which is stored in the register KEY and MCR.

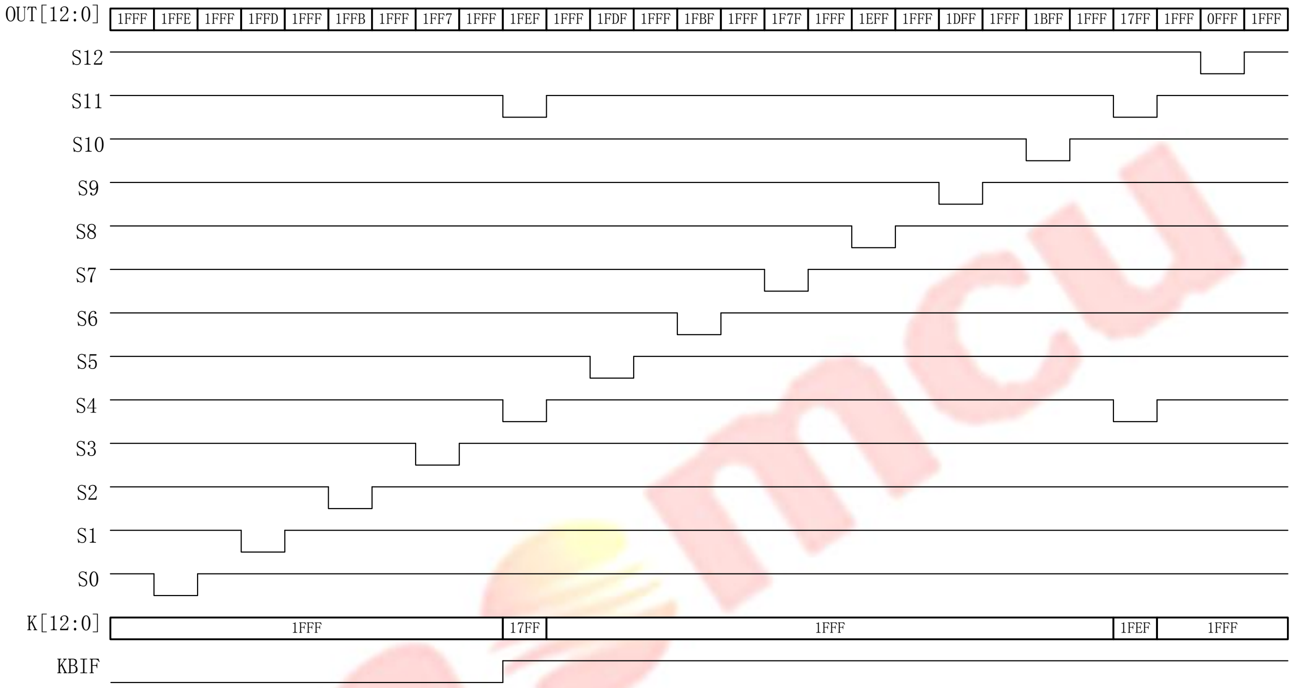
When the system found any key is pressed, the flag KBIF will be set, and the keyboard interrupt (KBI) will be requested. Refer to §7.2.

In addition, while MC10P11B working in STOP mode, the system will scanning keyboard automatically. If any key is pressed, KBI is triggered, and the system will be awaked from STOP mode. Refer to §8.1.

4.2 Explanation about Combination of Pressed Keys

This section describes the scanning process with waveform diagrams. It is a guide to determine which key is pressed.

(1) Single Key (not include GND) is pressed, e.g. K60 (S4-S11) is pressed

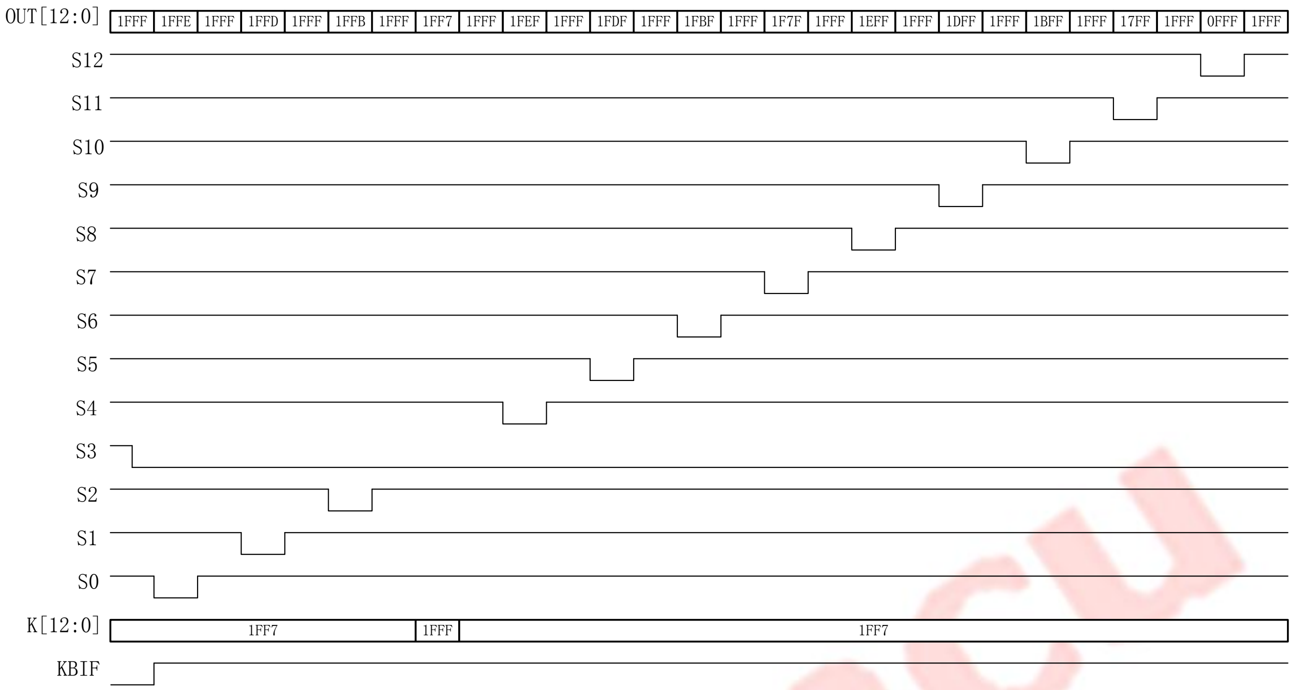


Note: OUT[12:0] are status of the control terminals of S0~S12 (refer to the diagram in§4.1); S0~S12 are the waveforms of port S0~S12; K[12:0] are the values of K12~K0; KBIF is the state of the flag KBIF.

If K60 (S4-S11) is pressed:

- ✧ While scanning S4, K[11]=0, i.e. K[12:0]=$17FF
- ✧ While scanning S11, K[4]=0, i.e. K[12:0]=$1FEF
- ✧ While scanning other ports, K[12:0]=$1FFF

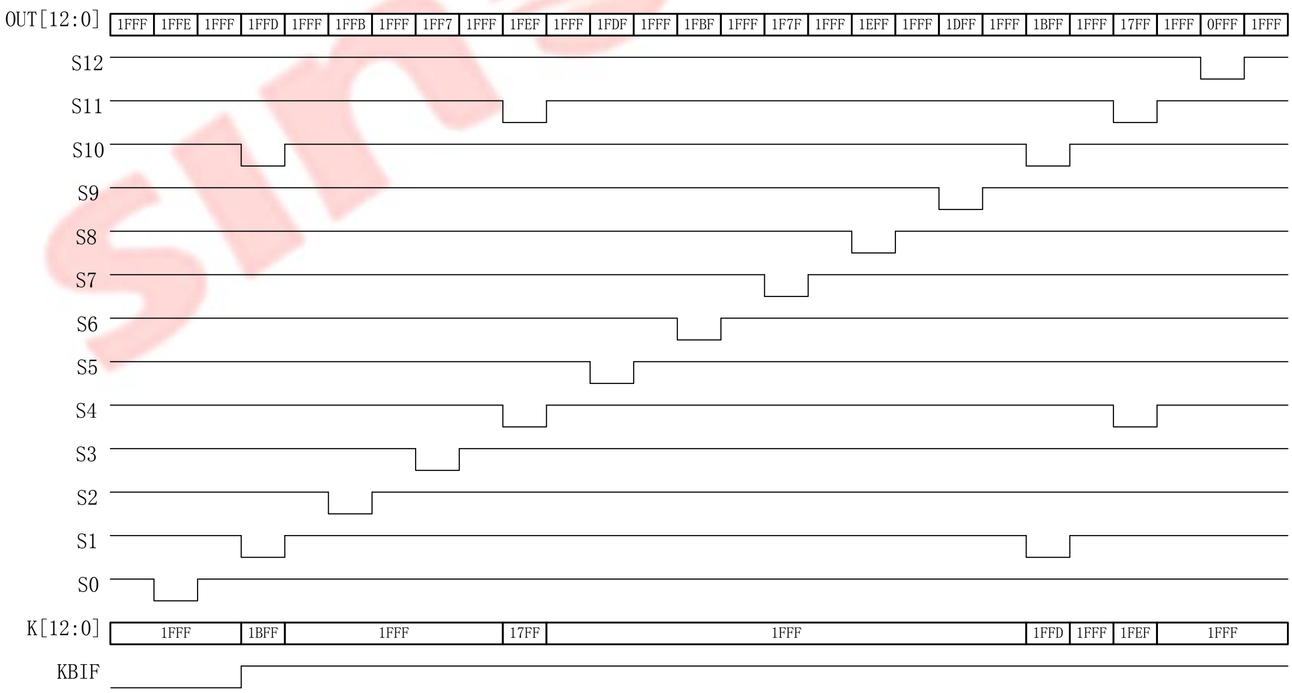
(2) Single key (one of ports if GND) is pressed, e.g. K82 (S3-GND) is pressed



If K82 (S3-GND) is pressed:

- ✧ While scanning S3, K[12:0]=$1FFF
- ✧ While scanning other ports, K[3]=0, i.e. K[12:0]=$1FF7

(3) Double keys (no reused port, and not include GND) are pressed, e.g. K60 (S4-S11) and K47 (S1-S10) are both pressed

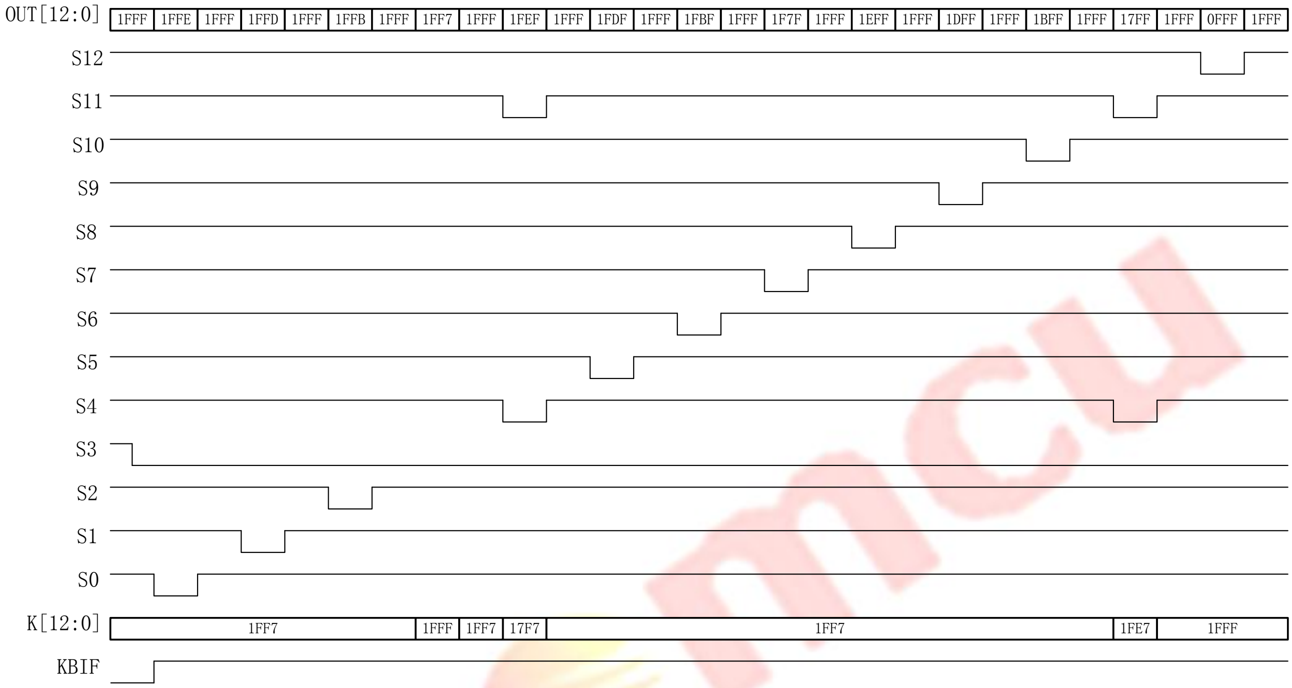


If K60 (S4-S11) and K47 (S1-S10) are both pressed:

- ✧ While scanning S1, K[10]=0, i.e. K[12:0]=$1BFF
- ✧ While scanning S4, K[11]=0, i.e. K[12:0]=$17FF

- ✧ While scanning S10, K[1]=0 i.e. K[12:0]=$1FFD
- ✧ While scanning S11, K[4]=0, i.e. K[12:0]=$1FEF
- ✧ While scanning other ports, K[12:0]=$1FFF

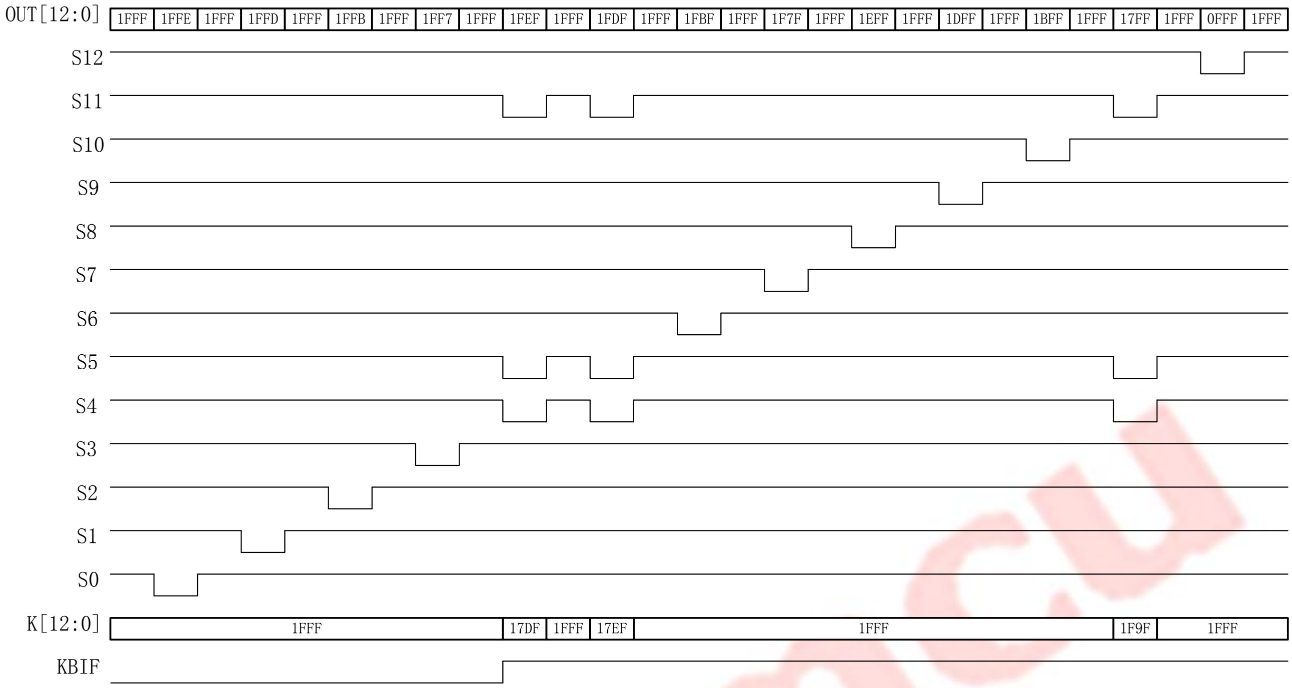
(4) Double keys (no reused port, one of ports is GND) are pressed, e.g. K60 (S4-S11) and K82 (S3-GND) are both pressed



If K60 (S4-S11) 和 K47 (S3-GND) are both pressed:

- ✧ While scanning S3, K[12:0]=$1FFF
- ✧ While scanning S4, K[11]=0 and K[3]=0, i.e. K[12:0]=$17F7
- ✧ While scanning S10, K[1]=0 and K[3]=0, i.e. K[12:0]=$1FE7
- ✧ While scanning other ports, K[3]=0, i.e. K[12:0]=$1FF7

（5）Double keys (one port is reused, not include GND) are pressed, e.g. K60 (S4-S11) and S61 (S5-S11) are both pressed

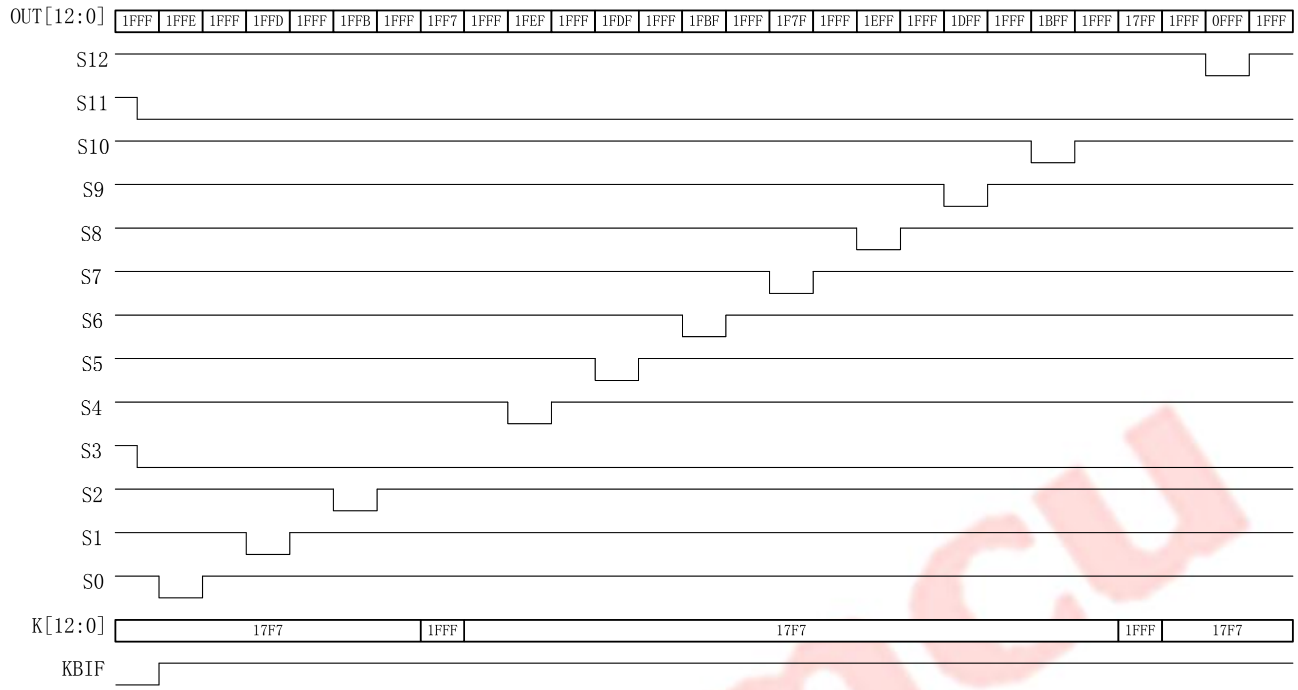


If K60 (S4-S11) 和 K61 (S5-S11) are both pressed:

- ✧ While scanning S4, K[5]=0 and K[11]=0, i.e. K[12:0]=$17DF
- ✧ While scanning S5, K[4]=0 and K[11]=0, i.e. K[12:0]=$17EF
- ✧ While scanning S11, K[4]=0 and K[5]=0, i.e. K[12:0]=$1F9F
- ✧ While scanning other ports, K[12:0]=$1FFF

Note: In fact, pressing K60 and K61 simultaneity causes the ports S4, S5 and S11 connecting together. The same effect will happen when K15 and K60 (or K15 and K60) are both pressed. That is to say it is impossible to distinguish these three situations. So, pressing double keys with a reused port is invalid when using T-shape keyboard method.

(6) Double keys (one port is reused, one of the other two is GND) are pressed, e.g. K82 (S3-GND) and K59 (S3-S11) are both pressed



If K82 (S3-GND) 和 K59 (S3-S11) are both pressed

- ✧ While scanning S3, K[12:0]=$1FFF
- ✧ While scanning S11, K[12:0]=$1FFF
- ✧ While scanning the other ports, K[3]=0 and K[11]=0, i.e. K[12:0]=$17F7

Note: In fact, pressing K82 and K59 simultaneity causes the ports S3, S11 and GND connecting together. The same effect will happen when K82 and K90 (or K59 and K90) are both pressed. That is to say it is impossible to distinguish these three situations. So, pressing double keys with a reused port is invalid when using T-shape keyboard method.

4.3 Relative Registers

This section describes the registers relative with keyboard scanning.

$00	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KEY	K7	K6	K5	K4	K3	K2	K1	K0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	u	u	u	u	u	u	u	u

BIT[7:0] **K[7:0]** – Keyboard scanning values of S7～S0.

Reading KEY can get the value of K[7:0]. Writing KEY can scan S0 to S12 sequentially. Refer to §4.1错误!未找到引用源。.

$01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	KBIE	KBIF	OUTC	K12	K11	K10	K9	K8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	1	u	u	u	u	u

BIT[4:0] **K[12:8]** –Keyboard scanning values of S12～S8.

5 GPIO

M10P11 has two general purpose input/output (GPIO) ports, whose names are P01 and P00. When the GPIO function is enabled, the corresponding key scanning function (S11 and S0) is disabled automatically. Each GPIOs is controlled by the corresponding Data Register bit (P01 and P00) and Direction Register bit (P01D and P00D)

R/W	P0xD	Function
W	0	The port is in input mode. Data is written into the output data latch.
W	1	Data is written into the output latch and output to the port.
R	0	The state of port is read.
R	1	The port is in output mode. The output data latch is read.

Here is the relative register of GPIO.

$02	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOR	P01M	P01U	P01D	P01	P00M	P00U	P00D	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

BIT[7] **P01M** – S11/P01 mode selection
0: Configured to GPIO function (P01)
1: Configured to key scanning function (S1)

BIT[6] **P01U** –P01 pull-up selection
0: No pull-up resistor is connected to P01
1: The internal pull-up resistor is connected to P01 (necessary condition: P01 is configured to GPIO function and in input mode)

BIT[5] **P01D** –P01 direction selection
0: Configured to input
1: Configured to output

BIT[4] **P01** –P01 data bit

BIT[3] **P00M** – S0/P00 mode selection
0: Configured to GPIO function (P00)
1: Configured to key scanning function (S0)

BIT[2] **P00U** –P00 pull-up selection
0: No pull-up resistor is connected to P00
1: The internal pull-up resistor is connected to P00 (necessary condition: P00 is configured to GPIO function and in input mode)

BIT[1] **P00D** –P00 direction selection
0: Configured to input
1: Configured to output

BIT[0] **P00** –P00 data bit

6 IROUT Port

IROUT has ability to sink large current. It can drive infrared emission diode directly. The state of IROUT port is controlled by OUTC bit in MCR register. The initial state is high-impedance.

$01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	KBIE	KBIF	OUTC	K12	K11	K10	K9	K8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	1	u	u	u	u	u

BIT[5] **OUTC** – IROUT state control bit
0：IROUT outputs low level
1：IROUT outputs high-impedance

Note: IROUT port is open-drain structure.

7 Interrupt

7.1 General Description

The interrupts of MC10P11B are keyboard interrupt (KBI) and software interrupt (SWI). KBI can be masked by I bit, which is in CPU status control register CCR, but SWI cannot be masked. Furthermore, SWI is also an instruct. For details about SWI, please refer to the datasheet “HC05 Instruction Set”.

The process of interrupt response is:

- ✧ While interrupt request occurring, CPU pushes all the relative registers (5 bytes altogether) to the system stack, set I bit to 1, and mask all the other interrupts. Differently from system reset, hardware interrupt does not terminate current instruction execution, but suspends itself until current instruction finished.
- ✧ While responding interrupt, firstly, CPU fetches the entrance address of the interrupt service subroutine from the corresponding interrupt vector, then jumps to the subroutine and executes.
- ✧ Each interrupts service subroutine needs an RTI instruct. When executing RTI, CPU pops all status registers from the system stack, and executes the instruct exactly after the interrupt happened.

The interrupt vectors are shown bellow. The priority is decreased from bottom to top in the list.

Vector Address	Interrupt
$1FF0:$1FF1	Reserved
$1FF2:$1FF3	Reserved
$1FF4:$1FF5	KBI
$1FF6:$1FF7	Reserved
$1FF8:$1FF9	Reserved
$1FFA:$1FFB	Reserved
$1FFC:$1FFD	SWI
$1FFE:$1FFF	RESET

7.2 Keyboard Interrupt

If any key is pressed, the flag KBIF will be set to 1 in the process of key scanning. If KBIE is 1 meanwhile, KBI interrupts request will occur. If KBIE is 0, KBI will not occur.

Here are the relative registers.

$01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	KBIE	KBIF	OUTC	K12	K11	K10	K9	K8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	1	u	u	u	u	u

BIT[7] **KBIE** – Keyboard interrupt enable bit

0: Keyboard interrupt is disabled

1: Keyboard interrupt is enabled

BIT[6] **KBIF** – Keyboard interrupt flag bit

0: No keyboard interrupt occurs

1: Keyboard interrupt occurs

Writing a "0" clears KBIF. Writing a "1" has no effect.

8 System Operation Modes

MC10P11B has two low power modes: STOP mode and WAIT mode.

8.1 STOP Mode

The instruct STOP makes MCU enter STOP mode, which has several effects bellow:

- ✧ System primary oscillator stops
- ✧ Clear I bit in CCR, and enable interrupt
- ✧ Data stored in RAM will be maintained
- ✧ All states of GPIO remain System primary oscillator stops
- ✧ All the internal operation stops, except keyboard scanning

If one of the following things happens, MCU will exit from STOP mode.

- ✧ KBI request occurs
- ✧ Any type of system reset occurs

While MCU works under STOP mode, almost all the operations terminate, so the power dissipation is very low.

8.2 WAIT Mode

The instruct WAIT makes MCU enter WAIT mode, which has several effects bellow:

- ✧ CPU clock stops
- ✧ CPU process and internal bus activities terminate
- ✧ Clear I bit in CCR, and enable interrupt
- ✧ Data stored in RAM will be maintained
- ✧ All states of GPIO remain
- ✧ All states of registers remain

If one of the following things happens, CPU clock will restarts and MCU will exit from WAIT mode.

- ✧ Any type of interrupt request occur
- ✧ Any type of system reset occurs

While MCU works under WAIT mode, activities of CPU stop, but the system primary oscillator still works, so the power dissipation is lower than under normal mode.

9 Electrical Specification

9.1 Absolute Rating

Rating	Symbol	Value	Unit
Supply Voltage	VDD	-0.3～6.5	V
Input Voltage	VIN	VSS-0.3～VDD+0.3	V
Operating Temperature	TA	-40～85	℃
Storage Temperature	Tstg	-65～150	℃

9.2 DC Electrical Characteristics

VDD=3V，T=25℃

Characteristics	Symbol	Pin	Condition	Min.	Typ.	Max.	Unit
Operating Voltage	VDD			1.8		5.5	V
Input Leakage Current	V_{leak}	All input ports	VIN=VDD,0			±1	uA
Input High Voltage	V_{ih}	P01、P00		0.7VDD		VDD	V
Input Low Voltage	V_{il}	P01、P00		0		0.3VDD	V
Pull-up Resistance	R_U	P01、P00			100		Kohm
Output High Current	I_{oh}	P01、P00	V_{oh}=2.7V	3	5		mA
Output Low Current	I_{ol1}	P01、P00	V_{ol}=0.3V	10	14		mA
Output Low Current	I_{ol2}	IROUT	V_{ol}=1.5V	300	400		mA
Idle Supply Current	I_{dds}	VDD	VDD=3V in STOP mode		1	3	uA
Dynamic Supply Current	I_{ddc}	VDD	VDD=3V no load			3	mA
LVR Voltage	V_{lvr}		T=0℃～40℃	1.25	1.5	1.75	V

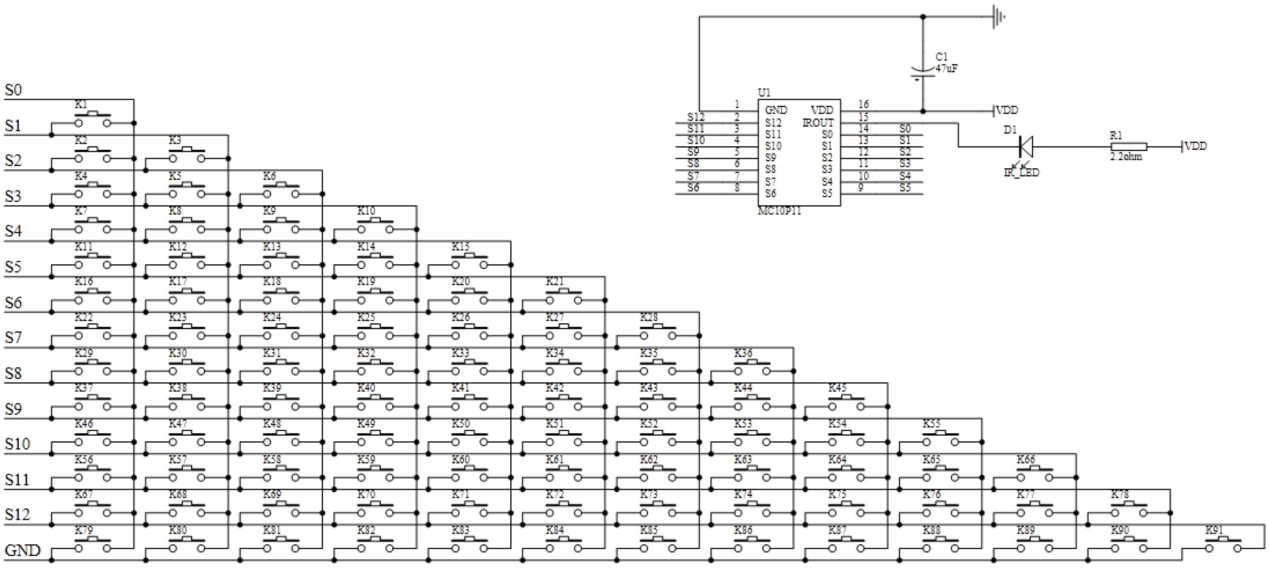
9.3 AC Electrical Characteristics

VDD=3V，T=25℃

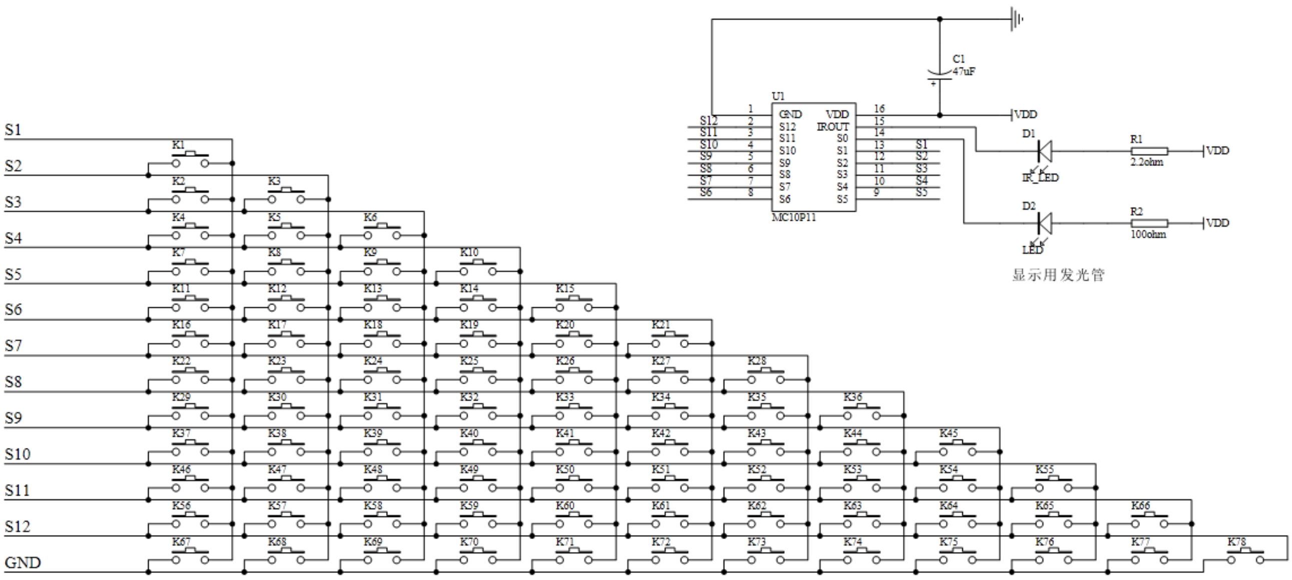
Characteristics	Symbol	Pin	Condition	Min.	Typ.	Max.
Internal RC Frequency	F_{hrc1}	T=25℃ VDD=3V	-1%	4	+1%	MHz
	F_{hrc2}	T=-20℃～70℃ VDD=1.8～5.5V	-2%	4	+2%	MHz

10 Typical Application Schematics

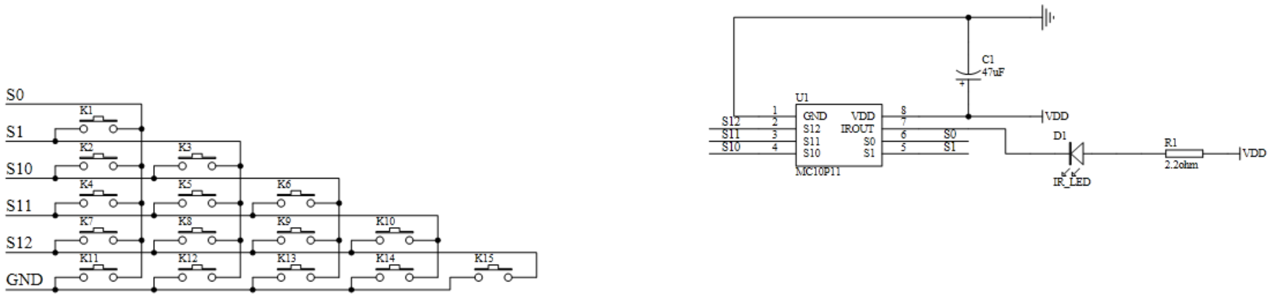
91 keys, without LED for indication



78 keys, with one LED for indication



15 keys, SOP8 package

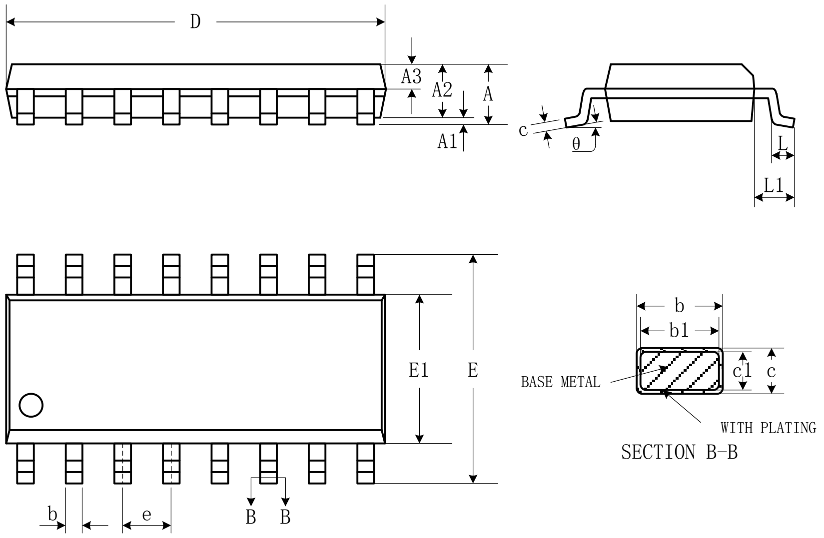


Note1: If using RAM data remaining function, the electrolytic capacitor C2 should not be omitted.

Note2: If the current of the infrared emission diode D1 is particularly large, the current-limiting resistor R1 should not be omitted.

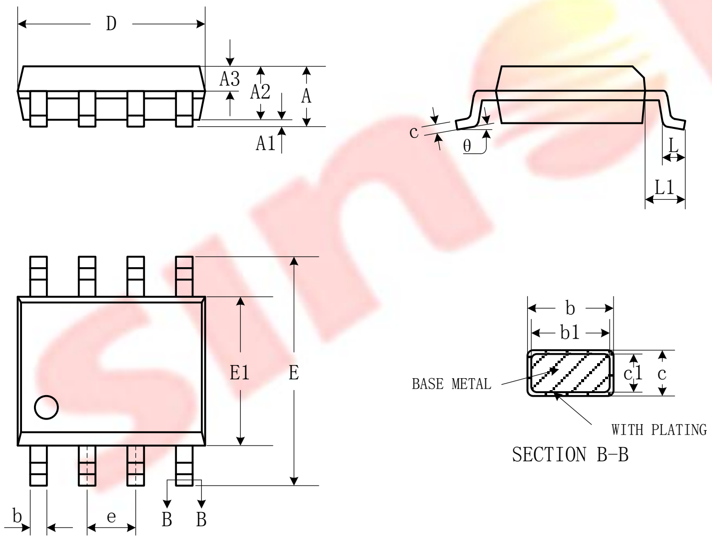
11 Dimension of Package

SOP16



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	9.70	9.90	10.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	-	8°

SOP8

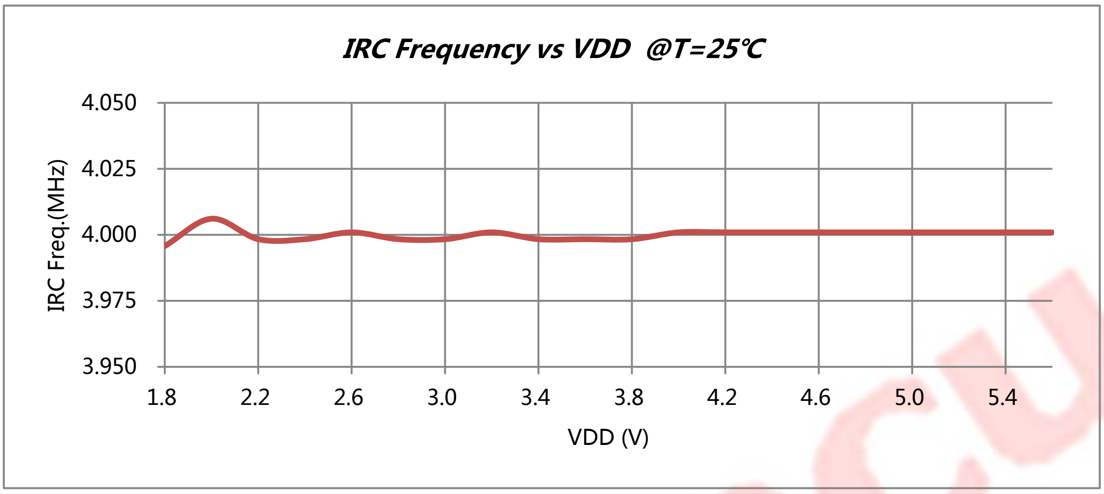


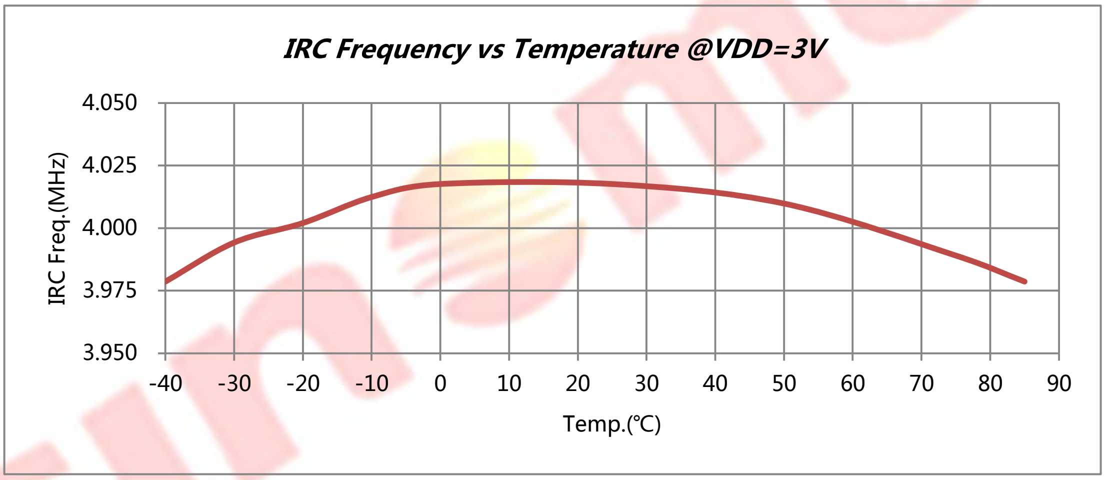
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	4.70	4.90	5.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	-	8°

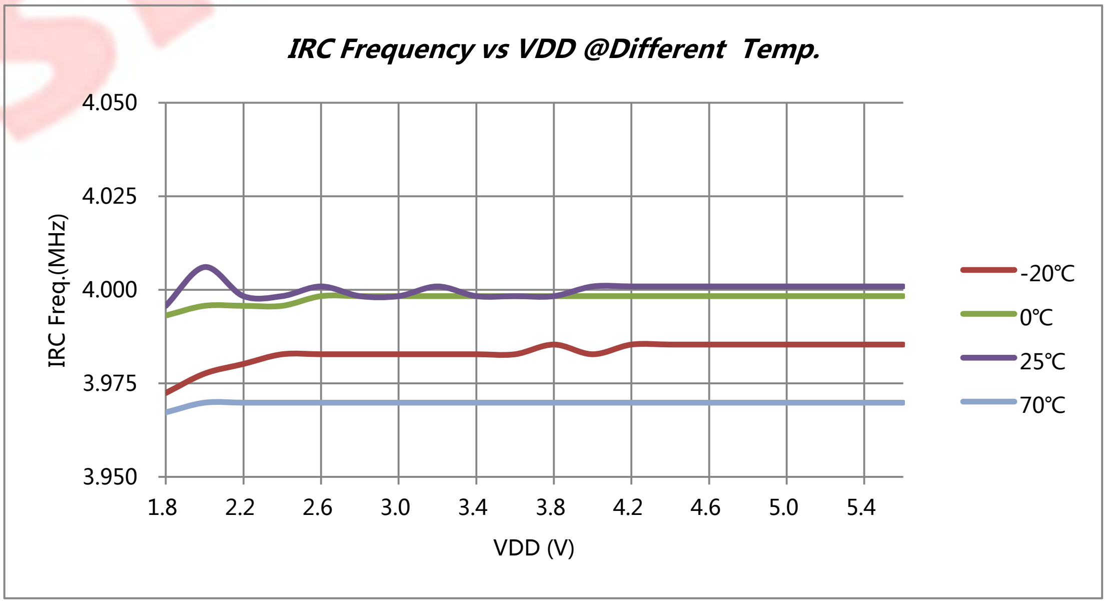
12 Appendix

Note: The contents of appendix is just for your reference.

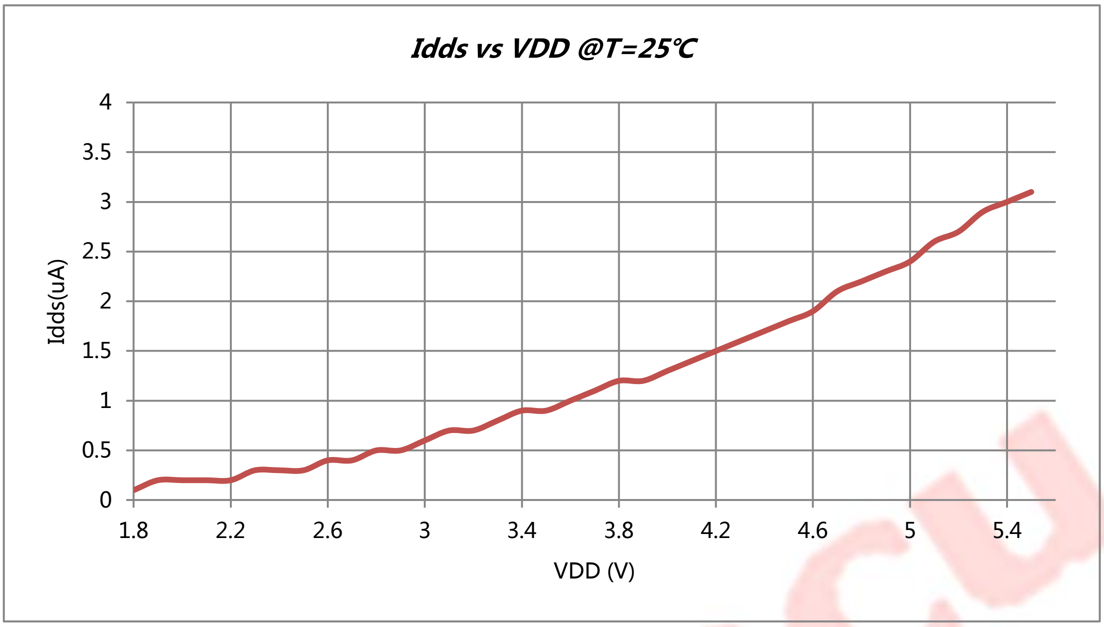
12.1 Internal RC Frequency







12.2 Idle Current



13 Revision History

Version	Date	Description
1.0	2012-3-27	First issued.
1.1	2013-4-28	§1.1 Modify description of RAM data remaining function, add SOP8 package §1.3 Add pin assignment of SOP8 package §10 Add application schematic for SOP8 package §11 Add dimension description of SOP8 package