

ADXL345 を用いた Shake 検出

by Tomoaki Tsuzuki

コンセプト

図 1 に ADXL345 の検出軸方向を示します。実際の使用環境では複数軸に同時に加速度が加えられる場合は殆どですが、このアプリケーションノートでは X 軸に加わる加速度のみを考えます。

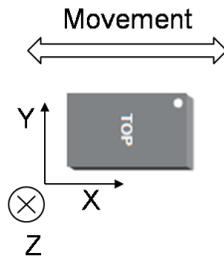


図 1. ADXL345 の検出軸方向

図 2 に Shake 動作時の加速度出力の実測値を示します。ADXL345 の設定は $\pm 16(g)$ 、最大分解能モード、100Hz データレートです。

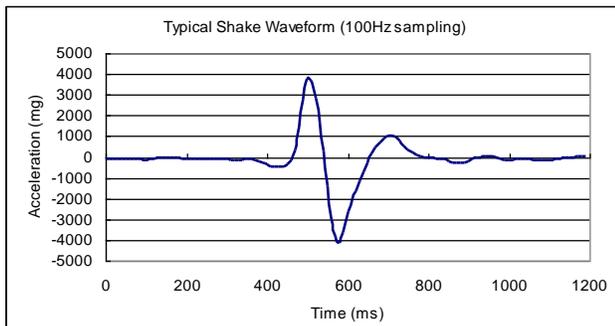


図 2. シェイク時の出力波形例

図 2 から、Shake 時の加速度波形は正と負のパルスで構成される事が解ります。パルスの振幅はどれだけ強く

Shake したかに依存しますが、殆どの場合で $\pm 3(g)$ ~ $\pm 8(g)$ 程度です。パルスの幅も Shake の強さに依存しますが、殆どの場合で 100ms 程度です。Shake 動作は正と負のパルス波形を認識する事で検出可能です。

アルゴリズム

Shake 検出のアルゴリズムはまず正か負のパルスを探します。パルスは、加速度が一定時間内にある閾値を超えて元に戻る波形として考えることができます。パルスが検出されたら、アルゴリズムは反対の極性のパルスを探します。一定時間 (TW) 内に反対極性のパルスが検出されれば、Shake として認識します。Shake の認識後は LATENCY と呼ばれる時間分だけ、アルゴリズムの動作を止めます。これは衝撃等による振動で Shake が連続して検出されるのを防ぐ役割を果たします。

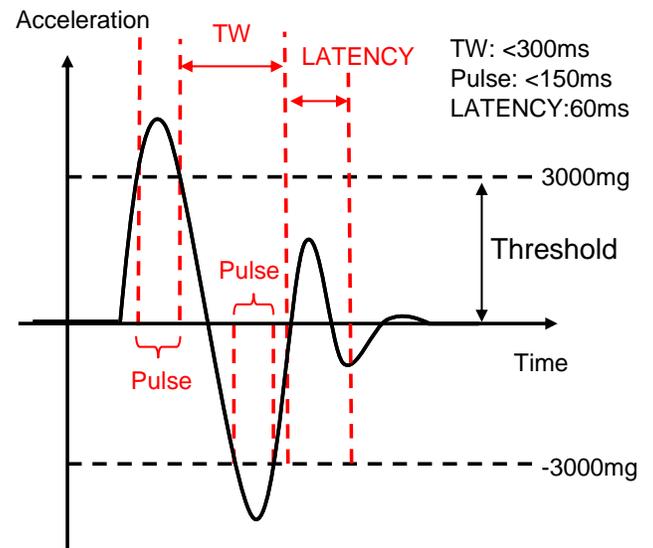


TABLE OF CONTENTS

コンセプト1
アルゴリズム1

ADXL345 の設定..... 3
シェイク検出アルゴリズム 4

ADXL345 の設定

表 1 にこのアプリケーションノートのアルゴリズムで動作を確認した ADXL345 の設定を示します。Auto-Sleep 機能を使うように設定されており、動きがない時は加速度センサが自動で低消費電力モードに移行します。THRESH_ACT, THRESH_INACT, TIME_INACT,

ACT_INACT_CTL は Auto-Sleep 機能の閾値の設定です。この設定は Application の仕様に依存しますが、表 2 に動作を確認した設定を記載します。FIFO を使用する事による消費電流の削減が可能です。FIFO の使用方法については AN-1025 を参照してください。

ADXL345 Setting			
Register Name	Address	Data	Meaning
BW/RATE	0x2C	0x0A	100Hz Data Rate
DATA_FORMAT	0x31	0x08	+/-16g, 13bit, Right justified data
POWER_CTL	0x2D	0x38	Sensor in measurement mode, Link and Auto-sleep enabled
INT_ENABLE	0x2E	0x98	Data Ready INT, Activity INT, Inactivity INT enabled

表 1. ADXL345 の設定

ADXL345 Setting			
Register Name	Address	Data	Meaning
THRESH_ACT	0x24	0x08	500mg
THRESH_INACT	0x25	0x08	500mg
TIME_INACT	0x26	0x02	2ms
ACT_INACT_CTL	0x27	0xFF	Use 3-axis, AC-coupled

表 2. ADXL345 の閾値の設定

図 4 はソフト処理全体のフローチャートです。プロセッサは割り込みの種類に応じて動作するように設定します。もし ACTIVITY か INACTIVITY 割り込みだった場合

は、ホストプロセッサは Shake アルゴリズムの変数を初期化して次の割り込みを待ちます。もし DRDY 割り込みだった場合は Shake アルゴリズムを走らせます。

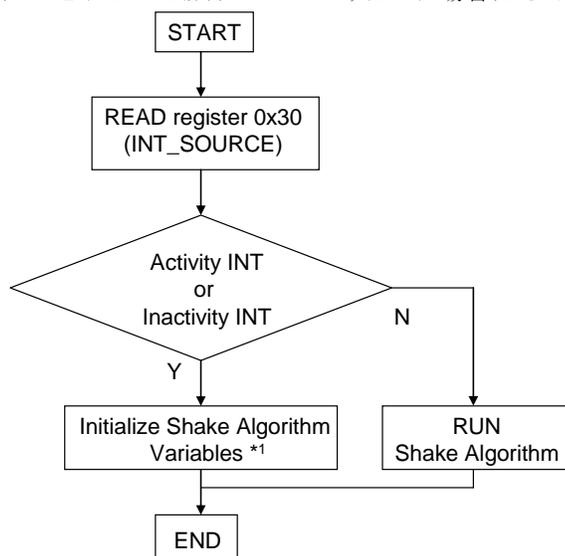


図 4. ソフト処理全体のフローチャート

シェイク検出アルゴリズム

本アプリケーションでは ADXL345 は 100Hz のデータレートに設定します。Shake のアルゴリズムは DRDY 割り込みをトリガとして走りますので、10ms に一回アルゴリズムを走らせる型になります。シェイク検出のアルゴリズムは下記のように実装できます。

1. 測定した加速度が閾値(TH)を超えたらタイマー動作を開始する
2. タイマーで加速度が閾値以下になるまでの時間

を測定する (パルス幅の測定)

3. もしパルス幅が設定した値(PW)以内であれば、パルスとして認識する
4. 一定時間(TWW)以内に反対極性のパルスが検出されれば Shake として認識する
5. 一定時間(LATENCY)の待ち時間を経過するまで待機

表 3 にアルゴリズムに使う定数と変数を示します。

Name	Type	Initial	Description
TW	Variable	0	Counting up Time Window
pFlag	Variable	0	Flag for positive pulse (become 1 when positive pulse is detected)
nFlag	Variable	0	Flag for negative pulse (become 1 when negative pulse is detected)
pTemp	Variable	0	Becomes 1 when in calculation of positive pulse
nTemp	Variable	0	Becomes 1 when in calculation of negative pulse
Pulse	Variable	0	Counting up pulse width
TH	Constant	768	Threshold for pulse recognition (3000 (mg))
PW	Constant	15	Threshold for pulse duration (150 (ms))
TWW	Constant	30	Threshold for Time window duration (300 (ms))
LATENCY	Variable	60	Dead Time after shake recognition it becomes 60 (600ms)
X	Variable	0	X axis acceleration (high-pass filtered)

表 3. アルゴリズムの変数と定数

図 5 と図 6 にアルゴリズムのフローチャートを示します。

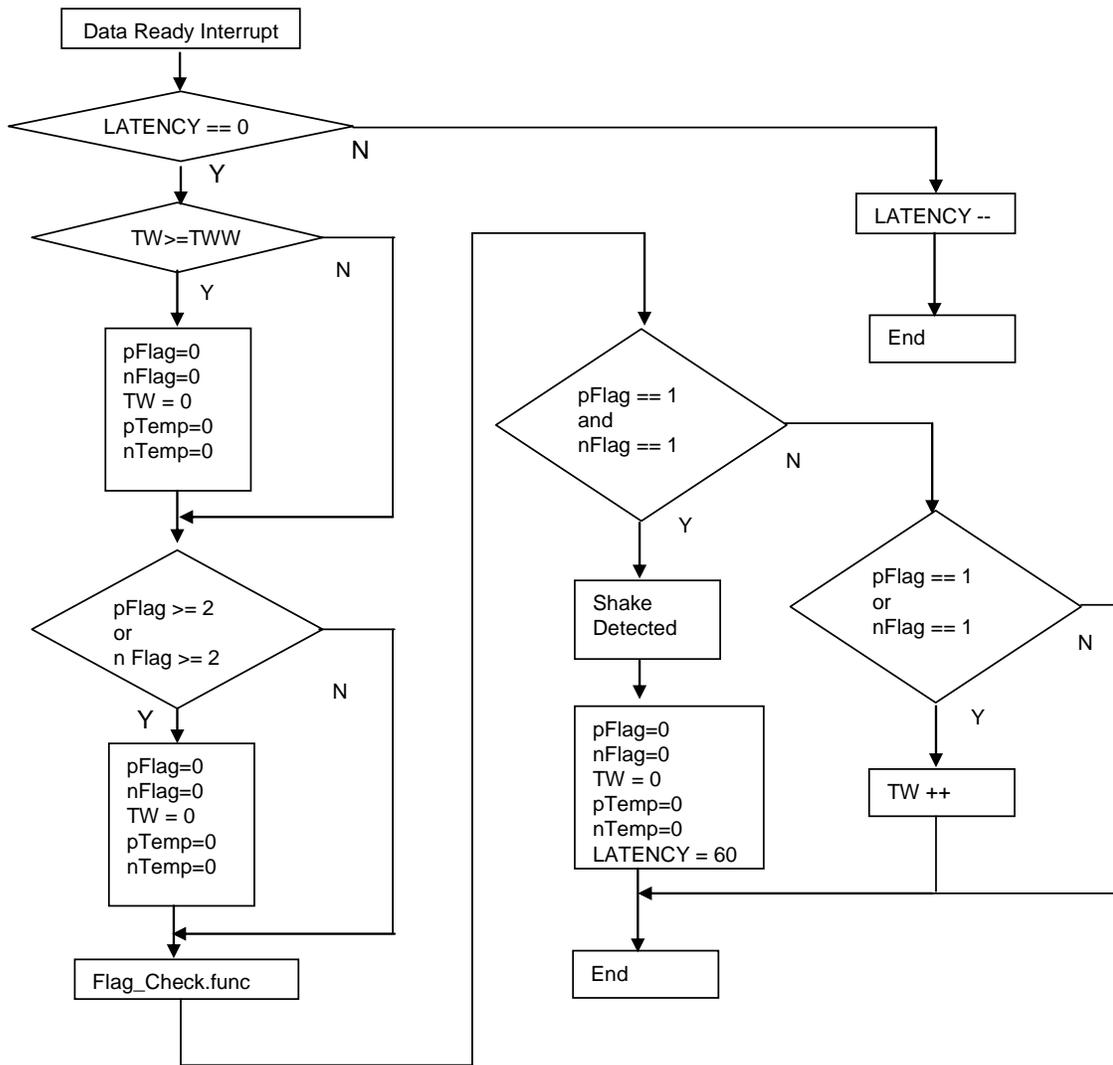


図 5. Shake アルゴリズム

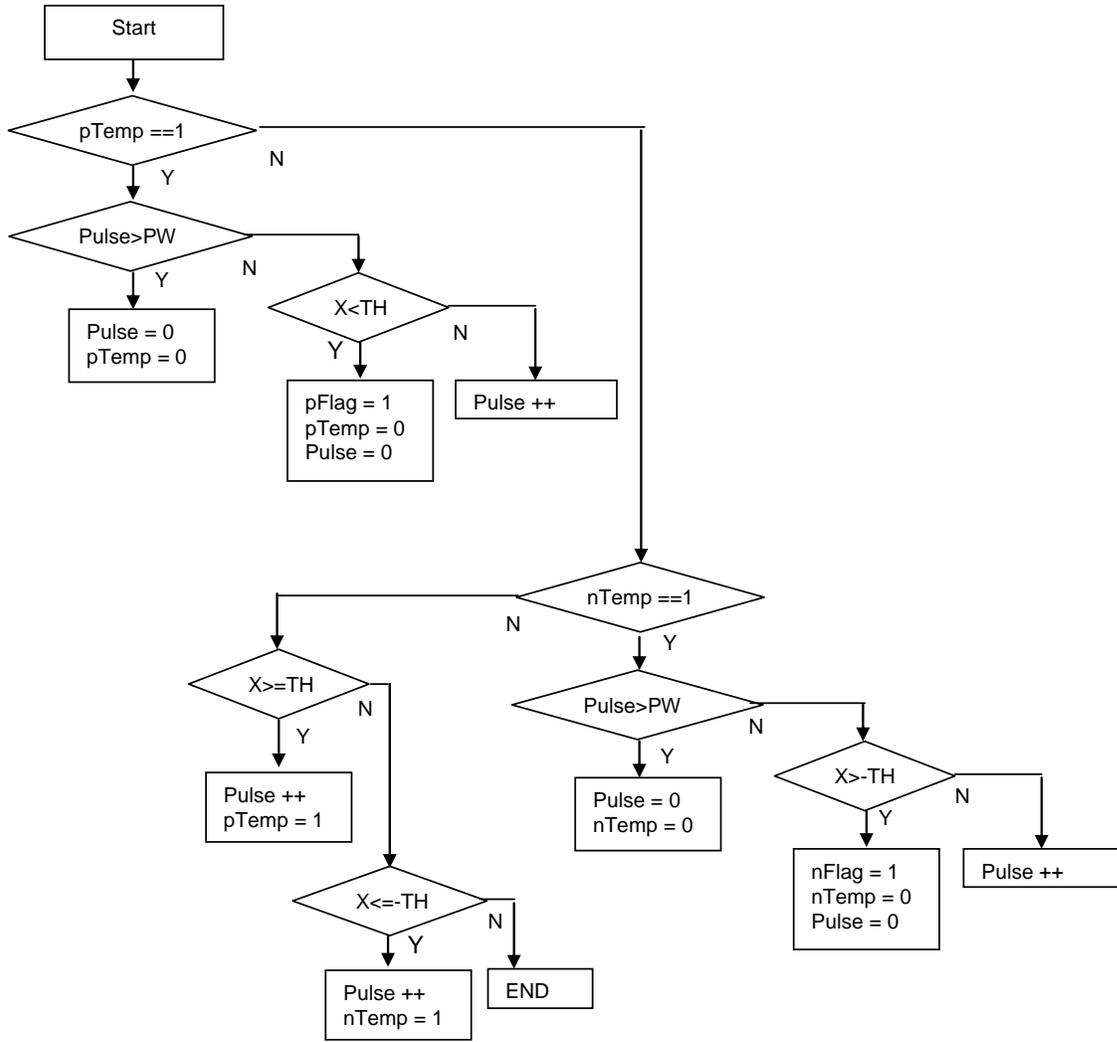


図 6. Pulse Detection 関数

NOTES

NOTES