

ADXL345 を用いた歩数計

by Tomoaki Tsuzuki

コンセプト

本アプリケーションノートに記載する歩数計のアルゴリズムは加速度センサーの設置向きによらず歩数をカウントできるアルゴリズムです。図 1 にアルゴリズムの概要を示します。

このアルゴリズムは正と負の極性のパルスを検出して歩数をカウントします。アルゴリズムは、3 軸分の加速度信号の RSS 値を計算し 32 データ分の移動平均を計算します。（ADXL345 の FIFO 機能を使用する事で、移動平均の計算は簡略化されます）RSS 値から移動平均の値を引いた値（この値を本アルゴリズムでは **SIGNAL** と呼びます）をパルスの検出に使用します。

SIGNAL が予め設定した閾値を超えた時点でタイマーの動作が開始します。タイマーで **SIGNAL** が閾値を下回るまでの時間を測定し、パルスの幅が設定した閾値の範囲

に収まっているかを確認します。パルスを検出した後に、アルゴリズムは反対極性のパルスを探します。一定時間内に反対極性のパルスが検出されれば、ステップとして認識します。ステップを認識した後は **LATENCY** と呼ぶ時間分だけデッドタイムを設けます。これは振動等によるノイズを排除する事が目的です。

最初の数ステップの検出では、アルゴリズムは歩行とは認識しません。**STEP_COUNTER** で設定したステップ数に一定時間内に達した時点でステップを歩行とみなします。一定時間内にステップが **STEP_COUNTER** の値に達しなかった場合は、それまでのステップは破棄されもう一度最初のステップを探します。またアルゴリズムはステップとステップの間の時間間隔を測定します。ステップとステップの間隔が長すぎたり短すぎたりした場合には、振動等によるミスカウントと判断し、それまでのステップを破棄します。

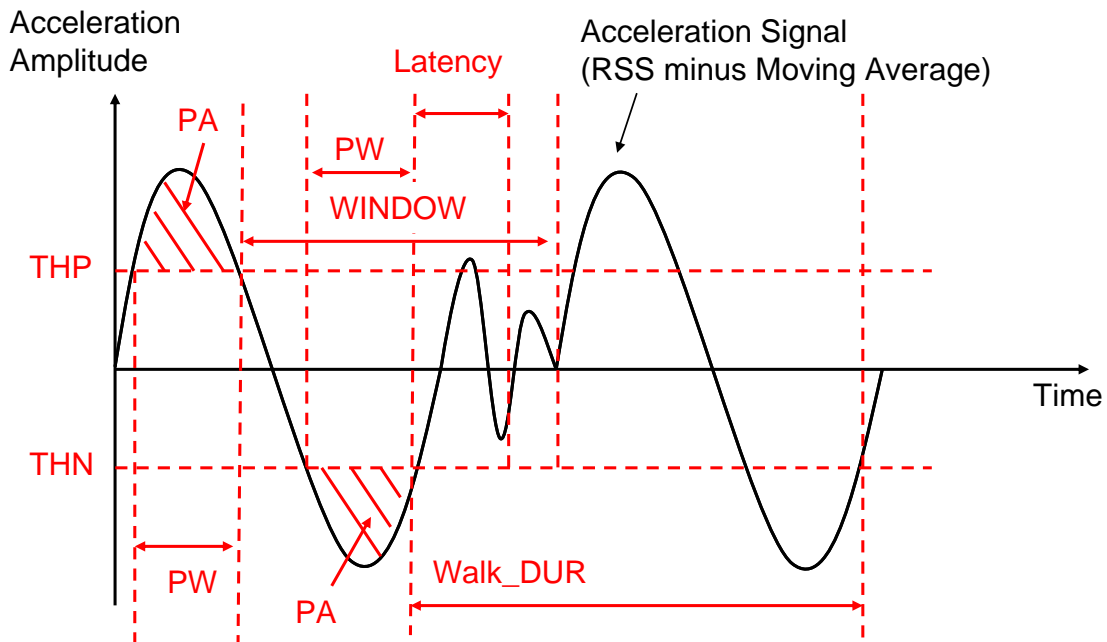


図 1. 歩数計アルゴリズム

Rev. 0

実装

本アルゴリズムは ADXL345 を表 1 のように設定しています。ADXL345 の Auto-sleep 機能は歩数計に最適です。動きが無い時は加速度センサーが自動的に低消費電力状態に移行するので、システム全体の不可が軽減されます。

表 2 の設定は Auto-sleep 機能を使用するための閾値の設定です。詳細は ADXL345 のデータシート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	0x1A	WM INT, Activity INT, Inactivity INT enabled
FIFO_CTL	0x38	5F	FIFO in FIFO mode, 32 samples

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. 閾値の設定

TABLE OF CONTENTS

コンセプト	1	歩数計アルゴリズム	5
実装	2		

図2にソフト処理全体の流れを示します。割り込み信号の発生を受けてプロセッサはINT_SOURCEレジスタを読みます。もしACTIVITY割り込みだった場合にはFIFOの内容をクリアして次の割り込みを待ちます。もし

INACTIVITY割り込みだった場合は、プロセッサは何もせずに次の割り込みを待ちます。もしWM割り込みだった場合にはプロセッサは歩数計のアルゴリズムを走らせます。

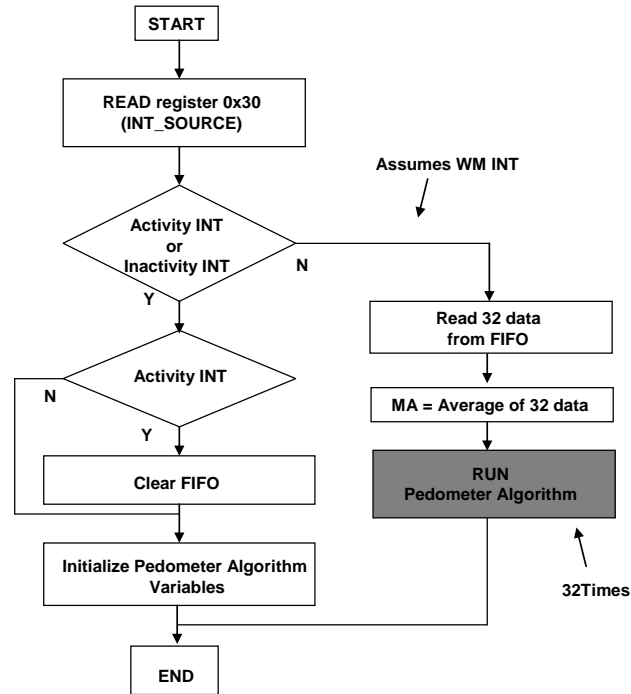


図2.ソフト処理全体のフローチャート

歩数計アルゴリズム

表3はアルゴリズムに使用する変数を示します。表4はアルゴリズムに使用する定数を示します。図3、図4、図5、図6、図7、図8に歩数計のアルゴリズムを示します。

Variable Name	Function	Initial State
Walk_Flag	1 when at walking state, 0 when at not walking	0
T_Walk_Flag	1 when at temporary walking state, 0 when at not temporary walking	0
RSS	RSS of three axis (unit is LSB)	0
MA	Average of 32 samples in FIFO	0
Signal	RSS minus MA	0
PA	Pulse area	0
PW	Pulse width	0
WINDOW	Shows how long time is passed after first pulse is detected	0
PPTEMP	1 when positive pulse detected and looking for negative pulse	0
NPTEMP	1 when negative pulse detected and looking for positive pulse	0
Pulse_Flag	0=looking for pulse, 1=positive pulse detected and looking for negative pulse, 2=negative pulse detected and looking for positive pulse, 4=in latency state	0
Latency	Time in which step count algorithm is not taken place after latest step is detected	0
WALK_DUR	Shows how long time is passed after latest step is detected	0
TEMP_STEP	Temporary step count. Cleared if WALK_DUR is above WALK_DUR_TH	0
STEP	Step counted	0

表3. 変数

Constants	Function	Initial Setting
Signal_TH	When signal is above this level, clear everything	2000
THP	Threshold for positive pulse	20
THN	Threshold for negative pulse	-20
PAMAX	Maximum pulse area	6000
PAMIN	Minimum pulse area	50
PWMAX	Maximum pulse width	80
PWMIN	Minimum pulse width	2
WINDOW_TH	Defines how long to look for second pulse after first pulses is detected	90
Time_Latency	Define how long Latency is	15
WALK_DUR_TH	Defines how long to look for another step after latest step is detected	100
TEMP_STEP_TH	Defines how many TEMP_STEP needed to recognize it as Walking	10

表4. 定数

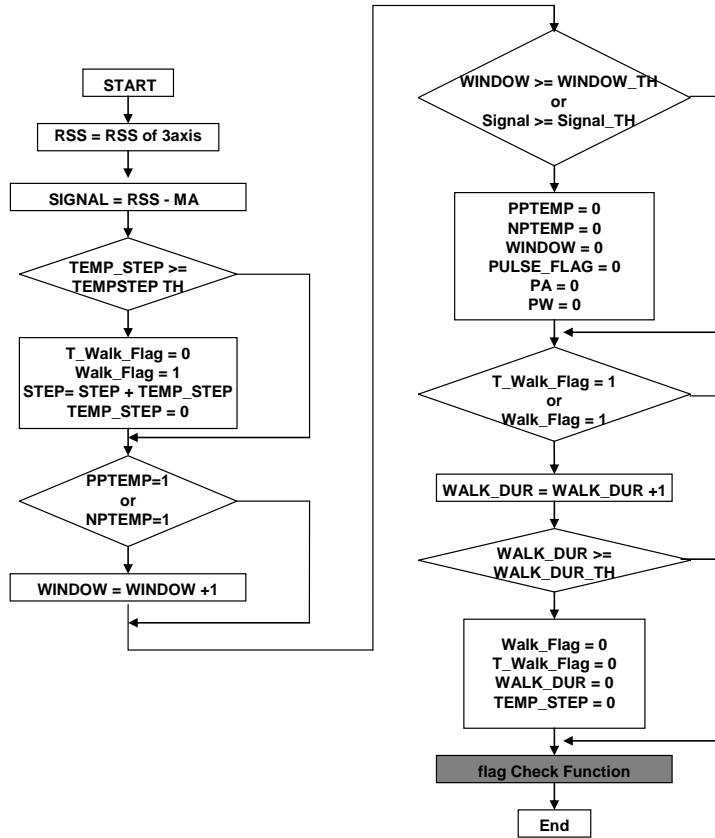


図 3. 歩数計アルゴリズム

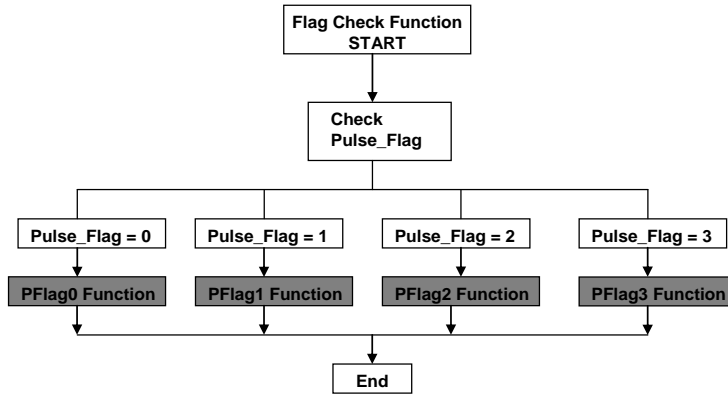


図 4. 歩数計アルゴリズム Flag Check 関数

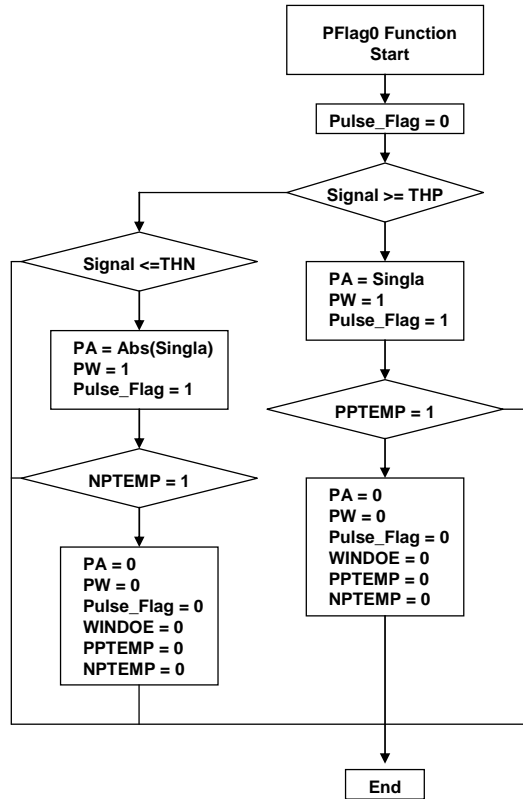


図 5. PFlag0 関数

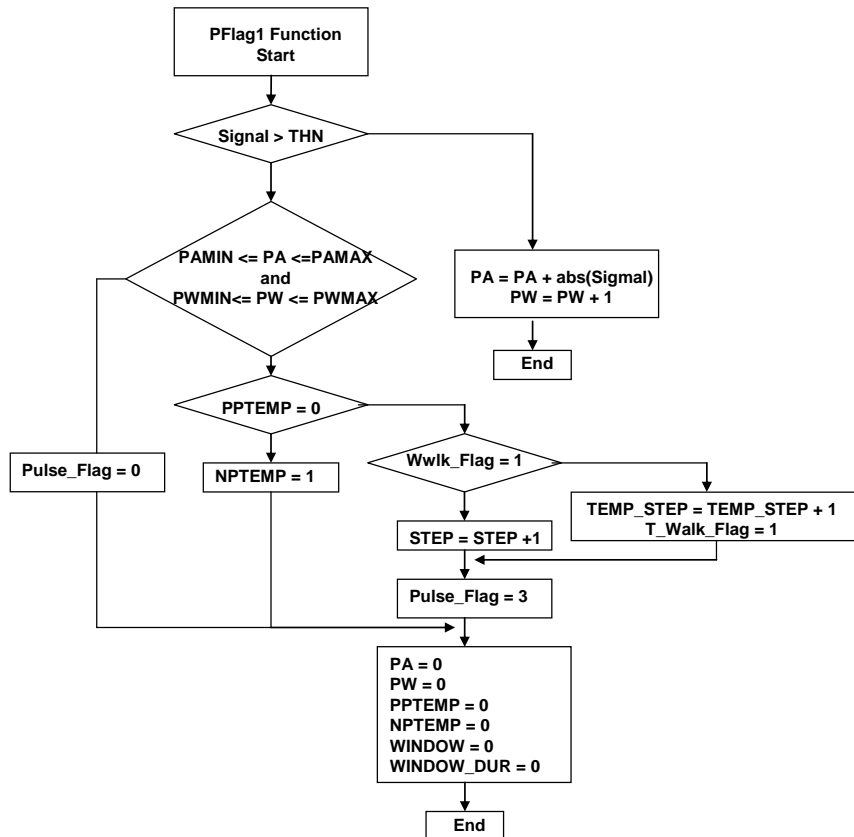


図 6. PFlag1 関数

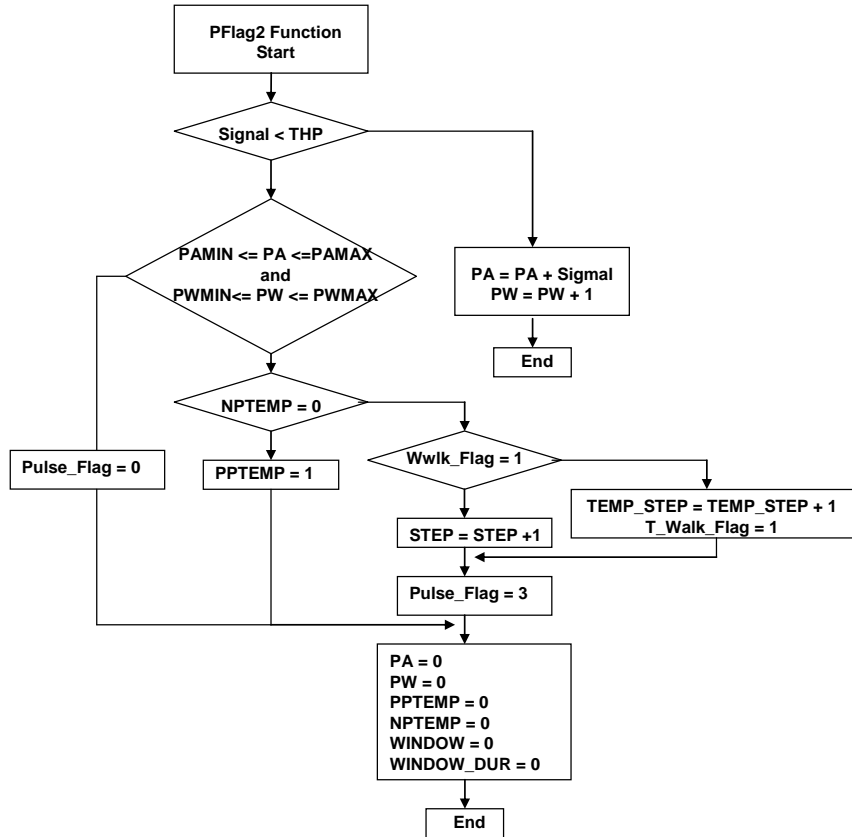


図 7. PFlag2 関数

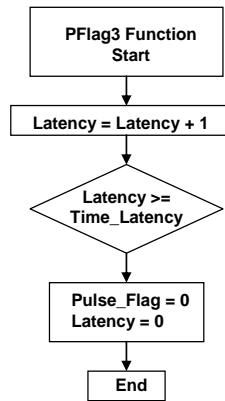


図 8. PFlag3 関数

NOTES

NOTES