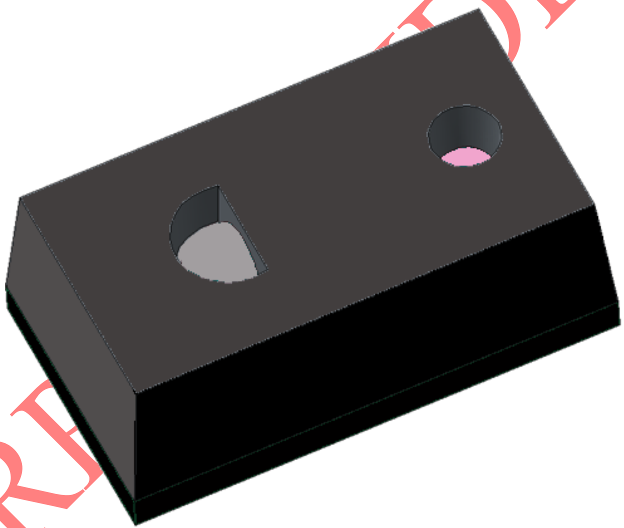
GP2AP03VT
SOFTWARE MANUAL
Time-of-Flight ranging Sensor



Contents

1. Introduction

GP2AP03VT is a distance measuring sensor in which a Time - of - Flight (ToF) type sensor is mounted in its own small package. This document is a description of how to implement software with user's product and factory calibration method.

Fig.1 shows the support range of Sharp's software. We provide sample code for MCU program. We also provide sample code for calibration to be carried out at the factory.

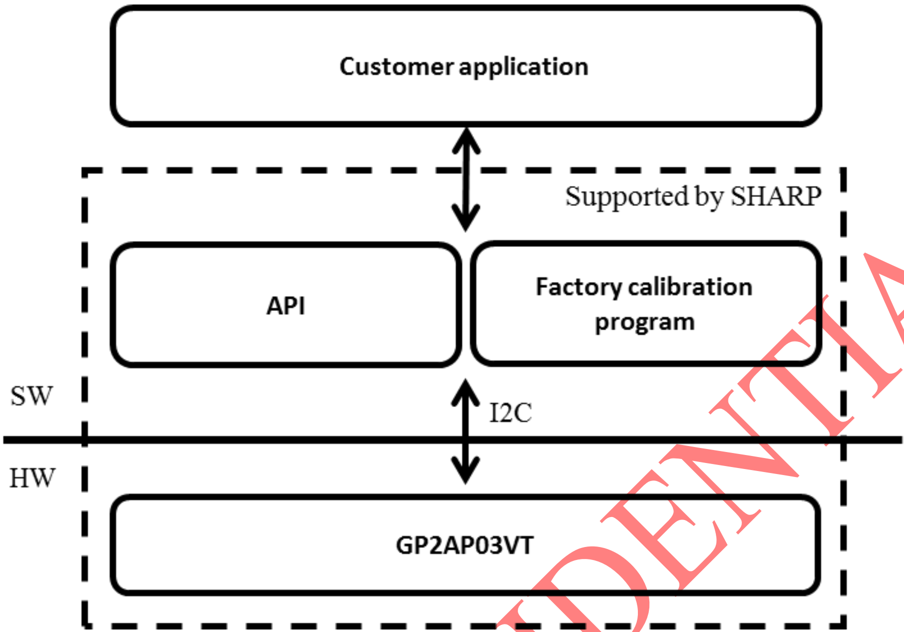


Fig.1 System strucure of software and GP2AP03VT

2. Sensor operation

2.1. Summary

Fig.2 shows the operation flow of the GP2AP03VT sensor.
(1) When the power supply voltage is applied to the sensor, the power-on reset circuit inside the IC circuit operates and goes into the shutdown state.
(2) By executing the initialization function via I^2C, set the value in the register of the sensor.
(3) Measurement is started by executing the measurement start function, and automatically returns to the shutdown state when the measurement is completed.
(4) The host can read the measurement result via I^2C and calculate the distance value by carrying out calculation processing. Approximately 6 msec of data reading time is required at I^2C communication speed 400 kHz.
(5) To continue measurement, execute the measurement start function again.

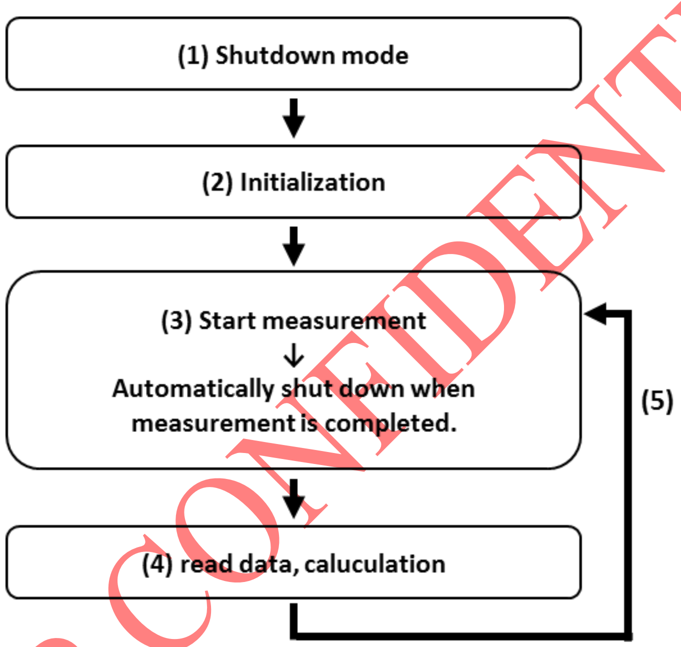


Fig.2 Operational flow

To suspend or pause the use of this sensor, perform the measurement end flow in section 2.4. Be sure to start from the Initalization flow (2) above when starting again.

2.2. Initialization

2.2.1. Initialization flow

Fig.3 shows the operation flow at initialization. When voltage is applied to the sensor, the sensor enters the shutdown state.

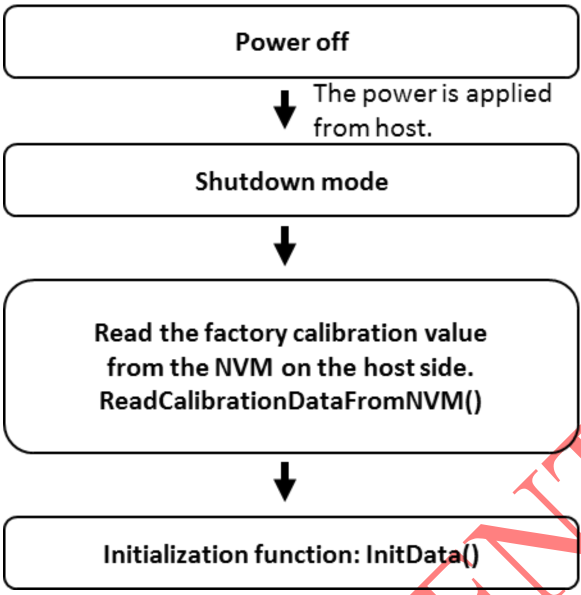


Fig.3 Initialization flow

If the factory calibration process has already been executed, execute the ReadCalibrationDataFromNVM () function and read the calibration value from the NVM on the host side. Calibration values include offset calibration value, crosstalk calibration value, factory calibration status. The offset calibration value is set to the offset1, offset2 variables. The crosstalk calibration value is set to data_xtalk variable. For details on factory calibration, see Chapter 3.
If calibration is not performed, set o offset1, offset2, data_xtalk variables to 0. After writing 0 to NVM, executing the ReadCalibrationDataFromNVM () function sets the variable to 0. Or directly assign 0 to the offset1, offset2, data_xtalk.

Table 1 Host side NVM address and program variable

NVM Address	Variable	Not perfomed	Perfomed
0x01	offset1	0	Read calibration value from NVM
0x02	offset2	0	
0x05, 0x06	data_xtalk	0	
0x07	factory_calibrated	0	0x03

In the sample program, the factory calibration information is saved in NVM address 0x05. When offset calibration is performed, the value of NVM address 0x05 is 0x01. When crosstalk calibration is performed, 0x02 is added to NVM address 0x05. It will be 0x03 if the calibration is completed normally.

Table 2 Factory calibratio status

NVM Address 0x07	
0x00	Not perfomed
0x01	Offset calibration
0x02	XTalk calibration
0x03	Both

Execute the InitData () function to initialize the sensor after setting the calibration value. At initialization, the trimming value is set in the register inside the IC. If the return value of the InitData () function is 0, the initialization function is executed normally. If the return value is -1, it is not executed normally. In that case, confirm that the GPIO terminal is in the open state.

Table 3 InitData() return value

InitData() Return value	Initialization status
0	Normal
-1	Error

2.2.2. **Measurement value acquisition time**

Two types of measurement time can be set. The measurement time when the communication speed of I^2C is 400 kHz is shown below.

Measurement time + data reading time (6 msec) is within 33.3 msec.

Depending on the usage of the customer, it may not be possible to satisfy the above measurement value acquisition time.

2.2.3. **Data acquisition**

Set the method to acquire data from the host. There are two types of measurement end interrupt and polling from the host. It is necessary to acquire measured values when this sensor is in shutdown state after the end of measurement.

After the measurement is completed, the INT pin becomes Low voltage. Also, the FLAG bit of register 01h becomes 1. By checking the INT terminal voltage or the FLAG bit, you can confirm that the measurement is completed. The CheckMeasurementEndFlag () function is provided to confirm that the measurement is completed.

Address	NAME	7	6	5	4	3	2	1	0
01h	COMMAND01H	-						FLAG	1/0

Fig.4 Regiseter at address 01h

Set the polling interval in polling mode to 33 msec or more when I^2C communication speed is 400 kHz.

2.2.4. **How to execute the initialization function**

For Normal mode, set the argument as follows.

InitData(dev, SHORT_RANGE_MODE, NORMAL_MODE)

2.3. Start measurement

Fig.5 shows the operation flow at the start of measurement. To start measurement, execute the following function.

StartMeasurement ()

The sensor automatically enters the shutdown mode when measurement is completed.

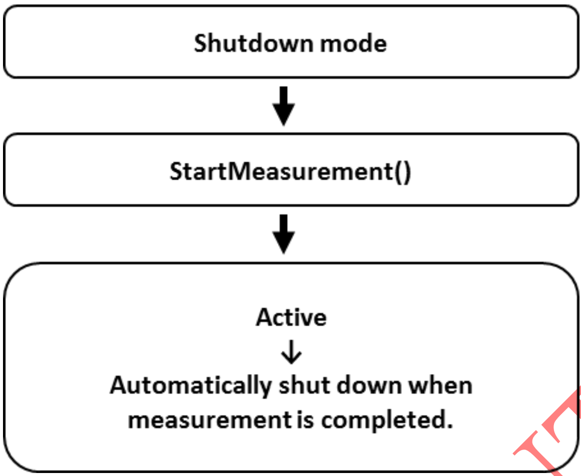


Fig.5 Operational flow at start of operation

2.4. Measurement end flow

Fig.6 shows the operation flow at the end of measurement. To terminate the sensor from the Active state, execute the following function. After interrupting the measurement, the sensor shifts to the shutdown mode.
StopMeasurement()
Be sure to execute the StopMeasurement () function when finishing measurement data acquisition.

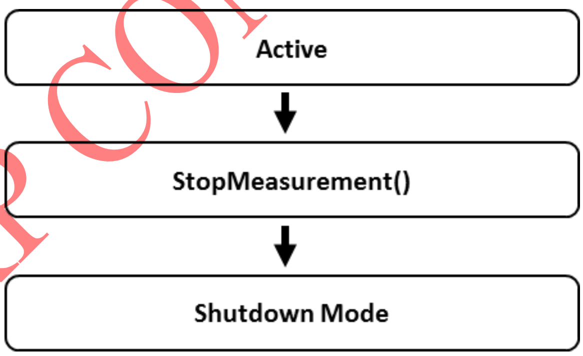


Fig.6 Measurement end flow

2.5. Acquisition of measurement data

Fig.7 shows an explanatory diagram of data acquisition processing at the occurrence of a measurement end interrupt or at the host's polling timer expiration.

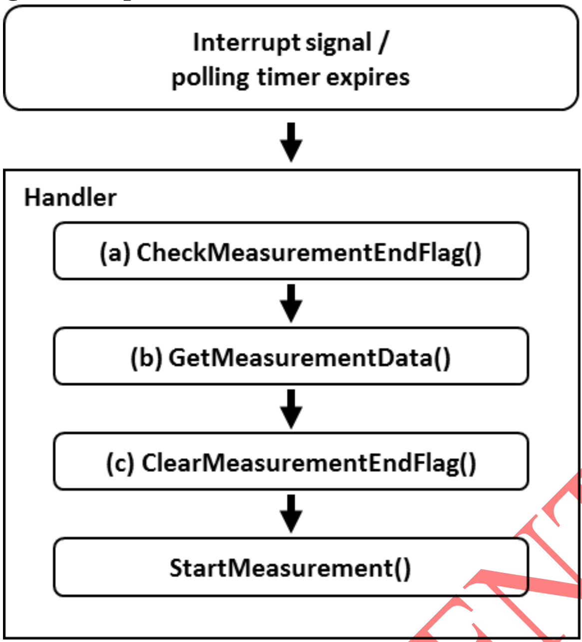


Fig.7 Data acquisition flow

2.5.1. (a) Check measurement end flag

To check the end of measurement, execute the CheckMeasurementEndFlag () function. If the return value of this function is 1, the measurement is completed. If it is 0, measurement has not been completed, wait until the measurement is completed and read the data.

2.5.2. (b) Acquisition of measurement data, calculation

Measurement data can be acquired by executing the following function. It takes about 6 msec when reading the measurement data from the sensor when the communication speed of I^2C is 400 kHz.

GetMeasurementData()

After acquiring measurement data, distance value can be obtained by performing calculation processing. The distance value is set in the variable range1, and the status flag is set in the range1_status. It is a valid distance value only when the status flag is 0x00. When the status flag is 0x80, it occurs when the crosstalk from the panel mounted on the whole sensor is large. In that case, the possibility that the panel is dirty is high. Otherwise an error occurs and "8888" meaning an error is output to range1.

Table 4 Status flag

Error Code	Error Status	Description
0x00	VALID_DATA	Valid data.
0x01	VCSEL_SHORT	When the VCSEL is short-circuited. If this error occurs, the VCSEL current will not flow inside the IC.
0x02	LOW_SIGNAL	The amount of reflected light obtained from the detected object is small
0x04	LOW_SN	The ratio of reflected light from the detected object and disturbance light is small.
0x08	TOO_MUCH_AMB	Disturbance light is large.
0x10	WAF	Wrap around error.
0x20	CAL_ERROR	Internal calculation error.
0x80	CROSSTALK_ERROR	Crosstalk from the panel is large.

2.5.3. (c) Clear measurement end flag

After finding the distance value, it is necessary to clear the measurement end flag. The measurement flag can be cleared by executing the following function. The INT terminal returns to H voltage. Also, the FLAG bit at address 01h of the register map becomes 0.

ClearMeasurementEndFlag()

2.5.4. How to check the setting parameters

Execute the CheckParams () function to check the setting parameters.

3. Factory calibration

3.1. Summary

This sensor is influenced by crosstalk from the panel by attaching a panel in front of the sensor. There are two kinds of effects of crosstalk from the panel.

- Shift distance value
- Distortion of distance value

In order to solve the above adverse effects, it is necessary to calibrate the sensor for each terminal at the user'sfactory. You can solve this by performing offset calibration for distance value deviation and crosstalk calibration for distance value distortion.

Fig.8 below shows the calibration flow. First place a white sheet at a distance of 100 mm from the sensor. After that, offset calibration can be performed by executing the PerformOffsetCalibration () function. Next, please do not put the object to be detected within 600 mm from the front of the sensor. Crosstalk calibration is performed by performing PerformXTalk Calibration ().The value obtained in the calibration process must be saved in the nonvolatile memory of the user product.

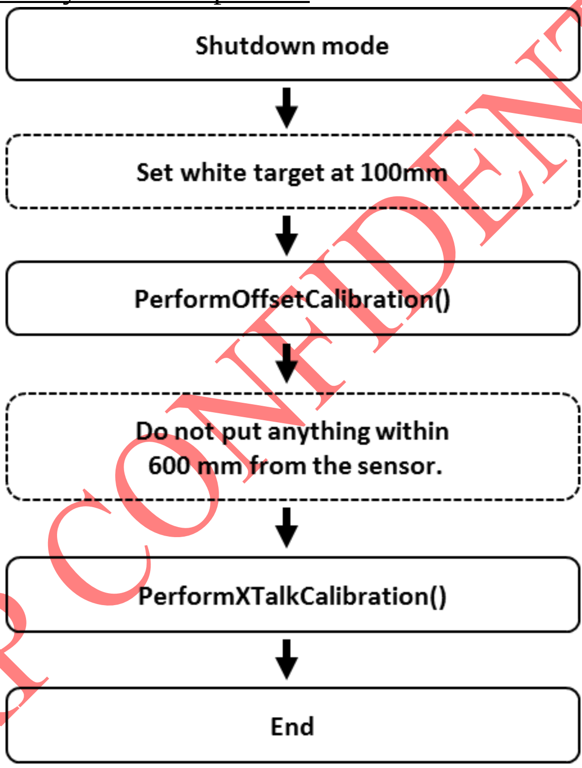


Fig.8 Calibration flow

This is an explanation table about NVM of the sample program. The bit width is 8 bits.。

Table 5 NVM of Sample Program

Address	Description
0x01	offset1
0x02	offset2
0x05	data_xtalk_lsb
0x06	data_xtalk_msb
0x07	factory_calibrated

Fig.9 shows the environment where the user's calibration is performed at the factory. Calibration should be performed in the absence of disturbance light. Please paste a light-absorbing sheet to prevent reflection of light from the wall. Room temperature should be from 20 ° to 25 °.

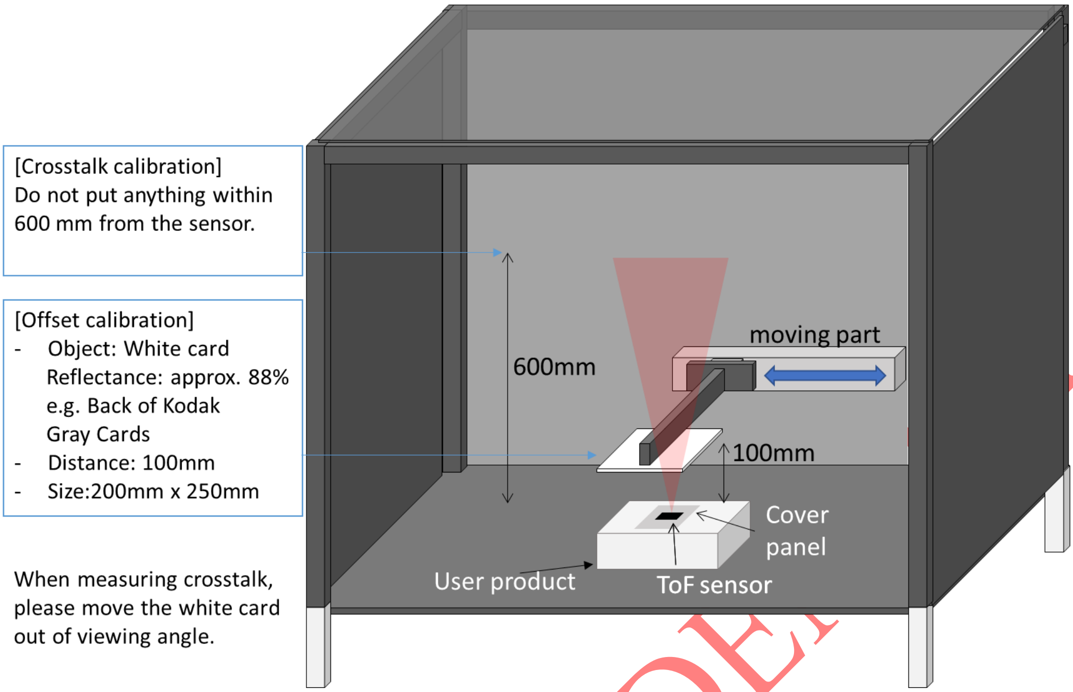


Fig.9 Calibration Environment at User Factory

3.2. Offset calibration

An image diagram is shown when the distance value is shifted by mounting the panel in front of the sensor. Please perform offset calibration to correct it.

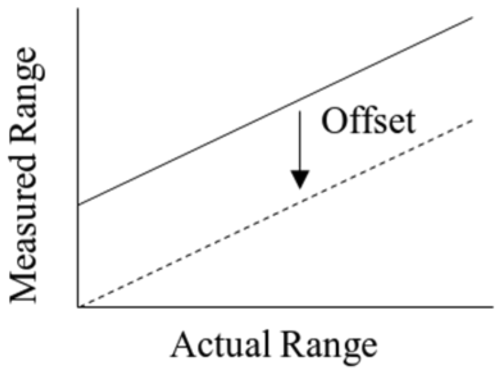


Fig. 10 Image diagram of offset calibration

Offset calibration method

1. Please put white paper at 100 mm from the sensor in an environment where disturbance light does not enter as shown in Fig.9.
2. Execute the PerformOffsetCalibration () function. Offset calibration is performed.
   Example: PerformOffsetCalibration(dev, DISTANCE_OFFSET_CALIB100MM)
3. The offset calibration values are saved in variables offset1, offset2.
   Offset value is normal within ± 75 mm.
4. Write the values of offset1, offset2 to the host's NVM.
5. In the sample program, 0x01 is written to the factory_calibrated variable indicating the calibration execution state. Execute the GetOffsetCalibrationData () function to check the value.

The values of offset1, offset2 are executed in the initialization function InitData () and the SetOffsetCalibrationData () function is executed and written to the register of the sensor.

3.3. Crosstalk calibration

Difference in distance value is caused by crosstalk from the cover glass. In order to compensate for it, it is necessary to perform crosstalk calibration.

Crosstalk calibration method

1. Do not place the object to be detected at 600 mm from the sensor in an environment where disturbance light does not enter.
2. Execute the PerformXTalkCalibration () function.
   Example: PerformXTalkCalibration(dev)
3. The offset calibration value is saved in the variable data_xtalk.
   Crosstalk value should be 0.00625or less.
   If the crosstalk value is larger than 0.00625, please review the optical design.
   The data_xtalk value is 16 bits wide and is output in floating point format. In that case, 0.00625 is 0x049A.
   If it is less than 0x049A, crosstalk calibration is successful.
4. Write the data_xtalk value to the host's NVM.
5. In the sample program, 0x02 is added to the factory_calibrated variable indicating the calibration execution state, and the value is 0x03.Execute the GetXTalkCalibrationData () function to check the value.

The data_xtalk value is executed in the initialization function InitData () and the SetXTalkCalibrationData () function is executed and written to the sensor register.