

OMRON

MEMS フローセンサ

形 D6F-A7D/-AB71D

ユーザーズマニュアル

MEMSフローセンサ



CDSC-038B

目次

1	概要	2
2	商品ラインナップ	2
3	外形寸法（単位：mm）	3
3.1	本体	3
3.2	アクセサリ（別売）	4
4	フローセンサの原理と構造	5
4.1	原理	5
4.2	構造	5
5	製品の特長	6
5.1	直線補正と温度補正	6
5.2	デジタル出力と流量誤差	6
6	性能/定格	7
7	使用方法	8
7.1	D6F-□□□D-000-□との配管接続方法	8
7.2	D6F-□□□D-000-□との電気的接続方法	9
8	通信仕様	10
8.1	I ² C 通信概要	10
8.2	インターフェースコンフィギュレーションレジスタ	10
8.2.1	アクセスアドレスレジスタ (00h – 01h)	12
8.2.2	シリアルコントロールレジスタ (02h)	12
8.2.3	ライトバッファレジスタ (03h – 06h)	13
8.2.4	リードバッファレジスタ (07h – 0Ah)	13
8.2.5	初期化レジスタ (0Bh)	14
8.2.6	パワーシーケンスレジスタ (0Dh)	14
8.2.7	I ² C アクセスコマンド例	15
8.3	レジスタ詳細説明	16
8.3.1	センサ制御レジスタ (D040h)	16
8.3.2	CRC 演算制御レジスタ (D049h)	16
8.3.3	データレジスタ群 (D051h-D068h)	18
9	出力データの説明	19
9.1	データ・アライメント	19
9.2	レジスタ内容	19
9.3	センサデータ例	19
10	センサ動作モード・フローチャート	20
10.1	センサ動作における I ² C 命令	21
10.2	CRC 演算機能有効化方法	23
10.3	リセット実行方法	24
10.4	I ² C バス特性	25
11	用語説明	29
12	サンプルソースコード	30
12.1	D6F_D_Sample.h	30
12.2	D6F_D_Sample.c	31
13	ご承諾事項	36

1 概要

本ユーザーズマニュアルでは、弊社 MEMS デジタル出力フローセンサ (D6F-□□□□D-000-□) のご使用方法、特記事項などを示すものです。なお本資料は製品カタログを補足するものであり、実際のご使用にあたっては製品カタログも合わせてご使用下さい。

2 商品ラインナップ[°]

MEMS デジタル出力フローセンサ (D6F-□□□□D-000-□) のラインナップを表 1 に示します。

表 1 ラインナップ

流量範囲	継手タイプ	形式
0~10 L/min	クイック継手 (P10)	D6F-10A7D-000-0
0~20 L/min		D6F-20A7D-000-0
0~50 L/min		D6F-50A7D-000-0
0~70 L/min	クイック継手 (P14)	D6F-70AB71D-000-0

アクセサリ (別売)を表 2 に示します。

表 2 アクセサリ (別売)

種類	形式
ケーブル	D6F-CABLE3
クイックファスナ	D6F-FASTENER-P10
変換継手(P10 - タケノコΦ10mm)	D6F-PLG1

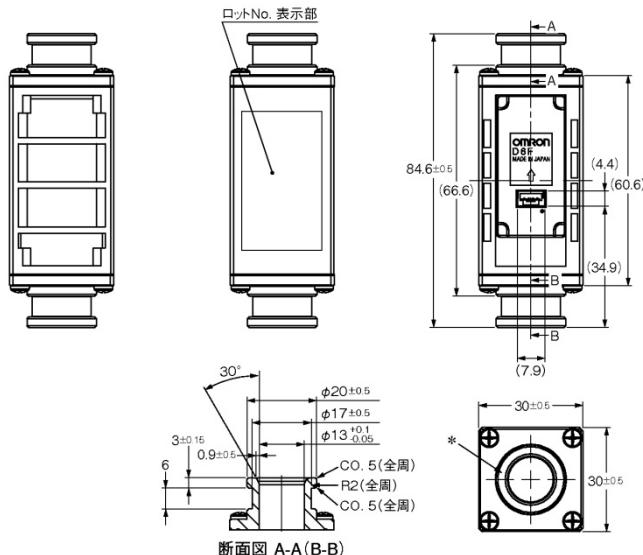
3 外形寸法 (単位: mm)

3.1 本体

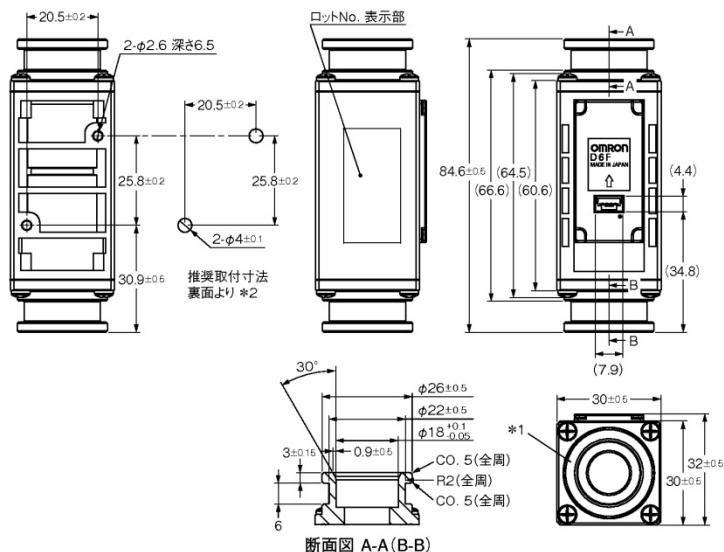
D6F-10A7D-000-0

D6F-20A7D-000-0

D6F-20A7D-000-0



D6F-70AB71D-000-0



*1 流量出入口の配管形状はクイック継手メス型 P10 または P14 に準拠する。

P10 または P14 とは JIS B 2401 に規定される O リングの呼び番号を示します。

オス型継手の O リング溝部形状は JIS B 2406 の P10 または P14 に準拠させてください。

*2 D6F-70AB71D-000-0 のφ2.6 穴を使用して本製品を固定する場合は呼び径 3mm の P タイプタッピングネジを使用し、締付トルク 1.2N・m 以下で締付けてください。ねじの嵌合長さは 5.5mm 以下してください。

本製品に接続する場合は以下のコネクタを使用してください。

コネクタ : GHR-04V-S (JST)

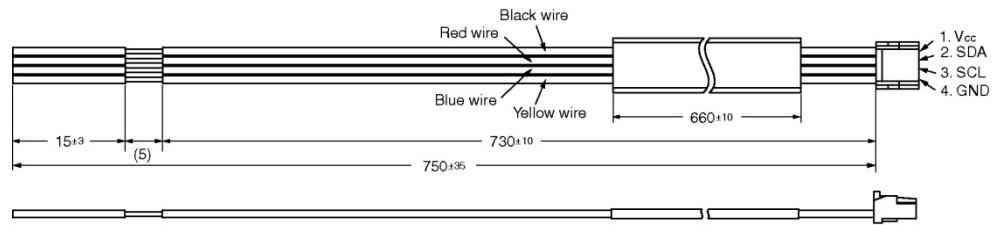
端子 : SSHL-002T-

導線 P0.2 (JST) : AWG26~30

3.2 アクセサリ (別売)

ケーブル

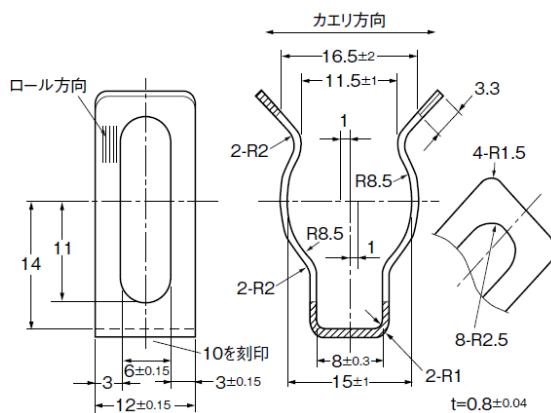
D6F-CABLE3



Contact :SSH-002T-P0.2 (Manufactured by J.S.T. Mfg. Co., Ltd.)
Housing :GHR-04V-S (Manufactured by J.S.T. Mfg. Co., Ltd.)
Wire :AWG#28

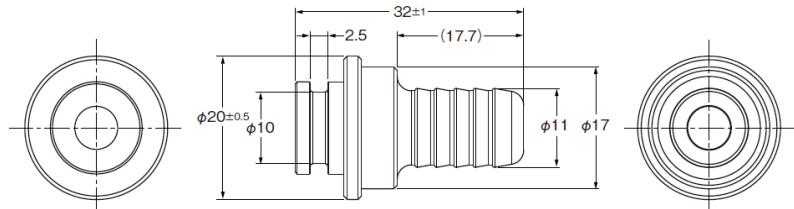
クイックファスナ

D6F-FASTENER-P10



変換継手

D6F-PLG1

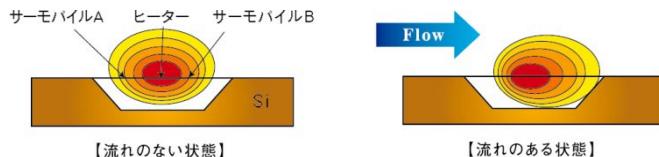


4 フローセンサの原理と構造

4.1 原理

MEMS デジタル出力フローセンサ (D6F-□□□□D-000-□) は熱式のマスフローセンサです。

シリコン基板上に形成された薄膜上にヒータが形成され、その両側にサーモバイルが配置されます。空気の流れにより、ヒータの上下流に設置されたサーモバイルの出力変化を流量として計測します。



4.2 構造

図 1 に MEMS デジタル出力フローセンサ (D6F-□□□□D-000-□) の内部断面図を示します。気体の流れはインレットからアウトレット方向に流れ、MEMS フローセンサチップの表面を気流が流れる形となります。

MEMS フローセンサチップはチップ表面を気流が通過する際に質量流量を測定します。より詳細な MEMS フローセンサチップに関する情報は、ユーザーズマニュアル(CDSC-023)をご覧下さい。

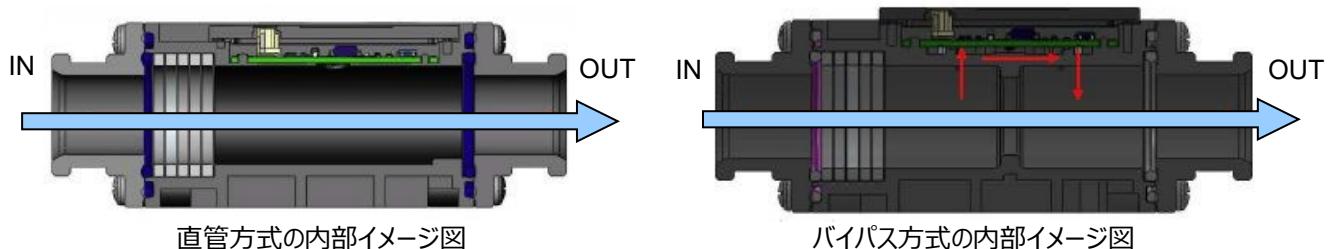


図 1 MEMS デジタルフローセンサの内部断面イメージ図

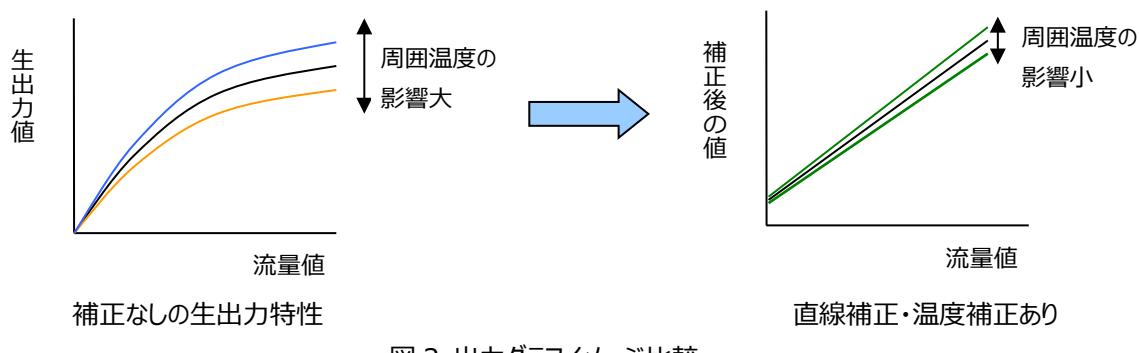
5 製品の特長

5.1 直線補正と温度補正

フローセンサチップからの生出力電圧値はガス流量の平方根にほぼ比例します。またフローセンサの生出力電圧値は周囲温度の影響を受けます。

このフローセンサは以下の補正を行う機能を実装しています。

- 1) 直線補正
- 2) 温度補正



5.2 デジタル出力と流量誤差

図 3 の左図はデジタルフローセンサの出力特性を示しています。このデジタルフローセンサは I²C インタフェースを通じて 16 ビットの測定流量値を出力します。上記の直線補正と温度補正を用いる事により、図 3 の右図の赤線に示す高精度な流量誤差を実現しています。

- $+/- 3 \%R.D : 25 - 100 \%F.S$
- $+/- 5 \%R.D : 10 - 25 \%F.S$

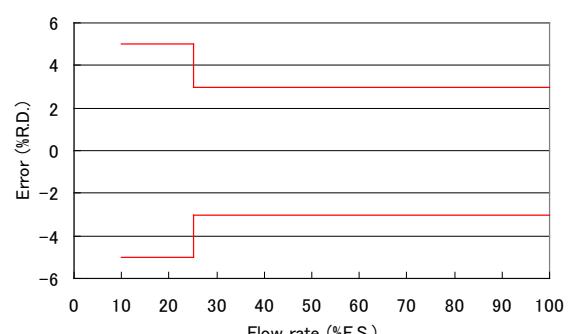
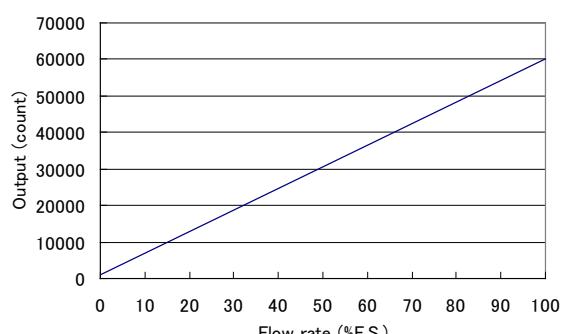


図 3 出力特性の例

6 性能/定格

形式	D6F-10A7D-000-0	D6F-20A7D-000-0	D6F-50A7D-000-0	D6F-70AB71D-000-0
流量範囲 (注 1)	0 to 10 L/min	0 to 20 L/min	0 to 50 L/min	0 to 70 L/min
適用媒体 (注 2)	Air			
継手形状	クイック継手 (P10)		クイック継手 (P14)	
端子仕様	4 端子コネクタ			
電源電圧	3.0 to 3.6 VDC			
分解能	15bit			
精度 (注 3)	$\pm 5\%RD$ (10%FS \leq 流量 $<$ 25%FS) $\pm 3\%RD$ (25%FS \leq 流量 \leq 100%FS)		$\pm 5\%RD$ (10L/min \leq 流量 $<$ 20L/min) $\pm 3\%RD$ (20L/min \leq 流量 \leq 70L/min)	
応答速度	90ms 以下			
再現性 (注 4)	0.3%RD	0.3%RD	0.5%RD	1.3%RD
通信方式	I ² C			
ケース材料	PPS			
保護構造	IEC IP40 (配管部を除く)			
最大許容耐圧	100kPa			
圧力損失 (注 4)	0.034kPa	0.083kPa	0.28kPa	0.57kPa
動作温度範囲 (注 6)	-10 to +60 °C			
動作湿度範囲 (注 6)	35 to 85%RH			
保存温度範囲 (注 6)	-30 to +80 °C			
保存湿度範囲 (注 6)	35 to 85%RH			
絶縁抵抗	センサ外壁とリード端子間: 20MΩ以上. (DC500V)			
耐電圧	センサ外壁とリード端子間: 500VAC, 50/60Hz min, 1 分間 (リーク電流: 1mA 以下)			
質量	57.3 g		64.4 g	

注 1 : 0°C, 101.3kPa の体積流量

注 2 : ダスト、オイルミストを含まない乾燥、清浄気体であること

注 3 : -10°C \leq 周囲温度 \leq 60°C

注 4 : 参考値 (代表値)

注 6 : 氷結、結露しないこと

7 使用方法

7.1 D6F-□□□□D-000-□との配管接続方法

P10 や P14 のクイックファスナを用いることにより、図 4 に示すように弊社デジタルフローセンサと簡単に接続することができる。

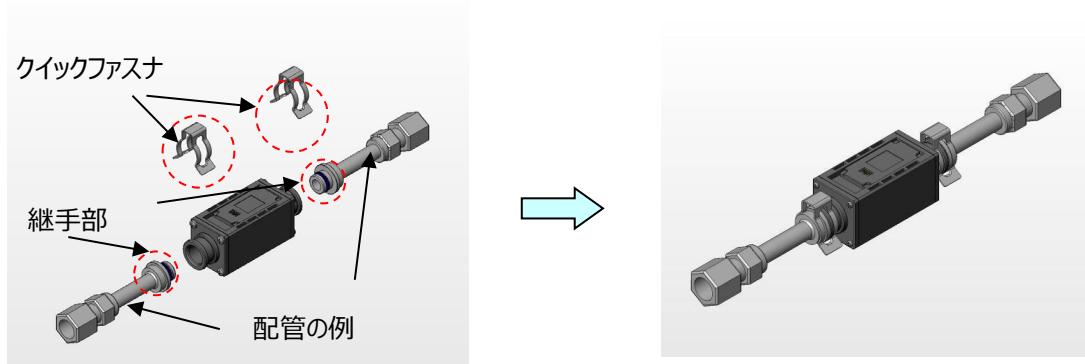
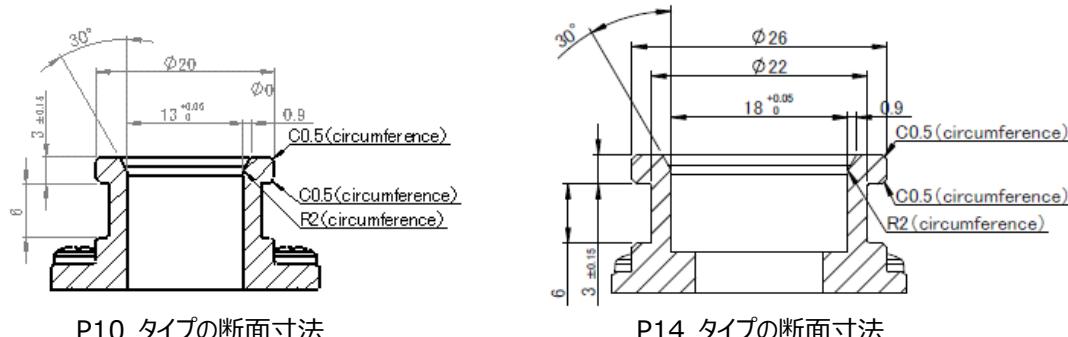


図 4 クイックファスを用いた配管接続例

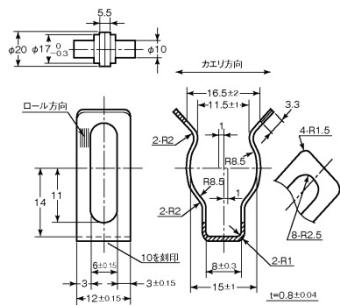
継手部と配管を接続する為のクイックファスナが使用できます。クイックファスナタイプはツールを用いずに手で配管の付け外しが可能です。現在、P10 と P14 の 2 種類のクイックファスナタイプをご用意しております。図 5 に P10 と P14 のクイックファスナの外形寸法図を示します。どちらのタイプのクイックファスナタイプをご使用になるかは、各々の製品データシートをご覧のうえご判断下さい。



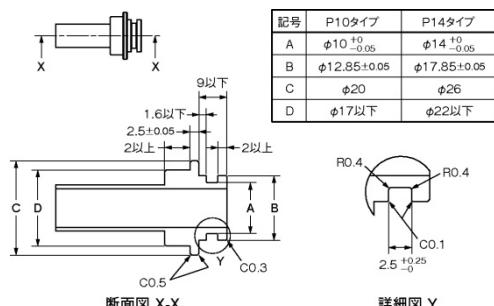
P10 タイプの断面寸法

P14 タイプの断面寸法

図 5 フローセンサ継手部の断面寸法図



ファスナの寸法図



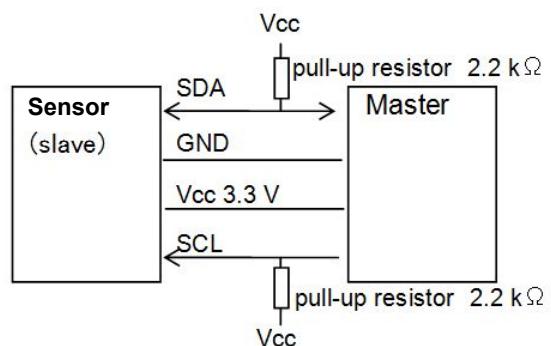
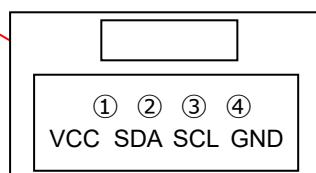
オス側継手寸法

図 6 P10 用クイックファスナの外形寸法図

7.2 D6F-□□□□D-000-□との電気的接続方法

デジタル I²C 出力方式の為、D6F-□□□□D-000-□と接続する際にはクロックライン（SCL）とデータライン（SDA）にプルアップ抵抗が各々必要となります。図 7 に示す様に、2.2 [kΩ]（推奨値）のプルアップ抵抗が VDD と各々の信号線の間に必要となります。

製品外観写真



コネクタ部ピン配置
図 7 コネクタ部のピン配置情報と電気的接続方法

※ フローセンサ接続に関するご注意点

アナログ出力タイプの製品に比べて、デジタル出力タイプ（I²C 通信出力）の製品はノイズ耐性が向上しておりますが、お客様でお使い頂く環境のノイズ影響により通信時にエラーが発生する可能性があります。その際には以下の点をご確認頂き、通信エラーを改善して頂く必要があります。

通信速度の確認

本製品は SCL 周波数が 400 kHz まで対応しておりますが、通信エラーが発生しやすい場合は、SCL 周波数を 100 kHz でご使用されることをお勧め致します。

配線ケーブルの確認

お客様制御マイコンと弊社フローセンサの間を接続するケーブル長が長い場合、よりノイズの影響を受けやすくなります。この場合はシールドケーブルのご使用をお勧め致します。

プルアップ抵抗値の確認

本製品の I²C 通信には、プルアップ抵抗が必要となります。推奨抵抗値は 2.2 [kΩ] しておりますが、お客様側制御マイコンと弊社フローセンサの間の接続ケーブル長に合わせて最適な抵抗値を選択して下さい。

なお、センサ側から ACK が返答されなければ、通信異常と判断して下さい。

ACK 返答時間は、SCL の 1 clock サイクル分になりますので、それ以上の時間 ACK 返答がない場合には通信異常となります。その場合は、一度電源を落としていただく必要があります。

8 通信仕様

8.1 I²C 通信概要

表 3 I²C 通信に関する基本機能

項目	説明
通信方式	I ² C
SCL 周波数	Max 400 kHz
出力フォーマット	Binary data (上位バイト、下位バイトの順)
スレーブアドレス	製品仕様書に記載 (例: 1101100b or 0x6C)

I²C スレーブアドレスは以下のように表現されます。(0x6C の例)

Bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
								R/W
値	1	1	0	1	1	0	0	1/0

Write 時 : スレーブアドレスの LSB を"0"にセットし、 D8h (1101_1000b)とする。

Read 時 : スレーブアドレスの LSB を"1"にセットし、 D9h (1101_1001b)とする。

8.2 インタフェースコンフィギュレーションレジスタ

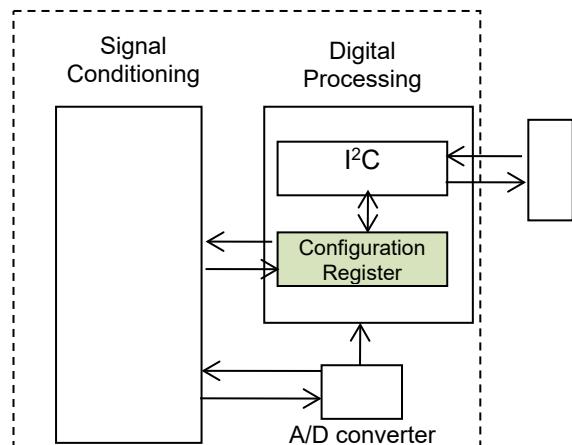


表 4 レジスタマップ

アドレス	レジスタ名	レジスタ概要
D040h	SENS_CTRL	センサ制御レジスタ
D049h	INT_CTRL	CRC 演算制御レジスタ
D051h	COMP_DATA1_H	補正流量値レジスタ
D052h	COMP_DATA1_L	
D061h	TMP_H	内部温度レジスタ
D062h	TMP_L	

図 8 コンフィギュレーションレジスタ概念図

D6F-□□□□D-000-□は、インターフェースコンフィギュレーションレジスタを介して実行されます。レジスタへアクセスする際には、インターフェースコンフィギュレーションレジスタの 00h と 01h にアクセス対象内部レジスタアドレスを書き込む必要があります。

インターフェースコンフィギュレーションレジスタ

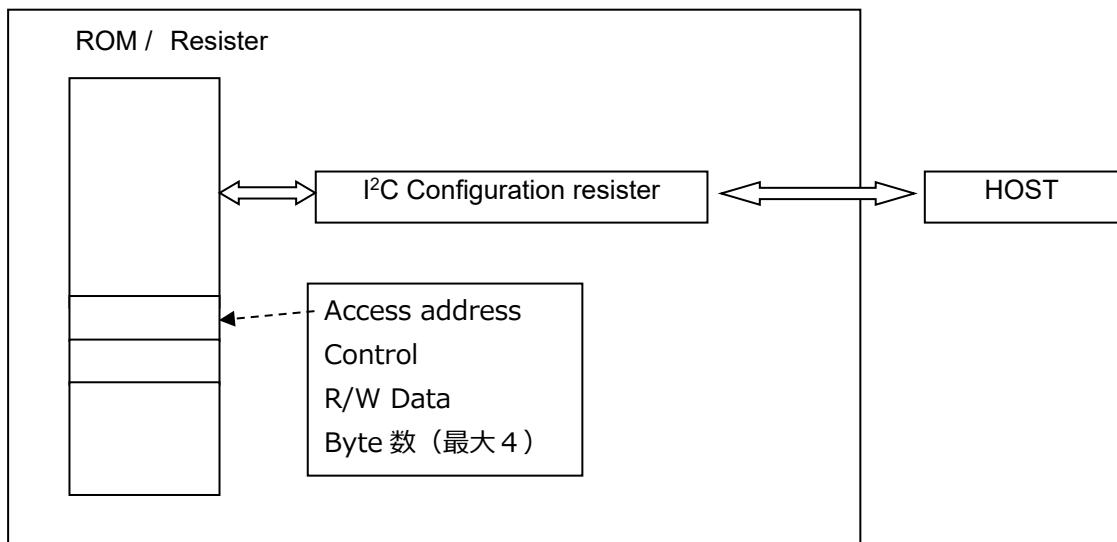


図 9 インターフェースコンフィグレーション

表 5 インターフェースコンフィギュレーションレジスター一覧

Configuration アドレス	機能	補足説明
00h	アクセスアドレス 1 (上位バイト)	最初のアクセスアドレスの上位バイト
01h	アクセスアドレス 2 (下位バイト)	最初のアクセスアドレスの下位バイト
02h	シリアルコントロール	ライト・リードアクセス制御
03h	ライトバッファ 0	アクセスアドレスに書かれるデータ
04h	ライトバッファ 1	アクセスアドレス + 1 に書かれるデータ
05h	ライトバッファ 2	アクセスアドレス + 2 に書かれるデータ
06h	ライトバッファ 3	アクセスアドレス + 3 に書かれるデータ
07h	リードバッファ 0	アクセスアドレスからの読み出しデータ
08h	リードバッファ 1	アクセスアドレス + 1 からの読み出しデータ
09h	リードバッファ 2	アクセスアドレス + 2 からの読み出しデータ
0Ah	リードバッファ 2	アクセスアドレス + 2 からの読み出しデータ
0Bh	初期化レジスタ	
0Dh	パワーシーケンス	ハードウェアリセット機能制御

上位バイト : 16 ビットデータの内ビット[15:8]を示す。下位バイト : 16 ビットデータの内ビット[7:0]を示す。

8.2.1 アクセスアドレスレジスタ (00h – 01h)

アクセスレジスタアドレスは、センサレジスタマップ、ADC レジスタマップ、内部メモリマップを含む内部レジスタブロックにアクセスする為に使用します。このレジスタは複数バイト転送時にはアドレスが自動インクリメントされるので、スタートアドレスを指定します。

表 6 アクセスアドレスレジスタ

アドレス	MSB D7	D6	D5	D4		D3	D2	D1	LSB D0
00h	A15	A14	A13	A12		A11	A10	A9	A8
01h	A7	A6	A5	A4		A3	A2	A1	A0

8.2.2 シリアルコントロールレジスタ (02h)

シリアルコントロールレジスタには様々なシリアルアクセスを制御するビットがあります。

表 7 シリアルコントロールレジスタ (02h)

アドレス	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
02h	D_byte_ cnt[3]	D_byte_ cnt[2]	D_byte_ cnt[1]	D_byte_ cnt[0]	Req	R_WZ	Acc_ctl[1]	Acc_ctl[0]

- Acc_ctl [1 : 0] – アクセスコントロールビット:
 - 0 0 = 16 ビットアドレスアクセス(A15-A0) (内部 ROM やレジスタなど)
 - 0 1 = reserved
 - 1 0 = reserved
 - 1 1 = reserved
- R_WZ – リード/ライト 選択ビット
 - 0 = ライトアクセス
 - 1 = リードアクセス
- Req- リクエストビット
 - 0 = 直前のアクセスリクエストは完了
 - 1 = 新規リクエスト 内部のシリアルバスブリッジ制御回路がリクエスト処理を完了すると、このReq フラグを"0"にクリアします。ライトアクセスリクエストの場合、内部バスブリッジ制御回路はライトバッファ内のデータをアクセスアドレスで指定されたレジスタに転送します。リードアクセスの場合、内部バスブリッジ制御回路は指定されたアドレスのデータをリードバッファに格納します。
- D_byte_cnt [3 : 0]
 - 転送バイト数指定 1, 2, 3, 4 バイト転送のみサポートします。

8.2.3 ライトバッファレジスタ (03h – 06h)

アドレス 03h – 06h が 4 つのライトバッファとなります。

ホストマイコンは以下の 2 通りの方法で 1 バイトをレジスタに書き込む事ができます。

ホストマイコンから 00h をスタートアドレスとして、コンフィギュレーションレジスタに対して

A[15:8]、A[7:0]、18h、data[0]とバースト書き込みすることで data[0]を Address=A[15:0] のレジスタに書き込む事が出来ます。

もしくはシリアルコンフィギュレーションレジスタに以下の 4 回実行する事で 1 バイト書き込みアクセス可能です。

- ・シリアルコンフィギュレーションレジスタのアドレス 00h にアクセスアドレスの A[15:8]をライト
 - ・シリアルコンフィギュレーションレジスタのアドレス 01h にアクセスアドレスの A[7:0]をライト
 - ・シリアルコンフィギュレーションレジスタのアドレス 03h にライトデータの data[0]をライト
 - ・シリアルコンフィギュレーションレジスタのアドレス 02h に 18h をライト(1 バイトの新規書き込み)
- [補足] シリアルコンフィギュレーションレジスタの 02h をリードすると、ビット[3]が"0"ならライトアクセス完了を意味します。

詳細はセクション 8.2.7 I²C アクセスコマンド例をご覧ください。

8.2.4 リードバッファレジスタ (07h – 0Ah)

アドレス 07h – 0Ah が 4 つのリードバッファとなります。

ホストマイコンは以下の 2 通りの方法で内部レジスタ値に読み出す事ができます。

内部シリアルバスコントローラのリードリクエスト処理が完了すると、リクエストフラグが"0"クリアされ読み出し対象レジスタの値がコンフィギュレーションレジスタのアドレス 07h – 0Ah にあるリードバッファに格納されます。

1 バイトリードの場合、ホストマイコンから 00h をスタートアドレスとして、コンフィギュレーションレジスタに対して A[15:8]、A[7:0]、1Ch とバースト書き込みする事で読み出しリクエストを発行します。その後ホストマイコンはリクエストビットが"0"クリアされるのを待って、コンフィギュレーションレジスタの 07h アドレスにあるリードバッファを読み出すと読み出し対象レジスタの値を読み出せます。

もしくはシリアルコンフィギュレーションレジスタに以下の 4 回実行する事でも 1 バイト読み出し可能です。

- ・シリアルコンフィギュレーションレジスタのアドレス 00h にアクセスアドレスの A[15:8]をライト
 - ・シリアルコンフィギュレーションレジスタのアドレス 01h にアクセスアドレスの A[7:0]をライト
 - ・シリアルコンフィギュレーションレジスタのアドレス 02h に 1ch をライト(1 バイトの新規読み出し)
- [補足] シリアルコンフィギュレーションレジスタの 02h をリードすると、ビット[3]が"0"ならリードアクセス完了を意味し、07h をリードすれば読み出し対象レジスタの値を読み出せます。

詳細はセクション 8.2.7 I²C アクセスコマンド例をご覧ください。

8.2.5 初期化レジスタ (0Bh)

電源投入後の初期化処理。NVM のトリミングデータをロードする為に、実行必須です。

- 初期化レジスタ(0Bh)に 00h をライト

詳細はセクション 8.2.7 I²C アクセスコマンド例をご覧ください。

8.2.6 パワーシーケンスレジスタ (0Dh)

センサの状態を監視する表示します。

表 8 パワーシーケンスレジスタ

アドレス	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
0Dh	Hard_R eset	ADC_st ate	ADC_st ate	ADC_st ate	Pwr_seq _state5	Pwr_seq _state5	Pwr_seq _state5	Pwr_seq _state5

表 9 パワーシーケンス状態の説明

Bit	Name	R/W	Description
[3:0]	Pwr_seq_state5	R	パワーシーケンスのステート状態値を示します。デバッグ目的で使用するものです。 h0(0000b): Idle h2(0010b): Active h9(1001b): Execute
[6:4]	ADC_state	R	ADC を制御するステート状態
[7]	Hard_Reset	R/W	1 -> ハードウェアリセットを実行（実行後自動クリア） 0 -> ハードウェアリセットを実行しない。

初期化処理後、あるいは電源リセット時には h0
D040 に 06h を書きこむと h2
演算中は h9
となります。

ハードウェアリセットが実行された場合、ハードウェアリセットビットはリセット実行後に自動で"0"クリアされ、内部レジスタはデフォルト値に戻り、内部トリミング値は不揮発性メモリからリロードされます。このハードウェアリセット機能は、電源リセットと同様の機能となります。

注記

- このリセットビットにアクセスする際には、デバイスをアンロックしたりアクセスする為のパスワードは不要です。
- ハードウェアリセット使用時、0～6 ビットは 0 としてください。

8.2.7 I²C アクセスコマンド例

以下に I²C アクセスのコマンド例を示します。青く着色された ACK と読み出しデータ部分は、フローセンサの出力データとなります。その他の着色されていない部分はホストマイコンが制御する部分となります。

I²C Command Examples

・I²C コマンド : I²C write

コンフィギュレーション

レジスタのスタートアドレス

コンフィギュレーションレジスタの

“00h”と“01h”にセットされるアドレス

START	Slave Address	ACK	Access Address	ACK	Reg Address H	ACK	Reg Address L	ACK
S	D8h (6Ch (7b)+ 0)	A	00h	A	D0h	A	40h	A

コンフィギュレーションレジスタの
“02h”と“03h”にセットされるデータ

・I²C コマンド : I²C read

コンフィギュレーションレジスタの

アドレス(Read Buffer 0 を指定)

START	Slave Address	ACK	Access Address	ACK
S	D8h (6Ch (7b)+ 0)	A	07h	A

コンフィギュレーションレジスタの
“07h”と“08h”からの読み出し値
(Read Buffer 0 / Read Buffer 1)

Re-Start	Slave Address	ACK	Read Data H	ACK	Read Data L	ACK	STOP
RS	D9h (6Ch (7b)+ 1)	A	xxh	A	xxh	NA	P

8.3 レジスタ詳細説明

センサモジュールの内部メモリやレジスタアクセスは、コンフィギュレーションレジスタを介してアクセスされます。

注記：NoneとかかれたビットはDefault値としてください。

8.3.1 センサ制御レジスタ (D040h)

表 10 SENS_CTRL レジスタ

アドレス	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
D040h						MS	DV_PWR [1]	DV_PWR [0]
Write Access	None	None	None	None	None	Host & MCU	Host & MCU	Host & MCU
Default	0	0	0	0	0	0	0	0

- DV_PWR[1 : 0] – メインデバイスパワーモード設定
 - 0 0 = スタンバイ – 全てのブロックはパワーダウン状態です。
 - 1 0 = MCU オン – MCU ブロックを動作時に必要です。 アナログ部、メモリ部への電源がオング状態となり、MCU へのクロック供給が開始されます。
- 補足：このレジスタは測定中に状態を変更しないで下さい。
- MS – MCU スタート – DV_PWR の状態に応じて、MCU モードを実行します。
 - 0 = ストップ 測定処理シーケンスがストップし、各ブロックが電源オフ状態となります。
 - 1 = スタート MCU のクロック供給が開始され、設定に応じた MCU モードが実行されます。

8.3.2 CRC 演算制御レジスタ (D049h)

表 11 INT_CTRL

アドレス	MSB D7	D6	D5	D4	D3	D2	D1	LSB D0
D049h							CRC_EN	
Write Access	NONE	NONE	NONE	NONE	NONE	NONE	Host & MCU	NONE
Default	0	0	0	0	0	0	1	0

- CRC_EN – CRC チェック機能選択 (CRC 演算の詳細は以下を参照して下さい。)
 - 0 = CRC チェック演算機能を無効にする
 - 1 = CRC チェック演算機能を有効にする

CRC 演算に関する補足説明

・CRC 概要

CRC はデータ通信におけるエラー検出方法として用いられます。弊社フローセンサには CRC8 多項式 $x^8 + x^5 + x^4 + 1$ を用いています。CRC 機能を有効とした時の 2 バイト読み出しの I²C アクセス例を以下に示します。

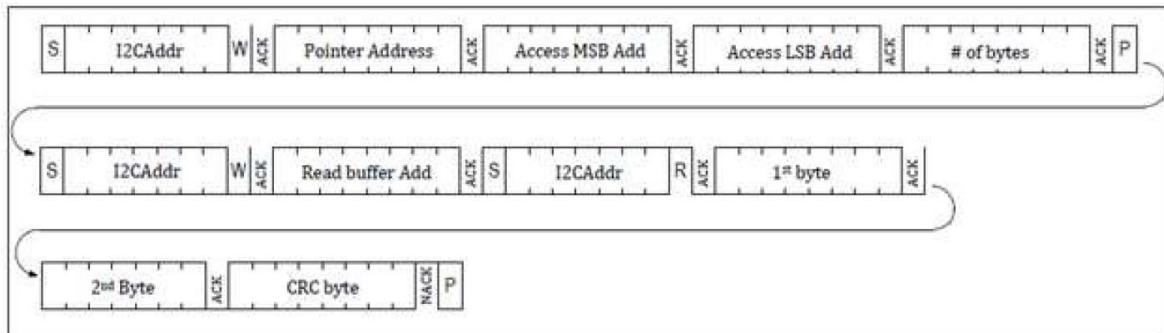


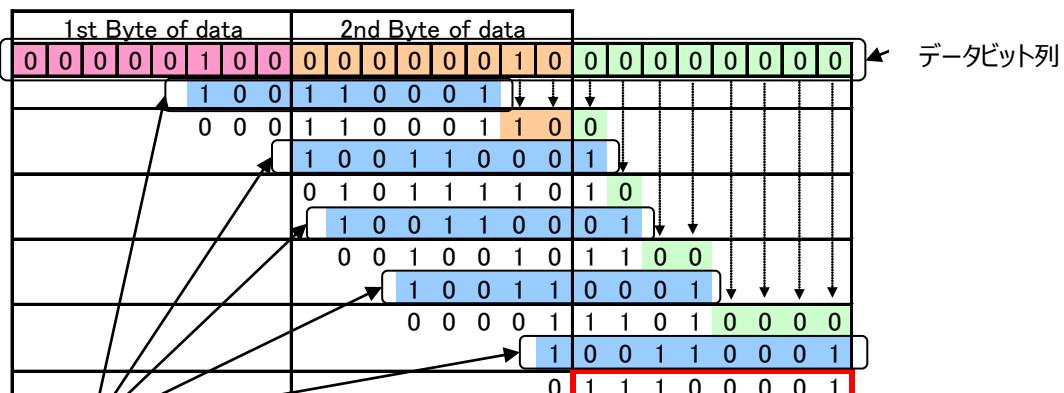
図 10 CRC 機能を有効とした時の 2 バイトデータ + CRC バイト読み出し例

・ビット単位 CRC-8 計算方法

1. データビット列を一列に並べます。
2. 多項式ビット列を、データビット列の下に並べます。
3. データビット列の一番左側のビットが“0”ならば、多項式ビット列を 1 ビット右にシフトします。データビット列の一番左側のビットが“1”ならば、多項式ビット列とで XOR 演算を実行します。その後多項式ビット列を 1 ビット右にシフトします。
4. 上記 1.～3. のステップを多項式ビット列がデータビット列の右端に到達するまで繰り返します。

以下に、XOR 演算に基づく CRC バイト算出例を示します。

	hex	bin
1st Byte of data	04h	00000100
2nd Byte of data	02h	00000010
Polynomial ($x^8 + x^5 + x^4 + 1$)	131h	100110001
CRC-byte checksum	225h	11100001



多項式ビット列シーケンス

図 11 CRC-8 での XOR 演算例

8.3.3 データレジスタ群 (D051h-D068h)

表 12 16 ビット データレジスタマップ

アドレス	レジスタ名称	MSB	D7	D6	D5	D4	D3	D2	D1	LSB	D0	説明
D051h	COMP_DATA1_H											補正後流量データ
D052h	COMP_DATA1_L											
D061h	TMP_H											内部温度センサ
D062h	TMP_L											

詳細はセクション 9. 出力データ説明をご覧下さい。

9 出力データの説明

測定値は予め指定されたレジスタに格納されます。測定されたデータは専用レジスタに格納されます。各専用レジスタは各々測定対象センサ出力信号を上位バイトと下位バイトに分けて格納します。

例えば補正後の流量値は、COMP_DATA1_HとCOMP_DATA1_Lをビット連接した符号無し16ビットとなります。

9.1 データ・アライメント

測定データは16ビットデータとして表現されます。またこのデータは連続する2バイトデータとしてビッグエンディアンフォーマットでフローセンサ・レジスタ群に格納されます。

ビッグエンディアンフォーマットとは、データの上位側バイトは下位側アドレスのレジスタに格納され、データの下位側バイトは上位側アドレスのレジスタに格納されます。

9.2 レジスタ内容

- COMP_DATA1_H と COMP_DATA1_L [D051h – D052h] : 補正後の流量データ (符号無し)
これらのレジスタには選択された補正方式の流量データが格納されています。

$$Fc [L/min] = (Fv - 1024) * RANGE / 60000$$

ここで、Fcは換算後の流量データ (L/min) で、Fvは流量データレジスタに格納される値です。

- TMP_H と TMP_L [D061h – D062h] : 温度データ (符号付き)

これらのレジスタにはASIC内部温度センサの温度データが格納されています。

レジスタの読み値から温度データへの換算には以下の式を用いてください。

$$Tv [^{\circ}C] = (Rv - 10214) / 37.39$$

ここで、Tvは換算後の温度データ (°C) で、Rvは温度データレジスタに格納される値です。

注記: 温度データは参考値としてのみご使用下さい。温度精度に関しては仕様上保証されていません。

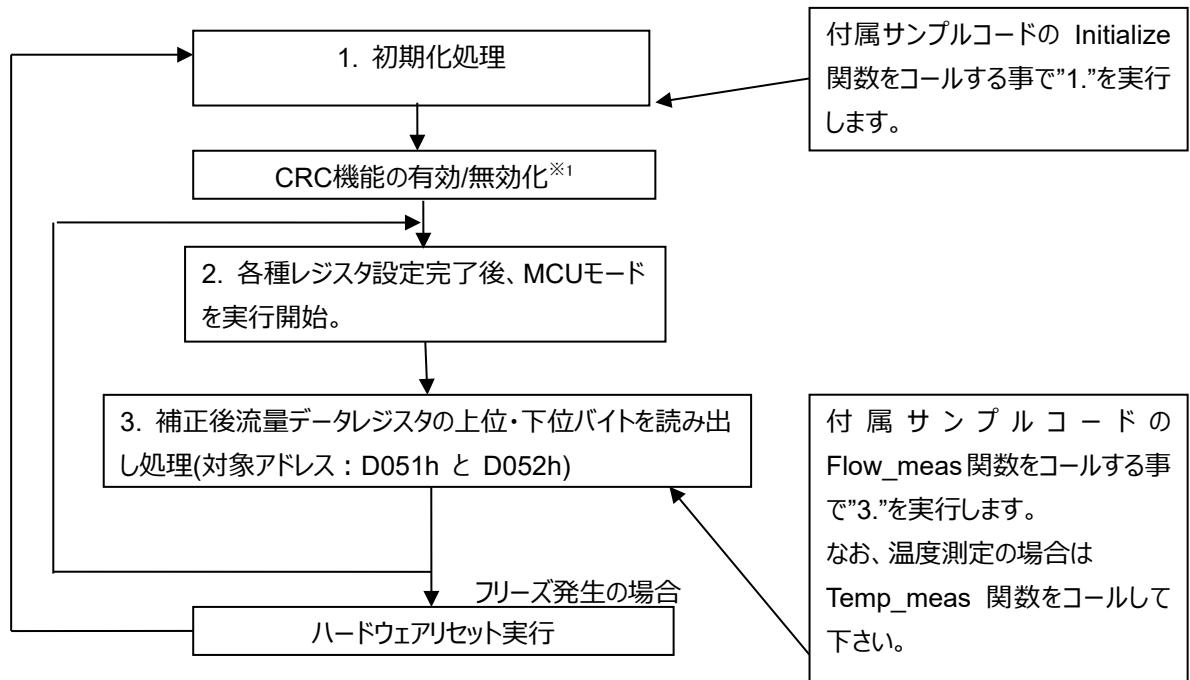
9.3 センサデータ例

以下の表は、ある流量と温度におけるフローセンサ・レジスタ読み値の例を表したものです。この表に記載された値は、デバイスで完全に補正が行われた前提ものです。(つまり、オフセットなし、ゲインエラーなし)

表 13 温度データレジスタ値と温度値比較

レジスタアドレス		レジスタ値		換算後の温度
TMP_H	TMP_L	HEX	DEC	
D061h	D062h	2B8Dh	11149	25.0 °C
2Eh	FFh	2EFFh	12031	48.6 °C
26h	BBh	26BBh	9915	-8.0 °C

10 センサ動作モード・フローチャート



*1 もしCRC演算機能をお使いになる場合、ページ25を参考にして制御コマンド発行して下さい。

図 12 フローセンサ動作モード・フローチャート

・通信待ち時間に関して

項目	記号	補足事項
応答時間	α	$\alpha \geq 120 \text{ ms}$
サンプリング間隔	β	$\beta > \alpha$

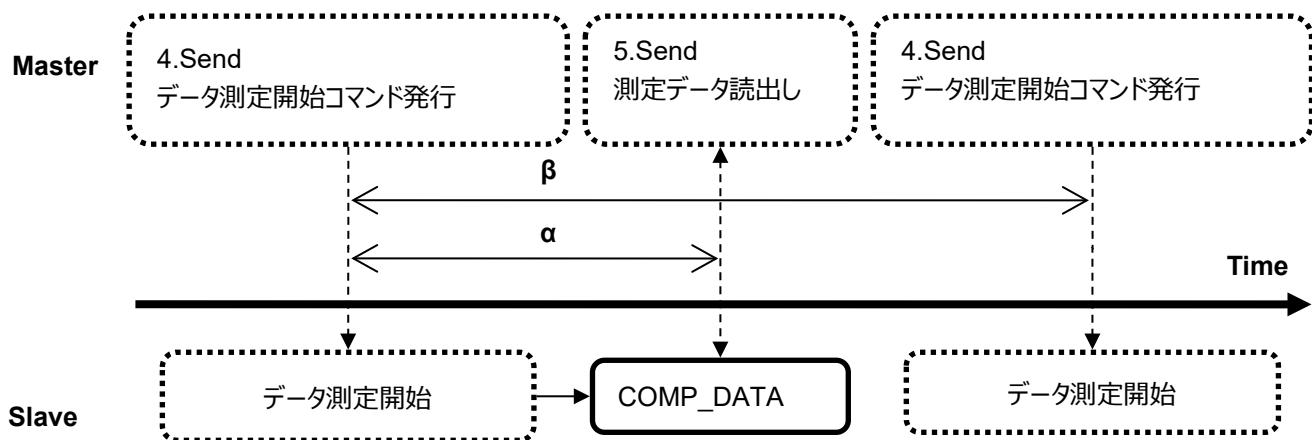


図 13 通信待ち時間関係イメージ図

10.1 センサ動作における I²C 命令

電源投入後の初期化処理 [実行必須です]

I²C コマンド: NVM のトリミングデータをロードする為に、初期化レジスタ(0Bh)に 00h をライト、

しかし MCU は非リセット状態を保持して下さい。

本コマンドの発行は電源投入後 200μs 後に実行してください。

表 14 初期化処理

START	Slave Address	ACK	Access Address	ACK	Write Data	ACK	STOP
S	D8h (6Ch (7b)+ 0)	A	0Bh	A	00h	A	P

* もし CRC 演算機能をお使いになる場合、ページ 25 を参考にして制御コマンド発行して下さい。

各種レジスタ設定完了後、MCU モード実行開始

センサコントロールレジスタ (D040h) に 06h を書き込むことで、セクション 6 で概要を示した補正などを実行する為に必要な各種設定と共に MCU モードが実行されます。センサコントロールレジスタに 06h を書き込み後にこのレジスタを読み出すことで、MCU で選択された MUX の状態を読み出す事が出来ます。処理実行後に、MS ビットが"0"にセットされます。

[注意]: MCU 動作中はデバイスにアクセスしないで下さい。120msec 経過後にライト・リードアクセスすれば大丈夫です。

I²C コマンド: センサコントロールレジスタ (D040h) に 06h 書き込み (MS=1 & MCU_on)

表 15 MCU モード実行処理

START	Slave Address	ACK	Access Address	ACK	Reg Address H	ACK	Reg Address L	ACK
S	D8h (6Ch (7b)+ 0)	A	00h	A	D0h	A	40h	A

Serial Ctrl	ACK	Write Data	ACK	STOP
18h	A	06h	A	P

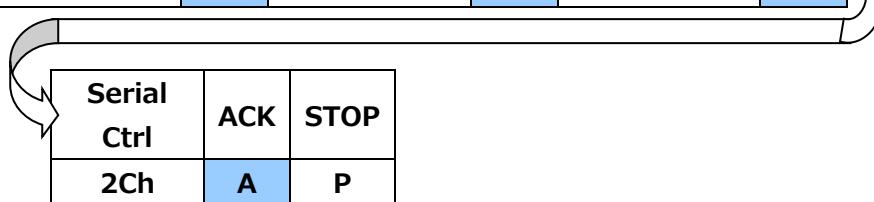
補正後流量レジスタの上位・下位バイトデータレジスタの読み出し(D051h and D052h)

初期化処理後、最初のデータは流量データではありません。無効処理をしてください。

I²C コマンド: 補正後の流量データレジスタ (D051hとD052h) を読み出す為に、シリアルコンフィギュレーションレジスタ (アドレス2h) に2Ch (2バイト読み出しアクセス) を書き込み。

表 16 補正後流量データの読み出し (Step1)

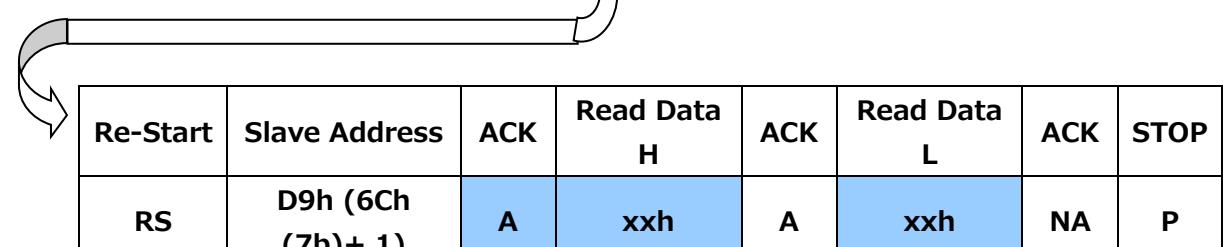
START	Slave Address	ACK	Access Address	ACK	Reg Address H	ACK	Reg Address L	ACK
S	D8h (6Ch (7b)+ 0)	A	00h	A	D0h	A	51h	A



I²C コマンド: リードバッファ 0 (アドレス 7h) とリードバッファ 1 (アドレス 8h) を介して補正後の流量データレジスタ 2 バイトを読み出す。

表 17 補正後流量データの読み出し (Step2)

START	Slave Address	ACK	Access Address	ACK				
S	D8h (6Ch (7b)+ 0)	A	07h	A				

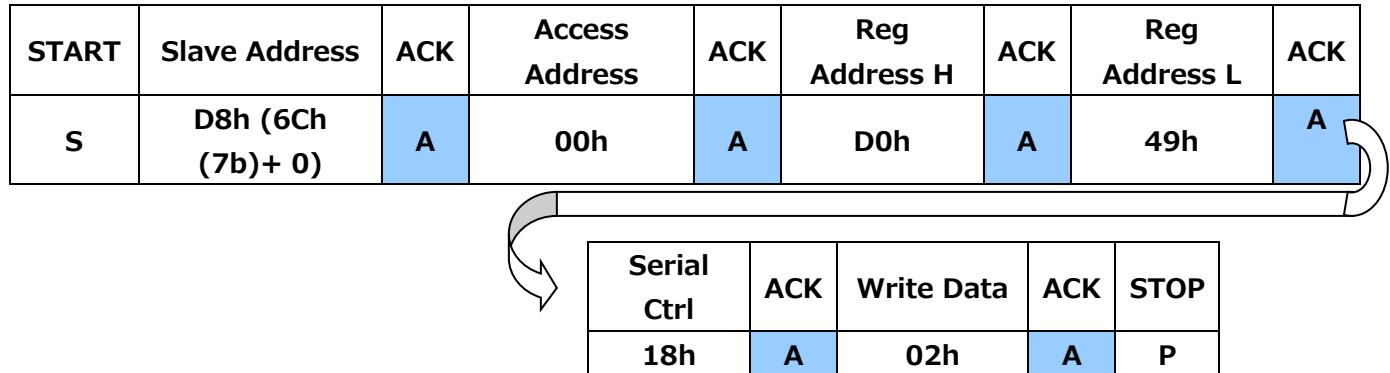


10.2 CRC 演算機能有効化方法

1. 割り込み制御レジスタのビット1を"1"にセットします。

I²C コマンド: 割り込みコントロールレジスタ (D049h) に 02h 書き込み (CRC_EN = 1)

表 18 CRC 演算有効化処理



なお、CRC 値読み出しシーケンスに関しては、8.3.2 CRC 演算制御レジスタの図 10 をご参照下さい。

10.3 リセット実行方法

センサが何らかの原因でフリーズしたときの対処としてリセットする方法が準備されています。

MCU リセット

初期化レジスタ 0Bh のビット 2 を"1"にセットします。これにより、MCU の実行を停止し、Active 状態に戻ります。パワーシーケンスレジスタの値が 9 のままの時は、この処理をすることで解消することができます。

I²C コマンド：初期化レジスタ(0Bh)に 04h を書き込み

表 19 MCU リセット

START	Slave Address	ACK	Access Address	ACK	Write Data	ACK	STOP
S	D8h (6Ch (7b)+ 0)	A	0Bh	A	04h	A	P

MCU リセット実行後は、初期化レジスタのビット 2 は自動で"0"クリアされます。

ソフトリセット

初期化レジスタ 0Bh のビット 3 を"1"にセットします。これにより、センサは電源投入状態に戻ります。パワーシーケンスレジスタの値が 9、2、0 以外の時は、この処理で Idle に戻すことができます。

I²C コマンド：初期化レジスタ(0Bh)に 08h を書き込み

表 20 ソフトリセット

START	Slave Address	ACK	Access Address	ACK	Write Data	ACK	STOP
S	D8h (6Ch (7b)+ 0)	A	0Bh	A	08h	A	P

MCU リセット実行後は、初期化レジスタのビット 3 は自動で"0"クリアされます。

ハードリセット

パワーシーケンスレジスタのビット 7 を"1"にセットします。これにより、センサは電源投入状態に戻ります。

I²C コマンド：パワーシーケンスレジスタ (0Dh) に 80h を書き込み (Hard_Reset = 1)

表 21 ハードリセット

START	Slave Address	ACK	Access Address	ACK	Write Data	ACK	STOP
S	D8h (6Ch (7b)+ 0)	A	0Dh	A	80h	A	P

ハードウェアリセット実行後は、パワーシーケンスレジスタのビット 7 は自動で"0"クリアされます。

10.4 I²C バス特性

このバスは異なる IC 間の通信を目的としています。2 本の信号線があります：双方向のデータ信号（SDA）とクロック信号（SCK）です。SDA と SCL の両方の信号ラインはプルアップ抵抗を介して電源電圧に接続されなければなりません。以下のプロトコルが定義されています。

- バスが非ビジー状態のときのみ、データ転送が開始できます。
- データ転送中は、データ信号はクロック信号が“H”の時は安定しなければなりません。
- クロック信号が“H”の区間でデータ信号が変化すると、制御信号として解釈されます。

従って、以下のバス状態が定義されています。

バス非ビジー状態

データ信号（SDA）とクロック信号（SCL）の両方が“H”的状態です。

データ転送スタート

クロック信号（SCL）が“H”的間にデータ信号（SDA）が“H”から“L”に変化する状態は、スタート条件として定義されています。

データ転送ストップ

クロック信号（SCL）が“H”的間にデータ信号（SDA）が“L”から“H”に変化する状態は、ストップ条件として定義されています。

有効データ

スタート条件後、データ信号（SDA）は有効な状態となります。データ信号はクロック信号が“H”的区間では安定した状態となります。データ信号はクロック信号が“L”的区間で変化する事になります。1 クロック当たりに 1 ビットのデータとなります。各データ転送はスタート条件で初期化され、ストップ条件で終了となります。スタート条件とストップ条件の間に転送されるバイト数に制限はありません。各転送データはバイト単位データと 9 ビット目データとしての受信側アクリッジで構成されます。

メッセージを送信する側は“トランシミッター”と呼ばれ、メッセージを受信するデバイスが“レシーバ”と呼ぶと定義されています。メッセージを制御する側が“マスター”と呼ばれます。また、マスターから制御される側を“スレーブ”呼びます。

アクノリッジ

各 8 ビットつまり 1 バイトに続き、1 ビットのアクノリッジビットとなります。このアクノリッジビットはマスタがアクノリッジビット用のクロックパルスを生成している区間に、レシーバ側がバスレベルを "L" にドライブで成立します。指定されたスレーブレシーバは、クロック同期した各バイトデータ受信後にアクノリッジビットを発行しなければなりません。

スレーブデバイスはアクノリッジビットに対応したクロック周期でクロックが "H" の区間に SDA ラインを安定して "L" にドライブしなければなりません。勿論、セットアップ・ホールド時間を考慮する必要があります。受信側マスタは、送信側スレーブから出力された最終バイトの後に非アクノリッジを発行しなければなりません。この場合マスタがストップコンディションを生成可能な様に、送信側は "H" のままにしておく必要があります。

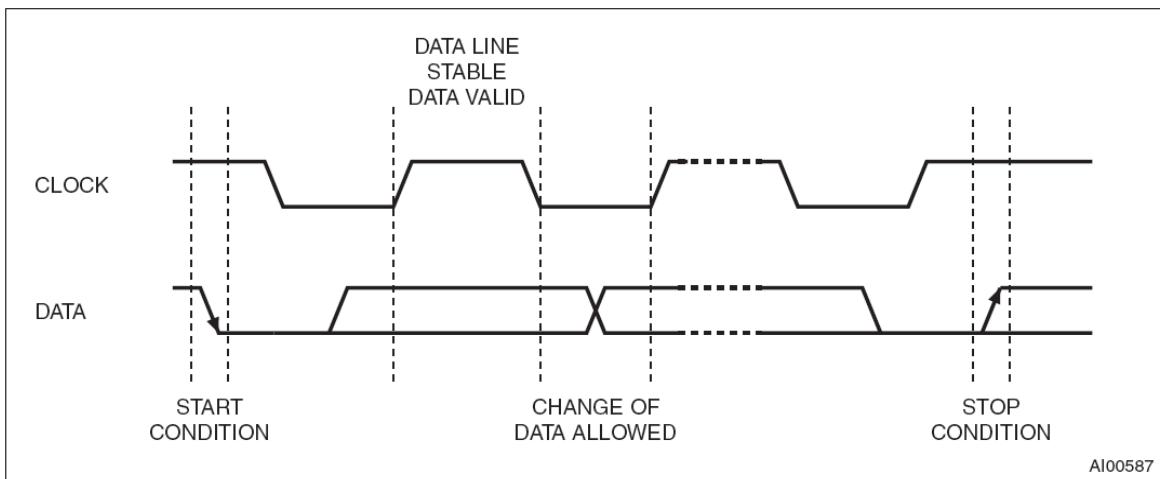


図 14 シリアルバスデータ送信シーケンス

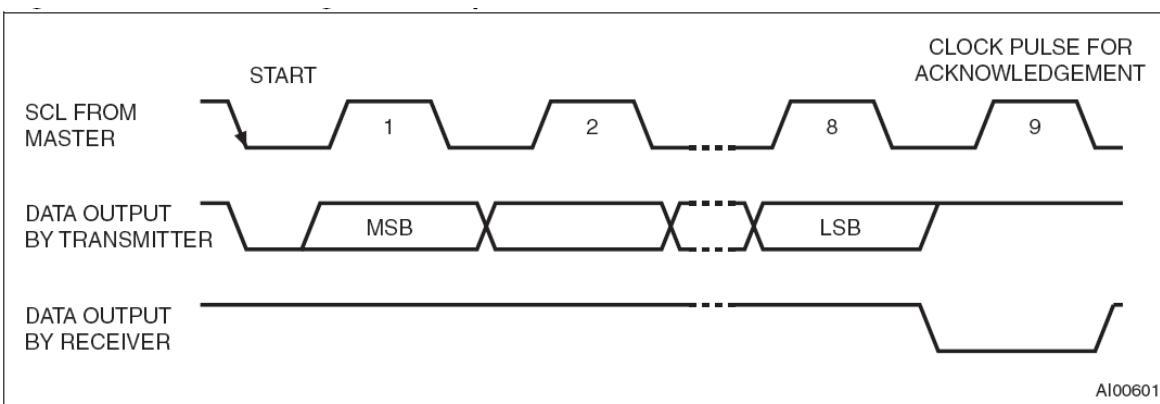


図 15 アクノリッジシーケンス

I²C 読み出しモード

このモードではスレーブアドレス設定完了後に、マスタがデジタルフローセンサスレーブを読み出します(図 9 参照)。ライトモード制御ビット (R/W = 0) とアクノリッジビットに続き、ワードアドレス"An"がセンサチップ内部のアドレスポインタにセットされます。次に、スタート条件とスレーブアドレスが再度発行され、リードモード制御ビット (R/W=1) が続きます。この時点でマスタトランスマッターは、マスタレシーバとなります。指定されたアドレスのバイトデータが送信され、マスタレシーバがスレーブトランスマッタに対してアクノリッジを発行します。アドレスポインタはアクノリッジビットに対応するクロック受信時にのみインクリメントされます。スレーブトランスマッタとしてのデジタルフローセンサはアドレス An+1 のデータをデータラインに送信し、マスタレシーバがこのデータを受け取りアクノリッジを発行、次のバイトと認識してアドレスポインタは An+2 にインクリメントされます。

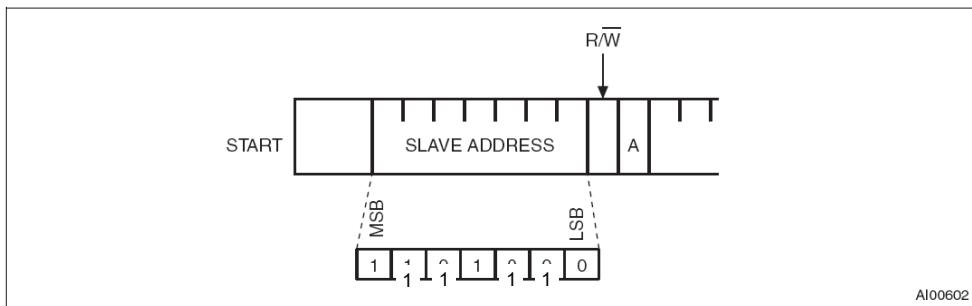


図 16 スレーブアドレス配置

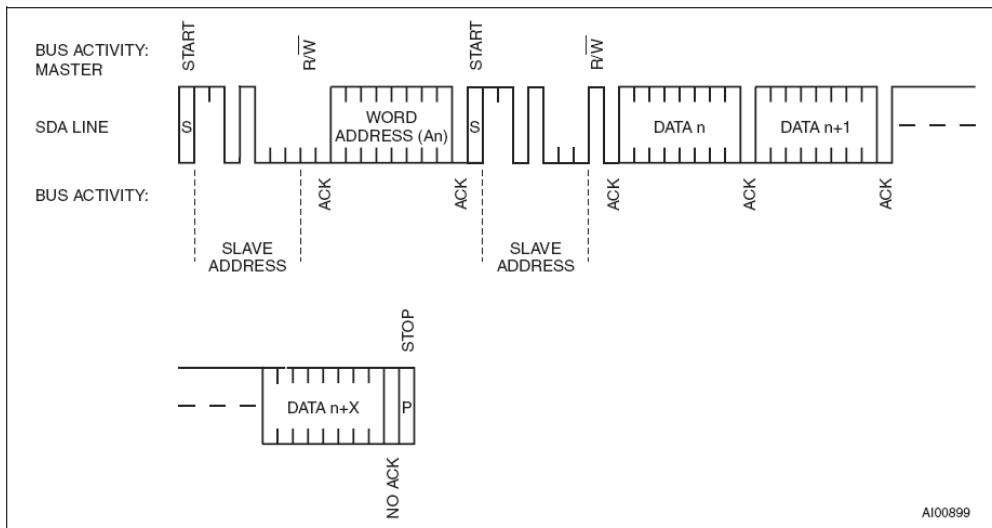


図 17 読み出しモードシーケンス

I²C 書き込みモード

このモードでは、マスタトランスマッターがデジタルフローセンサスレーブシーバに対してデータ転送します。バスプロトコルを図 18 に示します。スタート条件とスレーブアドレスに続き、バスをロジック"0" (R/W=0) 状態にします。これにより指定されたアドレスのデバイスの内部アドレスポインタに"An"が書き込まれます。内部メモリに書き込まれるワードデータが次にバス上にロードされ、アクノリッジビットに対応するクロック受信時に内部アドレスポインタが次のアドレスを示すようインクリメントされます。デジタルフローセンサはスレーブシーバとして、スレーブアドレス・ワードアドレス・書き込みデータ受信毎にアクノリッジをマスタトランスマッターに発行します。

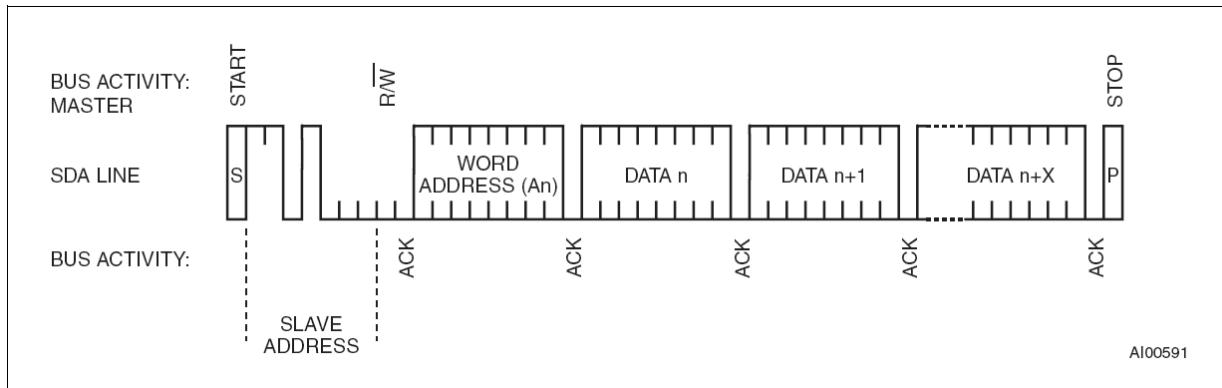


図 18 書き込みモードシーケンス

11 用語説明

- I²C (Inter – Integrated Circuit)

Philips 社が提唱したシリアル通信用プロトコルです。

シリアル・データ・ライン（SDA）とシリアル・クロック・ライン（SCL）の 2 本のバス・ラインのみで構成されています。このバスに接続されている各デバイスには各々固有のアドレス（スレーブアドレス）を持ち、それを元に制御側ホスト MCU のソフトウェアによる各デバイスのアドレス指定が可能です。I²C に関するより詳細な情報は、NXP Semiconductors 社のホームページの UM10204 の資料をご確認下さい。

- スレーブアドレス

I²C 通信における通信相手の”住所”を示すものです。

通常 7 ビットで表現されており、弊社デジタルフローセンサでは”110_1100b” または “6Ch”となっております。実際の通信の際には、最下位ビットにライト・リードを意味する 1 ビットが付加されて、8 ビットとなります。ライト時には”0”を付加し、“1101_1000b” または”D8h”となり、リード時には”1”を負付加し、“1101_1001b” または ”D9h”となります。お客様でご使用される制御側ホスト MCU により、ソフトウェア上でスレーブアドレスを 7 ビットで表現するのか 8 ビットで表現するのかが異なります。ご使用される制御側 MCU の I²C 通信ソフトウェアライブラリ仕様をご確認下さい。

12 サンプルソースコード

以下のサンプルソースコードは、STM32 マイクロコントローラを用いた場合の D6F-□□□□D-000-□向けのものです。I²C 制御部分は、お客様がお使いになるマイクロコントローラに合わせて修正が必要となります。

12.1 D6F_D_Sample.h

```
/*=====
 * D6F Digital Type Flow Sensor Header file (using STM32)
 * :Copyright: (C) OMRON Corporation, Microdevice H.Q.
 * :Auther   : Kenichi Handa
 * :Revision: $Rev$
 * :Id:      $Id$
 * :Date:    $Date$
 *
 * All Rights Reserved
 * OMRON Confidential
 * OMRON Proprietary Right
 *=====
 */
/*== for General ==*/
#define SA_7      0x6F // for 7bit Slave Address
#define RANGE_MODE 100 // Full Range 100[L/min]
/*== for Measure Mode ==*/
#define F       1 // Flow rate mode
#define T       2 // Temperature mode
/*== for CRC & HW_reset ==*/
#define ON      1 // CRC ON / HW_reset execute
#define OFF     2 // CRC OFF / HW_reset release

/* Function prototypes -----*/
void Initialize( void );
short Flow_meas( void );
short Temp_meas( void );
int  CRC_enable(char mode);
void HW_reset(char mode);
unsigned char GetCRC8(unsigned char data[], unsigned char num);
/* Private Functions -----*/
int  I2C_WR(unsigned char add, char *dbuf, unsigned char n);
uint8_t I2C_RD_8(unsigned char add, char *dbuf, unsigned char n);
short I2C_RD_16(unsigned char add, char *dbuf, unsigned char n);
unsigned short I2C_RD_u16(unsigned char add, char *dbuf, unsigned char n);
```

12.2 D6F_D_Sample.c

```
/*=====
/* D6F Digital Flow Sensor Sample Code (using STM32)
 * :Copyright: (C) OMRON Corporation, Microdevice H.Q.
 * :Auther   : Kenichi Handa
 * :Revision: $Rev$
 * :Id:      $Id$
 * :Date:    $Date$
 *
 * All Rights Reserved
 * OMRON Confidential
 * OMRON Proprietary Right
 *=====
#include "stm32f10x_i2c.h"
#include "D6F_D_Sample.h"

#define I2C1_SCL_PIN          GPIO_Pin_6
#define I2C1_SDA_PIN          GPIO_Pin_7
#define  POLYNOMIAL_CRC8 0x131

typedef unsigned char  uint8;
typedef unsigned short uint16;
typedef unsigned long  uint32;

    short  RD_FIFO;    /* 16bit data width */
unsigned short uRD_FIFO;    /* 16bit data width */
    uint8_t RD_REG;    /* 8bit data width */
    char  setting_done_flag = 0;
    char  CRC_on = 0;  /* CRC function flag */
    uint8_t det_crc;

// Dummy wait routine
void adc_wait(volatile unsigned long delay)
{
    while(delay) delay--;
}

/*=====
/* Initialize Function
/* Usage      : Initialize( void )
/* Argument   : Null
/* Return value : Null
/*=====
void Initialize( void )
{
```

```

/* EEPROM Control <= 00h */
char send1[] = {0x0B, 0x00};
I2C_WR(SA_7, send1, 2);
}

/*=====
/* Flow rate measure Function
/* Usage      : Flow_meas( void )          */
/* Argument   : NULL                      */
/* Return value : Compensated Flow rate value(short) */
=====*/
short Flow_meas(void)
{
    short rd_fifo;
    short rd_flow;
    unsigned long  wait_time;
    unsigned char  calc_crc8;
    uint8_t rd_data[2] = {0, 0};

    calc_crc8 = 0; /* initialize */
    /* [D040] <= 06h */
    char send2[] = {0x00, 0xD0, 0x40, 0x18, 0x06};
    I2C_WR(SA_7, send2, 5);

    /* wait time depend on resolution mode */
    wait_time = 120; /* 120msec wait */
    adc_wait(wait_time);

    /* [D051/D052] => Read Compensated Flow value */
    char send3[] = {0x00, 0xD0, 0x51, 0x2C, 0x07};
    uRD_FIFO = I2C_RD_u16(SA_7, send3, 5);

    if (CRC_on){
        rd_data[0] = (uRD_FIFO >> 8) & 0xFF;
        rd_data[1] = (uRD_FIFO      ) & 0xFF;
        calc_crc8 = GetCRC8(rd_data, 2);

        if(calc_crc8 != det_crc){ /* CRC value mismatch !! */
            /* Please type your error routine */
        }
    }
    rd_flow = ((uRD_FIFO - 1024) * RANGE_MODE *10/ 60000); /* convert to [L/min](x10) */
    return rd_flow;
}

```

```

/*=====
/* Temperature measure Function */
/* Usage      : Temp_meas() */
/* Argument   : NULL */
/* Return value : x10 Temperature */
/*=====*/
short Temp_meas(void)
{
    short    rd_temp;
    unsigned long wait_time;

    /* [D040] <= 06h */
    char send2[] = {0x00, 0xD0, 0x40, 0x18, 0x06};
    I2C_WR(SA_7, send2, 5);

    /* wait time depend on resolution mode */
    wait_time = 50 + 15; /* 65msec wait */
    adc_wait(wait_time);

    /* [D061/D062] => Read TMP_H/TMP_L value */
    char send3[] = {0x00, 0xD0, 0x61, 0x2C, 0x07};
    RD_FIFO = I2C_RD_16 (SA_7, send3, 5);

    rd_temp = ((RD_FIFO -10214)*1000 / 3739); // convert to degree-C(x10)
    return rd_temp;
}

/*=====
/* CRC Control Function */
/* Usage      : CRC_enable(ON) */
/* Argument1  : ON  : CRC function enable */
/*           : OFF : CRC function disable */
/* Return value : */
/*=====*/
int CRC_enable(char mode)
{
    uint8_t    int_reg_rd;
    uint8_t    int_reg_mod;

    /* [D049] => Pre-Read INT_CTRL Reg value */
    char send1[] = {0x00, 0xD0, 0x49, 0x1C, 0x07};
    int_reg_rd = I2C_RD_8(SA_7, send1, 5);

    switch(mode){
        case ON:int_reg_mod = int_reg_rd | 0x02; /* set INT_CTRL[1] */
}

```

```

/* [D049][2] <= 1:CRC ON Mode */
char send2[] = {0x00, 0xD0, 0x49, 0x18, int_reg_mod};
I2C_WR (SA_7, send2, 5);
CRC_on = 1;
break;
case OFF:int_reg_mod = int_reg_rd & 0xFD; /* clear INT_CTRL[1] */
/* [D049][2] <= 0:CRC OFF Mode */
char send3[] = {0x00, 0xD0, 0x49, 0x18, int_reg_mod};
I2C_WR (SA_7, send3, 5);
CRC_on = 0;
break;
default:
    return 1; /* illegal argument detect */
break;
}
return 0; /* normal finish */
}

/*=====
/* HW reset Control Function
/* Usage      : HW_reset(ON) */
/* Argument1  : ON  : HW reset execute
/*           : OFF : HW reset release
/* Return value : */
=====*/
void HW_reset(char mode)
{
    uint8_t    pow_reg_mod;

    switch(mode){
        case ON :pow_reg_mod = 0x80; /* Power Sequence Register <= 80h */
        break;
        case OFF:pow_reg_mod = 0x00; /* Power Sequence Register <= 00h */
        break;
        default :return 1; /* illegal argument detect */
        break;
    }

    char send1[] = {0x0D, pow_reg_mod};/* Power Sequence Register <= 80h */
    I2C_WR(SA_7, send1, 2);
}

```

```
/*=====
/* CRC8 calculation Control Function */
/* Usage      : GetCRC8(data, num) */
/* Argument1  : data : taget data for CRC8 calculate */
/* Argument2  : num  : size of data */
/* Return value : CRC value of data */
/*=====*/
unsigned char GetCRC8(unsigned char data[], unsigned char num)
{
    unsigned char crc = 0;
    unsigned char bytcnt;
    unsigned char bitcnt;

    for (bytcnt = 0; bytcnt < num; bytcnt++){
        crc ^= (data[bytcnt]);

        for (bitcnt = 8; bitcnt > 0; bitcnt--){
            if (crc & 0x80){
                crc = (crc << 1) ^ POLYNOMIAL_CRC8;
            }
            else{
                crc = (crc <<1);
            }
        }
    }
    return crc;
}
```

13 ご承諾事項

平素はオムロン株式会社(以下「当社」)の商品をご愛用いただき誠にありがとうございます。「当社商品」のご購入について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。ご承諾のうえご注文ください。

1. 定義

本ご承諾事項中の用語の定義は次のとおりです。

- (1) 「当社商品」： 「当社」の FA システム機器、汎用制御機器、センシング機器、電子・機構部品
- (2) 「カタログ等」：「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含みます。
- (3) 「利用条件等」：「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取扱い方法、利用上の注意、禁止事項その他
- (4) 「お客様用途」：「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み又は利用を含みます
- (5) 「適合性等」： 「お客様用途」での「当社商品」の(a) 適合性、(b) 動作、(c) 第三者の知的財産の非侵害、(d) 法令の遵守および(e) 各種規格の遵守

2. 記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- (1) 定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- (2) 参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- (3) 利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- (4) 「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

3. ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- (1) 定格・性能ほか「利用条件等」を遵守ご利用ください。
- (2) お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。「当社」は「適合性等」を一切保証いたしかねます。
- (3) 「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- (4) 「当社商品」をご使用の際には、(i) 定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii) 「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii) 利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv) 「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- (5) 「当社」は DDoS 攻撃（分散型 DoS 攻撃）、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスにより、「当社商品」、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害

その他の費用について一切責任を負わないものとします。お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。

- (6) 「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても、「当社」の意図した特別な商品用途の場合や特別の合意がある場合は除きます。
- (a) 高い安全性が必要とされる用途（例：原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娛樂設備、医用機器、安全装置、その他生命・身体に危険が及びうる用途）
 - (b) 高い信頼性が必要な用途（例：ガス・水道・電気等の供給システム、24 時間連続運転システム、決済システムほか権利・財産を取扱う用途など）
 - (c) 厳しい条件または環境での用途（例：屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など）
 - (d) 「カタログ等」に記載のない条件や環境での用途

4. 保証条件

「当社商品」の保証条件は次のとおりです。

- (1) 保証期間 ご購入後 1 年間といたします。
(ただし「カタログ等」に別途記載がある場合を除きます。)
- (2) 保証内容 故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。
 - (a) 当社保守サービス拠点における故障した「当社商品」の無償修理
(ただし、電子・機構部品については、修理対応は行いません。)
 - (b) 故障した「当社商品」と同数の代替品の無償提供
- (3) 保証対象外 故障の原因が次のいずれかに該当する場合は、保証いたしません。
 - (a) 「当社商品」本来の使い方以外のご利用
 - (b) 「利用条件等」から外れたご利用
 - (c) 本ご承諾事項「3. ご利用にあたってのご注意」に反するご利用
 - (d) 「当社」以外による改造、修理による場合
 - (e) 「当社」以外の者によるソフトウェアプログラムによる場合
 - (f) 「当社」からの出荷時の科学・技術の水準では予見できなかった原因
 - (g) 上記のほか「当社」または「当社商品」以外の原因（天災等の不可抗力を含む）

5. 責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。

「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

6. 輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

製品に関するお問い合わせ先

<p>お客様 相談室</p> <p>■ フリー通話 0120-919-066</p> <p>携帯電話の場合、 📞 055-982-5015 (有料) をご利用ください。</p> <p>受付時間：9:00～17:00 (土・日・12/31～1/3を除く)</p>	<p>クイック オムロン</p> <p>技術相談員にチャットでお問い合わせいただけます。(I-Webメンバーズ限定)</p> <p>受付時間：平日9:00～12:00 / 13:00～17:00 (土日祝日・年末年始・当社休業日を除く)</p> <p>※受付時間・営業日は変更の可能性がございます。最新情報はリンク先をご確認ください。</p>	<p> www.fa.omron.co.jp/contact/tech/chat/</p>
---	---	---

その他のお問い合わせ：納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。

 オムロン制御機器の最新情報をご覧いただけます。緊急時のご購入にもご利用ください。 www.fa.omron.co.jp

- 本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娛樂機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
- 本製品の内、外國為替及び外國貿易法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です。

オムロン商品のご用命は