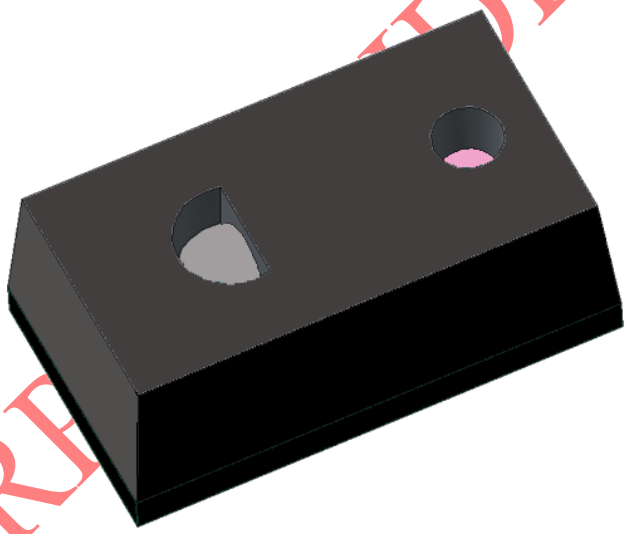
GP2AP03VT
SOFTWARE MANUAL
Time-of-Flight ranging Sensor



目次

1. イントロダクション

GP2AP03VT は Time-of-Flight(ToF)方式のセンサを独自の小型パッケージに実装した測距センサです。

このドキュメントはユーザー製品に TOF センサを組み込むためのソフトウェア実装方法と工場出荷時のキャリブレーション方法が書かれています。

図 1 はシャープのソフトウェアのサポート範囲を示しています。MCU 用サンプルコードを提供します。また、工場で実施するキャリブレーション用のサンプルコードも提供します。

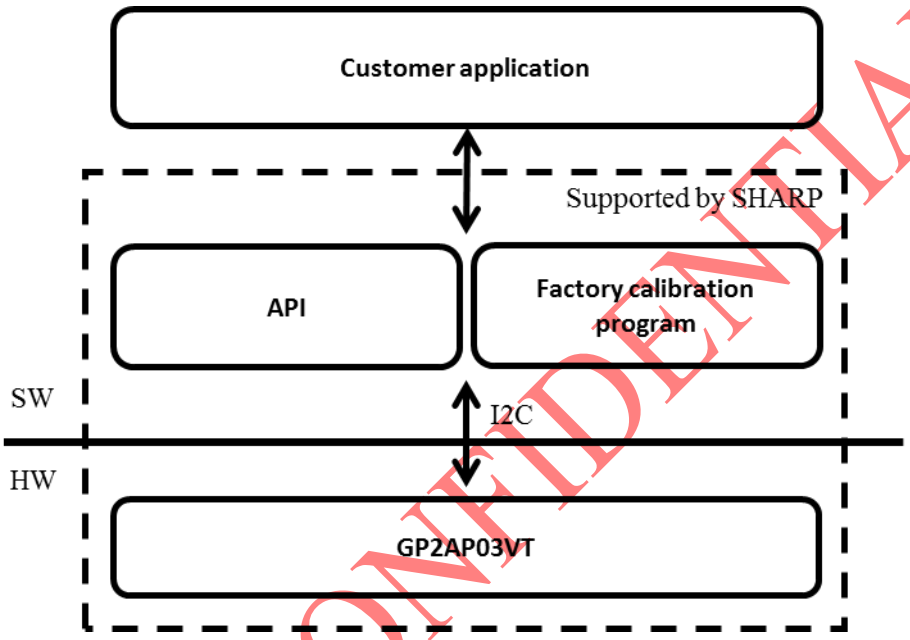


図 1　System strucure of software and GP2AP03VT

2. センサの動作

2.1. 概要

図 2 に GP2AP03VT センサの動作フローを示します。

（1）センサに電源電圧を印加すると IC 回路内部のパワーオンリセット回路が動作してシャットダウン状態になります。

（2）初期化関数を実行することにより、I²C 経由でセンサのレジスタに値を設定します。

（3）測定開始関数を実行することにより測定を開始します。測定が終了すると自動でシャットダウン状態に戻ります。

（4）ホストは I²C 経由で測定結果を読み込み、演算処理を実施することで距離値を求めることができます。I²C の通信時間は通信速度 400kHz の場合、約 6msec のデータ読み込み時間が必要になります。

（5）続けて測定したい場合は、再度測定開始関数を実行してください。

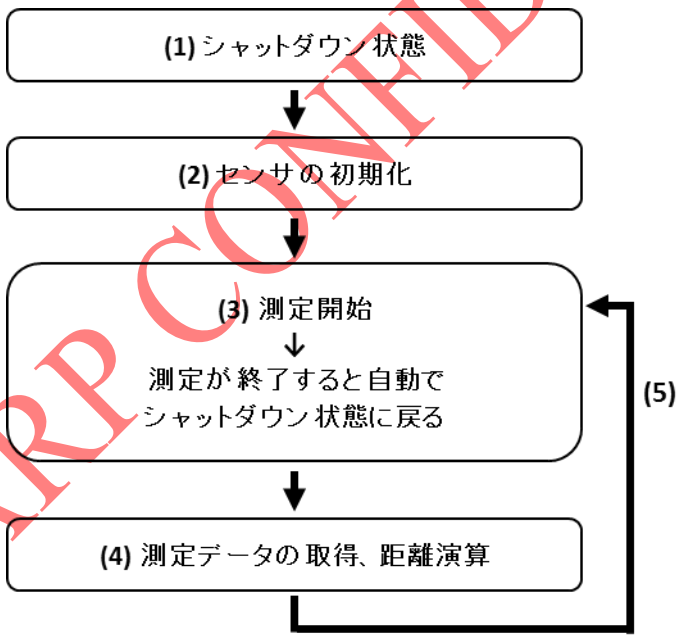


図２　動作フロー

本センサの使用を停止、休止するときは 2.4 章の測定終了フローを実施してください。再度スタートするときは必ず上記「(2)センサの初期化」から実施してください。

2.2. センサの初期化

2.2.1. 初期化フロー

図 3 は初期化時の動作フローです。センサに電圧が与えられると、シャットダウン状態になります。

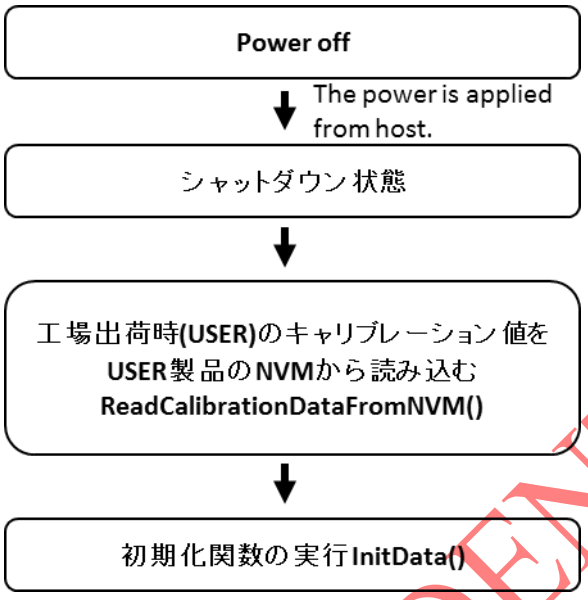


図 3　初期化フロー

工場でのキャリブレーション工程が実施済みの場合は、ReadCalibrationDataFromNVM()関数を実行して、キャリブレーション値をホスト側の NVM から読み込んでください。キャリブレーション値はオフセットキャリブレーション値、クロストークキャリブレーション値、工場キャリブレーション実施フラグがあります。オフセットキャリブレーション値は offset1, offset2 変数に設定してください。クロストークキャリブレーション値は data_xtalk 変数に設定してください。工場キャリブレーション実施フラグは factory_calibrated 変数に設定してください。キャリブレーションの詳細については 3 章を見てください。

キャリブレーション未実施の場合は、offset1, offset2, data_xtalk, factory_calibrated 変数に 0 を設定する必要があります。ホスト側の NVM に 0 を書き込んだ後、ReadCalibrationDataFromNVM()関数を実行すると変数に 0 がセットされます。もしくは直接 offset1, offset2, data_xtalk, factory_calibrated 変数に直接 0 を代入してください。

表 1　ホスト側の NVM アドレスとプログラムの変数設定

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

サンプルプログラムでは工場キャリブレーションを実施した情報を NVM アドレス 0x7 へ保存しています。オフセットキャリブレーションを実施した場合は NVM アドレス 0x07 の値は 0x01 です。クロストークキャリブレーションを実施した場合は NVM アドレス 0x07 に 0x02 が上書きされ、正常にキャリブレーションが完了した場合は 0x03 がセットされます。

表 2 工場キャリブーレーション実施情報

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

キャリブレーション値がセットされた後に、InitData()関数を実行してセンサを初期化してください。初期化時に IC 内部でトリミング値をレジスタに設定する作業が行われます。

InitData()関数の戻り値が 0 の場合は、初期化関数は正常に実行されています。戻り値が-1 の場合正常に実施されていません。その場合は、GPIO 端子がオープン状態になっていることを確認してください。

表 3　InitData()関数の戻り値

InitData() 戻り値	初期化の状態
0	正常に初期化が完了
−1	正常ではない。

2.2.2. 測定値取得時間

測定時間は２種類指定でき、I²C の通信速度が 400kHz の場合の測定時間を下記に示します。

測定時間＋データ読み込み時間(6msec)・・・33.3msec 以内

お客様の使用状況によっては上記の測定値取得時間を満たせないことがあります。

2.2.3. データ取得方法

ホストからデータを取得する方法は下記の２種類があります。

・測定終了割り込み

・ホストからのポーリング

このセンサは必ず測定終了後のシャットダウン状態の時に測定値を取得する必要があります。測定終了後に INT 端子が Low 電圧になります。また、レジスタ 01h 番地の FLAG ビットが 1 になります。

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

図４　01h 番地のレジスタ

測定が終了したことを確認するための CheckMeasurementEndFlag()関数を用意しています。

ポーリングモードのポーリング間隔は、I²C の通信速度が 400kHz の場合 33.msec 以上に設定してください。

2.2.4. 初期化関数の実行方法

Normal mode の場合は下記のように InitData()の引数を設定してください。

InitData(dev, SHORT_RANGE_MODE, NORMAL_MODE)

2.3. 測定開始

図 5 は測定開始時の動作フローです。測定開始させるには下記関数を実行してください。

StartMeasurement()

測定が終了すると自動でシャットダウンモードへ移行します。

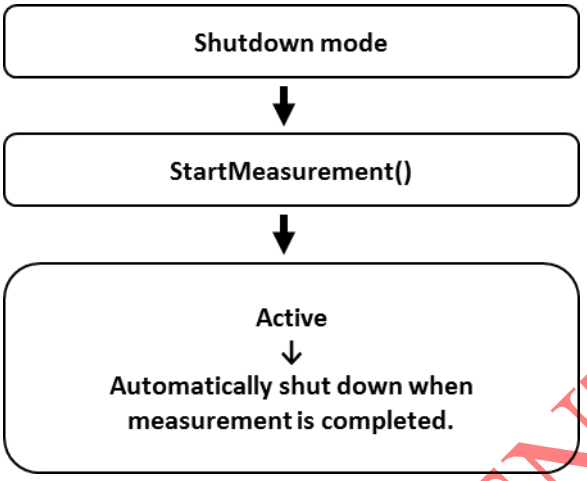


図 5　動作開始時の動作フロー

2.4. 測定終了

図 6 は測定終了時の動作フローです。センサを Active 状態から終了させるには下記関数を実行してください。センサは測定を中断したのち、シャットダウンモードへ移行します。

StopMeasurement()

測定データ取得を終了するときは必ず StopMeasurement()関数を実行してください。

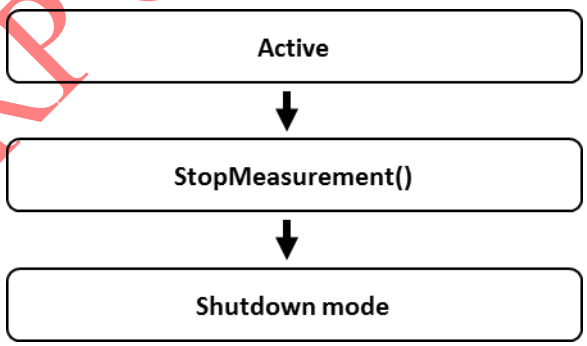


図 6　動作終了時の動作フロー

2.5. 測定データの取得、距離演算

測定終了割り込み発生時、もしくはホストでのポーリングタイマー満了時のデータ取得処理フローを図 7 に示します。

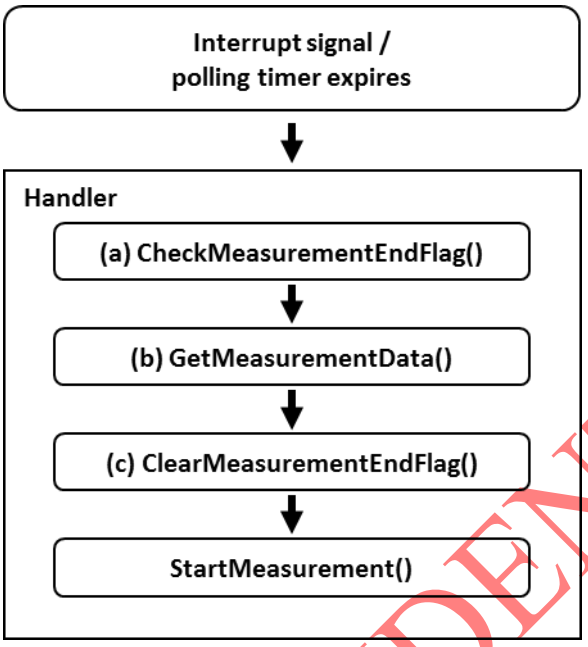


図 7　割り込み時のデータ取得フロー

2.5.1. (a) 測定終了フラグの確認

測定終了の確認をするには CheckMeasurementEndFlag()関数を実行する必要があります。この関数の戻り値が 1 の場合は測定終了です。0 の場合は測定が終了していませんので、測定が終了するまで待ってください。

2.5.2. (b) 距離データの取得、演算

CheckMeasurementEndFlag()関数の戻り値が 1 の場合、下記関数を実行することで測定データの読み込み、距離値演算を実施します。

GetMeasurementData()

センサから測定データの読み込みには、I^2C の通信速度が 400kHz の場合は約 6msec かかります。

測定データの取得後、演算処理を実施することで距離値が求まります。変数 range1 に距離値が、range1_status にステータスフラグがセットされます。ステータスフラグが 0x00 のときのみ有効なデータになります。ステータスフラグが 0x80 の場合はセンサ全面に装着されているパネルからのクロストークが大きい場合に発生します。その場合はパネルが汚れている可能性が高いです。それ以外はエラーとなり range1 にはエラーを意味する”8888”が出力されます。

表 4 range status

Error Code	Error Status	Description
0x00	NO ERROR	正確な距離が得られた場合
0x01	VCSEL_SHORT	VCSELがショートした場合。このエラーが発生した場合、IC内部ではVCSEL電流値をOFFにするように動作します。
0x02	LOW_SIGNAL	検知物から得られる反射光の量が少ない場合
0x04	LOW_SN	検知物から得られる反射光と外乱光の比が小さい場合
0x08	TOO_MUCH_AMB	外乱光が大きい場合
0x10	WAF	Wrap around errorの場合
0x20	CAL_ERROR	内部演算エラー
0x80	CROSSTALK_ERROR	パネルからのクロストークが大きい場合

2.5.3. (c) 測定終了フラグのクリア

最後に測定終了フラグをクリアする必要があります。下記関数を実行することで測定フラグをクリアすることができます。INT 端子は H 電圧に戻ります。また、レジスタマップの 01h 番地の FLAG ビットが 0 になります。

ClearMeasurementEndFlag()

2.5.4. 設定パラメータの確認方法

CheckParams()関数を実行することで、設定パラメータの確認ができます。

3. 工場出荷時のキャリブレーション

3.1. 概要

本センサはセンサの前にパネルを付けることにより、パネルからのクロストークの影響を受けます。

パネルからのクロストークの影響は２種類あります。

(a)　距離値のシフト

(b)　距離値の歪

上記の悪影響を解決するため、ユーザーの工場で端末一台ごとにセンサのキャリブレーションが必要です。(a)距離値のシフトにはオフセットキャリブレーション、(b)距離値の歪にはクロストークキャリブレーションを実施することで解決できます。

図 8 にキャリブレーションをフローを示します。最初にセンサから距離 100mm の場所に白紙を配置してください。その後、PerformOffsetCalibration()関数を実行することでオフセットキャリブレーションが実施できます。その次に、センサの前から 600mm 以内に検知物を置かない状態してください。そこで、PerformXTalkCalibration()関数を実施することでクロストークキャリブレーションが実施されます。キャリブレーション工程で求めた値はユーザー製品の不揮発メモリに保存する必要があります。

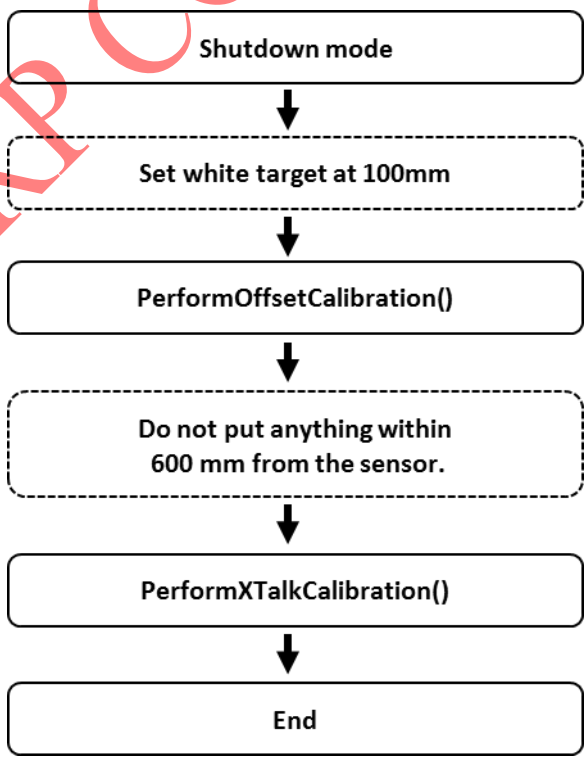


図 8 キャリブレーションフロー

サンプルプログラムの NVM についての説明表です。ビット幅は 8bit です。

表 5　サンプルプログラムの NVM

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

ユーザーの工場でのキャリブレーションを実施する環境を図 9 に示します。キャリブレーションは外乱光がない状態で実施する必要があります。また壁からの光の反射を防ぐため吸光シートを貼ってください。室温は 20~25°で実施してください。

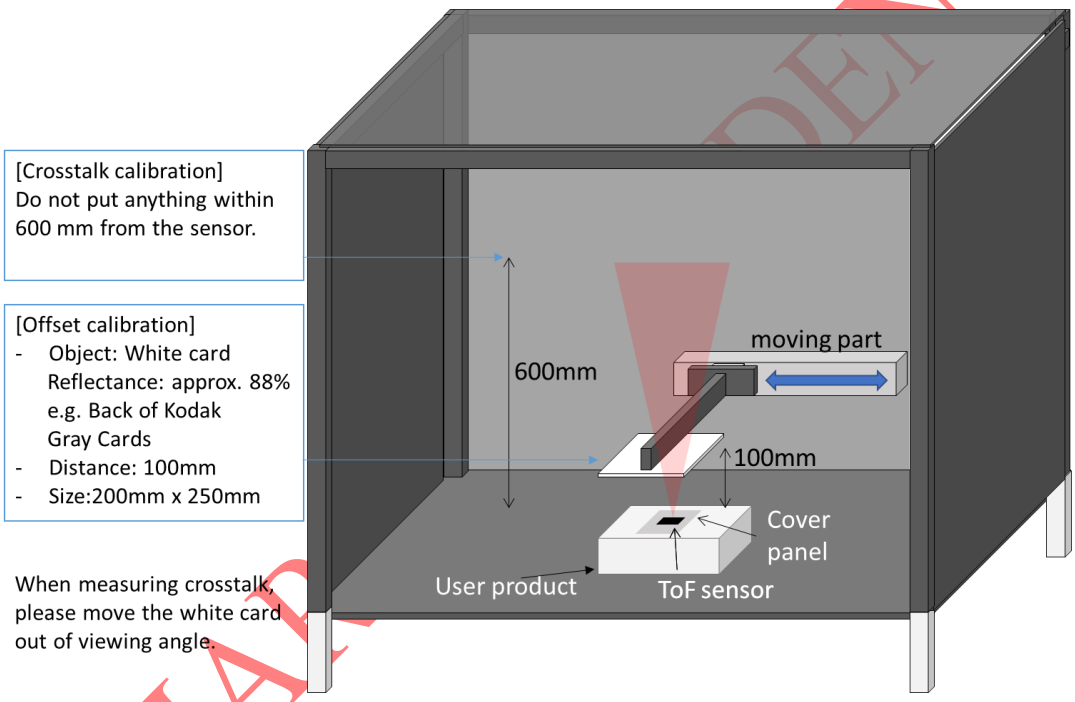


図 9　ユーザー工場でのキャリブレーション環境

3.2. オフセットキャリブレーション

センサの前にパネルを装着することで距離値がシフトした場合のイメージを図 10 に示します。それを補正するためにオフセットキャリブレーションを実施します。

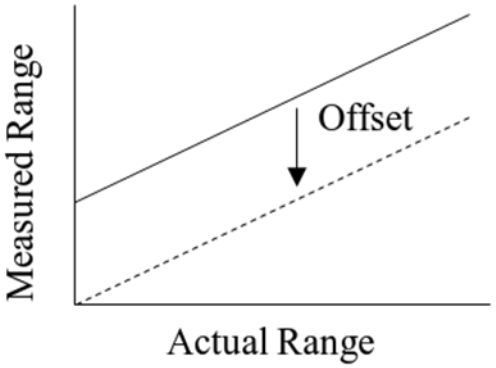


図 10　オフセットキャリブレーションの説明図

オフセットキャリブレーション方法

1．　図 9 のように外乱光が入らない環境で、センサから 100mm のところに白い紙を置いてください。

2．　PerformOffsetCalibration()関数を実行してください。オフセットキャリブレーションが実行されます。

実行例：PerformOffsetCalibration(dev, DISTANCE_OFFSET_CALIB100MM)

3．　変数 offset1, offset2 にオフセットキャリブレーション値が保存されます。オフセット値は±75mm 以内は正常です。

4．　offset1, offset2 の値をホストの NVM へ書き込んでください。

5．　サンプルプログラムではキャリブレーション実施状態を示す factory_calibrated 変数に 0x01 が書き込まれます。オフセットキャリブレーション値を確認するには GetOffsetCalibrationData()関数を実行してください。

offset1, offset2 の値は初期化関数 InitData()内で SetOffsetCalibrationData()関数が実行され、センサのレジスタに書き込まれます。

3.3. クロストークキャリブレーション

カバーガラスからのクロストークにより距離値に歪みが生じます。その歪を補正するためにクロストークキャリブレーションを実施する必要があります。

クロストークキャリブレーション方法

1． 外乱光が入らない環境で、センサから 600mm のところに検知物を配置しないでください。

2． PerformXTalkCalibration()関数を実行してください。

実行例：PerformXTalkCalibration(dev)

3． 変数 data_xtalk にオフセットキャリブレーション値が保存されます。クロストーク値は 0.00625 以下にする必要があります。クロストーク値が 0.00625 より大きい場合は光学設計を見直してください。data_xtalk 値は 16bit 幅で、浮動小数点形式で出力されます。0.00625 を浮動小数点形式で表すと 0x049A となります。0x049A より小さい値の場合はクロストークキャリブレーションが成功したことになります。

4． data_xtalk 値をホストの NVM へ書き込んでください。

5． サンプルプログラムではキャリブレーション実施状態を示す factory_calibrated 変数に 0x02 が足されて、値は 0x03 になります。クロストークキャリブレーション値を確認するには GetXTalkCalibrationData()関数を実行してください。

data_xtalk 値は初期化関数 InitData()内で SetXTalkCalibrationData()関数が実行され、センサのレジスタに書き込まれます。