DATA SHEET

GPY0130A

Monolithic and Wafer Level Packaged Three-Axis Accelerometer

Oct. 27, 2017

Version 1.0

GENERALPLUS TECHNOLOGY INC. reserves the right to change this documentation without prior notice. Information provided by GENERALPLUS TECHNOLOGY INC. is believed to be accurate and reliable. However, GENERALPLUS TECHNOLOGY INC. makes no warranty for any errors which may appear in this document. Contact GENERALPLUS TECHNOLOGY INC. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by GENERALPLUS TECHNOLOGY INC. for any infringement of patent or other rights of third parties which may result from its use. In addition, GENERALPLUS 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 Generalplus.

TABLE of CONTENTS

MONOLITHIC AND WAFER LEVEL PACKAGED THREE-AXIS ACCELEROMETER

1 DESCRIPTION

The Three-Axis Thermal Accelerometer is based on Proprietary thermal technology that is built with 0.18µm standard CMOS process and the most advanced Wafer Level Packaging. This device contains no moving sensor part and thus eliminates field reliability and repeatability issues associated with competitive products. For example, there is no measurable resonance (immunity to vibration), no stiction, and no detectable hysteresis. It also eliminates the "click" sounds typically heard in ball based orientation sensors. Shock survival of the MEMS sensing structure is greater than 200,000g. This sensor provides X/Y/Z axis acceleration signals with very low 0g offset, and temperature signal with high accuracy. In addition, it detects six orientation positions, X/Y shake and shake directions.

2 FEATURES

- Most Advantaged Technology in Industry
- Monolithically-Integrated Single Chip MEMS Sensor with On-Chip Signal Processing
- >200,000g Shock Survival Rating of Sensing Structure
- 12-bit Signal Output for X, Y and Z Axes
- Full Scale Range ±2g, ±4g and ±8g
- 8-bit Temperature Output (-50°C to +100°C)
- Smallest Wafer Level Package (WLP) Footprint 1.18mm×1.70mm×0.85mm
- 6-Position Orientation Detection
- Shake Detection with Interrupt
- Programmable Shake Detection Threshold
- I^{2}C Slave, FAST (≤400 KHz) Mode Interface
- 1.8~3.6V Single Supply Continuous Operation
- 1.8V Compatible I/O
- Embedded Power Up/Down Mode
- Eight I^{2}C Addresses Pre-settable by User
- RoHS Compliant

3 APPLICATIONS

- Consumer – LCD Projectors, Pedometers, DSC/DVC, MP3/MP4
- Gaming – Joystick, Toys
- Household Safety – Heating Fan, Iron, Cooling Fan

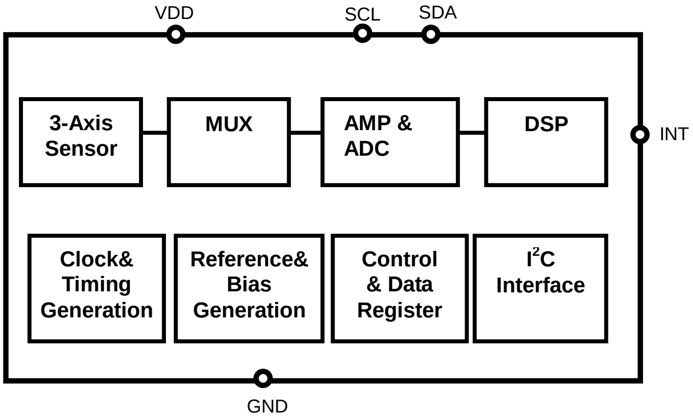


Figure 1: GPY0130A Functional block diagram

GPY0130A is the World's only monolithic and WLP three-axis accelerometer with smallest profile of 1.18mm×1.70mm×0.85mm. It is available in operating temperature range of -40°C to +85°C.

GPY0130A provides I^{2}C digital output with 400 KHz fast mode operation. And an interrupt pin (INT) is provided for orientation and X/Y shake detections. The device also has a power-down mode enabled through the I^{2}C interface.

4 SPECIFICATIONS

(Measurements @ 25°C, ±2g full scale range, acceleration = 0 g unless otherwise noted; VDD=1.8V unless otherwise specified)

Table 1: Parameter specification

PARAMETER	Conditions	MIN	TYP	MAX	UNITS
Full Scale Ranges (FSR) [1]	FSR[1:0] set to 0	-2		+2	g
	FSR[1:0] set to 1	-4		+4	
	FSR[1:0] set to 2	-8		+8	
Nonlinearity	Best fit straight line		±0.5		% FSR
Alignment Error [2]				±0.01	degree
Cross Axis Sensitivity [3]			±2.0		%
Sensitivity	±2g FSR		1024		LSB/g
	±4g FSR		512		
	±8g FSR		256		
Sensitivity Accuracy			±2		%
Sensitivity TC [4]	∆ from 25°C		±0.1		%/°C
Zero *g* Offset Bias Level [4]			±80		mg
Zero *g* Offset TC [4]	∆ from 25°C		±0.8		mg/°C
Frequency Response	@ -3dB		10		Hz
Output Data Rates (ODR)			100		Hz
Operating Voltage (VDD)		1.62	1.8	3.6	V
Operating Voltage Rise Time				8	mS
I^2C Bus Voltage (VIO)		1.2		VDD	V
Temperature Sensor Sensitivity			0.586		°C/LSB
Temperature Sensor Offset			0		LSB
Supply Current			1.6		mA
Power Down Current			0.8	1.0	µA
Operating Temperature Range		-40		+85	°C

Note:[1]: Guaranteed by measurement of initial offset and sensitivity

[2]: Alignment error is specified as the angle between the true and indicated axis of sensitivity

[3]: Relative contribution between any two of the three axis

[4]: With Generalplus's software driver (include)

5 ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (VDD)	-0.5 V to +5.0V
Storage Temperature	-40°C to +150°C
Acceleration	200,000g
ESD	2kV (HBM)

*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; the functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

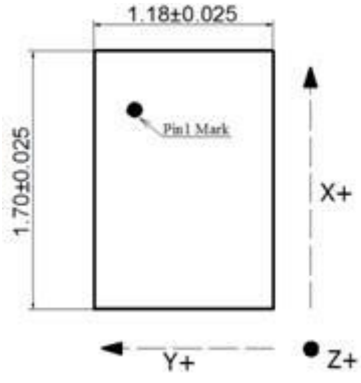
6 PIN DESCRIPTION: WLP PACKAGE

Table 2: Pin description

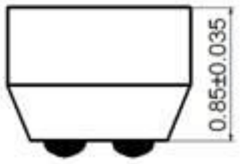
Pin	Name	Description	I/O
1	SCL	Serial Clock Line for I^2C bus	I
2	NC	No Connection	-
3	GND	Connected to Ground	I
4	INT	Interrupt Output	O
5	SDA	Serial Data Line for I^2C bus	I/O
6	VDD	1.62 V to 3.6 V	I

Caution: Electro Static Discharge (ESD) sensitive device.

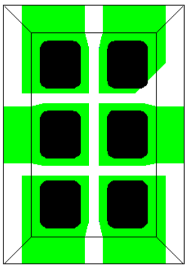
7 PACKAGE ILLUSTRATION



Top View



Side View



Bottom View

Figure 2: Package illustration

8 THEORY OF OPERATION

This device is a complete three-axis acceleration measurement system fabricated on a monolithic CMOS process. The device operation is based on proprietary sensor design gas convection.

A heat source, surrounded by thermopiles in the silicon chip, is suspended across a cavity. This heat source creates a field of heated gas around it, commonly referred as hot air bubble. The thermopiles capable of detecting changes in the temperature field around the heater. Applied acceleration disturbs the temperature profile. The temperature, and hence voltage output of thermopiles will then be different. The differential voltage at the thermopile outputs is directly proportional to the acceleration.

A signal path on the GPY0130A measures accelerations at three directions and temperature sequentially.

9 PIN DESCRIPTIONS

VDD – Supply input for the circuits and the sensor heater in the accelerometer. The DC voltage should be between 1.62 and 3.6 volts.

GND – Accelerometer ground pin.

NC – Should be left open.

INT – Interrupt output. The logic level on this pin reflects the state of the INT bit in the STATUS register. INT is set when the orientation differs from the last orientation read by the processor or an X/Y shake event is detected. INT needs to be pulled up to VIO. INT output is active low, and is cleared upon register INT_CLRx was written 1 by the I^2C master. Refer to the description of user register 0x01 in the "GPY0130A USER REGISTERS" section.

VIO – I^2C bus line.

SDA – I^2C serial data line that operates in FAST mode.

SCL – I^2C serial clock line that operates in FAST mode.

10 HARDWARE DESIGN CONSIDERATION

1. R1, R2, and R3 are pull-up resistors. The value can be determined by user according to the requirement of the host device; recommend using 2.7Kohm.
2. If INT is not used, it can be kept disconnected.
3. It is necessary to keep VDD voltage clean for best noise performance; suggested to keep VDD noise ≤ 1mV RMS. A low-ESR bypass cap is required and recommended value is 1~4.7μF. It should be placed close to the device as much as possible.

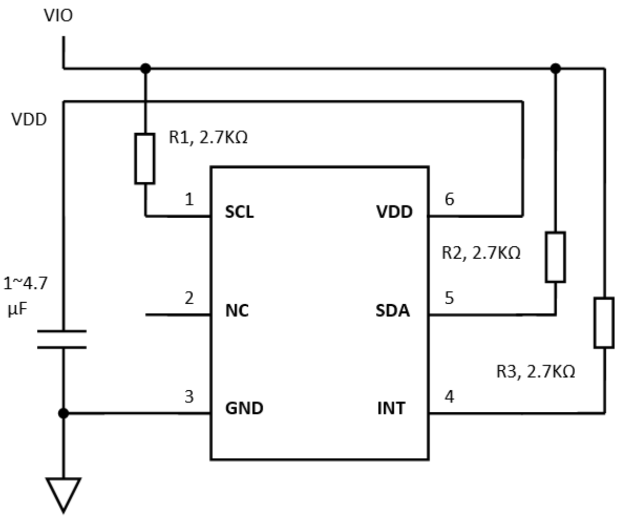


Figure 4: Connection diagram

11 LAND PATTERN

1. The device should be put into the center area of the PCB. Edge area is not recommended.
2. The device routing should be symmetric.
3. Solder mask defined PCB is preferred.
4. Recommended land pattern of PCB is shown in Figure 5 and recommended stencil opening size is shown in Figure 6.
5. Thickness of stencil should be 0.08mm ~ 0.1mm.
6. For ultra-small packages such as the GPY0130A, the bond pads and the amount of solder are relatively small. Therefore, strength of solder joints are not as high as compared with standard conventional SMT devices. For applications that require more robustness to mechanical shock (such as experienced in drop tests >1m), GENERALPLUS recommends the use of underfill material (e.g., Namics 1589-1, Henkel-UF3810, or equivalent).

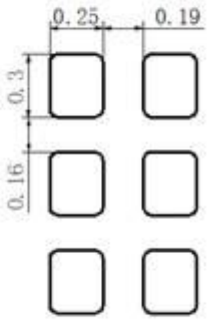


Figure 5: Recommended land pattern (unit: mm)

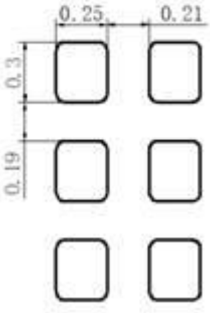


Figure 6: Recommended stencil opening size (unit: mm)

12 SOLDER REFLOW PROFILE

1. Reflow is limited by 2 times. Second reflow should be applied after device has cooled down to room temperature (25℃).
2. Recommended reflow profile for Pb free process is shown in Figure 7. The time duration of peak temperature (260℃) should be limited to 10 seconds.
3. Type 4 solder paste is recommended for a better SMT quality.

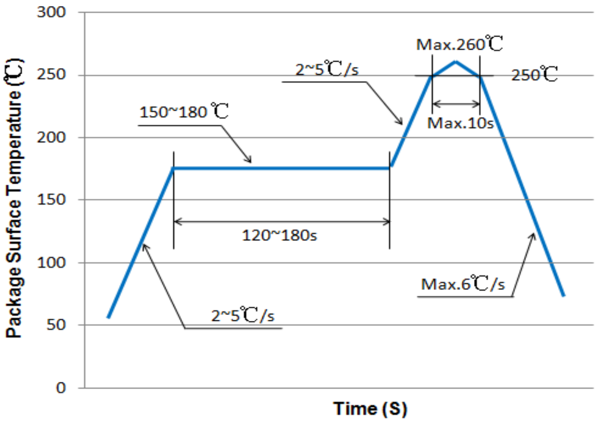


Figure 7: Recommended solder reflow profile

13 MANUAL SOLDERING

1. When soldering/repairing GPY0130A manually via solder iron or heater gun, the temperature on the device should not

exceed 260℃ and the time should be limited to 10 seconds.

2. Avoid bending or torqueing the PCB after the sensor is assembled.

14 DIGITAL INTERFACE

Table 3: Digital interface, I/V

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNITS
V_{IL}	Logic Input Low Level			0.3*VIO	V
V_{IH}	Logic Input High Level		0.7*VIO		V
V_{hys}	Hysteresis of Schmitt input		0.1		V
V_{OL}	Logic Output Low Level			0.2*VIO	V
V_{OH}	Logic Output High Level		0.8*VIO		V
I_i	Input Leakage Current	$0.1V_{DD}<V_{in}<0.9V_{DD}$	-10	10	µA

Table 4: Digital interface, timing

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
f_{SCL}	SCL Clock Frequency		0		400	kHz
$t_{HD;STA}$	START Hold Time		0.6			µS
$t_{SU;STA}$	START Setup Time		0.6			µS
t_{LOW}	LOW period of SCL		1.3			µS
t_{HIGH}	HIGH period of SCL		0.6			µS
$t_{HD;DAT}$	Data Hold Time		0		0.9	µS
$t_{SU;DAT}$	Data Setup Time		0.1			µS
t_r	Rise Time	From V_{IL} toV_{IH}			0.3	µS
t_f	Fall Time	From V_{IH} toV_{IL}			0.3	µS
t_{BUF}	Bus Free Time Between STOP and START		1.3			µS
$t_{SU;STO}$	STOP Setup Time		0.6			µS

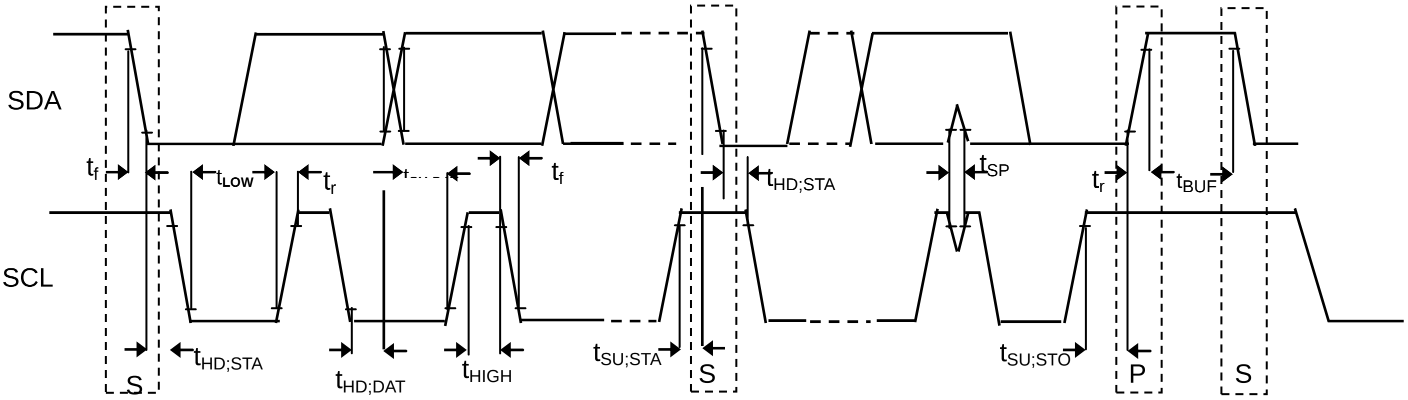


Figure 8: I^{2}C timing

Note: Both SCL and SDA should not be low for longer than 10ms

15 I^{2}C INTERFACE DESCRIPTION

A slave mode I^{2}C interface, capable of operating in standard or fast mode, is implemented on the GPY0130A. The interface uses a serial data line (SDA) and a serial clock line (SCL) to achieve bi-directional communication between master and slave devices. A master (typically a microprocessor) initiates all data transfers to and from the device, and generates the SCL clock that synchronizes the data transfer. The SDA pin on the GPY0130A operates both as an input and an open drain output. Since the GPY0130A only operates as a slave device, the SCL pin is always an input. There are external pull-up resistors on the I^{2}C bus lines. Devices that drive the I^{2}C bus lines do so through open-drain n-channel driver transistors, creating a wired NOR type arrangement.

Data on SDA is only allowed to change when SCL is low. A high to low transition on SDA when SCL is high is indicative of a START condition, whereas a low to high transition on SDA when SCL is high is indicative of a STOP condition**.** When the interface is not busy, both SCL and SDA are high. A data transmission is initiated by the master pulling SDA low while SCL is high, generating a START condition. The data transmission occurs serially in 8 bit bytes, with the MSB transmitted first. During each byte of transmitted data, the master will generate 9 clock pulses. The first 8 clock pulses are used to clock the data, the 9th clock pulse is for the acknowledge bit. After the 8 bits of data are clocked in, the transmitting device releases SDA, and the receiving device pulls it down so that it is stable low during the entire 9th clock pulse. By doing this, the receiving device "acknowledges" that it has received the transmitted byte. If the slave receiver does not generate an acknowledge, then the master device can generate a STOP condition and abort the transfer. If the master is the receiver in a data transfer, then it must signal the end of data to the slave by not generating an acknowledge on the last byte that was clocked out of the slave. The slave must release SDA to allow the master to generate a STOP or repeated START condition.

The master initiates a data transfer by generating a START condition. After a data transmission is complete, the master may terminate the data transfer by generating a STOP condition. The bus is considered to be free again a certain time after the STOP condition. Alternatively, the master can keep the bus busy by generating a repeated START condition instead of a STOP condition. This repeated START condition is functionally identical to a START condition that follows a STOP. Each device that sits on the I^{2}C bus has a unique 7-bit address (15H).

The first byte transmitted by the master following a START is used to address the slave device. The first 7 bits contain the address of the slave device, and the 8th bit is the R/W* bit (read = 1, write = 0; the asterisk indicates active low, and is used instead of a bar). If the transmitted address matches up to that of the GPY0130A, then the GPY0130A will acknowledge receipt of the address, and prepare to receive or send data. **Here, 7-bit I^{2}C Device Address is 15H. So device address byte is 0x2B for read mode and 0x2A for write mode.**

If the master is writing to the GPY0130A, then the next byte that the GPY0130A receives, following the address byte, is loaded into the address counter internal to the GPY0130A. The contents of the address counter indicate which register on the GPY0130A is being accessed. If the master now wants to write data to the GPY0130A, it just continues to send 8-bit bytes. Each byte of data is latched into the register on the GPY0130A that the address counter points to. The address counter is incremented after the transmission of each byte.

If the master wants to read data from the GPY0130A, it first needs to write the address of the register it wants to begin reading data from to the GPY0130A address counter. It does this by generating a START, followed by the address byte containing the GPY0130A address, with R/W* = 0. The next transmitted byte is then loaded into the GPY0130A address counter. Then, the master repeats the START condition and re-transmits the GPY0130A address, but this time with the R/W* bit set to 1. During the next transmission period, a byte of data from the GPY0130A register that is addressed by the contents of the address counter will be transmitted from the GPY0130A to the master. As in the case of the master writing to the GPY0130A, the contents of the address counter will be incremented after the transmission of each byte. The protocol for multiple byte reads and writes between a master and a slave device is depicted as follows.

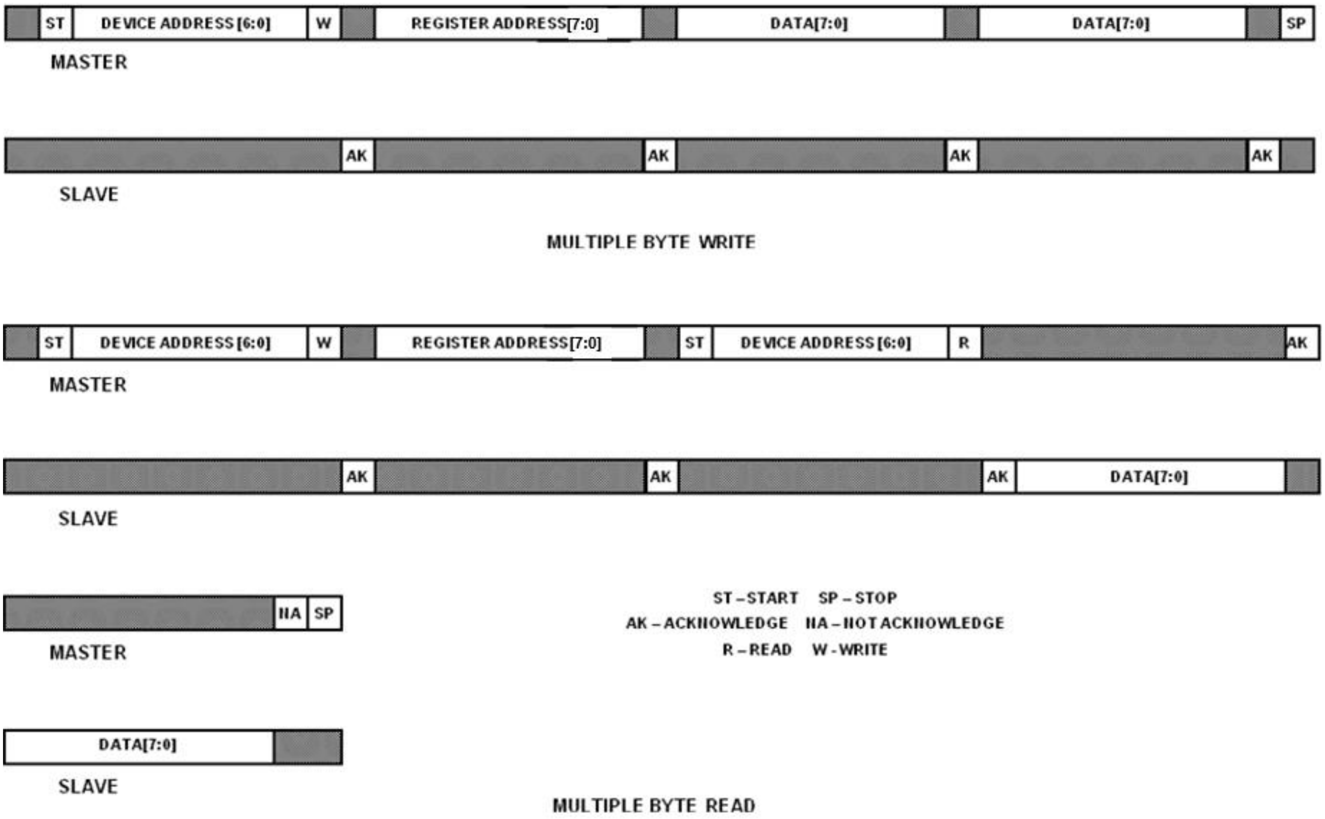


Figure 9: I^{2}C protocol

GPY0130A I^{2}C interface allows I^{2}C interface voltage VIO to be lower than the supply voltage VDD. VIO can be as low as 1.2 Volts. In order to achieve reliable operation avoid the situation when both SCL and SDA pins are low for longer than 10ms. Please contact GENERALPLUS if you plan on using I^{2}C interface voltage VIO greater than VDD.

16 GPY0130A USER REGISTERS

GPY0130A has 16 8-bit USER registers with addresses of 0x00 ~ 0x0F.

0x00: INT_SRC0 - Shake & change in orientation interrupt sources (read only)

Table 5: Shake & change in orientation interrupt sources, register 0x00 (read only)

D7	D6	D5	D4	D3	D2	D1	D0
CHORZ	CHORXY	0	0	SHYM	SHYP	SHXM	SHXP

0x00: INT_CLR0 - Shake & change in orientation interrupt clear (write only)

Table 6: Shake & change in orientation interrupt clear, register 0x00 (write only)

D7	D6	D5	D4	D3	D2	D1	D0
ORZC	ORXYC	-	-	SHYMC	SHYPC	SHXMC	SHXPC

0x01: INT_SRC1 - Data ready interrupt sources, tilt & orientation status (read only)

Table 7: Data ready interrupt sources, tilt & orientation, register 0x01 (read only)

D7	D6	D5	D4	D3	D2	D1	D0
TILT	ORZ	ORXY[1]	ORXY[0]	0	0	0	DRDY

0x01: INT_CLR1 - Data ready, tilt & data ready interrupt clear (write only)

Table 8: Data ready, tilt & data ready interrupt clear, register 0x01 (write only)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	SW_RST	-	-	-	DRDYC

The INT_SRCx and INT_CLRx are read-only and write-only registers, respectively, sharing the same respective I^{2}C addresses (0x00, 0x01). The INT_SRCx registers contain information indicating the source of interrupts produced on the INT output pin. When an event occurs which generates an interrupt output, the corresponding bit in the INT_SRCx register is set to 1. The I^{2}C master can poll the INT_SRCx registers to determine the cause of the interrupt, then read the appropriate registers to determine status, read data, and perform any necessary actions to service the interrupt. When interrupt servicing for that interrupt source is completed, the master writes a 1 into the corresponding bit location of the INT_CLRx register. GPY0130A then clears that bit of the INT_SRCx register, then automatically clears the same bit in the INT_CLRx register. GPY0130A does not clear any INT_SRCx register bits whose corresponding bit in the INT_CLRx register is set to 0.

The SHXP, SHXM, SHYP, or SHYM, bit in the INT_SRC0 register is set to 1 whenever a shake event is detected in the +X, −X, +Y, or −Y direction, respectively. If a shake event is detected in any axis and the appropriate bit in the INT_SRC0 register is set, any further detection of shake events in that axis is disabled until the shake bit for that axis is cleared by writing 1 to the corresponding bit in the INT_CLR0 register. Refer to the description of the DETECTION register for more information on shake detection.

The CHORXY bit in the INT_SRC0 register is set to 1 whenever a change in the X/Y plane orientation is detected. The CHORZ bit in the INT_SRC0 register is set to 1 whenever a change in the Z axis orientation is detected. Refer to the description of the DETECTION register for more information on orientation detection.

The ORXY[1:0] and ORZ[1:0] bits in INT_SRC1 indicate the orientation of GPY0130A. ORXY indicates orientation of the XY plane, and ORZ indicates direction of the Z axis. Mapping of the orientation bits is as follows:

Table 9: X and Y Orientation

ORXY[1]	ORXY[0]	Orientation
0	0	+X (X=+1g)
0	1	+Y (Y=+1g)
1	0	−X (X=−1g)
1	1	−Y (Y=−1g)

Orientation bits are determined by monitoring the magnitudes and signs of the X, Y, and Z acceleration outputs. The ORXY[1:0] bits are indicative of current orientation only if the off-axis tilt angle (angle between the XY plane and earth gravity vector) is small enough that there is sufficient acceleration amplitude on the X or Y axis to determine a valid XY orientation. For ORXY[1:0] to be updated, the larger of the X and Y axis acceleration magnitudes (a_x, a_y) must be greater than 0.375g. If this is not the case, the ORXY[1:0] bits retain their last values determined when there was sufficient acceleration amplitude to make a valid orientation decision.

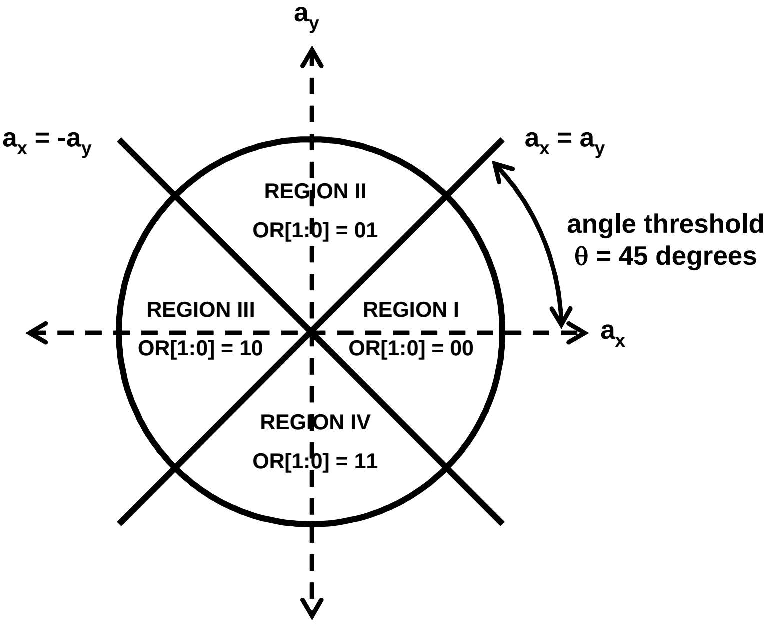


Figure 10: Orientation map

Table 10: Z Orientation

TILT/ORZ[1]	ORZ[0]	Orientation
0	0	positive Z, tilt not detected
0	1	negative Z, tilt not detected
1	0	positive Z, tilt detected
1	1	negative Z, tilt detected

The ORZ[1:0] bits indicate the Z axis orientation as shown in the table above. The vertical/horizontal Z axis orientation is determined by the same criteria used to determine the XY-plane off-axis tilt angle. When the XY plane has a sufficiently small off-axis tilt angle, XY orientation detection is valid (and continues to be updated), and the Z axis is detected as horizontal. ORZ[1] is also the bit for this TILT condition. When off-axis tilt angle exceeds the threshold discussed above, the Z axis is detected as either vertical up or vertical down, depending on the sign of the Z axis acceleration output.

To prevent flickering of the ORXY[1:0] and ORZ[1:0] bits near the orientation region boundaries, a valid measurement of the new orientation must be measured a consecutive number of times determined by the setting of the ORC[1:0] (orientation count) bits in the DETECTION register. This provides a low-pass filtering and hysteresis effect which prevents the orientation bits from rapidly changing.

The DRDY (data ready) bit in the INT_SRC1 register is set to 1 whenever a new measurement has been made, and new outputs are ready in the XOUT, YOUT, ZOUT and TOUT registers.

The TILT/ORZ[1] bit indicates the part is tilted too much in the X/Y plane, and the X/Y signal is too small to reliably determine orientation. In this case ORXY[1:0] updates are disabled, the orientation remains at its last value.

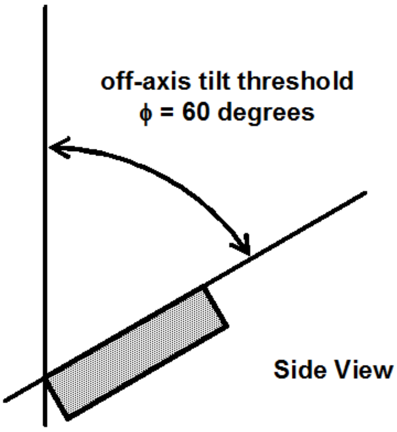


Figure 11: Off-axis tilt

Writing a 1 to the SW_RST bit will reset the part as if it had just powered up. The reset remains active as long as the current I^2C transfer is active. Accordingly, the I^2C logic itself is excluded, or it would not properly acknowledge (ACK) the current command.

0x02: STATUS - Instantaneous orientation status (read only)

Table 11: Instantaneous orientation status, register 0x02 (read only)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	ORD	ORIZ[1]	ORIZ[0]	ORIXY[1]	ORIXY[0]

The ORIXY[1:0] and ORIZ[1:0] bits are identical to ORXY[1:0] and ORZ[1:0], respectively, except that they are not subject to the same low-pass filtering as ORXY[1:0] and ORZ[1:0]. ORIXY[1:0] and ORIZ[1:0] are not normally used in the end application but are provided as outputs for information purposes.

ORD indicates that the OTP memory has been read. If a controller were to read a device before the OTP memory was read in, the results would be inaccurate, since the calibration data has not yet been loaded. Afterward, this bit will be 1 and cannot be reset by the user.

0x03: XOUT Upper - X-axis acceleration output MSB

Table 12: XOUT Upper, register 0x03

D7	D6	D5	D4	D3	D2	D1	D0
XOUT[11]	XOUT[10]	XOUT[9]	XOUT[8]	XOUT[7]	XOUT[6]	XOUT[5]	XOUT[4]

0x04: XOUT Lower - X-axis acceleration output LSB

Table 13: XOUT Lower, register 0x04

D7	D6	D5	D4	D3	D2	D1	D0
XOUT[3]	XOUT[2]	XOUT[1]	XOUT[0]	0	0	0	0

XOUT[11:0] is the 12-bit X-axis acceleration output. The output is in 2's complement format, with a range of −2048 to +2047.

0x05: YOUT Upper - Y-axis acceleration output MSB

Table 14: YOUT Upper, register 0x05

D7	D6	D5	D4	D3	D2	D1	D0
YOUT[11]	YOUT[10]	YOUT[9]	YOUT[8]	YOUT[7]	YOUT[6]	YOUT[5]	YOUT[4]

0x06: YOUT Lower - Y-axis acceleration output LSB

Table 15: YOUT Lower, register 0x06

D7	D6	D5	D4	D3	D2	D1	D0
YOUT[3]	YOUT[2]	YOUT[1]	YOUT[0]	0	0	0	0

YOUT[11:0] is the 12-bit y-axis acceleration output. Format is identical to the XOUT[11:0] output.

0x07: ZOUT Upper - Z-axis acceleration output MSB

Table 16: ZOUT Upper, register 0x07

D7	D6	D5	D4	D3	D2	D1	D0
ZOUT[11]	ZOUT[10]	ZOUT[9]	ZOUT[8]	ZOUT[7]	ZOUT[6]	ZOUT[5]	ZOUT[4]

0x08: ZOUT Lower - Z-axis acceleration output LSB

Table 17: ZOUT Lower, register 0x08

D7	D6	D5	D4	D3	D2	D1	D0
ZOUT[3]	ZOUT[2]	ZOUT[1]	ZOUT[0]	0	0	0	0

ZOUT[11:0] is the 12-bit z-axis acceleration output. Format is identical to the XOUT[7:0] output.

0x09: TOUT - Temperature output

Table 18: TOUT, register 0x09

D7	D6	D5	D4	D3	D2	D1	D0
TOUT[7]	TOUT[6]	TOUT[5]	TOUT[4]	TOUT[3]	TOUT[2]	TOUT[1]	TOUT[0]

GPY0130A contains an on-chip temperature sensor whose output can be read through the I^{2}C interface. The output is in 2's complement format. The nominal value of TOUT[7:0] is 0 at a temperature of 25°C, and the sensitivity of the output is approximately 0.586°C/LSB.

0x0A: INT_MASK0 - Shake & orientation detection interrupt mask (write only)

Table 19: Shake & orientation detection interrupt mask, register 0x0A (write only)

D7	D6	D5	D4	D3	D2	D1	D0
ORZE	ORXYE	-	-	SHYME	SHYPE	SHXME	SHXPE

0x0B: INT_MASK1 - Data ready, tilt & data ready interrupt mask (write only)

Table 20: Data ready interrupt mask, register 0x0B (write only)

D7	D6	D5	D4	D3	D2	D1	D0
0	-	0	0	-	-	-	DRDYE

The INT_MASKx registers are used to selectively enable or disable individual interrupts. When any bit in the INT_MASKx register is set to 1, the corresponding interrupt in the INT_SRCx register is enabled, allowing the event which set that bit in the INT_SRCx register to produce an interrupt signal on the INT output pin. The INT output is the logical NOR of all bits in the INT_SRC0 register and bits D2-D0 in the INT_SRC1 register which are enabled by the corresponding bits in the INT_MASKx register.

Note: The INT output is an active-low, open drain output. An external pull-up resistor is required on the INT pin when the INT output is used.

The bits in the INT_SRCx registers are set when the events corresponding to those bits occur, regardless of the state of the corresponding bits in the INT_MASKx register. This allows the I^2C master to poll for occurrence of events, rather than to respond to interrupts created by the same events.

0x0C: DETECTION - Orientation and shake detection parameters (read/write)

Table 21: Orientation and shake detection parameters, register 0x0C (read/write)

D7	D6	D5	D4	D3	D2	D1	D0
SHM	SHTH[2]	SHTH[1]	SHTH[0]	SHC[1]	SHC[0]	ORC[1]	ORC[0]

ORC[1:0] sets the orientation count, which is the number of consecutive valid new orientation readings that must be made before a new orientation value is written into bits OR[1:0] in the STATUS register. The nominal rate at which readings are taken is 100 Hz. The number of consecutive valid orientation measurements required to effect a "long-term" orientation change is set by ORC[1:0] as follows: 00 – 16 readings, 01 – 32 readings, 10 – 64 readings, 11 – 128 readings.

SHC[1:0] sets the X/Y shake count, which determines the number of readings allowed between the first shake event (acceleration exceeding the threshold set by SHTH[2:0]) and the second shake event (acceleration breaking the threshold with opposite sign, if SHM = 0, or just reversing sign, if SHM = 1). The number of readings is set by SHC[1:0] as follows: 00 – 8 readings, 01 – 16 readings, 10 – 32 readings, 11 – 64 readings.

SHTH[2:0] sets the X/Y shake threshold that acceleration must exceed to trigger the first shake event. SHTH[2:0] sets the shake threshold for the X, Y, and Z axes to a value between 0.25g (for SHTH[2:0]=000) and 2.0g (for SHTH[2:0]=111); each increment of SHTH[2:0] by 1 LSB increases shake threshold by 0.25g.

SHM is the X/Y shake mode bit. If SHM = 0, then for a shake to be detected, the second shake event must exceed the threshold set by SHTH[2:0] with the opposite sign of the first shake event, within the number of readings set by SHC[1:0]. If SHM = 1, then the second shake event must just have the opposite sign of the first shake event within the number of readings set by SHC[1:0].

0x0D: CONTROL - Operating mode control and full-scale range (read/write)

Table 22: Operating mode control and full-scale range, register 0x0D (read/write)

D7	D6	D5	D4	D3	D2	D1	D0
-	FSR[1]	FSR[0]	Cksel	-	-	0	PD

PD is the power down bit. Setting PD = 1 places GPY0130A into a low-power, non-functional power down state.

Cksel allows to operate GPY0130A with external clock for test purposes. When this bit is high the SCL input is used to drive the clock generator instead of internal oscillator. This is a factory-only bit. Always write it a 0. The part may be damaged if this bit is set to a 1.

FSR[1:0] determine the full-scale input acceleration range of GPY0130A. The following table summarizes the full scale input ranges and output sensitivities for the XOUT, YOUT, and ZOUT acceleration outputs for each setting.

Reserve at least 15ms before reading X, Y, and Z registers that assures the values being read at the moment is steady.

Table 23: Full-scale Range and Sensitivity

FSR[1]	**FSR[0]**	**Range (g)**	**Sensitivity (LSB/g)**
0	0	±2	1024
0	1	±4	512
1	0	±8	256
1	1	Undefined	N/A

0x0E: Device ID, currently has a value of 0x02.

0x0F: Who_Am_I - is a read-only register to identify the GPY0130A.

Table 24: Who_Am_I, register 0x0F

D7	D6	D5	D4	D3	D2	D1	D0
WAI[7]	WAI[6]	WAI[5]	WAI[4]	WAI[3]	WAI[2]	WAI[1]	WAI[0]

WAI[7:0] is an 8-bit version code programmed by the factory into non-volatile memory on GPY0130A.

17 SOFTWARE DRIVER

1. Due to the initial value of Z-axis may not be accurate, an AOZ software driver is utilized to obtain a calibrated value from Z-axis and store it onto NVRAM. It will assure the accelerated speed of Z-axis falls within a reasonable range. The following section will introduce how to use an AOZ software driver and give a program example for user's reference.
2. To compute the position or angle using Z-axis, we must use Dynamin Calibration program to assure the correctness of Z-axis. For more information, please contact Generalplus.

18 FACTORY CALIBRATION FOR Z AXIS

18.1 Purpose

In order to make sure obtaining guaranteed performance of accelerometer, the entire G-sensor calibration operation should consider dynamically environmental changes including covering area by different glues, etc. Filtering the variable offsets will be greatly helpful to use this sensor, so this operation should be proceeded with setting adjusted argument into NVRAM (or any other storage can be saved) in the test program of factory process. With the completely integration of G-sensor calibration process, all possible movements such as Flipping, Shaking, Reversing or Waving will be precisely detected to output corresponding behaviors within all kinds of product.

18.2 Procedure

Three steps should be done as follow,

Before AOZ operation starts, GPY0130A should be faced up on a flat table and make sure the temperature is in the range of 15℃ ~35℃.

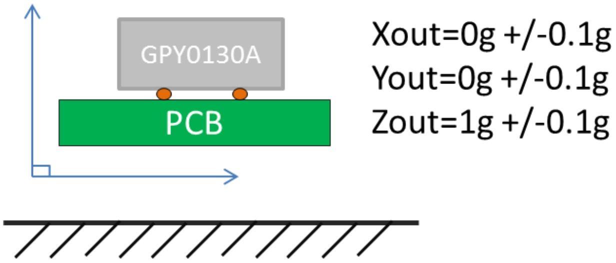
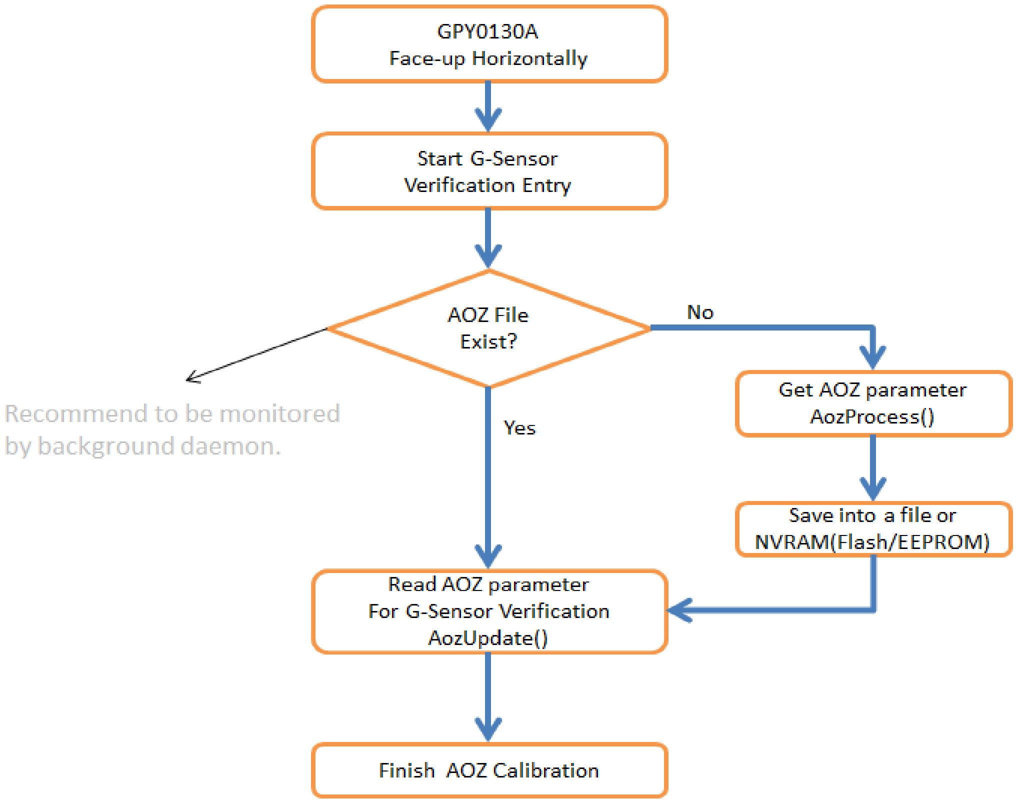


Figure: Calibration diagram

If it is the first time to do the operation, user can send a flag to the program to trigger "AozProcess()" function

Once the operation (AOZ parameters have been saved into a file or NVRAM (Flash or EEPROM)) is done, User should just call "AozUpdate()" function.

18.3 Flow chart



AOZ example C code

```
//int8;     // signed 8-bit number        (-128 to +127)
//uint8;      // unsigned 8-bit number    (+0 to +255)
//int16;      // signed 16-bt number       (-32,768 to +32,767)
//uint16;     // unsigned 16-bit number (+0 to +65,535)

uint16 AccBuff[4] = {0};

uint8 AOZ_high = 0;
uint8 AOZ_low = 0;

uint8 R_IIC_DataBuffer[11];

#define  D_AOZ_CalibPass     0
#define  D_AOZ_PosXOver    1
#define  D_AOZ_NegXOver    2
#define  D_AOZ_PosYOver    3
#define  D_AOZ_NegYOver 4
#define  D_AOZ_TempError 5
#define  D_AOZ_CalibFail       6

#define  D_AOZ_XLimit   100    //mg
#define  D_AOZ_YLimit   100    //mg
#define  D_AOZ_ZLimit   100    //mg
#define  D_AOZ_TempLimit  100      // =10.0℃

#define D_CalibrationTimes 5;    //Suggest Calibration times >=5
int main()
{
     F_ReadAozFromSPIFlash(); //Read AOZ_high & AOZ_low from SPI Flash

     //AOZ File Exist??
     if((AOZ_low != 0xff)&&(AOZ_low != 0)&&(AOZ_high != 0xff)&&(AOZ_high != 0))
     { //Yes
         AozUpdate();
     }
     else
     {//No
            AozProcess();

            // AOZ parameters(AOZ_high & AOZ_low) have been saved into a file or NVRAM (Flash or EEPROM)
            F_AozUpdateToSPIFlash();

            AozUpdate();
     }
}
/********************************************************************************/
/* AozProcess (void) : Get AOZ Parameters                                       */
/********************************************************************************/
void AozProcess(void)
{
    uint8     CalibrationTimes;
   uint8     DataReady;
   uint8     AOZ_StableCount = 0;
   uint16   AOZBuff[4];
   int16     AOZ_initial = 0;
   int16     AGZ_value = 0;
   float      fAOZ_ZData;
```

```
  float    fRawdataSum=0;
  float    fRawdataReal;
  float    fAOZ_delta;
  int16   sAOZ_delta;
  int16   AOZ_terminal;
  int16 X_Data,Y_Data,Z_Data,Temperature_Data;



CalibrationTimes = D_CalibrationTimes;

while(CalibrationTimes!=0)
{
        CalibrationTimes--;

  F_IIC_Write_B(0x0D,0x40);   //Set 8G Scale
  F_DelayMS(15);              //Delay 15 mS

  F_IIC_Write_B(0x0E,0x93);
  AOZBuff[0]=F_IIC_Read_B(0x23); //AOZ Low byte
  AOZBuff[1]=F_IIC_Read_B(0x24); //AOZ High byte
  AOZBuff[2]=F_IIC_Read_B(0x25); //AGZ Low byte
  AOZBuff[3]=F_IIC_Read_B(0x26); //AGZ High byte

  AOZ_initial = ((AOZBuff[1]&0x07)<<6) | ((AOZBuff[0]&0xfc)>>2);
  AGZ_value     = ((AOZBuff[3] & 0x03) << 3) | ((AOZBuff[2] & 0xe0) >> 5);

  if (AOZ_initial>255)
    AOZ_initial -= 512;

  if (AGZ_value>15)
    AGZ_value -= 32;

  while(1)
  {
    AOZ_StableCount++;

    do{
      F_IIC_Read_NByte(0x00,0x0A); //Data Update to R_IIC_DataBuffer[]

      DataReady=R_IIC_DataBuffer[1] & 0x01;
      DataReady&=0x01;

        }while(DataReady!=0x01);        //Wait Data Ready

    F_IIC_Write_B(0x01,0x01);     //Clear Data ready Flag

      X_Data=R_IIC_DataBuffer[3]*256+R_IIC_DataBuffer[4];
      Y_Data=R_IIC_DataBuffer[5]*256+R_IIC_DataBuffer[6];

      X_Data=(signed int)((signed long)X_Data*1000 >>12);
      Y_Data=(signed int)((signed long)Y_Data*1000 >>12);

      if(X_Data>D_AOZ_XLimit)
        return  D_AOZ_PosXOver;
      if(X_Data<-D_AOZ_XLimit)
        return  D_AOZ_NegXOver;
      if(Y_Data>D_AOZ_YLimit)
        return  D_AOZ_PosYOver;
      if(Y_Data<-D_AOZ_YLimit)
        return  D_AOZ_NegYOver;

      // Check Temperature at 15~35 degree
      if(R_IIC_DataBuffer[9]&0x80)
        Temperature_Data= 0xFF00 | R_IIC_DataBuffer[9];
      else
        Temperature_Data = R_IIC_DataBuffer[9];
      Temperature_Data=(signed int) (((signed long)Temperature_Data*586+25000)/100);
      if( (Temperature_Data>(250+D_AOZ_TempLimit)) || (Temperature_Data<(250-D_AOZ_TempLimit)) )
        return  D_AOZ_TempError;

      Z_Data=R_IIC_DataBuffer[7]*256+R_IIC_DataBuffer[8];
      fAOZ_ZData = (float)Z_Data /4096.0;

    fRawdataSum += fAOZ_ZData;

    if(AOZ_StableCount>=20)
    {
      fRawdataReal    = (float)fRawdataSum/20.0;

      fAOZ_delta = (int)(((fRawdataReal-1.0)*16)/((float)(1.0 + (float)AGZ_value*0.03125)));

      if(fAOZ_delta > 0){
        sAOZ_delta = (signed int)((fAOZ_delta *10 + 5)/10);
      }else if(fAOZ_delta < 0){
        sAOZ_delta = (signed int)((fAOZ_delta *10 - 5)/10);
      }else{
        sAOZ_delta = 0;
      }

      AOZ_terminal = AOZ_initial - sAOZ_delta;
      if (AOZ_terminal < 0){
        AOZ_terminal += 512;
      }

      AOZ_low = (AOZ_terminal & 0x3f) << 2;
      AOZ_low = AOZ_low | (AOZBuff[0] & 0x03);
      AOZ_high = (AOZ_terminal & 0x1c0) >> 6;
      AOZ_high = (AOZBuff[1]&0xf8) | AOZ_high;

                                              AozUpdate();

      break;
     } // if(AOZ_StableCount>=20)
   }   // while(1)
 }// while(CalibrationTimes!=0)

  for(AOZ_StableCount=0;AOZ_StableCount<20;AOZ_StableCount++)
  {
    do{
    DataReady=F_IIC_Read_NByte(0x00,0x0A);
    DataReady=R_IIC_DataBuffer[1] & 0x01;
    DataReady&=0x01;

    }while(DataReady!=0x01);     //Data not Ready
    F_IIC_Write_B(0x01,0x01);     //Clear Data ready Flag

    Z_Data=R_IIC_DataBuffer[7]*256+R_IIC_DataBuffer[8];
```

```
        fAOZ_ZData = (float)Z_Data /4096.0;
        fAOZ_ZRawdataSum += fAOZ_ZData;

    }
    Z_Data =(signed int) (fAOZ_ZRawdataSum*50.0);    //(fAOZ_ZRawdataSum/20.0*1000.0);

    if(    (Z_Data<=(1000+D_AOZ_ZLimit)) && (Z_Data>=(1000-D_AOZ_ZLimit))      )
        return D_AOZ_CalibPass;
    else
        return D_AOZ_CalibFail;

}
/******************************************************************************/
/* AozUpdate (void) : Update AOZ Parameters to GPY0130A      */
/******************************************************************************/
void AozUpdate(void)
{
         F_IIC_Write_B(0x0E,0x93);
         F_IIC_Write_B(0x23,AOZ_low);
         F_IIC_Write_B(0x24,AOZ_high);
}
/******************************************************************************/
/* F_AozUpdateToSPIFlash (void) : Update AOZ Parameters to SPI Flash    */
/******************************************************************************/
void F_AozUpdateToSPIFlash(void)
{

       // AOZ parameters have been saved into a file or NVRAM (Flash or EEPROM)
       // Step1. Erase SPI Flash
       // Step2. Write AOZ_low & AOZ_high to SPI Flash
}

/******************************************************************************/
/* F_ ReadAozFromSPIFlash (void) : Read AOZ Parameters from SPI Flash    */
/******************************************************************************/
void F_ReadAozFromSPIFlash(void)
{
       // Step1. Read Data from SPI Flash and Update to AOZ_low & AOZ_high
}
```

19 PACKAGE OUTLINE DRAWING

19.1 Order Information

Product Number	Package Type
GPY0130A – EN091	WLCSP 6pins Package(tape and reel, 3K/reel)

19.2 Package Information

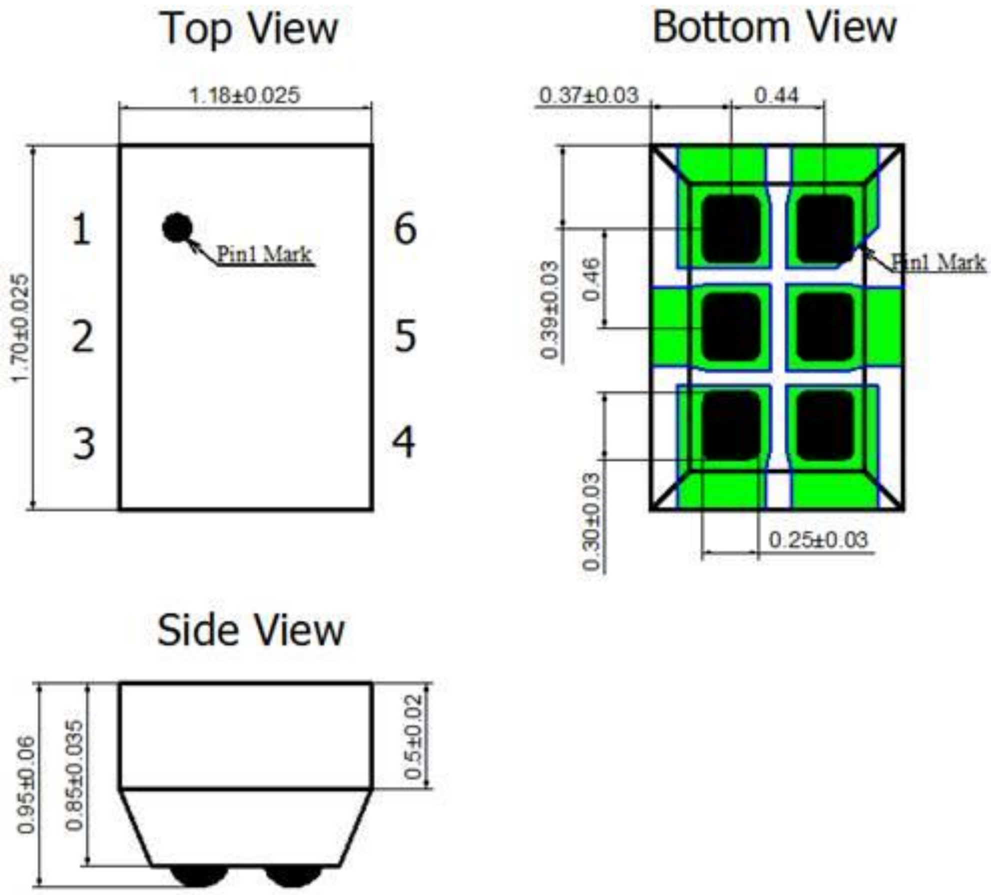
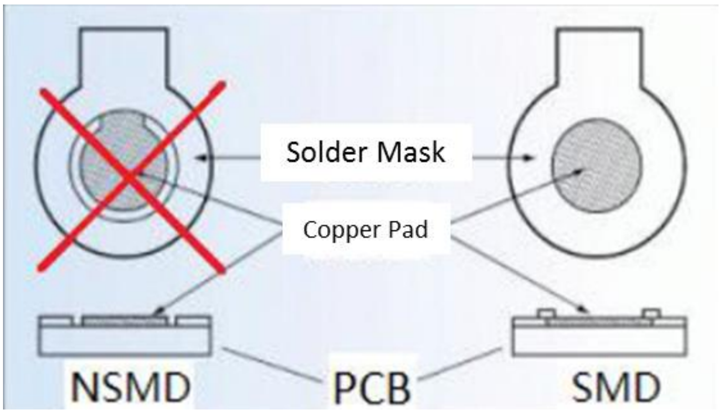


Figure 12: Mechanical package outline dimensions

PCB design requirement for GPY0130A

1. GPY0130A's substrate must be SMD (Solder mask define) design to match the size of GPY0130A's pad in order to increase the soldering reliability and yield rate.



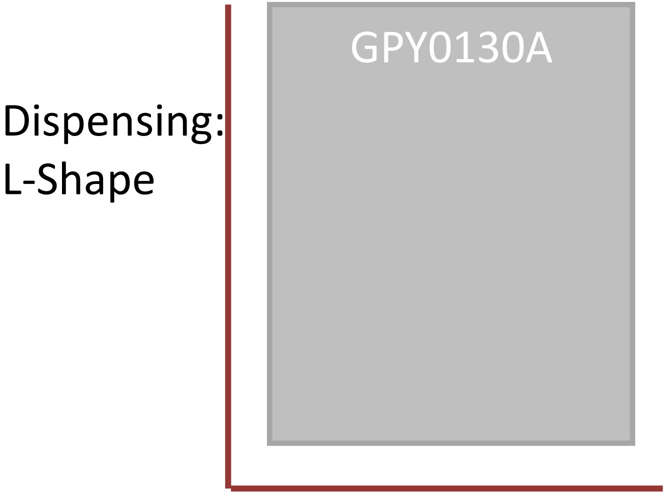
2. GPY0130A should be placed at the center, rather than the border, of a PCB to reduce the stress effect caused by deformation.
3. After SMT (Solder Mask technology), protect the component by filling underfill on its buttom. Please refer to the section "how to apply underfill to protect GPY0130A.
4. To avoid damaging components during SMT and underfilling, we suggest using masking tape for the protection. Also, clearly states and marks "Do not touch the area" warning in the respective document. Remove the masking tape before applying underfill. Avoid to damage the solder joint by carefully removing the tape.

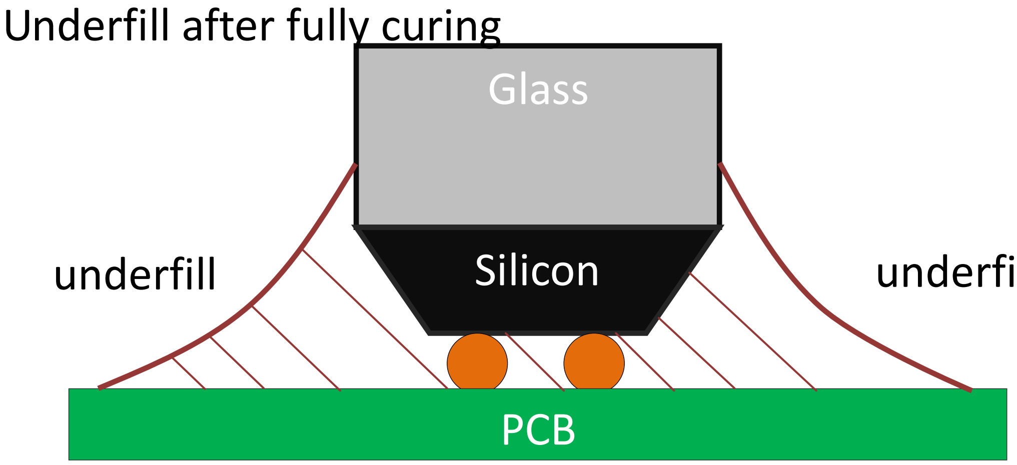
Suggestion for manual repairing

1. Assure the high temperature will not damage the surrounding components when removing defective component from PCB.
2. Clean the remaining of solder paste and solder mask from substrate
3. Soldering the solder paste on substrate. Make sure the solder paste applied on each substrate must be evenly distributed.
4. Precisely put the new component on pad on its respective location
5. When using heat gun, make sure the temperature does not exceed 260℃.

We suggest that all the steps mentioned above should be done under a 20X microscope.

How to apply underfill to protect GPY0130A:





1. The underfill must be fully filled the gap of silicon and PCB.

2. The underfill must fully filled to the side of glass and entire silicon.

3 For reworking, make sure all solder masks are completely cleaned while soldering.

4.Do not use the type of hard underfill (e.g. epoxy resins) to avoid unexpected outcome when distributing stress on it.

20 DISCLAIMER

The information appearing in this publication is believed to be accurate. Integrated circuits sold by Generalplus Technology are covered by the warranty and patent indemnification provisions stipulated in the terms of sale only. GENERALPLUS makes no warranty, express, statutory implied or by description regarding the information in this publication or regarding the freedom of the described chip(s) from patent infringement. FURTHERMORE, GENERALPLUS MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE. GENERALPLUS reserves the right to halt production or alter the specifications and prices at any time without notice. Accordingly, the reader is cautioned to verify that the data sheets and other information in this publication are current before placing orders. Products described herein are intended for use in normal commercial applications. Applications involving unusual environmental or reliability requirements, e.g. military equipment or medical life support equipment, are specifically not recommended without additional processing by GENERALPLUS for such applications. Please note that application circuits illustrated in this document are for reference purposes only.

21 REVISION HISTORY

Date	Revision #	Description	Page
Oct. 27, 2017	1.0	1. Add Software driver description. 2. Modify AOZ Software driver code. 3. Add PCB design request, Manual rework suggestion, How to protect GPY0130A with underfill.	
Jul. 29, 2016	0.2	Add AOZ operation	
Feb. 19, 2016	0.1	Original	18