



九齊科技股份有限公司
Nyquest Technology Co., Ltd.

DATA SHEET

NY9UP02A

Single Remote Controller with 13 I/O

Version 1.1

Oct. 16, 2020

NYQUEST TECHNOLOGY CO., Ltd. reserves the right to change this document without prior notice. Information provided by NYQUEST is believed to be accurate and reliable. However, NYQUEST makes no warranty for any errors which may appear in this document. Contact NYQUEST to obtain the latest version of device specifications before placing your orders. No responsibility is assumed by NYQUEST for any infringement of patent or other rights of third parties which may result from its use. In addition, NYQUEST products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of NYQUEST.

Revision History

<i>Version</i>	<i>Date</i>	<i>Description</i>	<i>Modified Page</i>
1.0	2020/07/31	Formal release	
1.1	2020/10/16	1. PAD description modified; 2. DC characteristics modified; 3. Remove NY9UP02A1 SOP-16.	7 37 -

Table of Contents

1. General Description	5
1.1 Features	5
1.2 Block Diagram	6
1.3 PACKAGE PIN ASSIGNMENT	6
1.4 Pin Description	7
2. Hardware Architecture	8
2.1 Function Block Diagram.....	8
2.2 Arithmetic Logic Unit.....	8
2.3 ALU Related Status Flag	8
2.4 Address Pointer	9
2.5 Program Counter (PC).....	9
2.6 Stack (STK)	9
2.7 Multi-function Register Pointer (RPT).....	9
3. Memory Organization	10
3.1 ROM	10
3.2 RAM	10
3.3 Memory Register	11
3.4 System Register	11
4. Function Description	12
4.1 System Reset	12
4.1.1 <i>System Power-On & Power-Down</i>	12
4.1.2 <i>Low Voltage Reset & Detection (LVR)</i>	12
4.1.3 <i>Watch-Dog Timer Reset (WDTR)</i>	12
4.2 System Control Register	12
4.2.1 <i>Introduction of System Control Register</i>	12
4.2.2 <i>System Control Register Address Map</i>	12
4.2.3 <i>Memory Register Address Map</i>	14
4.2.4 <i>Register without Memory Allocation Map</i>	14
4.3 RPT	14
4.4 RAM Control Register	15
4.4.1 <i>XMD</i>	15

4.5	POWF Register	15
4.6	INT	15
4.6.1	System Base Timer Polling	15
4.6.2	Interrupt Source.....	16
4.6.3	Flag Clear.....	16
4.7	I/O Ports	16
4.7.1	IO control register.....	17
4.8	IR Control Register	18
4.8.1	TM0CS	18
4.8.2	TM0_x / TM0_Dutx.....	18
4.8.3	IRCTRL and ONOFF.....	20
4.9	Power Saving Mode.....	21
4.9.1	Halt Mode.....	21
4.9.2	T-type Scan Mode.....	22
4.10	Internal Oscillator.....	22
5.	Instruction Set.....	23
5.1	Instruction Classified Table	23
5.2	Instruction Descriptions	26
5.2.1	Arithmetic Instructions	26
5.2.2	Conditional Instructions.....	32
5.2.3	Other Instructions.....	34
6.	Electrical Characteristics	39
6.1	Absolute Maximum Rating.....	39
6.2	DC Characteristics	39
7.	Die PAD Diagram	40
8.	APPLICATION.....	41
9.	PACKAGE DIMENSION.....	42
10.	ORDERING INFORMATION	42

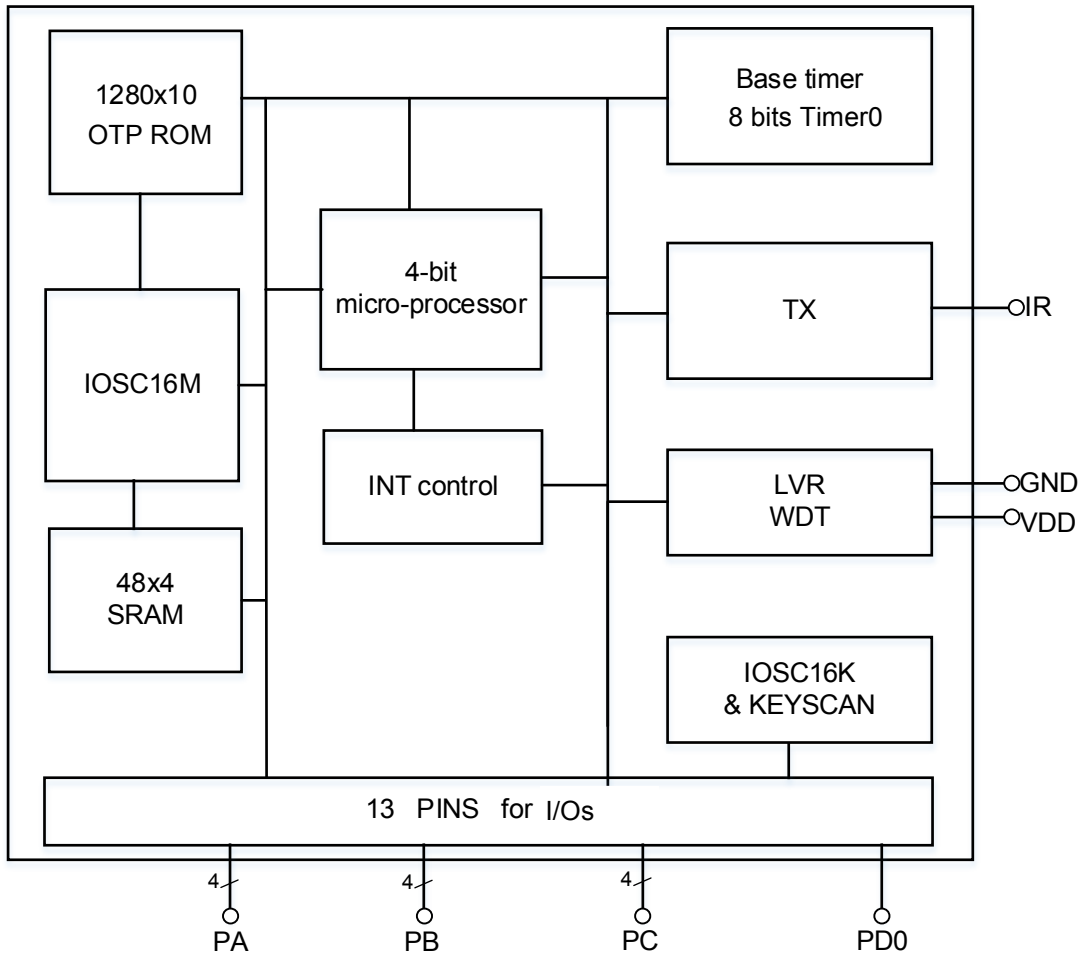
1. General Description

The NY9UP02A is a powerful 4-bit MCU with remote controller. It has 13 I/O ports and supports T-type key matrix. One large sink current IR port can fulfill transmitting function without any bipolar transistor. The RISC MCU architecture is very easy to use, and various applications can be easily implemented. There are 49 instructions, and most of them are executed in single cycle. Furthermore, it provides the HALT mode (sleep mode) to extend battery life.

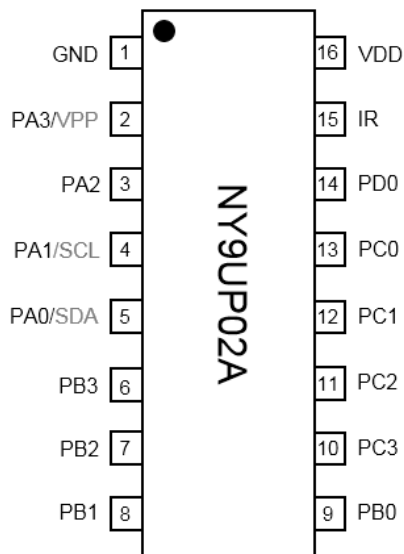
1.1 Features

- Operating voltage range: 1.8V to 3.6V.
- 4-bit RISC type micro-controller with 49 instructions.
- 1280x10-bit OTP ROM
- 48x4-bit RAM, indirect RAM addressing mode is supported.
- 2MHz/1MHz instruction frequency.
- HALT mode to save power, standby current <1uA @3V.
- Precisely embedded oscillator with build-in resistor, +/- 1% deviation in 1.8V~3.6V and -20°C~+70°C.
- Low voltage reset (LVR=1.5V) and watch-dog reset are all supported to protect the system.
- One entrance for interrupt operation with an independent stack, multiple interrupt sources.
- 13 flexible I/Os.
 - 13 I/Os of PAX, PBx, PCx and PD0 with function: bi-direction I/O with pull-high. initial output high · initial output low.
- M-Type, T-type or mixed type key wakeup supported.
- Infrared port provides TX application, optioned with large current IR carrier output or Normal(Sink/Drive) for TX.
- 8-bit readable timer with selectable timer clock source for IR TX carrier frequency.

1.2 Block Diagram



1.3 PACKAGE PIN ASSIGNMENT



1.4 Pin Description

Pad	ATTR.	Description
VDD	Power	Positive power
GND	Power	Negative power
IR	O	Infrared port (TX)
PA0/SDA	I/O	Bit 0 for Port A, or serial data input at programming mode. PA0: 1 flexible I/Os with function: bi-direction I/O with pull-high or Output IO with initial high/low. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode. And can be set as no waking up function IO through option.
PA1/SCL	I/O	Bit 1 for Port A, or serial clock input at programming mode. PA1: 1 flexible I/Os with function: bi-direction I/O with pull-high or Output IO with initial high/low. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode. And can be set as no waking up function IO through option.
PA2	I/O	PA2: 1 flexible I/Os with function: bi-direction I/O with pull-high or Output IO with initial high/low. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode. And can be set as no waking up function IO through option.
PA3/VPP	I/O	Bit 3 for Port A, or positive high power for programming. PA3: 1 flexible I/Os with function: bi-direction I/O with pull-high. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode. And can be set as no waking up function IO through option.
PB0~3	I/O	PB0~3, PC0~1: 6 flexible I/Os with function: bi-direction I/O with pull-high or Output IO with initial high/low. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode.
PC0~1	I/O	
PC2~3	I/O	PC2~3 : 2 flexible I/Os with function: bi-direction I/O with pull-high. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode.
PD0	I/O	PD0: A flexible I/O with function: bi-direction I/O with pull-high or Output IO with initial high/low. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode. And can be set as no waking up function IO through option.

2. Hardware Architecture

2.1 Function Block Diagram

The NY9UP02A belongs to the SRC family of the NYQUEST devices. Block diagram of the device is shown below.

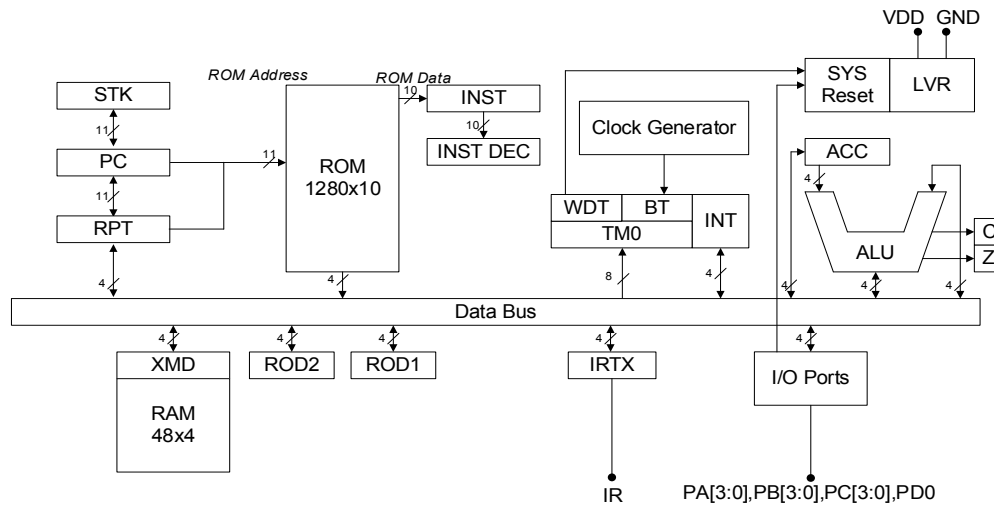


Figure 2-1 NY9UP02A Function Block Diagram

2.2 Arithmetic Logic Unit

The NY9UP02A provides a 4-bit arithmetic logic unit with a 4-bit accumulator to perform logic, unsigned arithmetic, data transfer and conditional branch operation. There are two flags (carry and zero) to indicate the result of the operation. One or two operands will be the data sources of the ALU operation. The operands can be ACC, RAM, register, or literal constant data.

2.3 ALU Related Status Flag

Besides CLRC and SETC commands directly assign the value of the carry flag, C is influenced by the arithmetic result. C means carry and also means the complement of borrow. If the addition operation larger than 0xF, C=1, and C=0 if the result $\leq 0xF$. If the subtraction operation smaller than 0, C=0, and C=1 if the result ≥ 0 .

Symbol	Flag	Description
C	Carry flag	C=1 if a carry-out occurs after an addition operation.
		C=0 if a borrow-in occurs after a subtraction operation.
Z	Zero flag	Z=1 if the result of an ALU operation is zero.

2.4 Address Pointer

The NY9UP02A micro-controller contains a program counter (PC), an interrupt dedicated stack (STK), and a multi-function register pointer (RPT). The length of each address pointer is 11-bit. Users have to keep in mind that the initial value of all the pointers is unknown, except the PC.

2.5 Program Counter (PC)

As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC starts from the reset vector (address 0x000) after the system reset, and its value is increased by one every instruction cycle unless changed by an interrupt or a branch instruction. The interrupt vector for timer & BT is at address 0x010.

2.6 Stack (STK)

One level hardware push/pop stack dedicated to the interrupt is available. When an interrupt occurs, the system pushes the PC to the STK automatically. When the program returns to the main program from the interrupt routine by IRET instruction, the system pops the STK back to the PC.

2.7 Multi-function Register Pointer (RPT)

As implied in the name, RPT are multi-function registers. Users have to operate RPT in coordination with instructions below.

When RD or RDI instruction is used, RPT value represents the ROM address and the read back ROM data is placed on ROD2[1:0], ROD1, ACC, ACC is 4 LSB of ROM data. After RD and RDI operation, the RPT with RDI will add one, but the RD will keep the present value.

Inst./Event	Function
CALL	Pushes PC+2 to RPT.
RJMP	Move RPT to PC
RBPC	Reads back PC+1 to RPT.
RD	Data Table Read to {ROD2[1:0], ROD1, ACC}
RDI	Data Table Read to {ROD2[1:0], ROD1, ACC}, and RPT = RPT+1
INCR	[RPT2[0], RPT1, RPT0]+1

3. Memory Organization

The NY9UP02A has 1280 words OTP ROM, 48 nibbles of RAM and some dedicated system control register. The registers are divided into normal system registers and 3 nibbles of Multi-function registers. The detail is shown in Figure 3-2 .

3.1 ROM

A program data single ROM is provided and its structure is shown below. The reserved region contains system information and can't be utilized by users.

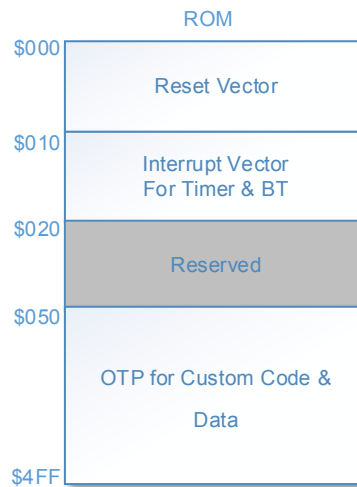


Figure 3-1 NY9UP02A ROM Map

3.2 RAM

NY9UP02A provide 48 nibbles RAM space. The address for RAM is 0x10~0x3F.

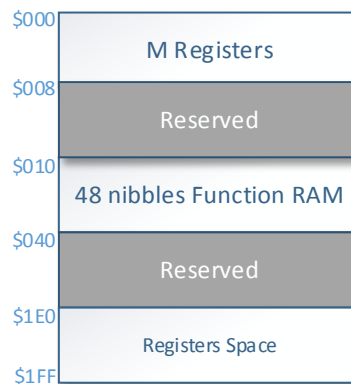


Figure 3-2 NY9UP02A RAM Map

In addition to the immediate addressing mode, the indexed addressing mode is also supported. The address of the indexed RAM should be stored into RPT1 and RPT0 first, and users can read from or write in the XMD0 memory register to realize the indexed RAM access.

3.3 Memory Register

The 6 memory registers share the address with RAM.

Address	Name	Bit	Description
0	RPT0	[3:0]	Multi-function register pointer bit [3:0]
1	RPT1	[3:0]	Multi-function register pointer bit [7:4]
2	RPT2	[2:0]	Multi-function register pointer bit [10:8]
3	--	--	Reserved
4	--	--	Reserved
5	ROD1	[3:0]	ROM data bit [7:4] access register
6	ROD2	[1:0]	ROM data bit [9:8] access register
7	XMD0	[3:0]	Indexed RAM data access register

3.4 System Register

The system registers of NY9UP02A is part of memory space. The memory operation instruction is compatible with register operation.

Address	Name	Bit	Description
0x1E4	PA	[3:0]	PA I/O port control register
0x1E5	PB	[3:0]	PB I/O port control register
0x1E6	PC	[3:0]	PC I/O port control register
0x1E7	PD	[0]	PD I/O port control register
0x1E8	TM0CS	[1:0]	Timer Clock Source selection
0x1E9	POWF	[1:0]	POWER flag register
0x1F3	TM0_L	[3:0]	TM0 counter register low 4 bits
0x1F5	TM0_H	[3:0]	TM0 counter register high 4 bits
0x1F6	TM0_DutL	[3:0]	Duty Low nibble
0x1F7	TM0_DutH	[3:0]	Duty High nibble
0x1FB	BTF	[3:0]	Base timer flag
0x1FC	IRCTRL	[3:0]	IR control register
0x1FD	ONOFF	[3:0]	IR and Timer control
0x1FE	INTF0	[1:0]	Interrupt flag
0x1FF	INT0	[1:0]	Enable/Disable interrupt

4. Function Description

4.1 System Reset

4.1.1 System Power-On & Power-Down

After power-on, the power-on reset initialization will automatically be set out. The system takes about **64ms** to leave from the reset initialization procedure and enters the normal operation and the program counter (PC) will start at the reset vector to execute the desired program.

4.1.2 Low Voltage Reset & Detection (LVR)

When the system enters the normal operation, the power voltage must be kept in an effective working voltage range. If the power voltage is lower than the effective working voltage range, the system will work improperly.

To prevent the system crash, NY9UP02A supplies Low Voltage Reset (LVR) detectors. Once the LVR detector detects a harmful low voltage supply, it will cause a low voltage reset. The so-called “low voltage reset” point of the NY9UP02A IC is about **1.5V**.

4.1.3 Watch-Dog Timer Reset (WDTR)

To recover from program malfunction, the NY9UP02A IC supports an embedded watch-dog timer reset. The WDTR function always works with the program executing. Users have to clear the WDT periodically to prevent from timing up with a reset generation. Typically, the minimum time-up period of the WDT is about 0.45s. The WDT can be cleared by instruction CWDT1 next to CWDT0 only.

4.2 System Control Register

4.2.1 Introduction of System Control Register

The TM0_L, TM0_H, TM0_DutL, TM0_DutH and TM0CS are timer0 control related registers. The IRCTRL is IR control related registers. The PA PB PC and PD are I/O ports registers. The ONOFF register is to turn on block function such as Timer0 and IR. INT0 register is used to enable interrupt entrance for BT, Timer0. INTF0 register are reading flag originated from those interrupt sources and written 0 to reset its flag individually. The combination of RPT0~2 are multi-function register pointer. The C and Z are arithmetic associated flags. The XMD0 is RAM access registers. The ROD1 and ROD2 registers are used to read the ROM data.

4.2.2 System Control Register Address Map

Address	Name	RSTV	R/W	Bit	Data	Description
0x1E4	PA	xxxx	R	[3:0]	0/1	Read port A input pad data
		1111	W	[3:0]	0/1	Write to port A data register

Address	Name	RSTV	R/W	Bit	Data	Description
0x1E5	PB	xxxx	R	[3:0]	0/1	Read port B input pad data
		1111	W	[3:0]	0/1	Write to port B data register
0x1E6	PC	xxxx	R	[3:0]	0/1	Read port C input pad data
		1111	W	[3:0]	0/1	Write to port C data register
0x1E7	PD	111x	R	[3:0]	0/1	Read port D input pad data
		1111	W	[3:0]	0/1	Write to port D data register
0x1E8	TM0CS	11	R/W	[1:0]	00	Timer Clock Source : 1M Hz
					01	Timer Clock Source : 2M Hz
					10	Timer Clock Source : 4M Hz
					11	Timer Clock Source : 8M Hz(Default)
0x1E9	POWF	0	R	[0]	0/1	Reserved
		x	R/W	[1]	0/1	PORF, Power on flag
		00	R	[3:2]	0/1	Reserved
0x1F3	TM0_L	1111	R/W	[3:0]	0/1	R: Read TM0 counter W: Read latch value to TM0
0x1F5	TM0_H	1111	R/W	[3:0]	0/1	(TM0_H Latch)
0x1F6	TM0_DutL	1111	R/W	[3:0]	0/1	Duty low nibble of carrier
0x1F7	TM0_DutH	1111	R/W	[3:0]	0/1	Duty high nibble of carrier (High Latch)
0x1FB	BTF	xxxx	R	[0]	0/1	System base timer 0.064 ms
				[1]	0/1	System base timer 0.128 ms
				[2]	0/1	System base timer 0.256 ms
				[3]	0/1	System base timer 16.348 ms
		0000	W	[3:0]	0000	Set base timer interrupt source = 0.064 ms
					0001	Set base timer interrupt source = 0.128 ms
					0010	Set base timer interrupt source = 0.256 ms
					0011	Set base timer interrupt source = 16.348 ms
				Others	Reserved	
0x1FC	IRCTRL	11	R/W	[0]	0/1	IRTXD, IR TX Data 0: On 1: Off
				[1]	0/1	IRCRY, IR Carrier 0: Off 1: On
		0	R	[2]	0/1	Reserved
		x	R	[3]	0/1	Read PRE_PIR
0x1FD	ONOFF	0000	R/W	[0]	0/1	TM0 Enable / Disable
				[1]	0	Reserved
				[2]	0	Reserved
				[3]	0/1	ADCRY, adjust carrier 0: no adjust 1:adjust to full carrier
0x1FE	INTF0	00xx	W	[0]	0/1	BT interrupt flag, write 0 to clear
				[1]	0/1	TM0 interrupt flag, write 0 to clear
				[2]	0	Reserved

Address	Name	RSTV	R/W	Bit	Data	Description
				[3]	0	Reserved
0x1FF	INT0	0000	R/W	[0]	0/1	Enable / Disable BT interrupt
				[1]	0/1	Enable / Disable TM0 interrupt
				[2]	0	Reserved
				[3]	0	Reserved

4.2.3 Memory Register Address Map

Addr	Name	R/W	Bit	Description	Initial	Wake-up
0	RPT0	R/W	[3:0]	Multi-function register pointer [3:0]	X	U
1	RPT1	R/W	[3:0]	Multi-function register pointer [7:4]	X	U
2	RPT2	R/W	[2:0]	Multi-function register pointer [10:8]	X	U
3	----	----	[3:0]	Reserved		
4	----	----	[3:0]	Reserved		
5	ROD1	R/W	[3:0]	ROM[7:4] data access register	X	U
6	ROD2	R/W	[1:0]	ROM[9:8] data access register	X	U
7	XMD0	R/W	[3:0]	Indexed RAM data access register	X	X

4.2.4 Register without Memory Allocation Map

Name	R/W	Bit	Description	Initial	Wake-up
C	-	1	Arithmetic carry flag	0	U
Z	-	1	Arithmetic zero flag	0	U

R : Can be read from the register

U : Unchanged (the same as before wake-up)

W : Can be written to the register

X : Unknown

4.3 RPT

The RPT of NY9UP02A is 11-bit, and the functions of RPT are listed in the section 2.7. Besides the instructions related to the XMD0 only access bit [8:0] of the RPT, others access all available bits.

The NY9UP02A provides the RD and RDI instruction to read the ROM data out. When RD or RDI is executed, the system takes the RPT as ROM address, and the ROM data is loaded to ROD2, ROD1, and ACC. Bit [9:8] of the ROM data is loaded to ROD2, bit [7:4] to ROD1, and bit [3:0] to ACC. Using RD or RDI to read the data of the reserved ROM area out is unacceptable. The RPT value keeps unchanged after RD, and RDI means the RPT adds 1.

The CALL instruction pushes the PC to the RPT and jump to the subroutine address of the operand 'a'. When the subroutine is finished, use RJMP to come back to the main program.

4.4 RAM Control Register

4.4.1 XMD

Users access XMD0 taking the RPT[8:0] = {RPT2[0], RPT1, RPT0} for NY9UP02A as the RAM address. Users have to watch out that the NY9UP02A does not support using XMD0 to access memory registers, so the RPT[8:0] can't be 0x000~0x007 when accessing XMD. And RPT[8:0] adds 1 automatically, when the instruction INCR is executed.

4.5 POWF Register

The POWF register is used to recognize the status of supply voltage. The POWF[1] is POR Flag and set to 1 by power on over 1.2v. The flag can be cleared by setting 0 to its register. If POR Flag is cleared, even the voltage is lower than 1.6V to cause LVR, the POR flag won't be forced to high unless the power disappears.

Category	Bit	Description
POWF	[1]	PORF: Write 0 (clear) only

4.6 INT

There is one hardware interrupt and it has 2 different sources in NY9UP02A. The interrupt event can be a fixed interval of the system base timer (BT), the timer overflow flag (TOF). The TOF can be selected as one of the sample rate timer's overflows by the register INT. There is a system base timer in the NY9UP02A IC. The NY9UP02A provide 4 fixed intervals from the system base timer for interrupt source: 0.064ms, 0.128ms, 0.256ms and 16.384ms.

As an interrupt occurs, NY9UP02A stores the accumulator (ACC), carry flag (C), zero flag (Z) and RPT automatically. Then move PC to STK, backup RPT. If the interrupt source is Timer0 or base timer, jump to the interrupt vector (0x0010). An interrupt routine finishes with an IRET instruction. The IC draws the ACC, C, Z and RPT back, and moves STK to PC back to jump back the main program.

The interrupt event has to be cleared by users after entering the interrupt routine.

4.6.1 System Base Timer Polling

Reading the 4-bit data of BTF acquires the value of the BT counter. The NY9UP02A provides 4 different base timer intervals for polling: 0.064ms, 0.128ms, 0.256ms and 16.384ms, shown as Figure 3-1. The value of time means the period, so polling a data toggle means half time of the interval. And the base timer restarts to count from 0, when CBT instruction is used.

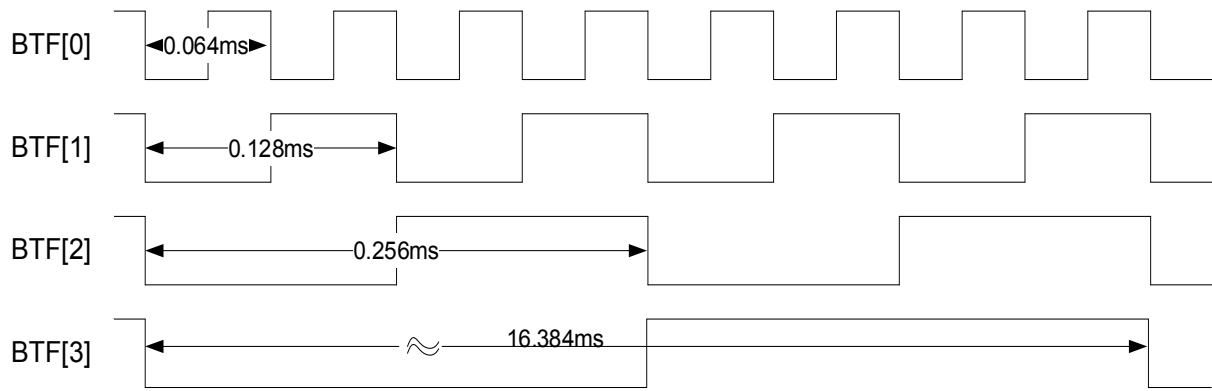


Figure 3-1 INT timing diagram

4.6.2 Interrupt Source

As mentioned in the section 4.6, the only one interrupt has 5 interrupt sources including 4 different BT intervals, the TOF, The INT0 register is interrupt enable/disable control register. The INTF0 is the interrupt flag, and it can be cleaned by writing 0 to related bit. The BTF is base timer interrupt source select by writing, and timer status can be read.

4.6.3 Flag Clear

- Using instruction CWDT0 and CWDT1 to clear the WDT.
- Writing 0x0 to the INTF0[0] clears the Base Timer flag.
- Writing 0x0 to the INTF0[1] clears the TOF.

4.7 I/O Ports

There are 13 I/O pins, designated as PAX, PBx, PCx, PD0, and x=0~3. All pins are IO (bi-direction) with pull-high, and users can set I/O functions by I/O registers. And when the chip is running, the status can be changed by I/O register control.

The table below shows the relation between them.

Category	Option	PX register write 1	PX register write 0
PAX/PBx/PCx/PD0 (x=0~3)	I/O with pull-high	input with pull-high	output low
PA0~2 PB0~3 PC0~1 PD0	initial output high	output high	output low output high
	initial output low	output high	output low

The pull-high resistor of all the I/O ports is about 125KΩ @3V for key matrix function usually.
PA[3:0] and PD0 can be controlled by option to enable or disable wakeup function.

4.7.1 IO control register

Each I/O port has its corresponding register PX. Reading from the register reveals the pad status, and writing to it means writing to the I/O register.

The four registers of PA, PB, PC and PD can configure the corresponding port status. The detail can check the section 4.7.

When IO is configured as a T-Scan application, the IO registers are used to switch I/O between input pull high and output pull low. When the register PX=1, it is an input with pull-high. When PX=0, it is an output and output low level. The register PX value of an output port simply means the output data. Users have to note that reading from an output port also getting the pad potential level, not the register value.

The pull-high resistor of all the I/O ports is only the strong pull-high, which is about 125KΩ @3V for key matrix function usually.

Example 3-1 Setting PA, PD

MVLA	0xC	; PA0 ∙ PA1 output low, PA2 ∙ PA3 input pull-high
MVAM	PA	
MVLA	0x3	; PB0 ∙ PB1 input pull-high, PB2 ∙ PB3 output low
MVAM	PB	

4.8 IR Control Register

The NY9UP02A provides an independent pin (IR) for infrared transmit block, which is used to send infrared signal. For the function of transmitter, users can set a variety of IR carrier frequency by the given clock source (TM0CS), 8-bit duty value (TM0_DutH, TM0_DutL) and 8-bit IR timer value (TM0_H, TM0_L).

4.8.1 TM0CS

The TM0CS register indicates the TM clock source of IR carrier frequency. The different clock sources combined with a variety of value of the 8-bit timer registers will cause different IR carrier frequency.

TM0CS	R/W	[1:0]	00	Timer Clock Source : 1M Hz
			01	Timer Clock Source : 2M Hz
			10	Timer Clock Source : 4M Hz
			11	Timer Clock Source : 8M Hz(Default)
	R/W	[3:2]		00: normal keyscan Halt 01: timer halt, IO floating & gating, wake up after about 0.5s 10: timer halt, IO floating & gating, wake up after about 1s 11: timer halt, IO floating & gating, wake up after about 2s

4.8.2 TM0_x / TM0_Dutx

The TM0_x register includes an 8-bit timer reload value latch and an 8-bit downward counter. Users can set initial timer value by writing TM0_L, TM0_H, and only update after writing TM0_H, and Timer counter value can be reloaded from initial value by CLR TM0. As the counter underflow, counter is automatically reloaded from the latch. For reading those register, users have to read TM0_H first and system will save LSB 4-bit data of timer counter value into TM0_L at same time. Due to the timer counter keeps running, the current counter of TM0_H might be not as same as the time for TM0_L is read back.

With the different counter value, the IR carrier frequency will be affected correspondingly; TM0_Dutx decides the duty of IR carrier. In addition, the timer will be stopped counting if the timer is turned off (ONOFF[0] equals to 0).

The TM0_Dutx register are used to store 8-bit duty value of timer0, writing TM0_DutH for reload duty value.

In order to generate IR carrier, users need to select timer source at first, then set the 8-bit value of timer and 8-bit value of duty. Then set IRCTRL[1:0]=2'b10 to enable IR TX and carrier. At last need to turn on timer0. At this time, the timer starts to countdown from set value, if counter meets the duty value, the IR pin start to sink current driving IRLED transmitting. When Timer downward counter reach 0, IR pin cancels sinking current status, until next timer value matching the value of duty, and IR pin starts to sink again. The carrier will generate automatically through IR pin with specific duty.

The equation of calculating the timer value to generate specific frequency carrier is shown below.

$$TM=(Ftcs/Fca)-1$$

TM: Timer0 value in decimal

Ftcs: Frequency of Timer0 clock

Fca: Frequency of carrier

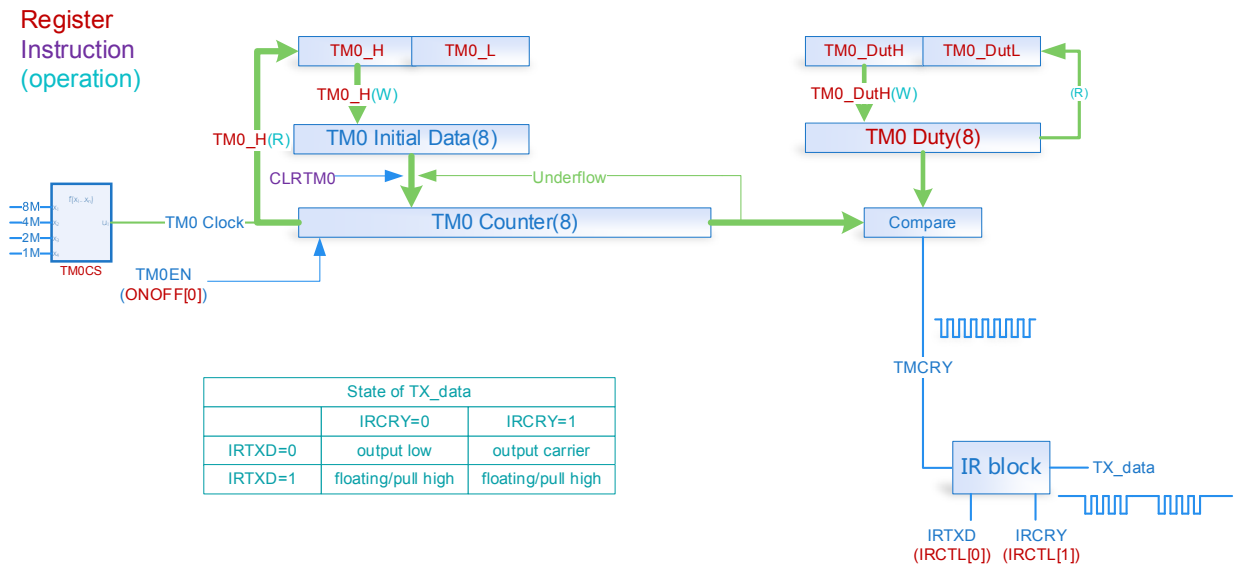


Figure 3-2 Timer diagram for TX

Example 3-2 Generate 38KHz (about 50% duty) carrier wave from IR

```

L_TST_IR:
    MVLA    0x3          ; Set timer0 source = 8M
    MVAM    TM0CS
    MVLA    0x2          ; Setting Timer Data low 4bits
    MVAM    TM0_L

    MVLA    0xD          ; Setting Timer Data high 4bits
    MVAM    TM0_H
    MVLA    0x9          ; Setting timer duty low 4bits
    MVAM    TM0_DutL
    MVLA    0x6          ; Setting timer duty high 4bits
    MVAM    TM0_DutH
    CLR TM0              ;Reset timer0

    MVLA    0x2          ;Enable IRTX and IR carrier
    MVAM    IRCTRL
    MVLA    0x1          ;Enable timer0
    MVAM    ONOFF
    
```

Example 3-3 Timer overflow

```

L_TST_INT_TOF:
    MVLA    0xF           ; Setting Timer Data low 4bits
    MVAM    TM0_L

    MVLA    0xF           ; Setting Timer Data high 4bits
    MVAM    TM0_H
    CLR TM0

    MVLA    0x1           ; Enable timer0
    MVAM    ONOFF
    MVLA    0x2           ; Set INT on, detect timer overflow in interrupt process
    MVAM    INT0
    
```

4.8.3 IRCTRL and ONOFF

The IRCTRL register includes the control of IR TX data enable / disable, the IR carrier disable / enable and read status of TX data from IR PAD(TX_data).

The ONOFF register includes the Timer0 enable/disable. The ONOFF[3] register can adjust the duty in the carrier frame to be complete since the TX data low period is not always divisible by the carrier frequency.

IRCTRL	R/W	[0]	0/1	IRTXD, IR TX Data 0: On 1: Off
		[1]	0/1	IRCRY, IR Carrier 0:off 1: On
	R	[2]	0/1	Reserved
	R	[3]	0/1	read TXD after Carrier adjustment
ONOFF	R/W	[0]	0/1	TM0 Enable / Disable
		[1]	0/1	Reserved
		[2]	0/1	Reserved
		[3]	0/1	ADCRY, adjust carrier 0: no adjust 1:adjust to full carrier

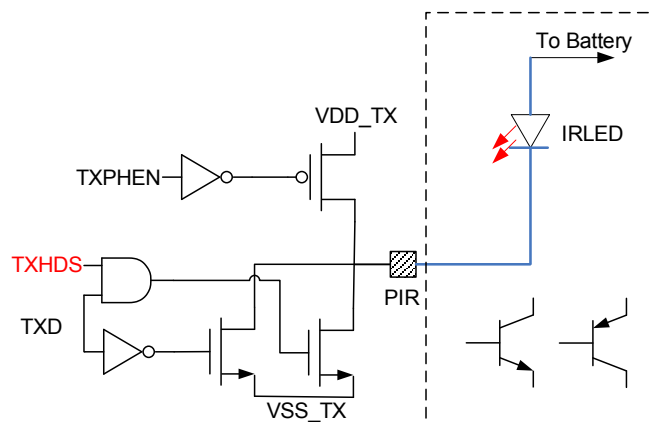


Figure 3-3 TX control & signal phase diagram

4.9 Power Saving Mode

The relationship between power saving mode, reset & normal mode is shown below.

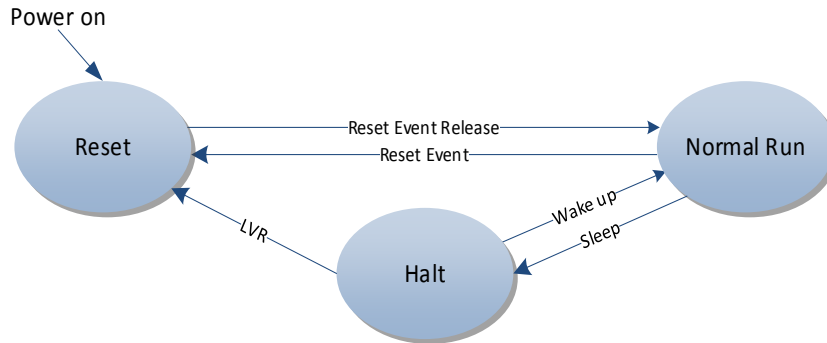


Figure 3-6 Power Saving Mode Flow Diagram

4.9.1 Halt Mode

The system enters the halt mode if the HALT command executed. The halt mode is also known as the sleep mode. As implied by the name, the IC falls asleep and the system clock is completely turned off, so all the IC functions are halted and it minimizes the power consumption.

The only way to wake-up the sleeping system is an input port wake-up. The IC keeps monitoring the input pads during the halt mode. If the input status of any input pad changes to low, the system will be woken-up. Then the succeeding instructions after the HALT instruction will be executed after the wake-up stable time (about 60us). So before executing the HALT instruction, users have to keep in mind that the input port status is high.

If the IC is waked-up from the halt mode by the occurrence of LVR, it goes into the reset procedure.

Example 3-9 Halt Mode operation

```

L_HALT_Loop:
    CWDT0
    CWDT1                                ; Clear watch dog
    MVMA    PA                            ; Confirm PA is high status
    SANL    0xF
    JMP     L_PB_CHK
    JMP     L_HALT_Proc
L_PB_CHK:
    MVMA    PB                            ; Confirm PB is high status
    SANL    0xF
    JMP     L_PC_CHK
    JMP     L_HALT_Proc
L_PC_CHK:
    MVMA    PC                            ; Confirm PC is high status
    SANL    0xF
    JMP     L_PD_CHK
    JMP     L_HALT_Proc
L_PD_CHK:
    MVMA    PD                            ; Confirm PD is high status
    SANL    0xF
    JMP     L_HALT_Loop
L_HALT_Proc:
    NOP
    NOP
    NOP
    HALT                                ; Enter Halt mode
    NOP                                ; If input status I/O change to low, wake up
    NOP
    .....

```

4.9.2 T-type Scan Mode

In T-type scanning application, each port (PA~PD) can be selected as scan key independently by option PXx (X=A~C, x=0~3) & PD0. It works as input with pull-high resistor and output fixed frequency low pulse in halt mode. Any of the keys touch would cause system wake-up.

Meanwhile, the frequency of key scan can be adjusted by option codes, such as about 15.625Hz, 31.25Hz, 62.5Hz and 125Hz.

4.10 Internal Oscillator

The system clocks are 1MHz and 2MHz option which is fast enough for most of applications. The clock generator is a Ring oscillator, and users can only select the internal resistor oscillation (INT-R). The INT-R oscillator accuracy is up to $\pm 0.5\%$, and the deviation is $\pm 1.0\%$ in the full range of 1.8V~3.6V VDD and -20°C~+70°C temperature.

5. Instruction Set
5.1 Instruction Classified Table

<i>Item</i>	<i>Inst.</i>	<i>Op1</i>	<i>Op2</i>	<i>Operation</i>	<i>Inst. Length</i>	<i>Exec. Cycle</i>	<i>Oper. Flag</i>	<i>Flag Affected</i>
Arithmetic Instructions								
1	INCM	6m		{C,M} = M + 1	1	1		C, Z
2	DECM	6m		{C,M} = M - 1	1	1		C, Z
3	ADDA	6m		{C,A} = A + M + C	1	1	C	C, Z
4	XORA	6m		A = A ^ M	1	1		Z
5	ANDA	6m		A = A & M	1	1		Z
6	ORA	6m		A = A M	1	1		Z
7	SUBA	6m		{C,A} = A - M - (B)	1	1	C	C, Z
8	MVAM	6m		Move A to M	1	1		
9	MVMA	6m		Move M to A	1	1		Z
10	RRM	6m		Right Rotate M with C	1	1	C	C, Z
11	RLM	6m		Left Rotate M with C	1	1	C	C, Z
12	SUBL	4L		{C,A} = A - L - (~B)	1	1	C	C, Z
13	ADDL	4L		{C,A} = A + L + C	1	1	C	C, Z
14	XORL	4L		A = A ^ L	1	1		Z
15	ANDL	4L		A = A & L	1	1		Z
16	ORL	4L		A = A L	1	1		Z
17	MVLA	4L		Move L to A	1	1		
18	RRA			Right Rotate A	1	1		
19	RLA			Left Rotate A	1	1		
20	RRC			Right Rotate A with C	1	1	C	C, Z
21	RLC			Left Rotate A with C	1	1	C	C, Z
22	CLRC			Clear Carry flag	1	1		C
23	SETC			Set Carry flag	1	1		C
24	INCA			A = A + 1	1	1		C, Z
25	DECA			A = A - 1	1	1		C, Z
Conditional Instructions								
26	SAGT	4L		Skip when A > L	1	1		
27	SALT	4L		Skip when A < L	1	1		
28	SANL	4L		Skip if A != L	1	1		
29	SCEZ			Skip if C == 0	1	1	C	
30	SCNZ			Skip if C != 0	1	1	C	
31	SZEZ			Skip if Z == 0	1	1	Z	
32	SZMZ			Skip if Z != 0	1	1	Z	
33	SBEZ	2b		Skip when A[b] = 0	1	1		
Other Instructions								
34	NOP			No operation	1	1		
35	CWDT0			Clear WDT Step1	1	1		

<i>Item</i>	<i>Inst.</i>	<i>Op1</i>	<i>Op2</i>	<i>Operation</i>	<i>Inst. Length</i>	<i>Exec. Cycle</i>	<i>Oper. Flag</i>	<i>Flag Affected</i>
36	CWDT1			Clear WDT Step2	1	1		
37	HALT			Enter HALT mode	1	1		
38	RD			Move ROM[RPT] to [ROD2, ROD1, ACC]	1	3		
39	RDI			Move ROM[RPT] to [ROD2, ROD1, ACC] and RPT + 1	1	3		
40	JMP	16a		Jump to Address	2	2		
41	CALL	16a		Jump to Address, and Move PC+2 to RPT	2	2		
42	IRET			Return from interrupt	2	2		
43	RJMP			Move RPT to PC	1	2		
44	RBPC			Move PC+1 to RPT	1	2		
45	SEI			Mask Interrupt	1	1		
46	CLI			Non-mask Interrupt	1	1		
47	INCR			RPT[8:0]+1	1	3		
48	CLRTM0			Timer0 reload latch value	1	1		
49	CBT			Clear Base timer	1	1		

A: 4-bit Accumulator data
B : 1-bit borrow flag data, shared with carry flag, $B=\sim C$
C: 1-bit carry flag data
M: 4-bit RAM or memory register data
R: 4-bit memory register data
L: 4-bit immediately literal data
T: 4-bit System register data
Z: 1-bit zero flag data
TCS: 2-bit clock source selection register of IR carrier frequency
RPT: Multi-function register data
TM: 12-bit timer data of IR carrier
ROM: 10-bit ROM data
ROD: ROM data access register data
PC: Program counter address pointer
STK: Interrupt dedicated stack address pointer

- a: ROM address
- b: bit address
- m: RAM or memory register address
- n: data address Plus/Not plus 1
- r: Memory register address
- t: System register address

5.2 Instruction Descriptions

5.2.1 Arithmetic Instructions

INCM m

Function: Add 1 to M of address m, and save the result back to M.

Operation: $M \leftarrow M + 1$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: None

Flags Affected: C, Z

Example: INCM m0

Before Instruction

M0=0x0

After Instruction

M0=0x1, C=0, Z=0

DECM m

Function: Subtract 1 from M of address m, and save the result back to M.

Operation: $M \leftarrow M - 1$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: None

Flags Affected: C, Z

Example: DECM m0

Before Instruction

M0=0x0

After Instruction

M0=0xF, C=0, Z=0

ADDA m

Function: Add M to A with C and the result is saved back to A.

Operation: $\{ C, A \} \leftarrow A + M + C$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: C

Flags Affected: C, Z

Example :ADDA m0

Before Instruction

A=0x7, M0=0xA, C=0

After Instruction

A=0x1, M0=0xA, C=1, Z=0

XORA m

Function: Exclusive OR A with M of address m, and the result is saved back to A.

Operation: $A \leftarrow A \wedge M$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags:None

Flags Affected: Z

Example: XORA m0

Before Instruction

A=0x3, M0=0xB

After Instruction

A=0x8, M0=0xB, Z=0

ANDA m

Function: AND A with M of address m, and save the result back to A.

Operation: $A \leftarrow A \& M$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: None

Flags Affected: Z

Example: ANDA m0

Before Instruction

A=0x7, M0=0xA

After Instruction

A=0x2, M0=0xA, Z=0

ORA m

Function: Inclusive OR A with M of address m, and save the result back to M.

Operation: $A \leftarrow A | M$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: None

Flags Affected: Z

Example: ORA m0

Before Instruction

A=0x3, M0=0x8

After Instruction

A=0xB, M0=0x8, Z=0

SUBA m

Function : Subtract M of address m from A with B, i.e. The (B) quantity effectively implements a borrow capability for multi-precision subtractions.

Operation : $\{ C, A \} = A - m - B$

Operand : $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: C

Flags Affected: C, Z

Example : SUBA m0

Before Instruction

A=0xA, M0=0x2, C=1

After Instruction

A=0x8, M0=0x2, Z=0, C=1

MVAM m

Function: Move A to M of address m.

Operation: $M \leftarrow A$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: None

Flags Affected: None

Example: MVAM m0

Before Instruction

A=0x8

After Instruction

M0=0x8

MVMA m

Function: Move M of address m to A.

Operation: $A \leftarrow M$

Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: None

Flags Affected: Z

Example: MVMA m0

Before Instruction

M0=0x8

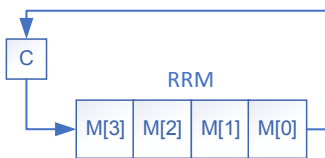
After Instruction

A=0x8

RRM m

Function: Right Rotate M with C.

Operation: $\{ C, M[3], M[2], M[1], M[0] \} \rightarrow \{ M[0], C, M[3], M[2], M[1] \}$



Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: C

Flags Affected: C, Z

Example: RRM m0

Before Instruction

m0=0x3, C=1

After Instruction

m0=0x9, C=1, Z=0

RLM m

Function: Left Rotate M with C.

Operation: $\{ C, M[3], M[2], M[1], M[0] \} \rightarrow \{ M[3], M[2], M[1], M[0], C \}$



Operand: $0x0 \leq m0 \leq 0x3F$
 $0x40 \leq m1 \leq 0x1FF$

Words: 1 (m0), 2 (m1)

Cycles: 1 (m0), 2 (m1)

Operative Flags: C

Flags Affected: C, Z

Example: RLM m0

Before Instruction

m0=0xE, C=1

After Instruction

m0=0xD, C=1, Z=0

SUBL L

Function : Subtract L from A with B, i.e. The (B) quantity effectively implements a borrow capability for multi-precision subtractions.

Operation : $\{ C, A \} = A - L - B$

Operand: $0x0 \leq L \leq 0xF$

Words : 1

Cycles : 1

Operative Flags: C

Flags Affected: C, Z

Example : SUBL 0x2

Before Instruction

A=0xA, L=0x2, C=1

After Instruction

A=0x8, Z=0, C=1

ADDL L

Function: Add L and C to A.

Operation: $A \leftarrow A + L + C$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1

Operative Flags: C

Flags Affected: C, Z

Example: ADDL 0x9

Before Instruction

A=0xB, C=1

After Instruction

A=0x5, C=1, Z=0

ANDL L

Function: AND A with L.

Operation: $A \leftarrow A \& L$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: ANDL 0xA

Before Instruction

A=0x0

After Instruction

A=0x0, Z=1

XORL L

Function: Exclusive OR A with L.

Operation: $A \leftarrow A \wedge L$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: XORL 0xA

Before Instruction

A=0x9

After Instruction

A=0x3, Z=0

ORL L

Function: Inclusive OR A with L.

Operation: $A \leftarrow A | L$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: ORL 0xA

Before Instruction

A=0x9

After Instruction

A=0xB, Z=0

MVLA L

Function: Move L to A.

Operation: $A \leftarrow L$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVLA 0x5

Before Instruction

A=0x8

After Instruction

A=0x5

RLA

Function : Left rotate A.

Operation: $\{ A[3], A[2], A[1], A[0] \} \rightarrow \{ A[2], A[1], A[0], A[3] \}$



Operand: None

Words : 1

Cycles : 1

Operative Flags: None

Flags Affected: None

Example :RLA

Before Instruction

A =0x3

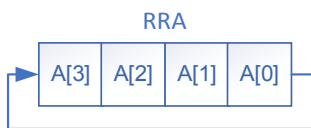
After Instruction

A =0x6

RRA

Function : Right rotate A.

Operation: $\{ A[3], A[2], A[1], A[0] \} \rightarrow \{ A[0], A[3], A[2], A[1] \}$



Operand: None

Words : 1

Cycles : 1

Operative Flags: None

Flags Affected: None

Example :RRA

Before Instruction

A =0xE

After Instruction

A =0x7

RRC

Function : Right rotate A with C.

Operation: $\{ C, A[3], A[2], A[1], A[0] \} \rightarrow \{ A[0], C, A[3], A[2], A[1] \}$



Operand: None

Words : 1

Cycles : 1

Operative Flags: C

Flags Affected: C, Z

Example :RRC

Before Instruction

A =0x3 ,C=1

After Instruction

A =0x9 ,C=1 ,Z=0

RLC

Function : Left rotate A with C.

Operation: { C, A[3], A[2], A[1], A[0]} → { A[3], A[2], A[1], A[0], C }



Operand: None

Words : 1

Cycles : 1

Operative Flags: C

Flags Affected: C, Z

Example :RLC

Before Instruction

A =0xE ,C=0

After Instruction

A =0xC ,C=1 ,Z=0

CLRC

Function: Clear C to 0.

Operation: C ← 0

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C

Example: CLRC

Before Instruction

C=1

After Instruction

C=0

SETC

Function: Set C to 1.

Operation: C ← 1

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C

Example: SETC

Before Instruction

C=0

After Instruction

C=1

INCA

Function: Add 1 to A.

Operation: A ← A + 1

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C, Z

Example: INCA

Before Instruction

A=0xF

After Instruction

A=0x0, C=1, Z=1

DECA

Function: Subtract 1 from A.

Operation: $A \leftarrow A - 1$

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C, Z

Example: DECA

Before Instruction

A=0x1

After Instruction

A=0x0, C=1, Z=1

5.2.2 Conditional Instructions
SAGT L

Function: Skip the next instruction if A greater than L.

Operation: Skip next if $A > L$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1, (2, 3)

Operative Flags: None

Flags Affected: None

Example: SAGT 0x8

Inst1

Inst2

After Instruction

If $A = (\text{or } <) 0x8$, 'Inst1' is executed.

If $A > 0x8$, 'Inst1' is discarded, and 'Inst2' is executed.

SALT L

Function: Skip the next instruction if A greater than L.

Operation: Skip next if $A < L$

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1, (2, 3)

Operative Flags: None

Flags Affected: None

Example: SALT 0x8

Inst1

Inst2

After Instruction

If $A = (\text{or } >) 0x8$, 'Inst1' is executed.

If $A < 0x8$, 'Inst1' is discarded, and 'Inst2' is executed.

SANL L

Function: Skip the next instruction if A not equal L.

Operation: Skip next if A != L

Operand: $0x0 \leq L \leq 0xF$

Words: 1

Cycles: 1, (2, 3)

Operative Flags: None

Flags Affected: None

Example: SANL 0x8

Inst1

Inst2

After Instruction

If $A \neq 0x8$, `Inst1` is discarded, and `Inst2` is executed

If $A = 0x8$, `Inst1` is executed.

SCNZ

Function: Skip the next instruction if C equal to 1.

Operation: Skip next if C = 1

Operand: None

Words: 1

Cycles: 1, (2, 3)

Operative Flags: C

Flags Affected: None

Example: SCNZ

Inst1

Inst2

After Instruction

If $C \neq 0x1$, `Inst1` is executed.

If $C = 0x1$, `Inst1` is discarded, and `Inst2` is executed.

SCEZ

Function: Skip the next instruction if C equal to 0.

Operation: Skip next if C=0

Operand: None

Words: 1

Cycles: 1, (2, 3)

Operative Flags: C

Flags Affected: None

Example: SCEZ

CALL a1

CALL a2

After Instruction

If $C \neq 0$ `CALL a1` is executed

If $C = 0$ `CALL a1` is discarded, and `CALL a2` is executed

SZEZ

Function: Skip the next instruction if Z equal to 0.

Operation: Skip next if Z=0

Operand: None

Words: 1

Cycles: 1, (2, 3)

Operative Flags: Z

Flags Affected: None

Example: SZEZ

CALL a1

CALL a2

After Instruction

If $Z \neq 0$ `CALL a1` is executed

If $Z = 0$ `CALL a1` is discarded, and `CALL a2` is executed

SZNZ

Function: Skip the next instruction if Z equal to 1

Operation: Skip next if Z = 1

Operand: None

Words: 1

Cycles: 1, (2, 3)

Operative Flags: Z

Flags Affected: None

Example: SZNZ

Inst1

Inst2

After Instruction

If $Z \neq 0x1$, `Inst1` is executed.

If $Z = 0x1$, `Inst1` is discarded, and `Inst2` is executed.

SBEZ b

Function: Skip the next instruction if A[b] is not set.

Operation: Skip next if $A[b]=0$.

Operand: $0x0 \leq b \leq 0x3$

Words: 1

Cycles: 1, (2, 3)

Example: SBEZ 0x3

Inst1

Inst2

After Instruction

If $A[3]=1$, `Inst1` is executed.

If $A[3]=0$, `Inst1` is discarded, and `Inst2` is executed.

5.2.3 Other Instructions
NOP

Function: No operation.

Operation: None

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: NOP

After Instruction

No operation for 1 cycle.

CWDT0

Function: Clear Watch Dog Timer Step1.

Operation: Step1 for clear Watch Dog Timer

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: CWDT0

Before Instruction

WDT counter = ???

After Instruction

WDT counter = ???

CWDT1

Function: Clear Watch Dog Timer Step2.

Operation: Watch dog counter \leftarrow 0x0

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: CWDT0

CWDT1

Before Instruction

WDT counter = ???

After Instruction

WDT counter = 0x0

Note : The CWDT0/CWDT1 instructions have to be executed step by step, otherwise the watch dog timer won' t be clear.

HALT

Function: Enter the halt (sleep) mode.

Operation: Stop system clock

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: HALT

After Instruction

The system enters the halt mode and the system clock is halted.

RD

Function: Read ROM data out to A and ROD using the RPT as address (data pointer).

Operation: A \leftarrow ROM data [3:0]

ROD1 \leftarrow ROM data [7:4]

ROD2 \leftarrow ROM data [9:8]

Operand: None

Words: 1

Cycles: 3

Operative Flags: None

Flags Affected: None

Example: RD

After Instruction

A=ROM[3:0] @ RPT

ROD1=ROM[7:4] @ RPT

ROD2=ROM[9:8] @ RPT

RDI

Function: Read ROM data out to A and ROD using the RPT as address (data pointer).

Operation: A \leftarrow ROM data [3:0]

ROD1 \leftarrow ROM data [7:4]

ROD2 \leftarrow ROM data [9:8]

RPT \leftarrow RPT + 1

Operand: None

Words: 1

Cycles: 3

Operative Flags: None

Flags Affected: None

Example: RDI

After Instruction

A=ROM[3:0] @ RPT

ROD1=ROM[7:4] @ RPT

ROD2=ROM[9:8] @ RPT

RPT = RPT +1

JMP a

Function: Unconditionally jump by a direct address a.

Operation: $PC \leftarrow a$

Operand: $0x0 \leq a \leq 0x4FF$

Words: 2

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: `JMP a1`

Before Instruction

PC=a0

After Instruction

PC=a1

IRET

Function: Return from the interrupt sub-routine.

Operation: $PC \leftarrow STK$

Operand: None

Words: 2

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: `IRET`

Before Instruction

PC=a0

After Instruction

PC=STK

ACC, PG, C, and Z are restored to the values that are backed up when entering the ISR.

CALL a

Function: Call subroutine by a direct address a, and save next address to RPT.

Operation: $RPT \leftarrow PC+2$

$PC \leftarrow a$

Operand: $0x0 \leq a \leq 0x4FF$

Words: 2

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: `CALL a1`

Before Instruction

PC=a0

After Instruction

PC=a1, RPT=a0+2

RJMP

Function: Load RPT to PC. Unconditionally jump by the indirect address RPT. The address should be loaded into RPT first.

Operation: $PC \leftarrow RPT$

Operand: None

Words: 1

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: `RJMP`

Before Instruction

RPT=0x0321

After Instruction

PC=0x0321

RBPC

Function: Read address in PC to RPT.

Operation: $RPT \leftarrow PC+1$

Operand: None

Words: 1

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: RBPC

Before Instruction

PC=0x0234

After Instruction

RPT=0x0235

CMKI

Function: Enable interrupt entrance

Operation: Non-mask interrupt

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: CMKI

Before Instruction

Interrupt entrance is disable

After Instruction

Interrupt entrance is enabled

SMKI

Function: Disable interrupt entrance

Operation: Mask all interrupt

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: SMKI

Before Instruction

Interrupt entrance is enable

After Instruction

Interrupt entrance is disable

INCR

Function: Increase RPT by 1.

Operation: $\{RPT2[0], RPT1, RPT0\} \leftarrow \{RPT2[0], RPT01, RPT0\} +1$

Operand: None

Words: 1

Cycles: 3

Operative Flags: None

Flags Affected: None

Example: INCR

Before Instruction

RPT=0x0134

After Instruction

RPT=0x0135

CLRTM0

Function: Reload Timer0 initial latch data to Timer0 counter.

Operation: Timer0 counter reload

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: CLRTM0

Before Instruction

Timer0 counter=0xAFF

Timer0 initial data latch=0x321

After Instruction

Timer0 counter=0x321

CBT

Function: Clear whole base timer counter to zero.

Operation: $BT \leftarrow 0$

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: CBT

Before Instruction

BT=?

After Instruction

BT=0

6. Electrical Characteristics

The following lists the electrical characteristics of the NY9UP02A OTP chip.

6.1 Absolute Maximum Rating

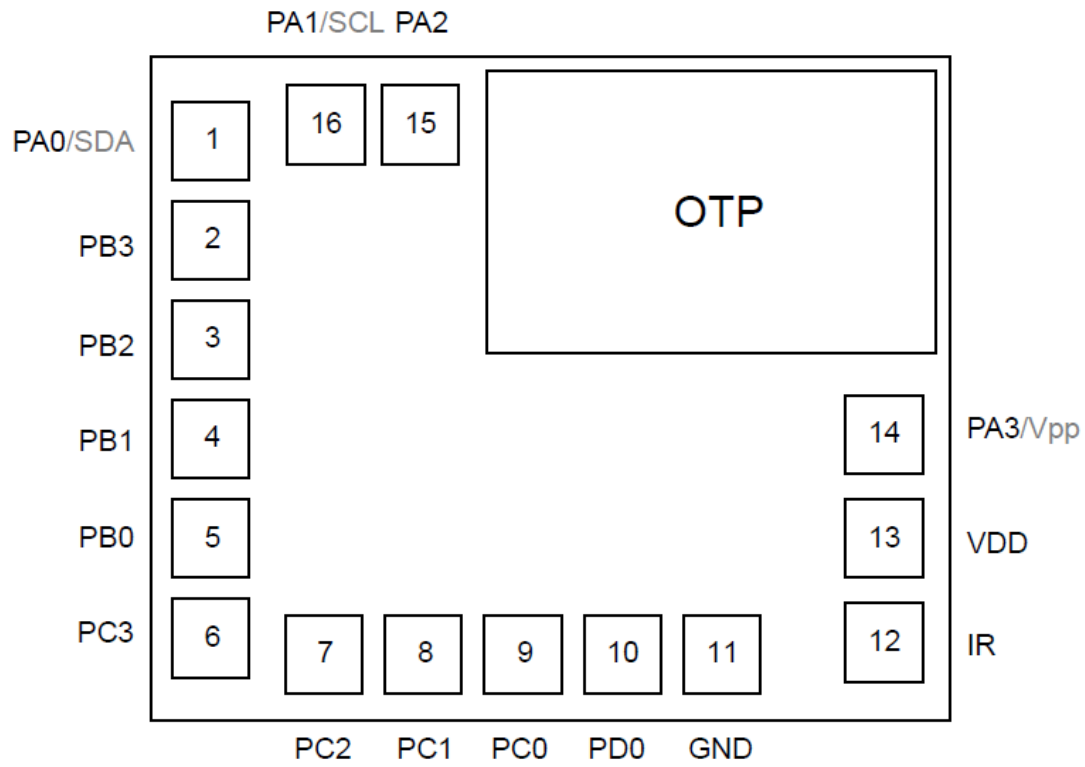
Symbol	Parameter	Rated Value	Unit
V _{DD} - GND	Supply voltage	-0.3 ~ +4.0	V
V _{IN}	Input voltage	GND-0.3V ~ V _{DD} +0.3	V
T _{OP}	Operating Temperature	-20 ~ +70	°C
T _{ST}	Storage Temperature	-40 ~ +85	°C

6.2 DC Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V _{DD}	Operating voltage	1.8	3	3.6	V	1 MHz / 2MHz
I _{SB}	Supply current			1	uA	Sleep, no load
I _{Scan}				2	uA	T-type key scan
I _{OP}			0.5		mA	2MHz, no load
I _{IL}	Input current (Internal 125KΩ pull-high)		24		uA	V _{IL} = 0V
I _{OH}	Output high current		7		mA	V _{OH} = 2.0V
I _{OL}	Output low current		16		mA	V _{OL} = 1.0V
V _{IL}	input low level		0.5*V _{DD}		V	V _{DD} =3.0V
V _{IH}	input high level		0.7*V _{DD}		V	V _{DD} =3.0V
I _{IR}	IR sink current		450		mA	V _{IR} = 1.5V
ΔF/F	Internal OSC frequency deviation	-1.5		1.5	%	V _{DD} : 1.8V ~ 3.6V, Temp: -20°C ~ +70°C

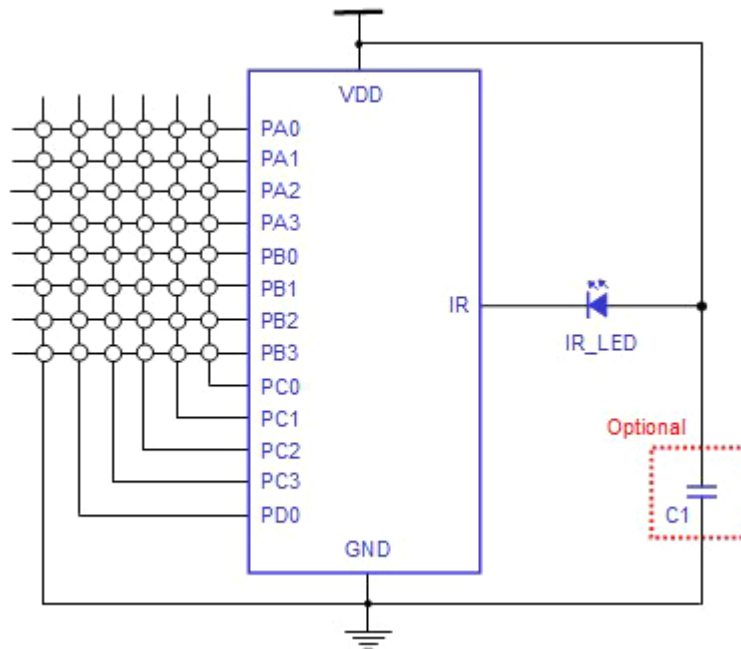
Note: (V_{DD}=3.0V, T_A=25°C, unless otherwise specified)

7. Die PAD Diagram

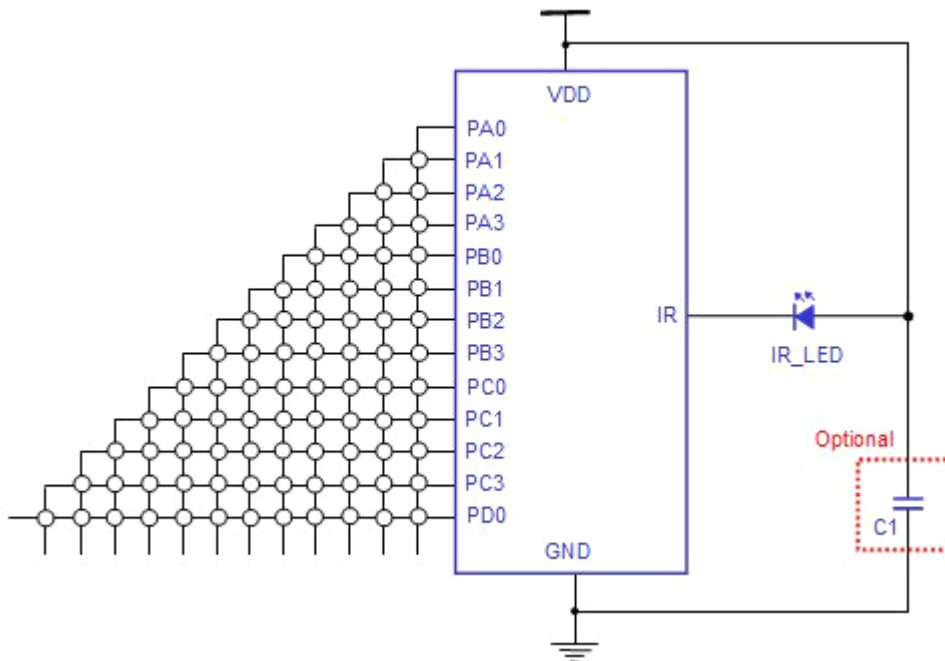


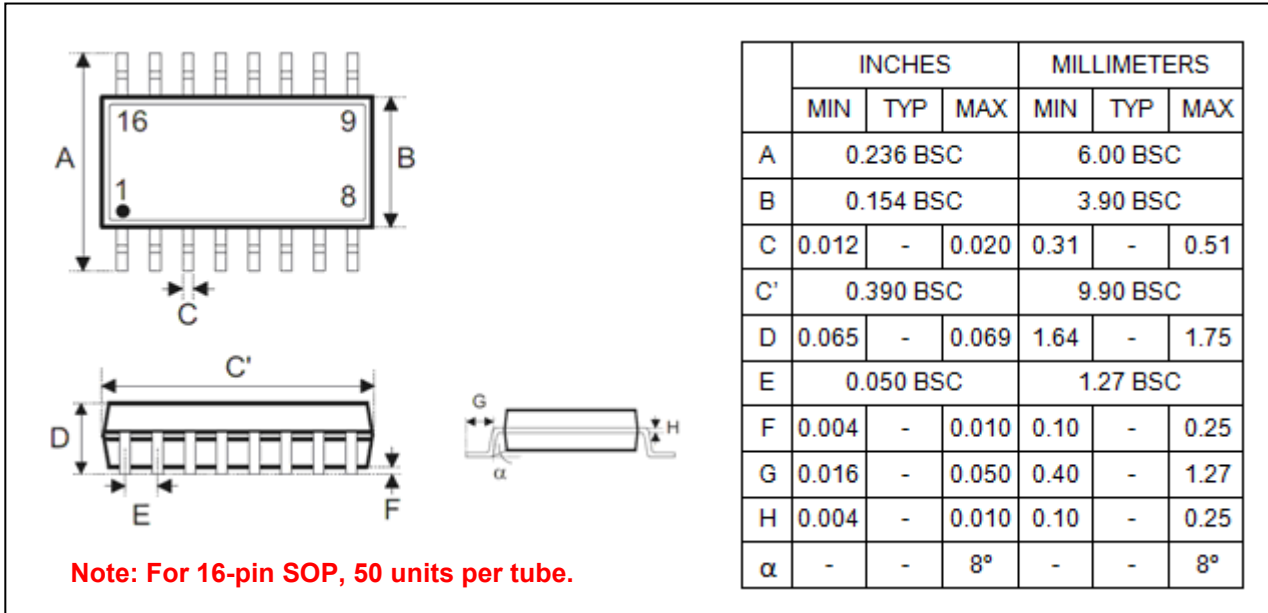
8. APPLICATION

(1) M-type Key Scan Keyboard (Matrix)



(2) T-type Key Scan Keyboard (T-Scan)



9. PACKAGE DIMENSION

10. ORDERING INFORMATION

<i>P/N</i>	<i>Package Type</i>	<i>Pin Count</i>	<i>Package Width</i>	<i>Shipping</i>
NY9UP02AW	Wafer	-		
NY9UP02A	SOP	16	150 mil	Tape & Reel: 2.5K pcs per Reel Tube: 50 pcs per Tube