PAJ7620U2

Integrated Gesture Recognition Sensor with I²C Interface

Datasheet

General Description

PAJ7620U2 integrates gesture recognition function with general I²C interface into a single chip. It can recognize 9 gestures including move up, move down, move left, move right, move forward, move backward, circle-clockwise, circle-counter clockwise, and wave. These gestures information can be simply accessed via the I²C bus. The PAJ7620U2 also offers built-in proximity detection for the purpose of sensing object approaching or departing. The PAJ7620U2 is designed with great flexibility in power-saving mechanism.

The PAJ7620U2 is designed to operate from 2.8V to 3.3V over -40°C to +85°C and the pull-up voltage for the I²C bus and interrupt line is from 1.8V to 3.3V.

Feature

- Typical supply voltage is 2.8V to 3.3V and I/O voltage is 1.8V~3.3V
- Nine gesture recognition (Up / Down / Left / Right / Push / Pull / CW / CCW / Wave)
- Gesture speed is 60°/s to 600°/s in Normal Mode and 60°/s to 1200°/s in Gaming Mode
- Ambient light immunity: < 100k Lux
- Built-in proximity detection
- Flexible power saving scheme
- I²C interface up to 400 kbit/s, Pull-up voltage from 1.8V to 3.3V
- Ambient light noise cancellation

Application

•PAD Phone
•Tablet Personal Computer
•Automobile Application

Pin Configuration

Pin NO.	Symbol	Type	Function
1	V_{BUS}	POWER	BUS power supply
2	I2C_SDA	IN/OUT (Open Drain)	I²C data pin
3	INT	OUT(Open Drain)	Interrupt pin (Active low)
4	TESTMD	IN	For Module Test Only
5	I2C_SCL	IN (Open Drain)	I²C clock pin
6	GND	GND	Ground
7	GPIO3	IN/OUT	For Module Test Only
8	GPIO2	IN/OUT	For Module Test Only
9	GPIO1	IN/OUT	For Module Test Only
10	GND	GND	Ground
11	V_{LED}	POWER	LED power input
12	V_{DD}	POWER	Main power supply
13	GPIO0	IN/OUT	For Module Test Only

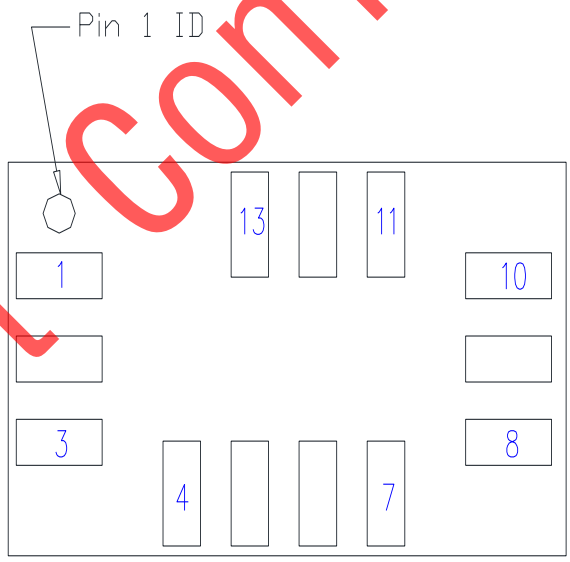


Figure 1. PAJ7620U2 Module Pin Configuration (BTM VIEW)

Ordering information

Part Number	Packing	Pin NO.	Package Type & Size	Options
PAJ7620U2	-	13	-	I²C Slave ID(using 7 bit addressing protocol): 0x73

Absolute Maximum Ratings, T_A = 27°C

Description	Symbol	Min.	Max.	Unit
Supply Voltage	V_{DD}	-	4	V
LED Supply Voltage	V_{LED}		4.6	V
LED Pulse Current *Note1.*	I_{LED}	-	2	A
I²C Pin, INT_N Pin Voltage (SCL, SDA, INT_N)	V_{BUS}	-0.3	V_{DD}+0.3	V
I²C Pin, INT_N Pin Current (SCL, SDA, INT_N)	I_{BUS}	-	10	mA
ESD, human body model	ESD_{HBM}	-	2	kV
ESD, Machine model	ESD_{MM}	-	200	V

Note1.Pulse Width < 500us, Duty Cycle < 5%

** Maximum Ratings are those values beyond which damage to the device may occur. Exposure to these conditions or conditions beyond those indicated may adversely affect device reliability. Functional operation under absolute maximum-rated conditions is not implied. Functional operation should be restricted to the Recommended Operating Conditions.*

Recommended Operating Condition

Description	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{DD}	2.8	-	3.6	V
LED Supply Voltage	V_{LED}	3	-	4.2	V
Peak LED Current Pulse Note1	I_{LED}	-	720	860	mA
I²C Pin, INT_N Pin Voltage (SCL, SDA, INT_N)	V_{BUS}	1.8	-	3.3	V
I²C Pin, INT_N Pin Current (SCL, SDA, INT_N)	I_{BUS}	-	-	5	mA
Operating Temperature	T_{op}	-20	-	70	°C

Note1.Pulse Width < 500us, Duty Cycle < 5%

General Specification, V_{DD}=2.8V, T_A = 27°C

Electrical Specifications						
Description	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Condition**
Supply Voltage	V_{DD}	2.8	-	3.6	V	
LED Supply Voltage	V_{LED}	3.0	-	4.2	V	LED Supply Voltage
I²C Pin, INT_N Pin Pull-up Voltage	V_{BUS}	1.8	-	3.3	V	
Current Consumption for Operation Modes	IDD	-	2.82	-	mA	1.Under Normal Mode 2.Including LED current (Peak = 760mA)
Suspend Current	IDD_SUS	-	15	-	uA	
Current Consumption for Standby State 1	IDD_ST1	-	2.3	-	mA	(Ref. to "Operating Principle") 1.Under Normal Mode 2. $S_{1,\ Response\ Factor}$ =0.5 3. $S_{2,\ Response\ Factor}$ =0.25 4.Including LED current (Peak = 760mA)
Current Consumption for Standby State 2	IDD_ST2	-	1.5	-	mA	
Current Consumption for Proximity Detection			0.2		mA	1.Detecting Rate = 10Hz 2.LED peak current = 600mA 3.LED on time = 6.8μs
I²C Bus Input High Voltage	VIH	0.7* V_{BUS}	-	V_{BUS} +0.3	V	I²C Bus Input High Voltage
I²C Bus Input Low Voltage	VIL	-0.3	-	0.3* V_{BUS}	V	I²C Bus Input Low Voltage
INT_N, SDA Output Low Voltage	VOL_SDA VOL_INT_N	-	-	0.1* V_{BUS}	V	INT_N, SDA Output Low Voltage
Gesture Function Specifications						
Description	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Condition**
Gesture Detecting Range	d_{OP}	5	-	15	cm	Calculated from PAJ7620U2 sensor center
Gesture Detecting Angle	θ_{OP}	-	60	-	degree	Calculated diagonally
Gesture Speed Response	ω	60		600	degree/s	Angular velocity under Normal Mode
		60		1200		Angular velocity under Gamming Mode
Gesture Update Rate	f_{Update}	-	120	240	Hz	120 Hz for Normal Mode 240 Hz for Gaming Mode
Sun Light Immunity	S_{LUX}	-	-	100k	Lux	Under Florescent light
LED View Angle	$2\theta_{1/2}$		60		degree	
LED Peak Wavelength	λ		940		nm	

Function Diagram

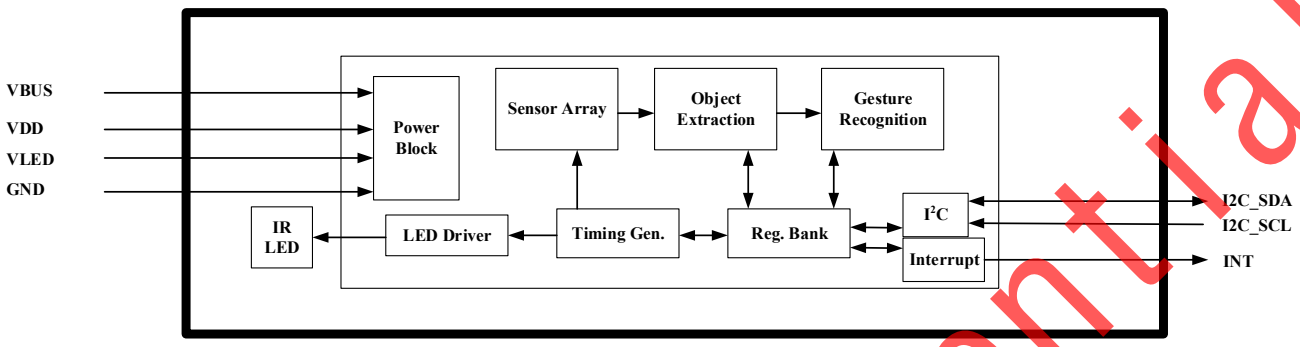


Figure 2. PAJ7620U2 Module Functional Block Diagram

Register Map and Function

Register Map:

Slave ID: 0x73 hex using 7 bit addressing protocol. Contact PixArt for other slave ID requirement.

Register Bank 0 (Switch to Register Bank 0 by setting Addr 0xEF to 00)

Address	Register Function	Access	Default
0x03	I²C suspend command (Write 0x01 to enter suspend state). I²C wake-up command is slave ID wake-up. Refer to topic "I²C Bus Timing Characteristics and Protocol"	W	0x01
0x41	Gesture detection interrupt flag mask	R/W	0xFF
0x42	Gesture/PS detection interrupt flag mask	R/W	0xFF
0x43	Gesture detection interrupt flag	R	-
0x44	Gesture/PS detection interrupt flag	R	-
0x45	State indicator for gesture detection (Only functional at gesture detection mode)	R	-
0x69	PS hysteresis high threshold (Only functional at proximity detection mode)	R/W	0xC8
0x6A	PS hysteresis low threshold (Only functional at proximity detection mode)	R/W	0x40
0x6B	PS approach state, Approach = 1 , (8 bits PS data >= PS high threshold) Not Approach = 0 , (8 bits PS data <= PS low threshold) (Only functional at proximity detection mode)	R	-
0x6C	PS 8 bit data (Only functional at proximity detection mode)	R	-
0xB0	Object Brightness (Max. 255)	R	
0xB1 0xB2	Object Size (Max. 900)	R	

Register Bank 1 (Switch to Register Bank 1 by setting Addr 0xEF to 01)

Address	Register Function	Access	Default
0x44	PS gain setting (Only functional at proximity detection mode)	R/W	0xA0
0x67 0x68	IDLE S1 Step, for setting the $S_{1,\ Response\ Factor}$	R/W	0x68 0x01
0x69 0x6A	IDLE S2 Step, for setting the $S_{2,\ Response\ Factor}$	R/W	0xD0 0x02
0x6B 0x6C	OPtoS1 Step, for setting the OPtoS1 time of operation state to standby 1 state	R/W	0xB0 0x04
0x6D 0x6E	S1toS2 Step, for setting the S1toS2 time of standby 1 state to standby 2 state	R/W	0x60 0x09
0x72	Enable/Disable PAJ7620U2	R/W	0x00

Register Function Description:

1. General Purpose:

Register Bank0, ADDR 0x03

Register Bank 0, ADDR 0x03, I²C Suspend Command		
NAME	**Reserved**	**Suspend**
BIT #	[7:1]	[0]
ACCESS	Write as 0	W
DEFAULT	0	1
	0x01	

NAME	FUNCTION/OPERATION
Suspend	Write 1: Enter suspend state (wake up by writing I2C slave ID (default: 0x73). Refer to topic "I²C Bus Timing Characteristics and Protocol"

Register Bank1, ADDR 0x72

Register Bank 1, ADDR 0x72, Enable/Disable PAJ7620U2		
NAME	**Reserved**	**Enable**
BIT #	[7:1]	[0]
ACCESS	Write as 0	R/W
DEFAULT	0	0
	0x00	

NAME	FUNCTION/OPERATION
Enable	1: Enable PAJ7620U2 0: Disable PAJ7620U2

To enter the suspend state, first disable the PAJ7620U2 by writing Register Bank 1, ADDR 0x72 with 0x00 then process the I²C suspend command by writing Register Bank 0, ADDR 0x03 with 0x01.

To exit the suspend state, first process the I²C wake-up command by writing the slave ID (Refer to topic "I²C Bus Timing Characteristics and Protocol") then enable the PAJ7620U2 by writing Register Bank 1, ADDR 0x72 with 0x01.

2. Gesture Detection Related:

Register Bank0, ADDR 0x41

	Register Bank 0, ADDR 0x41, Gesture Detection Interrupt Flag Mask							
NAME	**Counter-Clockwise Mask**	**Clockwise Mask**	**Backward Mask**	**Forward Mask**	**Right Mask**	**Left Mask**	**Down Mask**	**Up Mask**
BIT #	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	1	1	1	1	1	1	1	1
	0xFF							

NAME	FUNCTION/OPERATION
Counter Clockwise Mask	1: Counter clockwise gesture will generate an interrupt 0: Counter clockwise gesture will not generate an interrupt
Clockwise Mask	1: Clockwise gesture will generate an interrupt 0: Clockwise gesture will not generate an interrupt
Backward Mask	1: Backward gesture will generate an interrupt 0: Backward gesture will not generate an interrupt
Forward Mask	1: Forward gesture will generate an interrupt 0: Forward gesture will not generate an interrupt
Left Mask	1: Left gesture will generate an interrupt 0: Left gesture will not generate an interrupt
Right Mask	1: Right gesture will generate an interrupt 0: Right gesture will not generate an interrupt
Down Mask	1: Down gesture will generate an interrupt 0: Down gesture will not generate an interrupt
Up Mask	1: Up gesture will generate an interrupt 0: Up gesture will not generate an interrupt

Register Bank0, ADDR 0x42

	Register Bank 0, ADDR 0x42, Gesture Detection Interrupt Flag Mask	
NAME	**Reserved**	**Wave Mask**
BIT #	[7:1]	[0]
ACCESS	Write as 0000000	R/W
DEFAULT	1111111	1
	0xFF	

NAME	FUNCTION/OPERATION
Wave Mask	1: Wave gesture will generate an interrupt 0: Wave gesture will not generate an interrupt

Register Bank0, ADDR 0x43

	Register Bank 0, ADDR 0x43, Gesture Detection Interrupt Flag							
NAME	**Counter Clockwise**	**Clockwise**	**Backward**	**Forward**	**Right**	**Left**	**Down**	**Up**
BIT #	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
ACCESS	R	R	R	R	R	R	R	R
DEFAULT	-	-	-	-	-	-	-	-
	-							

NAME	FUNCTION/OPERATION
Counter Clockwise	1: Counter clockwise gesture be detected 0: No Counter clockwise gesture be detected
Clockwise	1: Clockwise gesture be detected 0: No Clockwise gesture be detected
Backward	1: Backward gesture be detected 0: No Backward gesture be detected
Forward	1: Forward gesture be detected 0: No Forward gesture be detected
Right	1: Right gesture be detected 0: No Right gesture be detected
Left	1: Left gesture be detected 0: No Left gesture be detected
Down	1: Down gesture be detected 0: No Down gesture be detected
Up	1: Up gesture be detected 0: No Up gesture be detected

Register Bank0, ADDR 0x44

	Register Bank 0, ADDR 0x44, Gesture Detection Interrupt Flag	
NAME	**Reserved**	**Wave**
BIT #	[7:1]	[0]
ACCESS	R	R
DEFAULT	-	-
	-	

NAME	FUNCTION/OPERATION
Wave	1: Wave gesture be detected 0: Wave gesture be detected

The gesture detection interrupt flag will be cleared by reading Register Bank 0, Addr 0x43 and 0x44.

Register Bank0, ADDR 0x45

Register Bank 0, ADDR 0x45, State Indicator for Gesture Detection		
NAME	**Reserved**	**State**
BIT #	[7:2]	[1:0]
ACCESS	R	R
DEFAULT	-	-
	-	

NAME	FUNCTION/OPERATION
State	0: Operation State 1: Standby 1 State 2: Standby 2 State

Refer to the “Gesture Detection Operating State and State Machine” in Operating Principle paragraph.

Register Bank1, ADDR 0x67, 0x68

Register Bank 1, ADDR 0x67, ADDR 0x68, IDLE S1 Step, for setting the $S_{1,\ Response\ Factor}$		
ADDRESS	ADDR 0x68	ADDR 0x67
NAME	IDLE S1 Step [15:8]	IDLE S1 Step [7:0]
BIT #	[7:0]	[7:0]
ACCESS	R/W	R/W
DEFAULT	0x01	0x68

NAME	FUNCTION/OPERATION
IDLE S1 Step	Normal Mode: $S_{1,\ Response\ Factor} = 8.333/(0.0323 \times IDLE\ S1Step + \alpha)$ Gaming Mode: $S_{1,\ Response\ Factor} = 4.167/(0.0323 \times IDLE\ S1\ Step + \alpha)$, $\alpha = 3.55$ for Near Mode $S_{1,\ Response\ Factor}$ represents the degrading factor of detection rate in Standby 1 State referenced to Normal Mode or Gaming Mode. Therefore, the object detecting rate in Standby1 State equals $S_{1,\ Response\ Factor}$ multiplied by the gesture update rate of Normal Mode or Gaming Mode in Operation State.

Register Bank1, ADDR 0x69, 0x6A

Register Bank 1, ADDR 0x69, ADDR 0x6A, IDLE S2 Step, for setting the $S_{2,\ Response\ Factor}$		
ADDRESS	ADDR 0x6A	ADDR 0x69
NAME	IDLE S2 Step [15:8]	IDLE S2 Step [7:0]
BIT #	[7:0]	[7:0]
ACCESS	R/W	R/W
DEFAULT	0x02	0xD0

NAME	FUNCTION/OPERATION
IDLE S2 Step	Normal Mode: $S_{2,\ Response\ Factor} = 8.333/(0.0645 \times IDLE\ S2Step + \alpha)$ Gaming Mode: $S_{2,\ Response\ Factor} = 4.167/(0.0645 \times IDLE\ S2\ Step + \alpha)$, $\alpha = 3.55$ for Near Mode $S_{2,\ Response\ Factor}$ represents the degrading factor of detecting rate in Standby 2 State referenced to Normal Mode or Gaming Mode. Therefore, the object detecting rate in Standby2 State equals $S_{2,\ Response\ Factor}$ multiplied by the gesture update rate of Normal Mode or Gaming Mode in Operation State.

Register Bank1, ADDR 0x6B, 0x6C

Register Bank 1, ADDR 0x6B, ADDR 0x6C, OPtoS1 Step, for setting the OPtoS1 time of operation state to standby 1 state		
ADDRESS	ADDR 0x6C	ADDR 0x6B
NAME	OPtoS1 Step [15:8]	OPtoS1 Step [7:0]
BIT #	[7:0]	[7:0]
ACCESS	R/W	R/W
DEFAULT	0x04	0xB0

NAME	FUNCTION/OPERATION
OPtoS1 Step	Normal Mode: OPtoS1 time = OPtoS1 step/120 Gaming Mode: OPtoS1 time = OPtoS1 step/240 The OPtoS1 time means the time that no object be detected from Operation State to Standby 1 State

Register Bank1, ADDR 0x6D, 0x6E

Register Bank 1, ADDR 0x6D, ADDR 0x6E, S1toS2 Step, for setting the S1toS2 time of standby 1 state to standby 2 state		
ADDRESS	ADDR 0x6E	ADDR 0x6D
NAME	S1toS2 Step [15:8]	S1toS2 Step [7:0]
BIT #	[7:0]	[7:0]
ACCESS	R/W	R/W
DEFAULT	0x09	0x60

NAME	FUNCTION/OPERATION
S1toS2 Step	Normal Mode: S1toS2 time = S1toS2 step/(60 $xS_{1,\ Response\ Factor}$) Gaming Mode: S1toS2 time = S1toS2 step/(120 $xS_{1,\ Response\ Factor}$) The S1toS2 time means the time that no object be detected from Standby 1 State to Standby 2 State

Register Bank0, ADDR 0xB0

Register Bank 0, ADDR 0xB0, Object Brightness, Report object brightness	
ADDRESS	ADDR 0xB0
NAME	ObjectAvgY[8:1]
BIT #	[7:0]
ACCESS	R
DEFAULT	-

NAME	FUNCTION/OPERATION
ObjectAvgY	Report Object Brightness (Max. value 255).

Register Bank0, ADDR 0xB1, 0xB2

Register Bank 0, ADDR 0xB1, ADDR 0xB2, Object Size		
ADDRESS	ADDR 0xB2	ADDR 0xB1
NAME	ObjectSize[11:8]	ObjectSize[7:0]
BIT #	[3:0]	[7:0]
ACCESS	R	R
DEFAULT		

NAME	FUNCTION/OPERATION
ObjectSize	Report Object Size (Max. value 900).

3. Proximity Detection Related:

Register Bank1, ADDR 0x44

	Register Bank 1, ADDR 0x44, Proximity gain setting		
NAME	Reserved	PS Gain	Reserved
BIT #	[7]	[6]	[5:0]
ACCESS	Write as 1	R	Write as 100000
DEFAULT	1	0	100000
	0xA0		

NAME	FUNCTION/OPERATION
PS Gain	0: 1x gain 1: 2x gain

Register Bank0, ADDR 0x42

	Register Bank 0, ADDR 0x42, Gesture Detection Interrupt Flag Mask		
NAME	Reserved	Proximity Mask	Reserved
BIT #	[7:2]	[1]	[0]
ACCESS	Write as 000000	R/W	Write as 0
DEFAULT	111111	1	1
	0xFF		

NAME	FUNCTION/OPERATION
Proximity Mask	1: Proximity detection will generate an interrupt 0: Proximity detection will not generate an interrupt

Register Bank0, ADDR 0x44

	Register Bank 0, ADDR 0x44, Proximity Detection Interrupt Flag		
NAME	Reserved	PS Int. Flag	Reserved
BIT #	[7:2]	[1]	[0]
ACCESS	R	R	R
DEFAULT	-	-	-
	-		

NAME	FUNCTION/OPERATION
PS Int. Flag	1: PS interrupt flag 0: No PS interrupt flag

The proximity detection interrupt flag will be cleared by reading Register Bank 0, Addr 0x44.

Register Bank0, ADDR 0x69

Register Bank 0, ADDR 0x69, PS Hysteresis High Threshold	
NAME	PS High Thd
BIT #	[7:0]
ACCESS	R/W
DEFAULT	0xC8

NAME	FUNCTION/OPERATION
PS High Thd	PS Hysteresis Interrupt Window High Threshold

Register Bank0, ADDR 0x6A

Register Bank 0, ADDR 0x6A, PS Hysteresis Low Threshold	
NAME	PS Low Thd
BIT #	[7:0]
ACCESS	R/W
DEFAULT	0x40

NAME	FUNCTION/OPERATION
PS Low Thd	PS Hysteresis Interrupt Window Low Threshold

Register Bank0, ADDR 0x6B

	Register Bank 0, ADDR 0x6B, Proximity Approach State	
NAME	**Reserved**	**PS Approach**
BIT #	[7:1]	[0]
ACCESS	R	R
DEFAULT	-	-
	-	

NAME	FUNCTION/OPERATION
PS Approach	PS approach state, Approach = 1 , (8 bits PS data >= PS high threshold) Not Approach = 0 , (8 bits PS data <= PS low threshold)

Register Bank0, ADDR 0x6C

Register Bank 0, ADDR 0x6C, Proximity Readout	
NAME	8 bits PS data [7:0]
BIT #	[7:0]
ACCESS	R
DEFAULT	-

NAME	FUNCTION/OPERATION
8 bits PS data [7:0]	PS Readout

Refer to Figure 3 for how to set the PS hysteresis window and the interrupt mechanism of proximity detection.

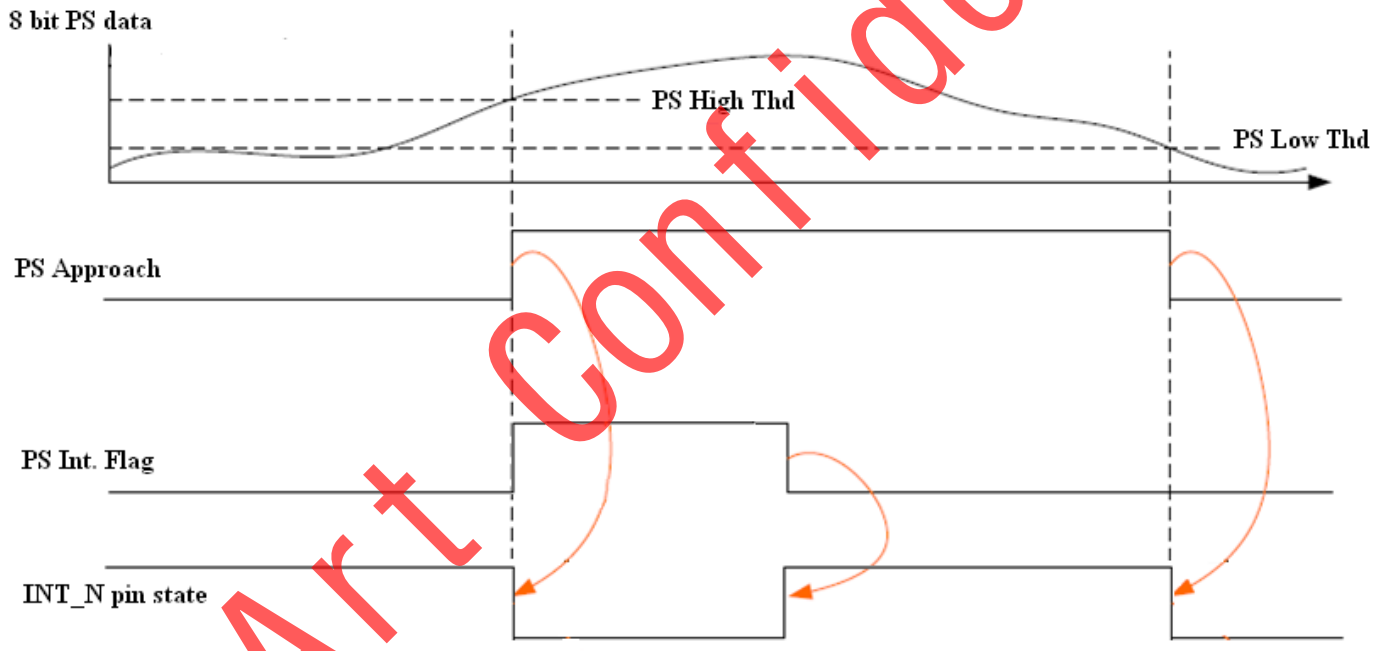


Figure 3. Proximity Sensing Functional Diagram

Operating Principle

1. Gesture Sensor Module Orientation

The PAJ7620U2 should be oriented as shown in Figure 4 for correct gesture detection.

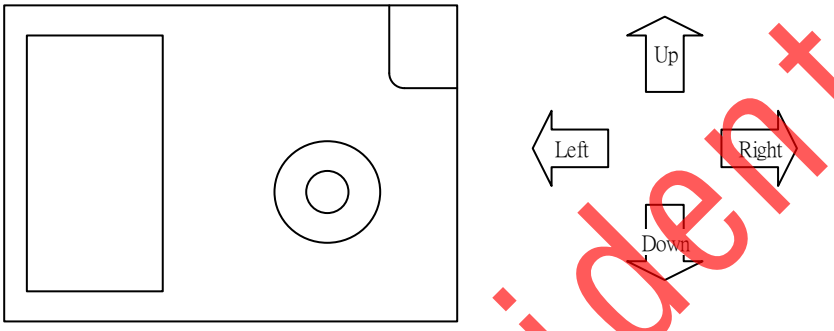


Figure 4. The Correct Gesture Sensor Module Orientation (Front View)

If rotate the gesture module, The direction of gesture detection interrupt flag mask (Register Bank 0, ADDR 0x41) and gesture detection interrupt flag (Register Bank 0, ADDR 0x43) needs to re-mapping.

Register Bank0, ADDR 0x41

	Register Bank 0, ADDR 0x41, Gesture Detection Interrupt Flag Mask							
NAME	Counter-Clockwise Mask	Clockwise Mask	Backward Mask	Forward Mask	Up Mask	Down Mask	Right Mask	Left Mask
BIT #	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]

Register Bank0, ADDR 0x43

	Register Bank 0, ADDR 0x43, Gesture Detection Interrupt Flag							
NAME	Counter Clockwise	Clockwise	Backward	Forward	Up	Down	Right	Left
BIT #	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]

2. Power-On Sequence

In the power-on sequence, The VBUS **Must** be power on before VDD. After power on, wait T_1 μs for PAJ7620U2 to stabilize and then write slave ID (0x73) to process I²C wake-up (Refer to topic "**I²C Bus Timing Characteristics and Protocol"**). After T_2 μs, write the initial settings and the different modes settings to PAJ7620U2. Lastly, enable PAJ7620U2 by writing Register Bank1, Addr0x72 with 0x01. The gesture data can now be accessed through the I²C bus.

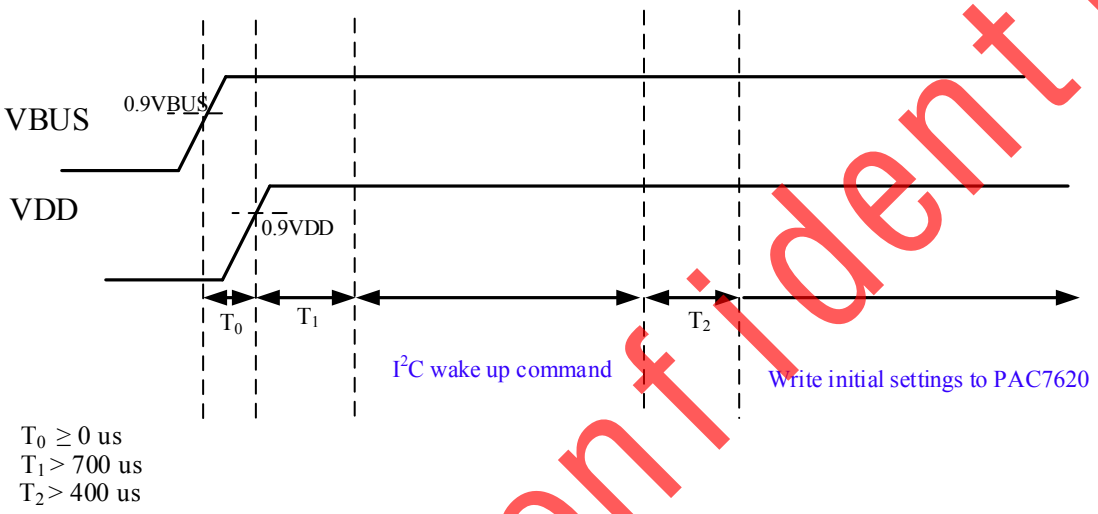


Figure 5. PAJ7620U2 Power-On Timing Diagram

3. Gesture Detection Operating State and State Machine

When in gesture detection, the state machine of PAJ7620U2 is in Figure13. Following is the detail description of each state.

i. Operation State (OP state)

When in operation state, the gesture update rate is 120Hz for Normal Mode and 240Hz for Gaming Mode respectively. The gesture result can be accessed by interrupt mechanism or continuous polling the gesture detection interrupt flag (Register Bank 0, Addr0x43).

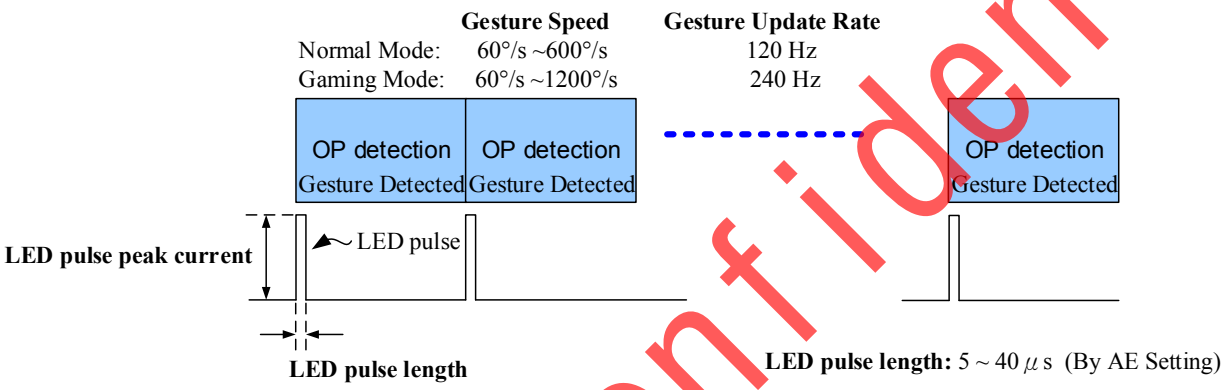


Figure 9. Operation State (OP state) Diagram

ii. Standby 1 State (S1 state)

When in Standby1 state, the object detection rate equals $S_{1,\ Response\ Factor}$ multiply the gesture update rate of Normal Mode or Gaming Mode.

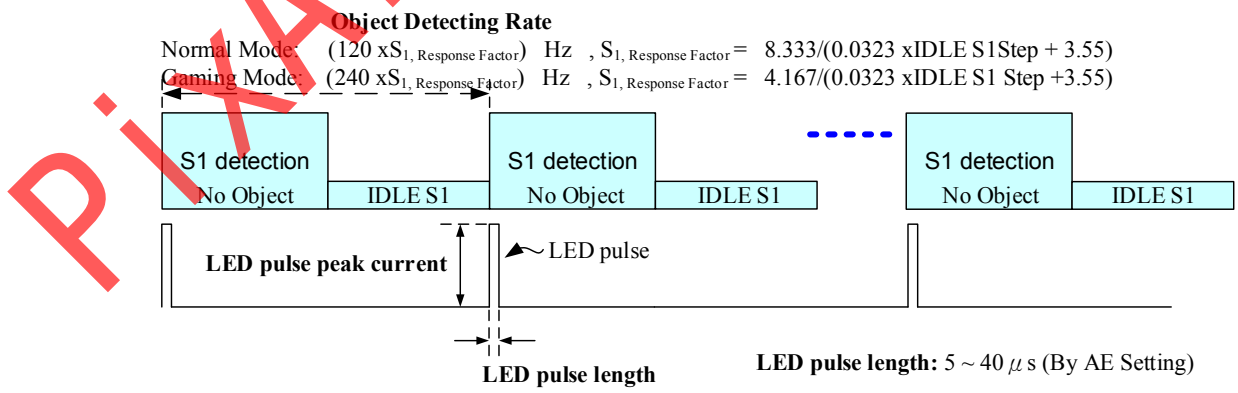


Figure 10. Standby 1 State (S1 state) Diagram

iii. Standby 2 State (S2 state)

When in Standby 2 state, the object detection rate equals $S_{2,\ Response\ Factor}$ multiply the gesture update rate of Normal Mode or Gaming Mode.

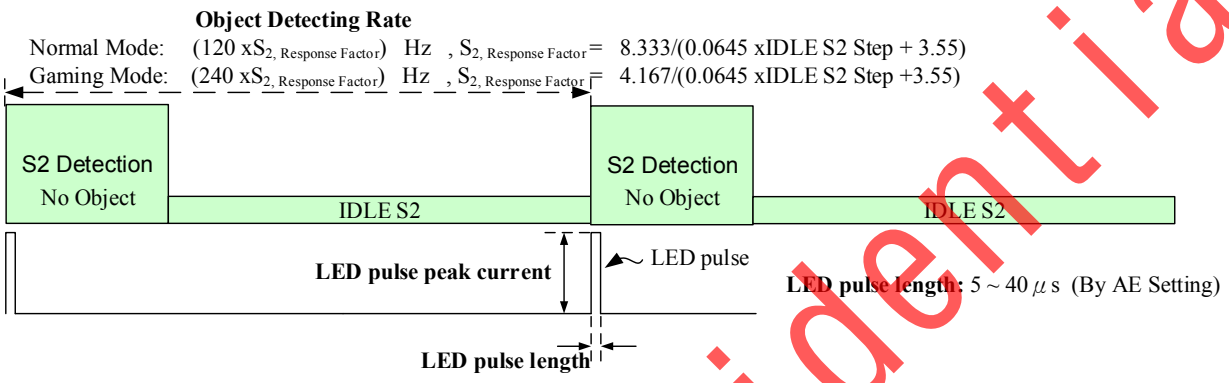


Figure 11. Standby 2 State (S2 state) Diagram

iv. Suspend State (SUS state)

To enter the suspend state, first disable the PAJ7620U2 by writing Register Bank 1, ADDR 0x72 with 0x00 then process the I²C suspend command by writing Register Bank 0, ADDR 0x03 with 0x01.

To exit the suspend state, first process the I²C wake-up command by writing the slave ID (Refer to topic "I²C Bus Timing Characteristics and Protocol") then enable the PAJ7620U2 by writing Register Bank 1, ADDR 0x72 with 0x01.

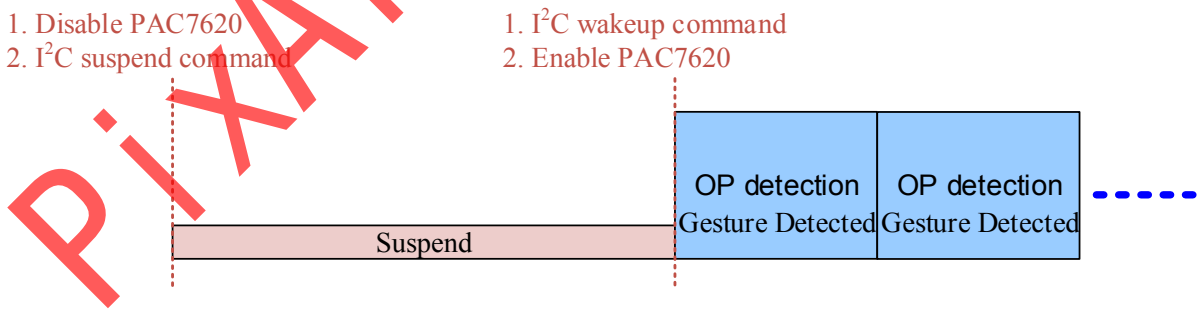
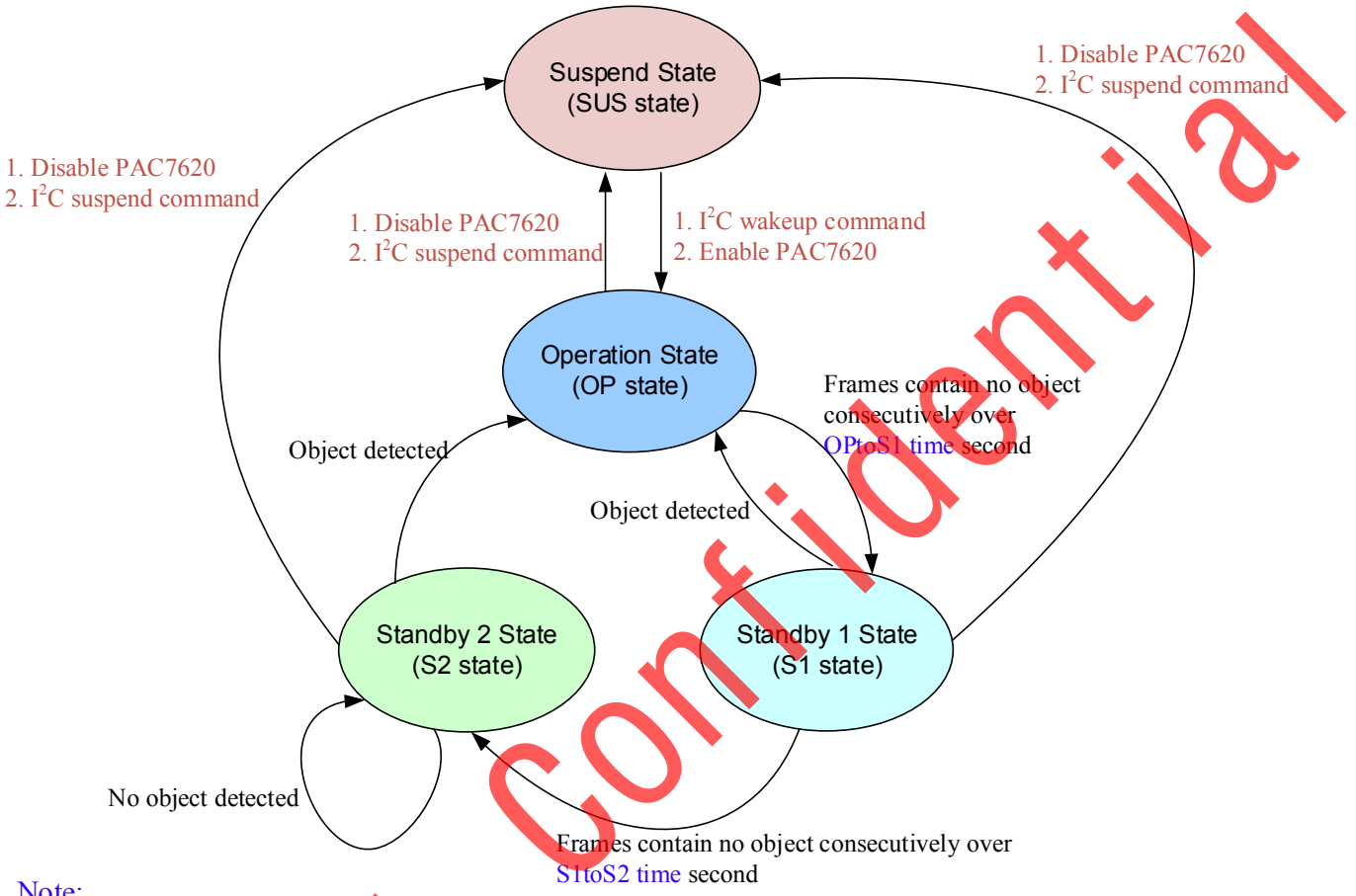


Figure 12. Suspend State (SUS state) Diagram

v. State Machine



Note:
OPtoS1 time = OPtoS1 step/120 @ Normal Mode
= OPtoS1 step/240 @ Gaming Mode

S1toS2 time = S1toS2 step/(60 $xS_{1,\ Response\ Factor}$) @ Normal Mode
= S1toS2 step/(120 $xS_{1,\ Response\ Factor}$) @ Gaming Mode

Figure 13. State Machine of Gesture Detection

4. Proximity Detection Operating Principle

When in proximity detection, the state machine of PAJ7620U2 is in Figure 16. Following is the detail description of each state.

i. Proximity Operation State (PS OP state)

When in operation state, the update rate is 10Hz and the LED on time is 8μs. The LED peak current is 760 mA.

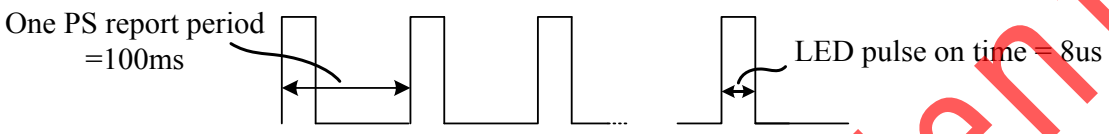


Figure 14. Proximity Operation State (PS OP state) Diagram

ii. Suspend State (SUS state)

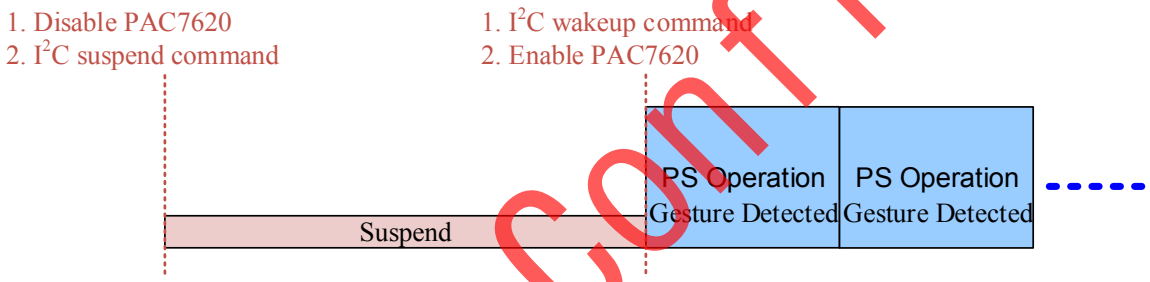


Figure 15. Suspend State (SUS state) Diagram

iii. State Machine

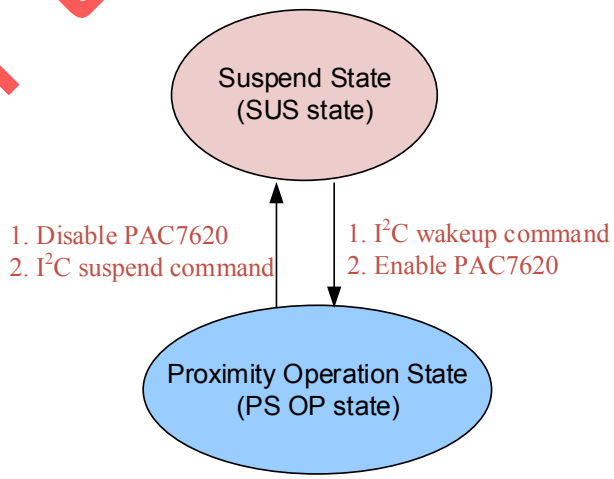


Figure 16. State Machine of Gesture Detection

I²C Bus Timing Characteristics and Protocol

i. I²C Timing Parameter

Parameter	Symbol	STANDARD MODE		FAST MODE		Unit
		Min.	Max.	Min.	Max.	
SCL clock frequency.	f_{scl}	10	100	10	400	kHz
Hold time for Start/Repeat Start. After this period, the first clock pulse is generated.	$t_{HD.STA}$	4		0.6		μs
Set-up time for a repeated Start.	$t_{SU.STA}$	4.7		0.6		μs
Low period of SCL clock.	t_{LOW}	4.7		1.3		μs
High period of SCL clock.	t_{HIGH}	4		0.6		μs
Data hold time.	$t_{HD.DAT}$	0		0		μs
Data set-up time.	$t_{SU.DAT}$	250		100		ns
Rise time of both SDA and SCL signals.	t_r		1000	-	300	ns
Fall time of both SDA and SCL signals.	t_f		300	-	300	ns
Set-up time for STOP condition.	$t_{SU.STO}$	4		0.6		μs
Bus free time between a STOP and START.	t_{BUF}	4.7		1.3		μs

** maximum current is 5mA and capacitance load spec. =100pF*

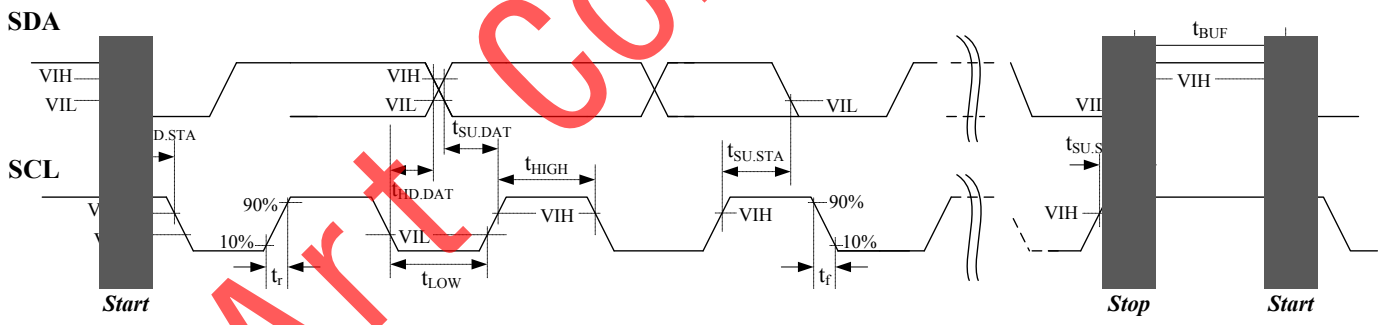


Figure 17. I²C Timing Diagram

ii. I²C General Description

- SDA (serial data) and SCL (serial clock) form a two-wire serial interface compatible with I²C. The PAJ7620U2 is implemented as a slave-only device so it never drives SCL. It drives SDA during (host) read cycles and transmission of the Acknowledge bit. PAJ7620U2 uses 7-bit addressing and does not support clock stretching. The SDA and SCL pins are open-drain structure requiring external pull-up resistors.

- Start and stop condition: SDA high to low transition while SCL is high defines a Start condition. SDA low to high transition while SCL is high defines a Stop condition. (Refer. to Figure 18)

- Valid data: The data on SDA line must be stable during high period of SCL. MSB is always transferred first for each byte. LSB of the first byte is Read / Write control bit. (Refer. to Figure 19)
- Both master and slave can transmit and receive data from the bus.
- Acknowledge: The Receiving device should pull down SDA during high period of the 9th clock (SCL) after a complete byte has been received from the transmitter. In the case of the master receiving data from the slave, the master does not generate an Acknowledge bit after the last byte to indicate the end of a master read cycle.

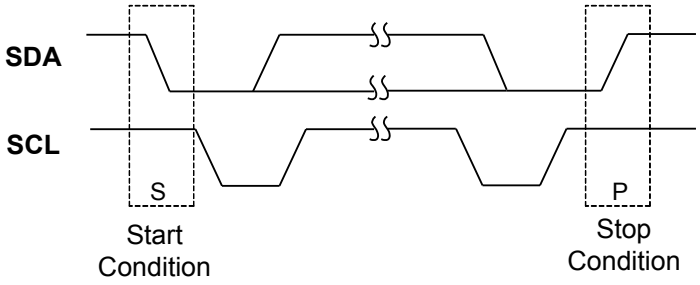


Figure 18. Start and Stop Conditions

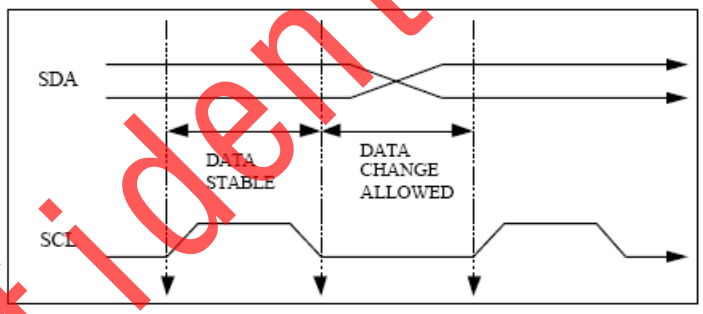
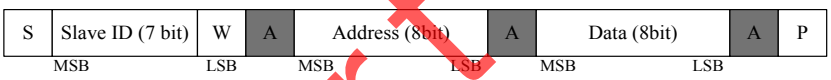


Figure 19. Valid Data

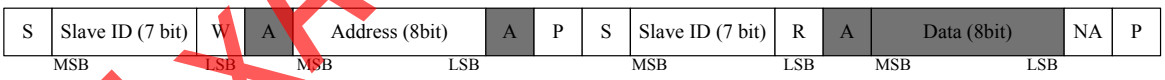
iii. I²C Protocol

The slave ID of PAJ7620U2 is ***0x73*** hex using 7 bit addressing protocol. Contact Pixart for other slave ID requirement.

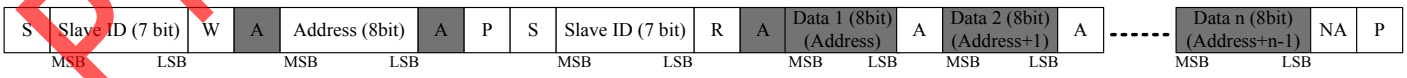
I. Single Write Protocol



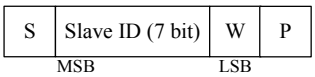
II. Single Read Protocol



III. Burst Read Protocol

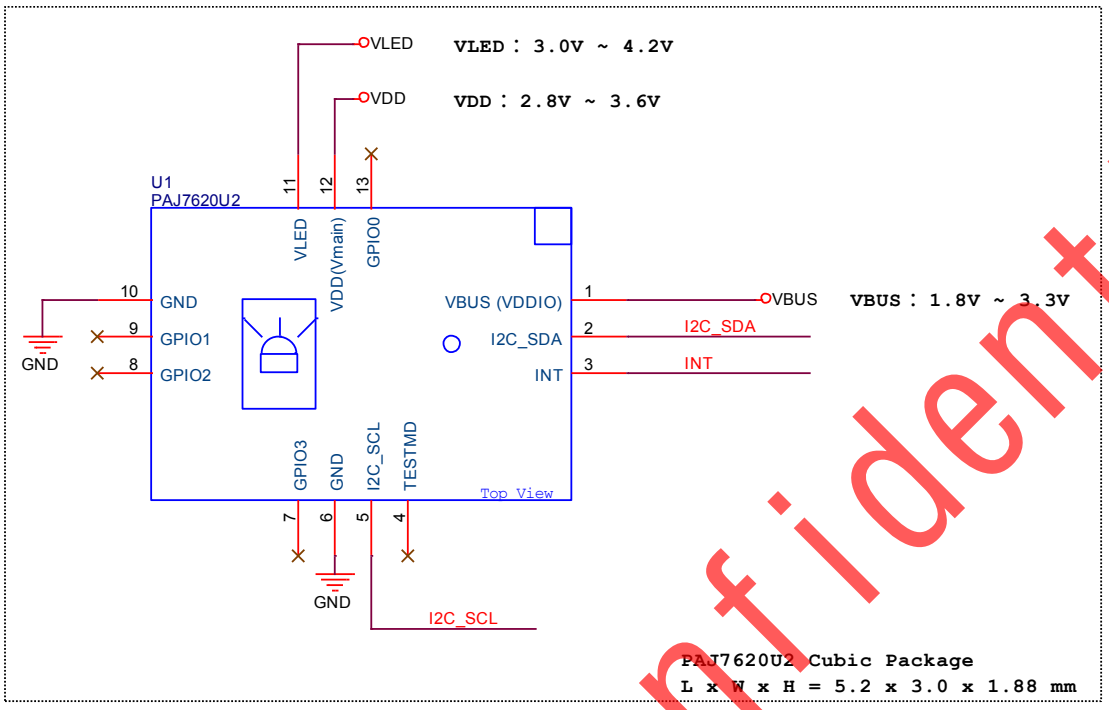


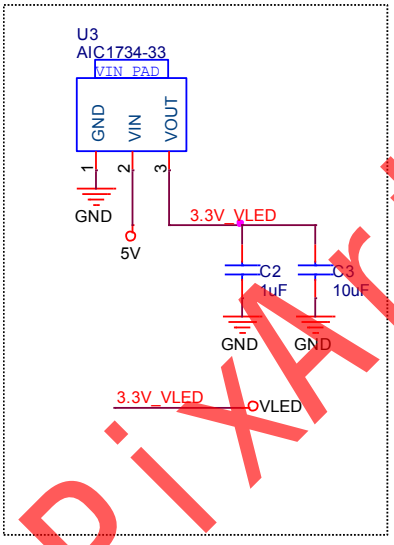
IV. I²C Wake-up command Protocol

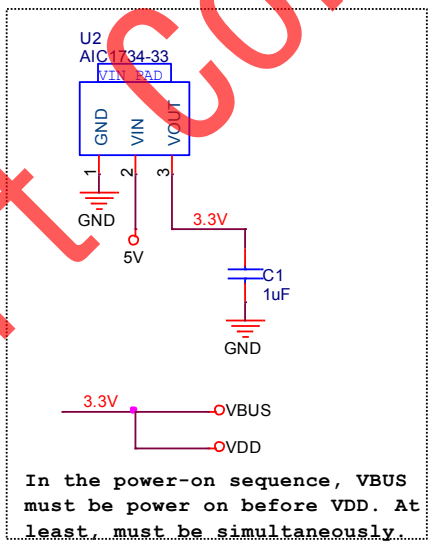


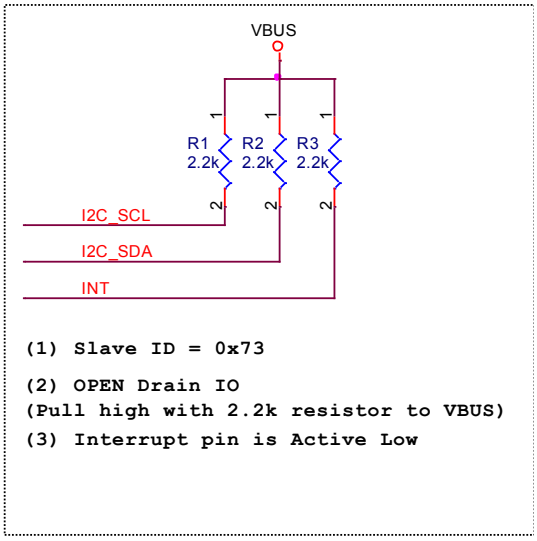
A Acknowledge (0 for an ACK)
NA Non-acknowledge (1 for an NACK)
P Stop condition
Master to Slave
S Start condition
W Write (0 for write)
R Read (1 for read)
Slave to Master

Reference Application Circuit

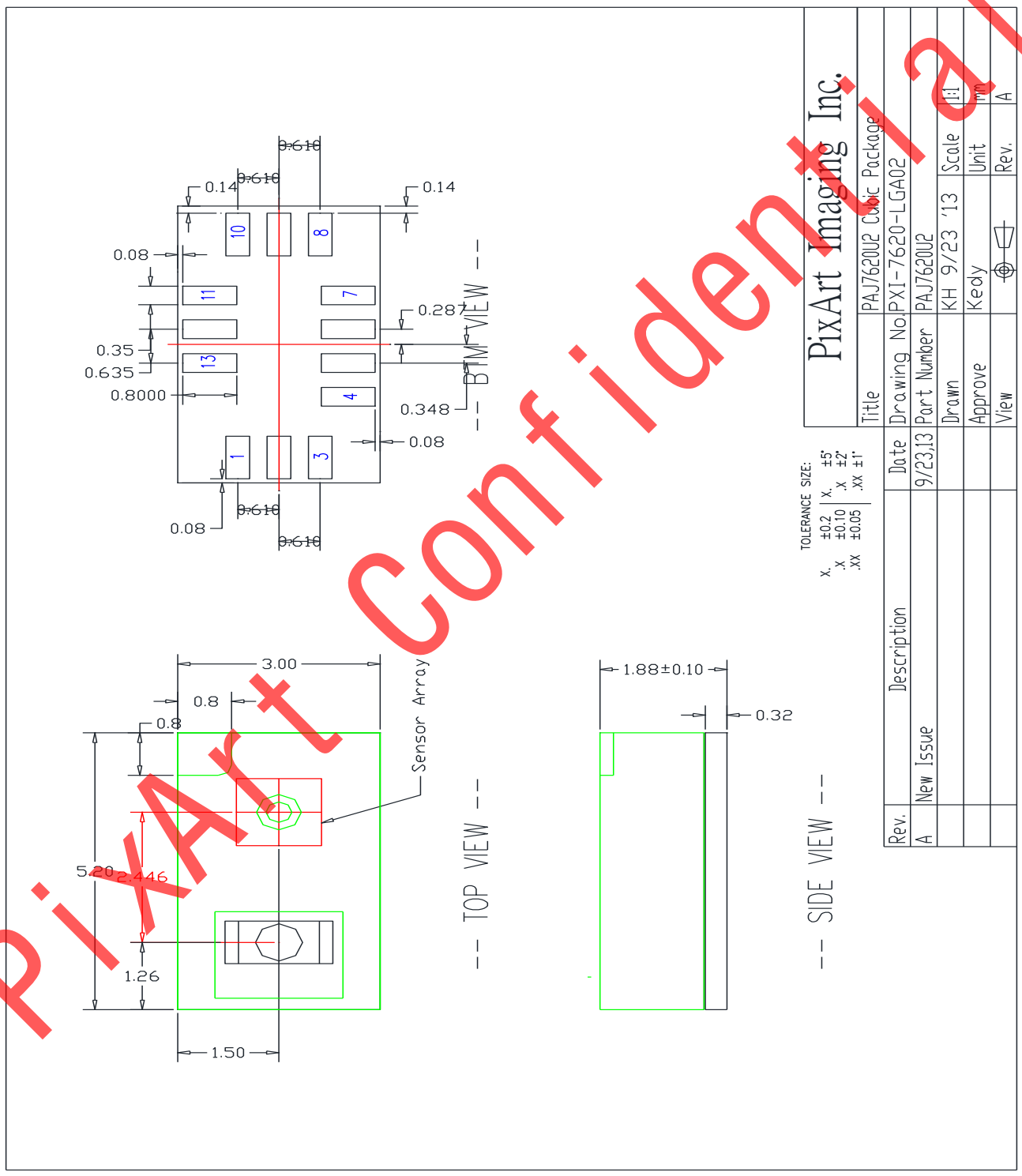




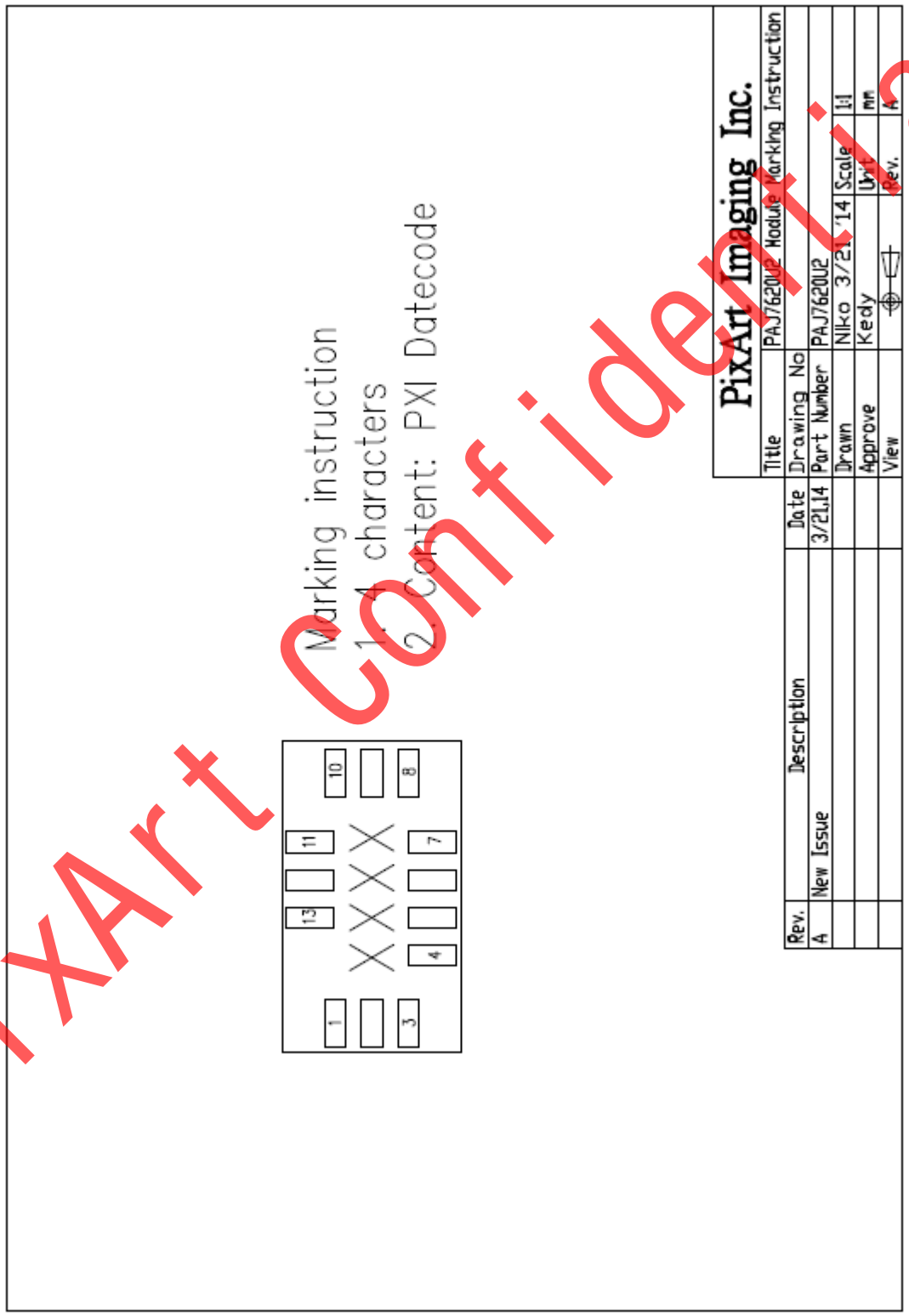




Outline Dimension



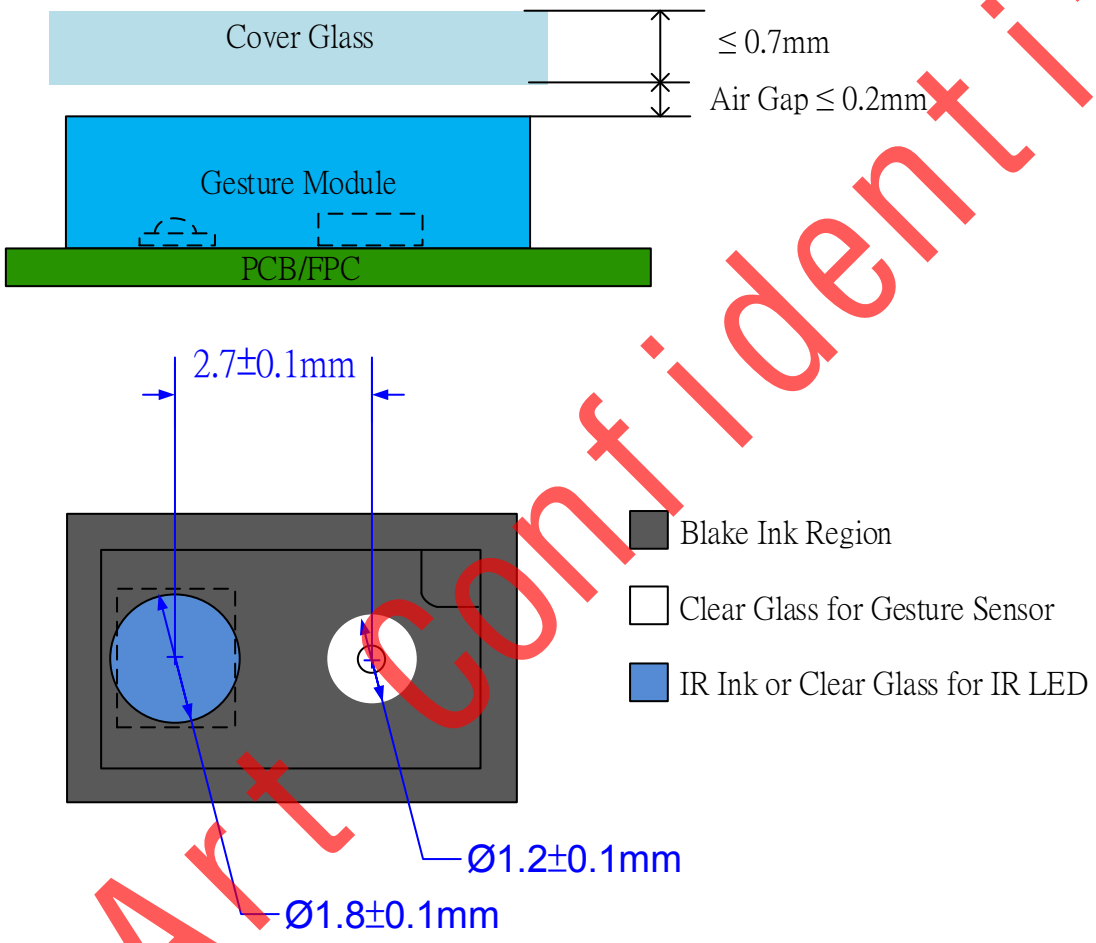
Marking Information



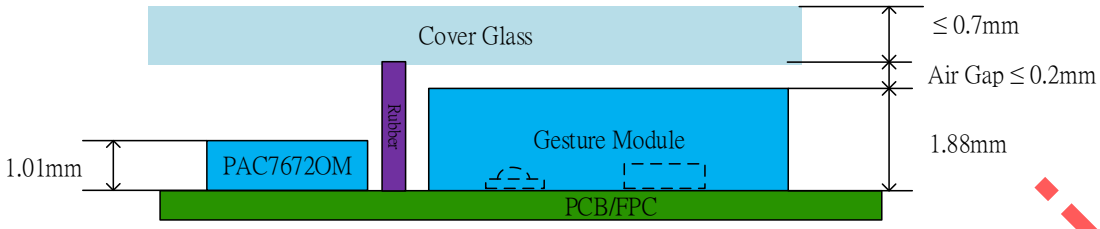
Mechanical Design Guide

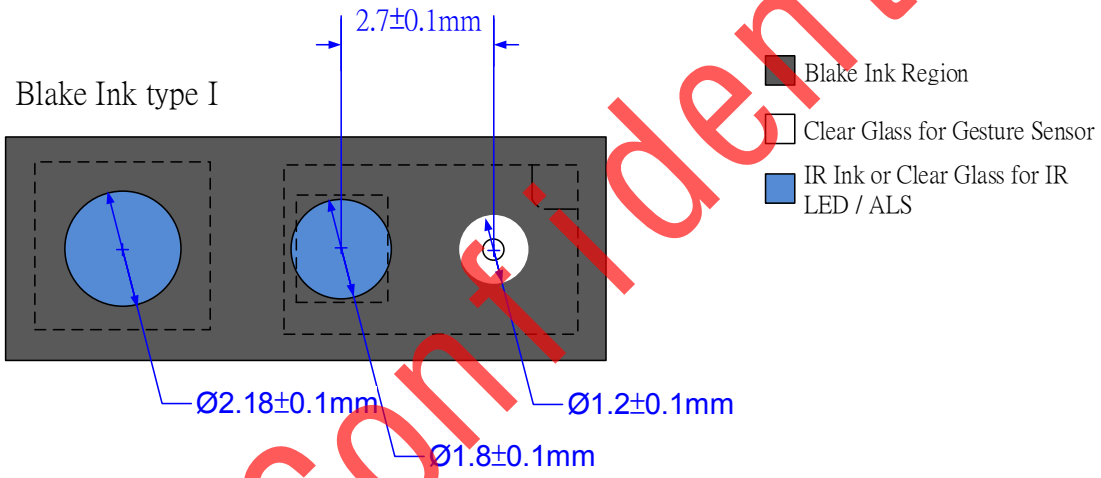
i. Gesture Module Only

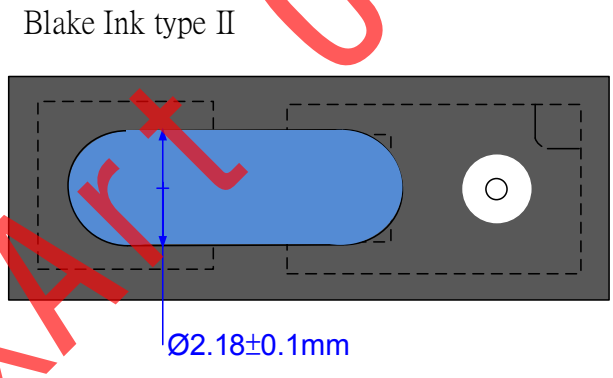
PXI suggest mechanical design as below for better performance.



ii. Gesture Module with ALS sensor (PAC7672OM)

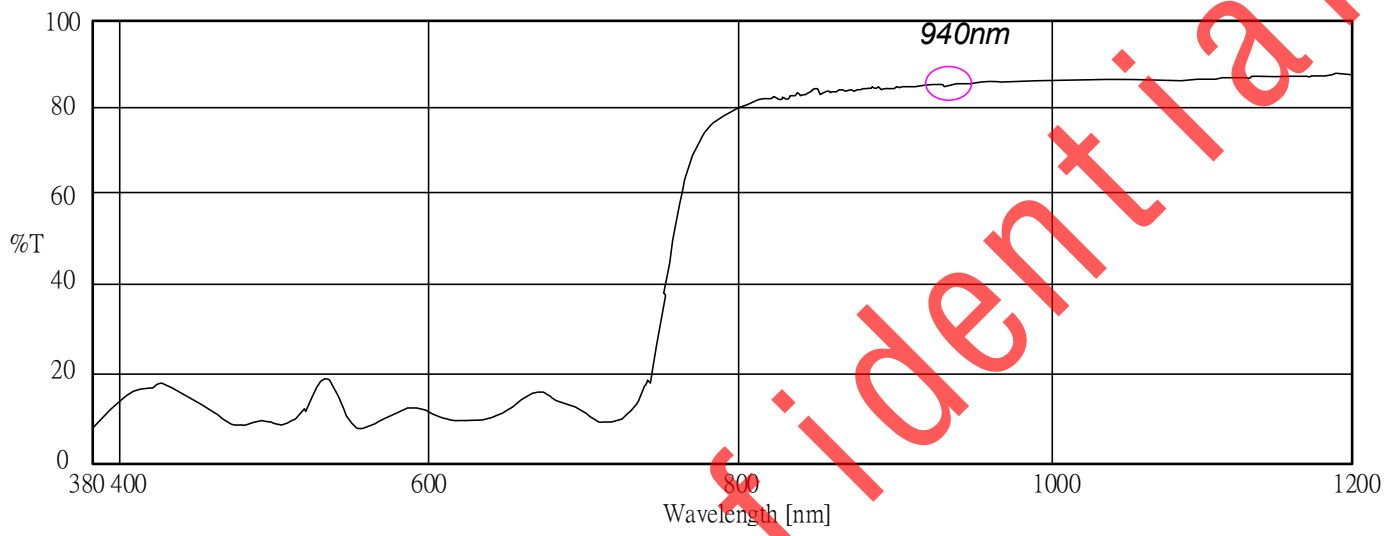






IR Ink Spectrum (Recommend)

IR Ink Vendor : Teikoku Ink



Programing Sequence and Function Application

This chapter describes how to implement firmware for PAJ7620U2 and function application.

i. Initial

Step 1：Power On, The V_{Bus} must be power on before V_{DD}.

Step 2：Wait 700us for PAJ7620U2 to stabilize.

Step 3：Write slave ID or I2C read command to process I²C wake-up.
It's recommend to read Reg_0x00. It will return "0x20" when wake-up finish.
By the way, There is no-ack from PAJ7620U2 before wake-up finish.

Step 4：Write initial setting to gesture.

```
unsigned char initial_register_array[][2] = {
    {0xEF,0x00},
    {0x37,0x07},
    {0x38,0x17},
    {0x39,0x06},
    {0x42,0x01},
    {0x46,0x2D},
    {0x47,0x0F},
    {0x48,0x3C},
    {0x49,0x00},
    {0x4A,0x1E},
    {0x4C,0x20},
    {0x51,0x10},
    {0x5E,0x10},
    {0x60,0x27},
    {0x80,0x42},
    {0x81,0x44},
    {0x82,0x04},
    {0x8B,0x01},
    {0x90,0x06},
    {0x95,0x0A},
    {0x96,0x0C},
    {0x97,0x05},
    {0x9A,0x14},
```

```
        {0x9C,0x3F},
        {0xA5,0x19},
        {0xCC,0x19},
        {0xCD,0x0B},
        {0xCE,0x13},
        {0xCF,0x64},
        {0xD0,0x21},
        {0xEF,0x01},
        {0x02,0x0F},
        {0x03,0x10},
        {0x04,0x02},
        {0x25,0x01},
        {0x27,0x39},
        {0x28,0x7F},
        {0x29,0x08},
        {0x3E,0xFF},
        {0x5E,0x3D},
        {0x65,0x96},
        {0x67,0x97},
        {0x69,0xCD},
        {0x6A,0x01},
        {0x6D,0x2C},
        {0x6E,0x01},
        {0x72,0x01},
        {0x73,0x35},
        {0x74,0x00},
        {0x77,0x01},
    };
```

ii. Get Gesture result

Step 1：Set Interrupt or I²C polling timer.

Step 2：Read Bank_0_Reg_0x43/0x44 for gesture result if interrupt or timer happen. Gesture result will be clean when I²C read finish.

iii. Change to PS mode

Step 1：Write PS mode setting to gesture.

```
unsigned char change_to_proximity_register_array[][2] = {
    {0xEF,0x00},
    {0x41,0x00},
    {0x42,0x02},
    {0x48,0x20},
    {0x49,0x00},
    {0x51,0x13},
    {0x83,0x00},
    {0x9F,0xF8},
    {0x69,0x96},
    {0x6A,0x02},
    {0xEF,0x01},
    {0x01,0x1E},
    {0x02,0x0F},
    {0x03,0x10},
    {0x04,0x02},
    {0x41,0x50},
    {0x43,0x34},
    {0x65,0xCE},
    {0x66,0x0B},
    {0x67,0xCE},
    {0x68,0x0B},
    {0x69,0xE9},
    {0x6A,0x05},
    {0x6B,0x50},
    {0x6C,0xC3},
    {0x6D,0x50},
    {0x6E,0xC3},
    {0x74,0x05},
};
```

iv. Get PS approach status

Step 1：Read Bank_0_Reg_0x6B for PS approach status or read Bank_0_Reg_0x6C for PS raw data.

v. Change to Gesture mode

Step 1：Write Gesture mode setting to gesture.

```
unsigned char change_to_gesture_register_array[][2] = {
    {0xEF,0x00},
    {0x41,0x00},
    {0x42,0x00},
    {0xEF,0x00},
    {0x48,0x3C},
    {0x49,0x00},
    {0x51,0x10},
    {0x83,0x20},
    {0x9f,0xf9},
    {0xEF,0x01},
    {0x01,0x1E},
    {0x02,0x0F},
    {0x03,0x10},
    {0x04,0x02},
    {0x41,0x40},
    {0x43,0x30},
    {0x65,0x96},
    {0x66,0x00},
    {0x67,0x97},
    {0x68,0x01},
    {0x69,0xCD},
    {0x6A,0x01},
    {0x6b,0xb0},
    {0x6c,0x04},
    {0x6D,0x2C},
    {0x6E,0x01},
    {0x74,0x00},
    {0xEF,0x00},
    {0x41,0xFF},
    {0x42,0x01},
};
```

vi. Enter Suspend mode

Step 1：Write Suspend mode setting to gesture.

```
unsigned char suspend_register_array[][2] = {
    {0xEF,0x01},
    {0x72,0x00},
    {0xEF,0x00},
    {0x03,0x01},
};
```

vii. Resume Gesture

Step 1：Write slave ID or I2C read command to process I²C wake-up.

It's recommend to read Reg_0x00. It will return "0x20" when wake-up finish.

By the way, There is no-ack from PAJ7620U2 before wake-up finish.

Step 2：Write Resume setting to gesture.

```
unsigned char resume_register_array[][2] = {
    {0xEF,0x01},
    {0x72,0x01},
};
```

Register Map

Bank	Address		Register Name	Bits	Access	Notes
	Hex	Dec				
0	3	3	SW_Suspend_EnL	[0]	W	Write 1: Enter suspend state (wake up by writing I2C slave ID (default: 0x73), Refer to topic "I2C Bus Timing Characteristics and Protocol"
0	32	50	R_CursorUseTop	[0]	R/W	1: Enable cursor center function
0	32	50	R_CursorInvertY	[2]	R/W	1: Horizontal cursor center inverse
0	32	50	R_CursorInvertX	[3]	R/W	1: Vertical cursor center inverse
0	37	55	R_CursorClampLeft[4:0]	[4:0]	R/W	Minimum cursor horizontal center value, if the data is less than this, clamp at 0. Otherwise, clamp at data-(R_CursorClampLeft<<R_PositionResolution).
0	38	56	R_CursorClampRight[4:0]	[4:0]	R/W	Maximum cursor horizontal center value, if the data is larger than this, clamp at this -1
0	39	57	R_CursorClampUp[4:0]	[4:0]	R/W	Minimum cursor vertical center value, if the data is less than this, clamp at 0 Otherwise, clamp at data-(R_CursorClampUp<<R_PositionResolution).
0	3A	58	R_CursorClampDown[4:0]	[4:0]	R/W	Maximum cursor vertical center value, if the data is larger than this, clamp at this -1
0	3B	59	CursorClampCenterX[7:0]	[7:0]	R	Clamping Center X
0	3C	60	CursorClampCenterX[11:8]	[3:0]	R	
0	3D	61	CursorClampCenterY[7:0]	[7:0]	R	Clamping Center Y
0	3E	62	CursorClampCenterY[11:8]	[3:0]	R	
0	40	64	R_MCU_IntFlagGClr	[1]	R/W	1: IntFlag1 and IntFlag2 clear enable
0	40	64	R_MCU_IntFlagInv	[4]	R/W	INT Pin Polarity: 1:1 stands for the interrupt event, 0:0 stands for the interrupt event
0	41	65	R_Int_1_En[7:0]	[7:0]	R/W	If the corresponding bit is 1: the corresponding interrupt event enable
0	42	66	R_Int_2_En[7:0]	[7:0]	R/W	If the corresponding bit is 1: the corresponding interrupt event enable
0	43	67	IntFlag_1[7:0]	[7:0]	R	When interrupt event happens, the corresponding bit is set to 1 bit 0: Up bit1: Down bit2: Left bit3: Right bit4: Forward bit5: Backward bit6: Clockwise bit7:Counterclockwise
0	44	68	IntFlag_2[7:0]	[7:0]	R	When interrupt event happens, the corresponding bit is set to 1 bit 0: Wave, wave mode use only bit1: Proximity, proximity mode use only bit2: Has Object, cursor mode use only bit3: Wake up trigger, trigger mode use only bit4: Confirm, confirm mode use only bit5: Abort, confirm mode use only bit6: N/A bit7:No Object, cursor mode use only
0	45	69	SleepMode_Status[1:0]	[1:0]	R	Indicate operation status, 0:normal, 1:weak sleep, 2:deep sleep
0	46	70	R_AELedOff_UB[7:0]	[7:0]	R/W	If OFF Frame average brightness > this x2, AE decrease
0	47	71	R_AELedOff_LB[7:0]	[7:0]	R/W	If OFF Frame average brightness < this x2, AE increase
0	48	72	R_AE_Exposure_UB[7:0]	[7:0]	R/W	Exposure up bound
0	49	73	R_AE_Exposure_UB[15:8]	[7:0]	R/W	
0	4A	74	R_AE_Exposure_LB[7:0]	[7:0]	R/W	Exposure low bound,

0	4B	75	R_AE_Exposure_LB[15:8]	[7:0]	R/W	
0	4C	76	R_AE_Gain_UB[7:0]	[7:0]	R/W	Gain stage up bound
0	4D	77	R_AE_Gain_LB[7:0]	[7:0]	R/W	Gain stage low bound
0	51	81	R_Manual_GG	[0]	R/W	Gain Manual Mode
0	51	81	R_Manual_Exposure	[1]	R/W	Exp Manual Mode
0	51	81	R_Manual_Exposure_Default	[2]	R/W	Exp Manual Mode
0	51	81	R_AE_EnH	[4]	R/W	1: AE Enable
0	5E	94	R_TG___CLK_manual	[0]	R/W	Set corresponding CLK in Manual Mode
0	5E	94	R_DMSP_CLK_manual	[1]	R/W	
0	5E	94	R_SEN__CLK_manual	[2]	R/W	
0	5E	94	R_OTHERCLK_manual	[3]	R/W	
0	5E	94	R_SRAM_CLK_manual	[4]	R/W	
0	5E	94	R_I2C__CLK_manual	[5]	R/W	
0	60	96	TS_osc_code[6:0]	[6:0]	R/W	OSC code in use
0	60	96	OSC_BIST_OK	[7]	R/W	OSC bist done identifier
0	69	105	R_Pox_UB[7:0]	[7:0]	R/W	Proximity Up Bound
0	6A	106	R_Pox_LB[7:0]	[7:0]	R/W	Proximity Low Bound
0	6B	107	S_State	[0]	R	Proximity state
0	6C	108	S_AvgY[8:1]	[7:0]	R	Proximity object average brightness
0	80	128	Tm_GPIO0_OEL	[1]	R/W	0: GPIO0 Output Enable
0	80	128	Tm_GPIO0_IEB	[2]	R/W	0: GPIO0 Input Enable
0	80	128	Tm_GPIO1_OEL	[5]	R/W	0: GPIO1 Output Enable
0	80	128	Tm_GPIO1_IEB	[6]	R/W	0: GPIO1 Input Enable
0	81	129	Tm_GPIO2_OEL	[1]	R/W	0: GPIO2 Output Enable
0	81	129	Tm_GPIO2_IEB	[2]	R/W	0: GPIO2 Input Enable
0	81	129	Tm_GPIO3_OEL	[5]	R/W	0: GPIO3 Output Enable
0	81	129	Tm_GPIO3_IEB	[6]	R/W	0: GPIO3 Input Enable
0	82	130	Im_INT	[0]	R	INT Pin Input Signal
0	82	130	Tm_INT_OEL	[1]	R/W	0: INT Output Enable
0	82	130	Tm_INT_IEB	[2]	R/W	0: INT Input Enable
0	83	131	R_LightThd[7:0]	[7:0]	R/W	Only if the pixel > this, it would be taken as the part of object
0	8B	139	R_Cursor_ObjectSizeTh[7:0]	[7:0]	R/W	The object size threshold for cursor mode
0	90	144	R_NoMotionCountThd[6:0]	[6:0]	R/W	No motion counter threshold to quit has motion state
0	93	147	R_XDirectionThd[4:0]	[4:0]	R/W	Gesture detection horizontal threshold
0	94	148	R_YDirectionThd[4:0]	[4:0]	R/W	Gesture detection vertical threshold
0	95	149	R_ZDirectionThd[4:0]	[4:0]	R/W	Gesture detection z direction threshold
0	96	150	R_ZDirectionXYThd[4:0]	[4:0]	R/W	Gesture detection x and y threshold to detect forward or backward
0	97	151	R_ZDirectionAngleThd[3:0]	[3:0]	R/W	Gesture detection angle threshold to detect forward or backward
0	9A	154	R_RotateXYThd[4:0]	[4:0]	R/W	Gesture detection x and y threshold to detect rotation
0	9C	156	R_FilterWeight[1:0]	[1:0]	R/W	IIR filter weight between frame position distance
0	9C	156	R_FilterDistThd[4:0]	[6:2]	R/W	IIR filter frame position distance threshold
0	9F	159	R_UseBGModel	[0]	R/W	Background model enable
0	9F	159	R_BGUseDiffWeight	[1]	R/W	During calculating object center, use the weight between background and pixel.
0	9F	159	R_BGUpdateAtProcess	[2]	R/W	Update background at process state

0	9F	159	R_BGUpdateMaxIntensity_En	[3]	R/W	Backgounrd up bound threshold enable
0	9F	159	R_RotateEnh	[4]	R/W	Rotate gesture detection enable
0	9F	159	R_ZDirectionEnh	[5]	R/W	Backward and Forward gesture detection enable
0	9F	159	R_YDirectionEnh	[6]	R/W	Up and down gesture detection enable
0	9F	159	R_XDirectionEnh	[7]	R/W	Left and right gesture detection enable
0	A5	165	R_FilterImage	[0]	R/W	Image filter enable
0	A5	165	R_FilterAverage_Mode	[3:2]	R/W	Image filter mode: 0: weak average, 1: strong average, 2: 3 out of 9 median average
0	A5	165	R_UseLightWeight	[4]	R/W	Use pixel brightness as weight to calaulate center enable
0	B0	176	ObjectAvgY[8:1]	[7:0]	R	Object brightness
0	B1	177	ObjectSize[7:0]	[7:0]	R	Object size
0	B2	178	ObjectSize[11:8]	[3:0]	R	
0	C7	199	AngleAcc[7:0]	[7:0]	R	Gesture angle accumulation
0	C8	200	AngleAcc[10:8]	[2:0]	R	
0	CC	204	R_YtoZSum[5:0]	[5:0]	R/W	Z direction mapping parameter
0	CD	205	R_YtoZFactor[5:0]	[5:0]	R/W	Z direction mapping parameter
0	CE	206	R_PositionFilterLength[2:0]	[2:0]	R/W	IIR Filter length for cursor object center
0	CE	206	R_ProcessFilterLength[2:0]	[6:4]	R/W	IIR Filter length for gesture object center
0	CF	207	R_WaveCountThd[3:0]	[3:0]	R/W	Wave gesture counter threshold
0	CF	207	R_WaveAngleThd[3:0]	[7:4]	R/W	Wave gesture angle threshold
0	D0	208	R_AbortCountThd[2:0]	[2:0]	R/W	Abort gesture counter threshold
0	D0	208	R_AbortXYRatio[4:0]	[7:3]	R/W	Abort gesture X and Y direction ratio
0	D3	211	PositionFilterCenterX[7:0]	[7:0]	R	Horizontal Object Center after IIR filter for cursor mode
0	D4	212	PositionFilterCenterX[11:8]	[3:0]	R	
0	D4	212	PositionFilterCenterY[11:8]	[7:4]	R	Vertical Object Center after IIR filter for cursor mode
0	D5	213	PositionFilterCenterY[7:0]	[7:0]	R	
0	D6	214	PositionFilterAvgY[7:0]	[7:0]	R	Object brightness after IIR filter for cursor mode
0	D7	215	PositionFilterAvgY[8]	[0]	R	
0	D7	215	PositionFilterSize[9:8]	[5:4]	R	Object size after IIR filter for cursor mode
0	D8	216	PositionFilterSize[7:0]	[7:0]	R	
1	0	0	Cmd_HSize[5:0]	[5:0]	R/W	horizontal size
1	1	1	Cmd_VSize[5:0]	[5:0]	R/W	veritcal size
1	2	2	Cmd_HStart[5:0]	[5:0]	R/W	horizontal start point
1	3	3	Cmd_VStart[5:0]	[5:0]	R/W	veritcal start point
1	4	4	R_LS_Comp_DAvg_V	[7]	R/W	Lens Shading for digital vertical average
1	4	4	R_LS_Comp_DAvg_H	[6]	R/W	Lens Shading for digital horizontal average (unused)
1	4	4	Cmd_ASkip_V	[5]	R/W	analog vertical skip
1	4	4	Cmd_ASkip_H	[4]	R/W	analog horizontal skip
1	4	4	Cmd_DAvg_V	[3]	R/W	digital vertical average
1	4	4	Cmd_DAvg_H	[2]	R/W	digital horizontal average (unused)
1	4	4	Cmd_VFlip	[1]	R/W	vertical flip
1	4	4	Cmd_HFlip	[0]	R/W	horizontal flip
1	25	37	R_LensShadingComp_EnH	[0]	R/W	Lens Shading compensation enable, active high
1	27	39	R_OffsetY[6:0]	[6:0]	R/W	vertical offset of lens (s+6, -63~63)
1	28	40	R_LSC[6:0]	[6:0]	R/W	A, R2 coeff, (un-signed, 0~127)

1	29	41	R_LSFT[3:0]	[3:0]	R/W	shift amount of A*R2(un-signed, 0~15)
1	3E	62	Cmd_DebugPattern[7:0]	[7:0]	R/W	debug pattern for TG output
1	41	65	R_dac_EnL	[7]	R/W	DAC enable control in normal mode
1	41	65	R_dac_EnL_ADC	[6]	R/W	DAC enable control in scan ADC mode
1	41	65	R_dac[2:0]	[5:3]	R/W	DAC gain code in normal mode
1	41	65	R_dac_ADC[2:0]	[2:0]	R/W	DAC gain code in scan ADC mode
1	43	67	R_pwrsv	[7]	R/W	manual value for TS_pwrsv
1	43	67	R_pgatest_EnH	[6]	R/W	PGA test mode enable in normal mode
1	43	67	R_pgatest_EnH_PGA	[5]	R/W	PGA test mode enable in scan PGA mode
1	43	67	R_pgatest_EnH_ADC	[4]	R/W	PGA test mode enable in scan ADC mode
1	43	67	R_pgatestinv_EnH	[3]	R/W	PGA test signal inversion
1	43	67	R_dacref_sel	[2]	R/W	DAC input signal inversion selection
1	43	67	R_pga_EnL	[0]	R/W	PGA enable control
1	44	68	R_ggh[1:0]	[7:6]	R/W	PGA global gain
1	44	68	R_pga_fast[1:0]	[5:4]	R/W	PGA total bias current option
1	44	68	R_adctest[3:0]	[3:0]	R/W	ADC test sweep code
1	5E	94	T_clamp_drv_ctl[1:0]	[7:6]	R/W	Constant gm clamp circuit driver NMOS number select
1	5E	94	T_vdda28comp_enh	[5]	R/W	
1	5E	94	T_vbgp2vdda_byp_EnH	[4]	R/W	Vbgp2vdda buffer bypass, 0: buffer enable, 1: buffer off
1	5E	94	T_vdday_lvl[2:0]	[3:1]	R/W	VDDAY voltage select:2.31, 2.36, 2.41, 2.46, 2.52, 2.59, 2.66, 2.75V
1	5E	94	T_vbgp2vdday_byp_EnH	[0]	R/W	Vbgp2vdday buffer bypass, 0: buffer enable, 1: buffer off
1	65	101	R_IDLE_TIME[7:0]	[7:0]	R/W	idle time for normal operation
1	66	102	R_IDLE_TIME[15:8]	[7:0]	R/W	
1	67	103	R_IDLE_TIME_SLEEP_1[7:0]	[7:0]	R/W	idle time for weak sleep
1	68	104	R_IDLE_TIME_SLEEP_1[15:8]	[7:0]	R/W	
1	69	105	R_IDLE_TIME_SLEEP_2[7:0]	[7:0]	R/W	idle time for deep sleep, 2X
1	6A	106	R_IDLE_TIME_SLEEP_2[15:8]	[7:0]	R/W	
1	6B	107	R_Obj_TIME_1[7:0]	[7:0]	R/W	Weak sleep enter time, unit: one report frame time.
1	6C	108	R_Obj_TIME_1[15:8]	[7:0]	R/W	
1	6D	109	R_Obj_TIME_2[7:0]	[7:0]	R/W	Deep sleep enter time, unit: two report frame time.
1	6E	110	R_Obj_TIME_2[15:8]	[7:0]	R/W	
1	72	114	Enable/Disable PAJ7620U2	[0]	R/W	1: Enable PAJ7620U2 0: Disable PAJ7620U2
1	73	115	R_AUTO_SLEEP_Mode1	[1:0]	R/W	The way goes to weak sleep, 0: No motion sleep, 1: No object sleep
1	74	116	R_WakeUpSig_Sel[1:0]	[5:4]	R/W	1: GPIO0 output TG_Start signal, 2: GPIO0 output TG_Finish signal
1	74	116	R_Control_Mode[2:0]	[2:0]	R/W	R_Control_Mode=0 (Gesture Mode) R_Control_Mode=3 (Cursor Mode) R_Control_Mode=5 (PS Mode)
1	74	116	R_GPIO_Probe_En	[7:6]	R/W	GPIO debug probe select, to use this probe, set GPIO as output.
1	77	119	R_SRAM_Read_EnH	[0]	R/W	SRAM read enable, active high

Recommended Guideline for PCB Assembly

Recommended vender and type for Pb-free solder paste

1. Almit LFM-48W TM-HP
2. Senju M705-GRN360-K

IR Reflow Soldering Profile:

Temperature profile is the most important control in reflow soldering. It must be fine tuned to establish a robust process. The typical recommended IR reflow profile is showed in figure 8 below.

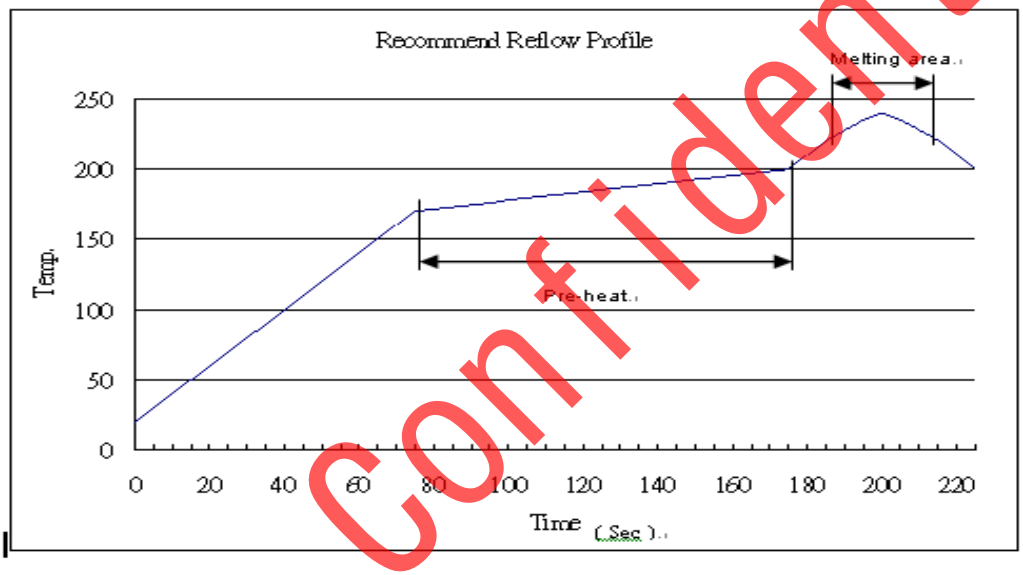


Fig. 8 IR Reflow Profile

Reflow Profile :

1. Average Ramp-up Rate (30°C to preheat zone): 1.5~ 2.5 Degree C/ Sec
2. Preheat zone:
   - 2.1 Temp ramp from 170~ 200 degree C
   - 2.2 Exposure time: 90 +/- 30 sec
3. Melting zone:
   - 3.1 Melting area temp > 220 degree C for at least 30 ~ 50 sec
   - 3.2 Peak temperature : 245 degree C.