

# TCAN5102-Q1 SPI、UART、または I<sup>2</sup>C コントローラ付き自動車用 CAN FD ライトレスポнда、プログラマブル PWM 搭載

## 1 特長

- AEC-Q100：車載アプリケーション認定済み
- CAN FD Light Responder は ISO 11898-1:202x に準拠
- TCAN4572-Q1 CAN FD Light Commander を使用する場合、最大 5Mbps の CAN FD Light データレートをサポートします
- 以下のような外部 CAN トランシーバを制御する能力：TCAN1162x-Q1、TCAN1043A、TCAN1463A
- CAN バス経由でプログラム可能
- 電源電圧：3.3V ~ 5V
- 最大 8 つのチップセレクトにより、最大 20MHz に対応する SPI コントローラをサポート
- 最大 2.5Mbaud をサポートする UART コントローラ
- I<sup>2</sup>C コントローラで高速モードプラス (1MHz) をサポート
- 他の機能と多重化可能な 13 の GPIO
- モーター制御用の台形波ランププロファイルをサポートするプログラマブル PWM 出力
- 20 ピン VSSOP (DGQ) パッケージ

## 2 アプリケーション

- ボディエレクトロニクスおよび照明
- ハイブリッド、電動、パワートレインシステム
- インフォテインメントおよびクラスタ
- 電化製品

## 3 説明

TCAN5102-Q1 は、CiA 604-1 および ISO 11898-1：2024 と互換性のあるコントロールエリアネットワーク (CAN) フレキシブルデータ (FD) ライトレスポндаデバイスで、専用の CAN FD ライトコマンドを使用した場合、最大 5Mbps のデータレートをサポートできます。このデバイスは、レスポндаノードプロセッサを必要としないコマンドレスポндаアーキテクチャにおいて CAN FD ライトレスポндаノードアプリケーションをサポートするように設計されています。レスポндаノードへの制御はすべて、コマンドノードプロセッサからの CAN バスを介して行われるため、レスポндаノードプロセッサやソフトウェアは不要です。

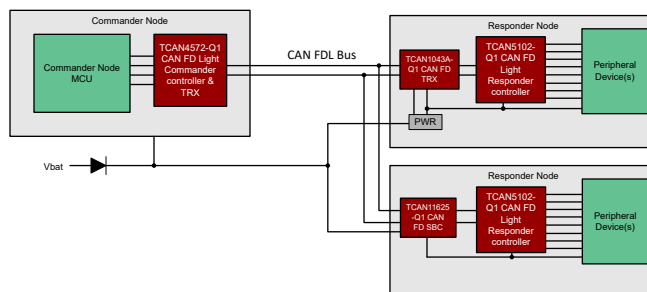
デバイスは CAN FD ライトコマンドノードからデータやコマンドを受信し、これらのコマンドをシリアルペリフェラルインターフェイス (SPI) コントローラ、UART コントローラ、I<sup>2</sup>C コントローラ、GPIO ピンのいずれかまたは両方に変換して、TCAN5102-Q1 が制御しているデバイスやペリフェラルデバイスと通信します。PWM 出力チャネルは、ステップモーターを制御するためのハードウェアで台形波ランププロファイルもサポートしています。デューティサイクルまたは周波数を上昇させることができます。外部水晶振動子やクロックは不要です。このデバイスは、外部の TCAN1162x-Q1、TCAN1043A-Q1、TCAN1463A-Q1 CAN FD (SIC) トランシーバを制御し、システムレベルの柔軟性を確保します。このデバイスは、CAN FD トランシーバ / SBC を利用してノードの電力を制御し、CAN RXD (CRXD) ピンを Low にラッチすることでウェイクアップ信号を TCAN5102-Q1 に伝達します。

### パッケージ情報

部品番号	パッケージ (1)	パッケージサイズ (2)
TCAN5102-Q1	20 ピン (VSSOP, DGQ)	5.1mm × 4.9mm

- (1) 詳細については、[セクション 11](#) を参照してください。  
 (2) パッケージサイズ (長さ × 幅) は公称値であり、該当する場合はピンも含まれます。





CAN FD ライトコマンダーレスポンス アプリケーション

## 目次

<b>1 特長</b> .....	<b>1</b>	7.4 デバイスの機能モード.....	<b>20</b>
<b>2 アプリケーション</b> .....	<b>1</b>	7.5 プログラミング.....	<b>21</b>
<b>3 説明</b> .....	<b>1</b>	7.6 レジスタ マップ.....	<b>58</b>
<b>4 ピン構成および機能</b> .....	<b>4</b>	<b>8 アプリケーションと実装</b> .....	<b>228</b>
<b>5 仕様</b> .....	<b>5</b>	8.1 アプリケーション情報.....	<b>228</b>
5.1 絶対最大定格.....	<b>5</b>	8.2 代表的なアプリケーション.....	<b>228</b>
5.2 ESD 定格.....	<b>5</b>	8.3 電源に関する推奨事項.....	<b>229</b>
5.3 推奨動作条件.....	<b>5</b>	8.4 レイアウト.....	<b>229</b>
5.4 熱に関する情報.....	<b>5</b>	<b>9 デバイスおよびドキュメントのサポート</b> .....	<b>231</b>
5.5 電源の特性.....	<b>6</b>	9.1 ドキュメントのサポート.....	<b>231</b>
5.6 電気的特性.....	<b>6</b>	9.2 ドキュメントの更新通知を受け取る方法.....	<b>231</b>
5.7 タイミング要件.....	<b>7</b>	9.3 サポート・リソース.....	<b>231</b>
5.8 スイッチング特性.....	<b>8</b>	9.4 商標.....	<b>231</b>
5.9 I <sup>2</sup> C バス タイミング要件.....	<b>10</b>	9.5 静電気放電に関する注意事項.....	<b>231</b>
<b>6 パラメータ測定情報</b> .....	<b>12</b>	9.6 用語集.....	<b>231</b>
<b>7 詳細説明</b> .....	<b>15</b>	<b>10 改訂履歴</b> .....	<b>231</b>
7.1 概要.....	<b>15</b>	<b>11 メカニカル、パッケージ、および注文情報</b> .....	<b>232</b>
7.2 機能ブロック図.....	<b>15</b>	11.1 メカニカルデータ.....	<b>233</b>
7.3 機能説明.....	<b>17</b>		

## 4 ピン構成および機能

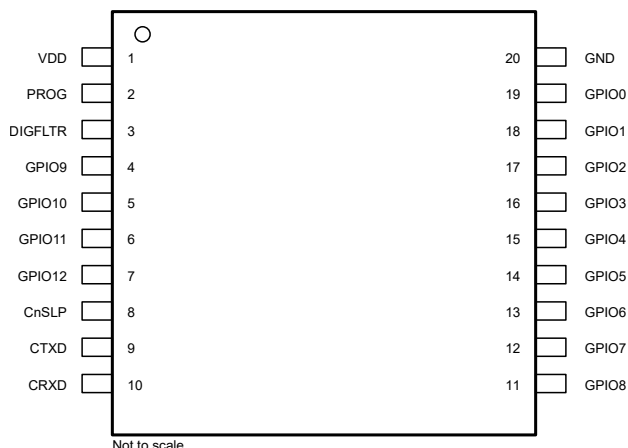


図 4-1. DGQ 20-PIN (上面図)

表 4-1. ピンの機能

ピン		タイプ <sup>(1)</sup>	説明
名称	番号		
VDD	1	P	デバイス電源入力、3.3V または 5V
PROG	2	I	デバイスの構成およびデバッグのための入力
DIGFLTR	3	P	デジタル電源フィルタ ピン
GPIO9/CS4/SCL	4	I/O	汎用入出力ピン、SPI チップ セレクト チャネル 4 出力、I <sup>2</sup> C クロック出力
GPIO10/CS5/SDA	5	I/O	汎用入出力ピン、SPI チップ セレクト チャネル 5 出力、I <sup>2</sup> C データピン
GPIO11/CS6/PWM0	6	I/O	汎用入出力ピン、SPI チップ セレクト チャネル 6 出力、または PWM0 出力
GPIO12/CS7/PWM1	7	I/O	汎用入出力ピン、SPI チップ セレクト チャネル 6 出力、または PWM1 出力
CnSLP	8	O	CAN トランシーバ制御出力ピン (nSLP)
CTXD	9	O	CAN トランシーバ TXD 出力ピン (トランシーバの TXD 入力ピンに接続)
CRXD	10	I	CAN トランシーバ RXD 入力ピン (トランシーバの RXD 出力ピンに接続)
GPIO8/URXD	11	I/O	汎用入出力ピンまたは UART RXD 入力
GPIO7/UTXD	12	I/O	汎用入出力ピンまたは UART TXD 出力
GPIO6/PICO	13	I/O	汎用入出力ピンまたは SPI ペリフェラル入力、コントローラ出力
GPIO5/POCI	14	I/O	汎用入出力ピンまたは SPI ペリフェラル出力、コントローラ入力
GPIO4/SCK	15	I/O	汎用入出力ピンまたは SPI クロック出力
GPIO3/CS0	16	I/O	汎用入出力ピンまたは SPI チップ セレクト チャネル 0 出力
GPIO2/CS1	17	I/O	汎用入出力ピンまたは SPI チップ セレクト チャネル 1 出力
GPIO1/CS2	18	I/O	汎用入出力ピンまたは SPI チップ セレクト チャネル 2 出力
GPIO0/CS3	19	I/O	汎用入出力ピンまたは SPI チップ セレクト チャネル 3 出力
GND	20	P	グラウンド ピン

(1) I = 入力、O = 出力、I/O = 入力または出力、G = グラウンド、P = 電源。

## 注

複数の機能を持つピンについては、ピンを構成するために IO\_CFG レジスタを参照し、ピンの多重化の詳細については、[セクション 7.3.4](#) を参照してください。

## 5 仕様

### 5.1 絶対最大定格

自由気流での動作温度範囲内 (特に記述のない限り) <sup>(1)</sup>

		最小値	最大値	単位
VDD	電源入力	-0.3	6	V
DIGFLTR	デジタルコアフィルタピン	-0.3	1.8	V
V <sub>LOGIC_IN</sub>	ロジックピン入力電圧範囲	-0.3	6	V
V <sub>LOGIC_OUT</sub>	ロジックピン出力電圧範囲	-0.5	6	V
I <sub>O(LOGIC)</sub>	ロジックピン出力電流		12	mA
T <sub>J</sub>	接合部温度	-40	150	°C
T <sub>stg</sub>	保存温度	-65	150	°C

(1) 「絶対最大定格」の範囲外の動作は、デバイスの永続的な損傷の原因となる可能性があります。絶対最大定格は、このデータシートの「推奨動作条件」に示された値を超える状態で本製品が正常に動作することを暗黙的に示すものではありません。絶対最大定格の範囲内であっても、推奨動作条件の範囲外で使用情况、本デバイスは完全に機能するとは限らず、このことが本デバイスの信頼性、機能、性能に影響を及ぼし、本デバイスの寿命を縮める可能性があります。

### 5.2 ESD 定格

			値	単位
V <sub>(ESD)</sub>	静電放電	人体モデル (HBM) 分類レベル 3A、その他のすべてのピン、AEC Q100-002 に準拠 <sup>(1)</sup>	±2000	V
		荷電デバイス モデル (CDM)、分類レベル C5、AEC Q100-011 に準拠	±750	
		角のピン その他のピン	±750	

(1) AEC Q100-002 は、HBM ストレス試験を ANSI / ESDA / JEDEC JS-001 仕様に従って実施しなければならないと規定しています。

### 5.3 推奨動作条件

外気温度範囲での動作時 (特に記述がない限り)

		最小値	公称値	最大値	単位
VDD	電源電圧	3.0		5.5	V
I <sub>OH(DO)</sub>	デジタル出力 High レベル電流	-12			mA
I <sub>OL(DO)</sub>	デジタル出力 Low レベル電流			12	mA
C <sub>(VDD)</sub>	VDD 電源容量	1			μF
C <sub>(DIGFLTR)</sub>	デジタルコアフィルタのピン容量	0.33	1		μF
ESR <sub>CO</sub>	電源ピンおよびフィルタピンの ESR 容量要件	0.001		1	Ω
T <sub>J</sub>	動作ジャンクション温度範囲	-40		125	°C

### 5.4 熱に関する情報

熱評価基準 <sup>(1)</sup>		TCAN5102-Q1	単位
		RGY (QFN)	
		24-PINS	
R <sub>θJA</sub>	接合部から周囲への熱抵抗	31.8	°C/W
R <sub>θJB</sub>	接合部から基板への熱抵抗	11.8	°C/W
R <sub>θJC(top)</sub>	接合部からケース (上面) への熱抵抗	21.4	°C/W
R <sub>θJC(bot)</sub>	接合部からケース (底面) への熱抵抗	2.8	°C/W
Ψ <sub>JT</sub>	接合部から上面への特性パラメータ	0.3	°C/W

## 5.4 熱に関する情報 (続き)

熱評価基準 <sup>(1)</sup>		TCAN5102-Q1	単位
		RGY (QFN)	
		24-PINS	
$\Psi_{JB}$	接合部から基板への特性パラメータ	11.8	°C/W

(1) 従来および最新の熱評価基準の詳細については、『[半導体および IC パッケージの熱評価基準](#)』アプリケーション レポートを参照してください。

## 5.5 電源の特性

外気温度範囲での動作時 (特に記述がない限り)

パラメータ		テスト条件	最小値	標準値	最大値	単位
<b>VDD</b>						
IDD	VDD 供給電流デバイス(通常モード時)	通常モード、VDD = 5V $I_I = 0$ , 0V または VDD SPI、UART、I2C 無効化		1.4	1.8	mA
		通常モード、VDD = 3.3V $I_I = 0$ , 0V または VDD SPI、UART、I2C 無効化		1.4	1.8	mA
IDD	VDD 供給電流デバイス(スリープモード時)	スリープモード、VDD = 5V $I_I = 0$ , 0V または VDD $-40^{\circ}\text{C} \leq T_J \leq 85^{\circ}\text{C}$		10	20	$\mu\text{A}$
		スリープモード、VDD = 5V $I_I = 0$ , 0V または VDD $T_J > 85^{\circ}\text{C}$		20	60	$\mu\text{A}$
		スリープモード、VDD = 3.3V $I_I = 0$ , 0V または VDD $-40^{\circ}\text{C} \leq T_J \leq 85^{\circ}\text{C}$		10	20	$\mu\text{A}$
		スリープモード、VDD = 3.3V $I_I = 0$ , 0V または VDD $T_J > 85^{\circ}\text{C}$		20	60	$\mu\text{A}$
UV <sub>DDR</sub>	電源低電圧立ち上がりスレッシュホールド	VDD 立ち上がり	2.35		2.95	V
UV <sub>DDF</sub>	電源低電圧立ち下がりスレッシュホールド	VDD 立ち下がり	2.1		2.7	V
UV <sub>DDHYS</sub>	電源低電圧検出ヒステリシス			350	430	mV

## 5.6 電気的特性

外気温度範囲での動作時 (特に記述がない限り)

パラメータ		テスト条件	最小値	標準値	最大値	単位
<b>GPIO (入力)</b>						
V <sub>IH</sub>	High レベル入力電圧	GPIOx ピン	0.7			VDD
		CRXD ピン	0.7			
		PROG ピン	0.7			
V <sub>IL</sub>	Low レベル入力電圧	GPIOx ピン			0.3	VDD
		CRXD ピン			0.3	
		PROG ピン			0.3	

## 5.6 電気的特性 (続き)

外気温度範囲での動作時 (特に記述がない限り)

パラメータ		テスト条件	最小値	標準値	最大値	単位
I <sub>IH</sub>	High レベル入力リーク電流	GPIOx ピン 入力 = VDD VDD プルアップ/プルダウン抵抗無効	-1		1	μA
		CRXD ピン 入力 = VDD	-1		1	
		PROG ピン 入力 = VDD		90	140	
I <sub>IL</sub>	Low レベル入力リーク電流	GPIOx ピン 入力 = 0V プルアップ/プルダウン抵抗無効	-1		1	μA
		PROG ピン 入力 = 0V	-1		1	
		CRXD 入力 = 0V	-140	-90		
C <sub>IN</sub>	入力容量	20 MHz、GPIOx ピン		4	10	pF
		1 MHz、CRXD ピン		4	10	
		1 MHz、PROG ピン		4	10	
I <sub>LKG(OFF)</sub>	電源がない場合のリーク電流です	GPIOx ピン 入力 = 5.5V、VDD = 0V	-1	0	1	μA
R <sub>PD</sub>	プルダウン抵抗	GPIOx (有効の場合)	40	60	80	kΩ
		PROG	40	60	80	
R <sub>PU</sub>	プルアップ抵抗	GPIOx (有効の場合)	40	60	80	kΩ
		CRXD	40	60	80	
GPIOs (出力)						
V <sub>OH</sub>	High レベル出力電圧	GPIOx、CTXD、CnSLP ピン I <sub>OH</sub> = 10mA	0.7	0.85		VDD
		GPIOx、CTXD、CnSLP ピン I <sub>OH</sub> = 2mA	0.8	0.9		
V <sub>OL</sub>	Low レベル出力電圧	GPIOx、CTXD、CnSLP ピン I <sub>OL</sub> = -10mA		0.15	0.3	VDD
		GPIOx、CTXD、CnSLP ピン I <sub>OL</sub> = -2mA		0.1	0.2	
I <sub>OH</sub>	High レベル入力電流	GPIOx、CTXD、CnSLP ピン V <sub>O</sub> = 0.7 VDD	10	12		mA
I <sub>OL</sub>	Low レベル出力電流	GPIOx、CTXD、CnSLP ピン V <sub>O</sub> = 0.3 VDD		-12	-10	mA
I <sub>LKG(OFF)</sub>	電源がない場合のリーク電流です	CTXD、CnSLP ピン VPIN = 5.5 V	-5		5	μA
サーマル シャットダウン						
TSD <sub>F</sub>	サーマル シャットダウン下降		155	171	180	°C
TSD <sub>R</sub>	サーマル シャットダウン上昇		160	176	190	°C
TSD <sub>HYS</sub>	サーマル シャットダウンヒステリシス		2	5	10	°C

## 5.7 タイミング要件

		最小値	標準値	最大値	単位
<b>電源</b>					
$t_{PWRUP}$	デバイスが起動するまでに、VDD が POR を超えた後の時間			2	ms
<b>モード変更</b>					

## 5.7 タイミング要件 (続き)

		最小値	標準値	最大値	単位
$t_{\text{MODE\_NOM\_SLP}}$	デバイスが CAN トランシーバをオフにする「Go-to-sleep」コマンドからの経過時間		100	600	$\mu\text{s}$
<b>デバイス タイミング</b>					
$t_{\text{MODE\_POR\_NOM}}$	POR から通常モードへの移行時間、CAN メッセージ受信準備完了		150	400	$\mu\text{s}$
<b>SPI のタイミング要件</b>					
$t_{\text{HD-DAT}}$	アクティブ SCK エッジ後の POCI 入力ホールド時間要件 <a href="#">SPI のタイミング特性</a> を参照してください	20			ns
$t_{\text{SU-DAT}}$	アクティブ SCK エッジ前の POCI 入力セットアップ時間要件 <a href="#">SPI タイミング特性</a> を参照してください	5			ns
<b>UART</b>					
$t_{\text{W(RX)}}$	パルス幅、受信 開始、停止、データビット <sup>(1)</sup>	0.97	1	1.04	U

(1) U = UART のボー時間 = 1/ プログラムされたボーレート

## 5.8 スイッチング特性

外気温度範囲での動作時 (特に記述がない限り)

パラメータ		テスト条件	最小値	標準値	最大値	単位
SPI スイッチング特性						
f <sub>SCK-MAX</sub>	SCK、最大 SPI クロック周波数 <sup>(1)</sup>	通常モード、C <sub>L</sub> = 20pF	20			MHz
t <sub>SCK-MIN</sub>	SCK、最小 SPI クロック周期 <sup>(1)</sup>	通常モード、C <sub>L</sub> = 20pF	50			ns
t <sub>SCKR</sub>	SCK の立ち上がり時間 <sup>(1)</sup>	通常モード、C <sub>L</sub> = 20pF	4	10	ns	
t <sub>SCKF</sub>	SCK の立ち下がり時間 <sup>(1)</sup>	通常モード、C <sub>L</sub> = 20pF	4	8	ns	
t <sub>SCKH</sub>	SCK、SPI クロック high <sup>(1)</sup>	通常モード	0.47	0.49	0.5	S <sup>(3)</sup>
t <sub>SCKL</sub>	SCK、SPI クロック low <sup>(1)</sup>	通常モード	0.5	0.51	0.53	S <sup>(3)</sup>
t <sub>SCK-CS</sub>	最後の SCK エッジから CS 非アクティブ状態までの遅延時間 <sup>(1)</sup>	通常モード SPI のタイミング特性を参照してください	0.44	0.5	0.52	S <sup>(3)</sup>
t <sub>CS-SCK</sub>	CS アクティブから最初の SCK エッジまでの遅延時間 <sup>(1)</sup>	通常モード SPI のタイミング特性を参照してください	0.49	0.5	0.57	S <sup>(3)</sup>
t <sub>HD-DAT</sub>	キャプチャ SCK エッジから PICO データ変更までのデータホールド時間 <sup>(1)</sup>	通常モード SPI のタイミング特性を参照してください	0.47	0.5	0.51	S <sup>(3)</sup>
t <sub>CS-DV</sub>	遅延時間、チップセレクトのアクティブエッジから PICO データ有効まで <sup>(1)</sup>	通常モード の SPI モード 0、2 SPI のタイミング特性を参照してください		0	5	ns
		通常モード の SPI モード 1、3 SPI のタイミング特性を参照してください		0.5	0.57	S <sup>(3)</sup>
I2C のスイッチング特性						
t <sub>STUCKBUS_I2C</sub>	バスラインがデバイスがバスをスタック状態と見なすまで「スタック」状態を維持する時間。		25	40	65	ms
f <sub>SB_SCLOUT_I2C</sub>	I2C スタックバス回復パターンバスをスタック解除しようとした際に生成されるクロック周波数		5.5	8.5	14	kHz
UART						



## 5.8 スイッチング特性 (続き)

外気温度範囲での動作時 (特に記述がない限り)

パラメータ		テスト条件	最小値	標準値	最大値	単位
$t_{\text{baud}}$	最大 UART ボーレート				2.5	MHz
$t_{\text{UARTTXD}}$	CAN コマンド完了から UART 送信開始までの時間			1	2	U
$t_{\text{UARTRXD}}$	UART 受信から CAN 読み取り準備完了までの時間 <sup>(1)</sup>	最後のデータ / パリティビットの終了を UART.RXN ビットは、 $C_L = 20\text{pF}$ を設定		0.5	2	U
$t_{\text{W(TX)}}$	パルス幅、送信開始、停止、データビット <sup>(1) (2)</sup>	$C_L = 20\text{pF}$	0.98		1.02	U

- (1) 設計により規定されています。  
 (2)  $U = \text{UART のボー時間} = 1 / \text{プログラムされたボーレート}$   
 (3)  $S = \text{SPI クロック周期} = 1 / \text{SPI クロック周波数}$

## 5.9 I<sup>2</sup>C バス タイミング要件

外気温度範囲での動作時 (特に記述がない限り)

			最小値	最大値	単位
I <sup>2</sup> C バス - スタンダード モード					
f <sub>scl</sub>	I <sup>2</sup> C クロック周波数		0	100	kHz
t <sub>sch</sub>	I <sup>2</sup> C クロックの High 時間 <sup>(1)</sup>		4		μs
t <sub>scl</sub>	I <sup>2</sup> C クロックの Low 時間 <sup>(1)</sup>		4.7		μs
t <sub>sp</sub>	I <sup>2</sup> C スパイク時間			50	ns
t <sub>sds</sub>	I <sup>2</sup> C シリアル データ セットアップ時間		250		ns
t <sub>sdh</sub>	I <sup>2</sup> C シリアル データ ホールド時間		0		ns
t <sub>icr</sub>	I <sup>2</sup> C 入力の立ち上がり時間 <sup>(1)</sup>			1000	ns
t <sub>icf</sub>	I <sup>2</sup> C 入力の立ち下がり時間			300	ns
t <sub>ocf</sub>	I <sup>2</sup> C 出力の立ち下がり時間	10pF ~ 400pF バス		300	ns
t <sub>buf</sub>	STOP と START 間の I <sup>2</sup> C バスのフリー時間		4.7		μs
t <sub>sts</sub>	I <sup>2</sup> C START または反復 START 条件の設定		4.7		μs
t <sub>sth</sub>	I <sup>2</sup> C START または反復 START 条件ホールド		4		μs
t <sub>sps</sub>	I <sup>2</sup> C STOP 条件の設定		4		μs
t <sub>vd(data)</sub>	有効データ時間	SCL Low から SDA 出力有効まで		3.45	μs
t <sub>vd(ack)</sub>	ACK 条件の有効データ時間	SCL Low から SDA (出力) Low への ACK 信号		3.45	μs
C <sub>b</sub>	I <sup>2</sup> C バスの容量性負荷			400	pF
I <sup>2</sup> C バス - ファスト モード					
f <sub>scl</sub>	I <sup>2</sup> C クロック周波数		0	400	kHz
t <sub>sch</sub>	I <sup>2</sup> C クロックの High 時間 <sup>(1)</sup>		0.6		μs
t <sub>scl</sub>	I <sup>2</sup> C クロックの Low 時間 <sup>(1)</sup>		1.3		μs
t <sub>sp</sub>	I <sup>2</sup> C スパイク時間			50	ns
t <sub>sds</sub>	I <sup>2</sup> C シリアル データ セットアップ時間		100		ns
t <sub>sdh</sub>	I <sup>2</sup> C シリアル データ ホールド時間		0		ns
t <sub>icr</sub>	I <sup>2</sup> C 入力の立ち上がり時間 <sup>(1)</sup>		20	300	ns
t <sub>icf</sub>	I <sup>2</sup> C 入力の立ち下がり時間		20 × (V <sub>CC</sub> / 5.5V)	300	ns
t <sub>ocf</sub>	I <sup>2</sup> C 出力の立ち下がり時間	10pF ~ 400pF バス	20 × (V <sub>CC</sub> / 5.5V)	300	ns
t <sub>buf</sub>	STOP と START 間の I <sup>2</sup> C バスのフリー時間		1.3		μs
t <sub>sts</sub>	I <sup>2</sup> C START または反復 START 条件の設定		0.6		μs
t <sub>sth</sub>	I <sup>2</sup> C START または反復 START 条件ホールド		0.6		μs
t <sub>sps</sub>	I <sup>2</sup> C STOP 条件の設定		0.6		μs
t <sub>vd(data)</sub>	有効データ時間	SCL Low から SDA 出力有効まで		0.9	μs
t <sub>vd(ack)</sub>	ACK 条件の有効データ時間	SCL Low から SDA (出力) Low への ACK 信号		0.9	μs
C <sub>b</sub>	I <sup>2</sup> C バスの容量性負荷			400	pF
I <sup>2</sup> C バス - 高速モード プラス					
f <sub>scl</sub>	I <sup>2</sup> C クロック周波数		0	1000	kHz
t <sub>sch</sub>	I <sup>2</sup> C クロックの High 時間 <sup>(1)</sup>		0.26		μs
t <sub>scl</sub>	I <sup>2</sup> C クロックの Low 時間 <sup>(1)</sup>		0.5		μs
t <sub>sp</sub>	I <sup>2</sup> C スパイク時間			50	ns

## 5.9 I<sup>2</sup>C バス タイミング要件 (続き)

外気温度範囲での動作時 (特に記述がない限り)

			最小値	最大値	単位
$t_{sds}$	I <sup>2</sup> C シリアル データ セットアップ時間		50		ns
$t_{sdh}$	I <sup>2</sup> C シリアル データ ホールド時間		0		ns
$t_{icr}$	I <sup>2</sup> C 入力の立ち上がり時間 <sup>(1)</sup>			120	ns
$t_{icf}$	I <sup>2</sup> C 入力の立ち下がり時間		$20 \times (V_{CC} / 5.5V)$	120	ns
$t_{ocf}$	I <sup>2</sup> C 出力の立ち下がり時間	10pF ~ 550pF / バス	$20 \times (V_{CC} / 5.5V)$	120	ns
$t_{buf}$	STOP と START 間の I <sup>2</sup> C バスのフリー時間		0.5		μs
$t_{sts}$	I <sup>2</sup> C START または反復 START 条件の設定		0.26		μs
$t_{sth}$	I <sup>2</sup> C START または反復 START 条件ホールド		0.26		μs
$t_{sps}$	I <sup>2</sup> C STOP 条件の設定		0.26		μs
$t_{vd(data)}$	有効データ時間	SCL Low から SDA 出力有効まで		0.45	μs
$t_{vd(ack)}$	ACK 条件の有効データ時間	SCL Low から SDA (出力) Low への ACK 信号		0.45	μs
$C_b$	I <sup>2</sup> C バスの容量性負荷			550	pF

- (1) このパラメータが満たされていることを確認するのは、システム設計者の責任です。この値は、バスの負荷とプルアップ抵抗によって異なります。

## 6 パラメータ測定情報

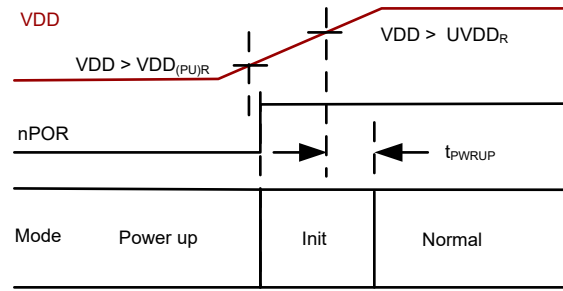


図 6-1. 起動タイミング

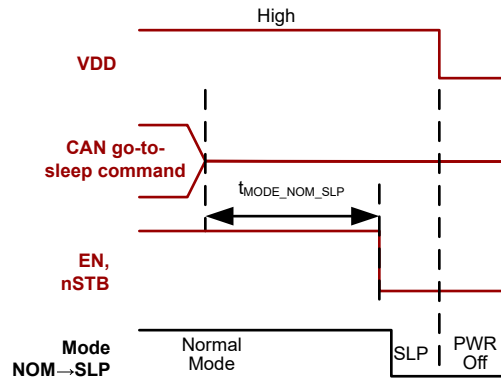


図 6-2. 通常モードからスリープモードへのタイミング

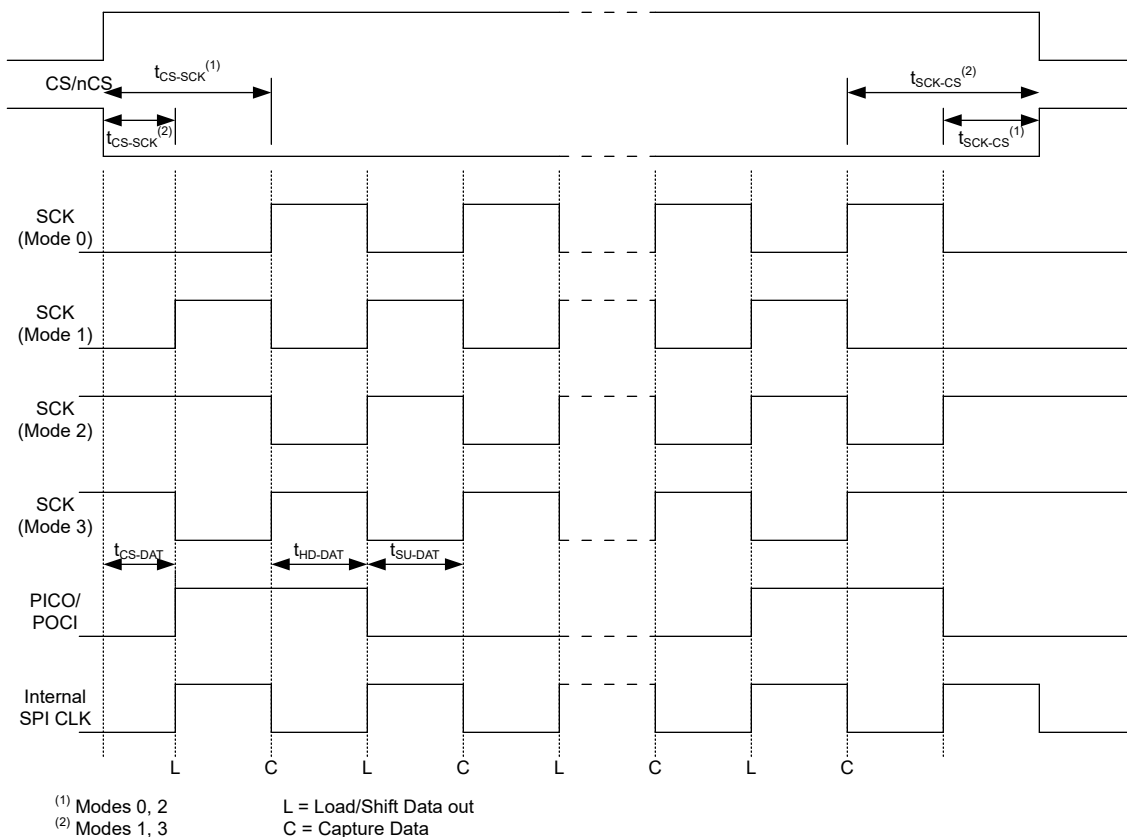


図 6-3. SPI タイミング特性

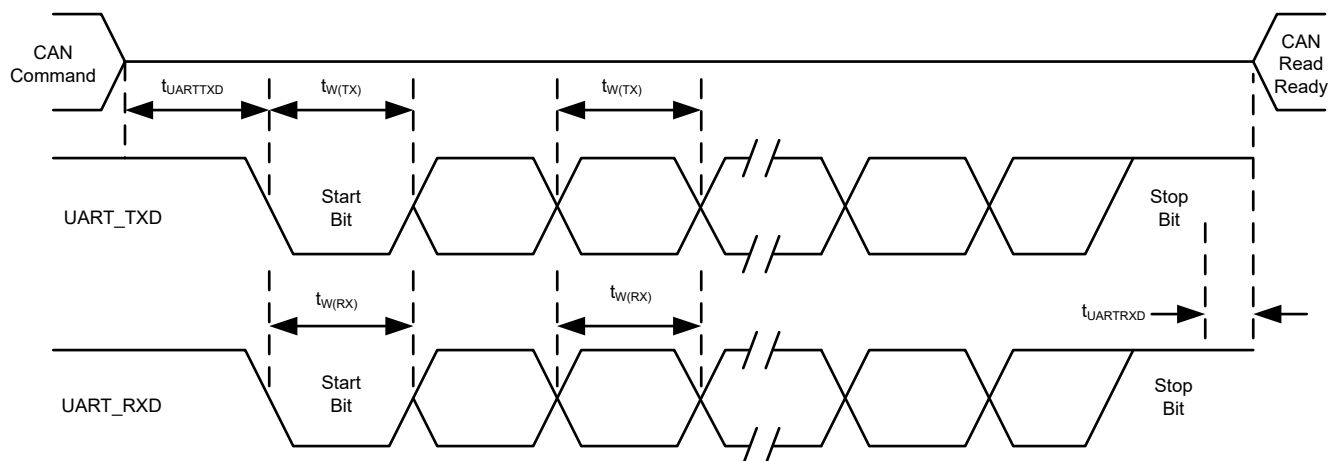
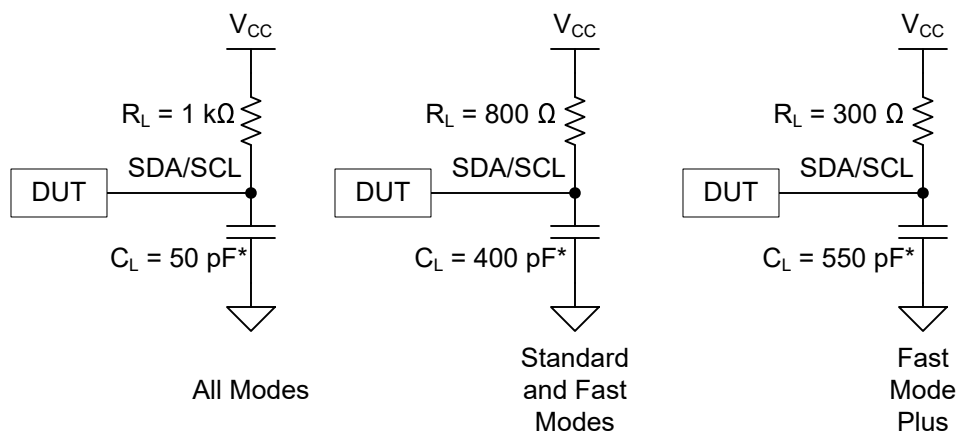


図 6-4. UART タイミング特性



\* $C_L$  includes probe and jig capacitance

図 6-5. I2C バス負荷

## 7 詳細説明

### 7.1 概要

TCAN5102-Q1 は、CiA 604-1 および ISO 11898-1 : 2024 と互換性のあるコントロール エリア ネットワーク (CAN) フレキシブル データ (FD) ライト レスポンダ デバイスで、専用の CAN FD ライト コマンドを使用した場合、最大 5Mbps のデータ レートをサポートできます。このデバイスは、レスポンダ ノード プロセッサを必要としないコマンド レスポンダ アーキテクチャにおいて CAN FD ライト レスポンダ ノード アプリケーションをサポートするように設計されています。レスポンダ ノードへの制御はすべて、コマンド ノード プロセッサからの CAN バスを介して行われるため、レスポンダ ノード プロセッサやソフトウェアは不要です。

デバイスは CAN FD ライト コマンド ノードからデータやコマンドを受信し、これらのコマンドをシリアル パリフェラル インターフェイス (SPI) コントローラ、UART コントローラ、I<sup>2</sup>C コントローラ、GPIO ピンのいずれかまたは両方に変換して、TCAN5102-Q1 が制御しているデバイスやパシフェラル デバイスと通信します。PWM 出力チャネルは、ステッパ モーターを制御するためのハードウェアで台形波ランプ プロファイルもサポートしています。デューティ サイクルまたは周波数を上昇させることができます。外部水晶振動子やクロックは不要です。このデバイスは、外部の TCAN1162x-Q1、TCAN1043A-Q1、TCAN1463A-Q1 CAN FD (SIC) トランシーバを制御し、システム レベルの柔軟性を確保します。このデバイスは、CAN FD トランシーバ / SBC を利用してノードの電力を制御し、CAN RXD (CRXD) ピンを Low にラッチすることでウェークアップ信号を TCAN5102-Q1 に伝達します。

### 7.2 機能ブロック図

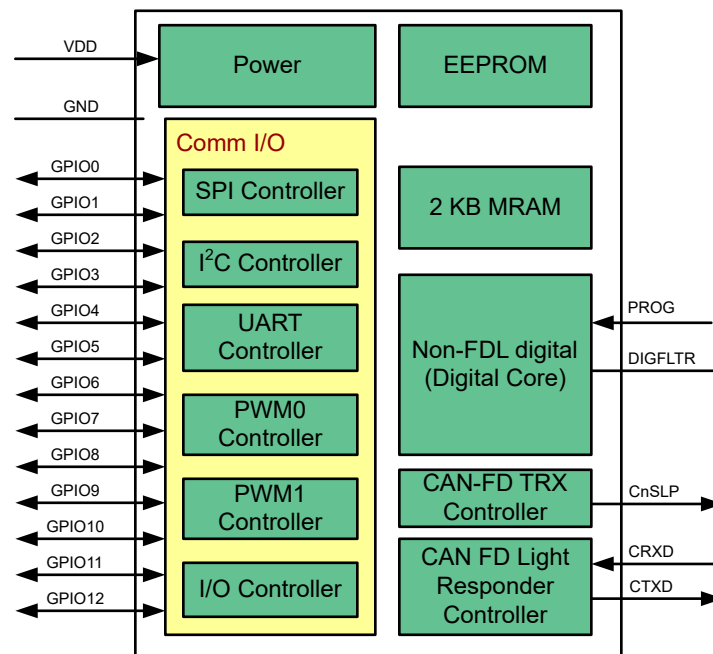


図 7-1. ハイレベルブロック図

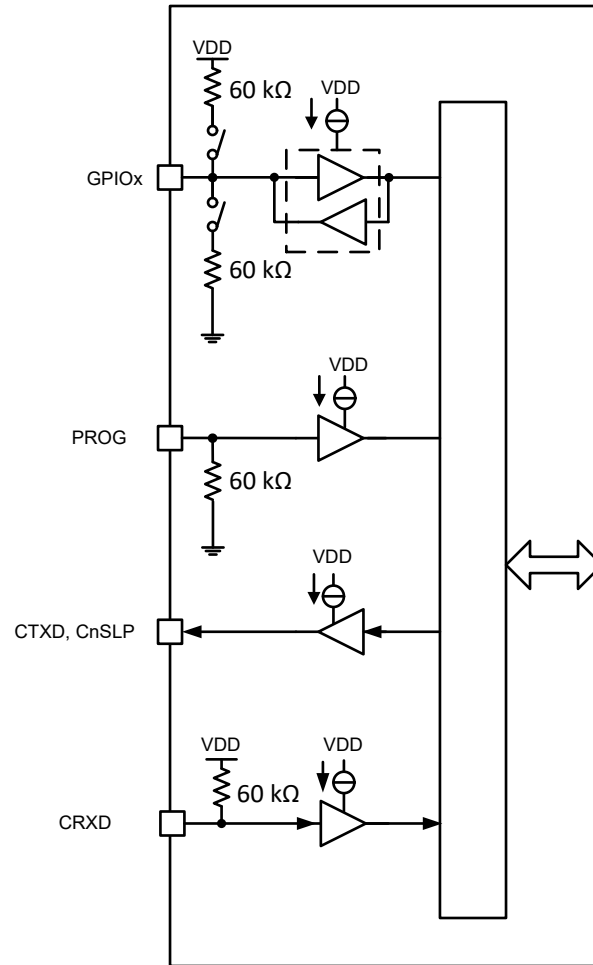


図 7-2. 入力および出力のブロック図



## 7.3 機能説明

### 7.3.1 VDD

VDD はデバイスの電源入力であり、デジタル入出力電圧レベルを 3.3V と 5V に設定します。

### 7.3.2 PROG ピン

PROG ピンは、ECU 開発フェーズで使用されます。電源投入時にこのピンを GND に設定すると、CAN バスを使用してライン終了プログラミング経由で TCAN5102-Q1 をプログラムできます。CAN バスのプログラミングデータ レートは、このプロセスでは 1Mbps に設定されています。

### 7.3.3 DIGFLTR ピン

このピンを使用して、内部デジタル コア レギュレータのフィルタリングを行います。フィルタ コンデンサの推奨値については、「推奨動作条件」表を参照してください。

### 7.3.4 GPIOx およびピンの多重化の特殊機能

TCAN5102-Q1 には 12 本の GPIO ピンがあり、複数の用途に構成できます。汎用 I/O として、このピンは入力、プッシュプル出力、またはオープン ドレイン出力に設定できます。また、これらのピンにはユーザーが有効なプルアップ / プルダウン抵抗も内蔵されています。

デバイス内のさまざまなシリアル インターフェイスと IP ブロックをサポートするために、複数のピンに GPIO と多重化するオプションがあります。追加機能はレジスタ設定内で「特殊機能」と呼ばれ、この場合ピンはもはや GPIO として動作せず、代わりに特殊な IP ブロックに接続されます。特殊機能用に構成している場合、このピンにはプルアップおよびプルダウン抵抗構成のみが使用されます。特殊機能を選択すると、出力 / 入力、プッシュプル / オープン ドレインなどのモードが自動的に設定されます。たとえば、SPI POCI に対応する GPIO が特殊機能として選択されている場合、その GPIO の設定が SPI IP によって上書きされ、その GPIO が入力になるように構成されます。必要に応じて、そのピンで内部プルアップまたはプルダウン抵抗を有効にすることで、既知のピン状態を実現できます。

#### 注

一部のピンには、2 つの特殊機能が割り当てられています。競合のイネーブルが発生した場合、プライオリティの高い IP が優先されます。たとえば、SPI IP ブロックが SPI チップセレクトチャンネル 4 (CS4) をイネーブルにしておき、I<sup>2</sup>C IP もイネーブルになっている場合、GPIO9 は I<sup>2</sup>C の SCL ピンに多重化されます。すべてのピンのデフォルトのプログラムされていない動作は、入力で高インピーダンスです。

GPIO のデフォルトのパワーオン状態を EEPROM に保存することで、パワーオン応答を行えます。電源オフ時、または POR スレッシュホールドを下回っているとき、これらのピンは高インピーダンス状態になります。デバイスの電源投入後、構成が EEPROM から GPIO レジスタにコピーされます。出力動作に影響を与える可能性のある構成レジスタのみが EEPROM に保存されます。保存されるレジスタは以下の通りです

1. IO\_OE\_0 および IO\_OE\_1
2. IO\_RE\_0 および IO\_RE\_1
3. IO\_PU\_0 および IO\_PU\_1
4. IO\_OUTPUT\_0 および IO\_OUTPUT\_1

表 7-1. 特殊機能向けのピンのマルチプレクシング

ピン	名称	低優先度	高優先度
4	GPIO9	SPI CS4	I <sup>2</sup> C SCL
5	GPIO10	SPI CS5	I <sup>2</sup> C SDA
6	GPIO11	SPI CS6	PWM0
7	GPIO12	SPI CS7	PWM1
11	GPIO8	UART RXD	-
12	GPIO7	UART TXD	-
13	GPIO6	SPI PICO	-

表 7-1. 特殊機能向けのピンのマルチプレクシング (続き)

ピン	名称	低優先度	高優先度
14	GPIO5	SPI POCI	-
15	GPIO4	SPI SCK	-
16	GPIO3	SPI CS0	-
17	GPIO2	SPI CS1	-
18	GPIO1	SPI CS2	-
19	GPIO0	SPI CS3	-

#### 7.3.4.1 GPIO 同期

出力として構成された GPIO は、同時に更新をサポートします。この GPIO 出力の更新を行うには、2 つのモードがあります

1. デフォルト：IO\_OUTPUT\_x レジスタが更新されると、GPIO は直ちに更新されます (各バイト)
2. 同期：更新は IO\_OUTPUT\_1 への書き込み後にのみ行われます。すべてを同期するには、まず IO\_OUTPUT\_0 に書き込み、次に IO\_OUTPUT\_1 への書き込みが完了すると、すべてのビットが同時に更新されます

この機能は、GPIO\_OUT\_SYNC ビットを 1 にセットすることで、MRAM\_IP\_CFG レジスタで有効になります。

#### 7.3.5 EEPROM

TCAN5102-Q1 は EEPROM を内蔵しており、電源オン構成時に以下の情報を保存するために使用されます：

1. CAN バスのデータレート
2. CAN ID
3. ブロードキャスト CAN ID およびマスク
4. 構成ロック (データ レートおよび CAN ID)
5. パワーオン リセット ピンの状態を設定するのに必要な GPIO 構成
  - a. IO\_OE\_0 および IO\_OE\_1
  - b. IO\_RE\_0 および IO\_RE\_1
  - c. IO\_PU\_0 および IO\_PU\_1
  - d. IO\_OUTPUT\_0 および IO\_OUTPUT\_1

#### 注

EEPROM には CRC フィールドがあり、CRC に故障が検出されると、デバイスは CAN インターフェイスを無効にし、GPIO を高インピーダンス状態に保持します

#### 7.3.6 SPI コントローラ

TCAN5102-Q1 は SPI コントローラを搭載しており、8 つのチップセレクトと、SPI チャネル 0 ~ 3 に異なる SPI 設定を備えています。最後の 4 チャネルは、SPI チャネル 3 と構成を共有します。4 つの最初のチャネルはそれぞれ、独立した SPI 速度、SPI モードを持つことができます。別のチップ セレクトを使用すると、共有の SPI POCI、PICO、SCK ラインとの通信を実行できます。チップ セレクトは、アクティブ High とアクティブ Low のどちらかに構成できます。メッセージはメッセージ RAM (MRAM) に格納され、IP が機能するには MRAM 内の領域を割り当てる必要があります。詳細については「[セクション 7.5.3](#)」を参照。

SPI コントローラの構成と使用方法の詳細については、[セクション 7.5.4](#) を参照してください。

#### 7.3.7 UART コントローラ

TCAN5102-Q1 は、UART コントローラを搭載しています。UART は、LIN、RS232、RS485、Flexwire など、さまざまなインターフェイスで使用する標準的なバイト指向プロトコルです。このコントローラは分数ボーレート分周器を備えており、多様なインターフェイスで利用できる幅広いボーレートに対応しています。

メッセージはメッセージ RAM (MRAM) に格納され、IP が機能するには MRAM 内の領域を割り当てる必要があります。詳細については「[セクション 7.5.3](#)」を参照。MRAM を使用せずに、UART IP をシングルバイト モードで実行できます。このモードには、UART 構成レジスタにイネーブルビットがあります。

UART コントローラの構成と使用方法の詳細については、[セクション 7.5.5](#) を参照してください。

### 7.3.8 I2C コントローラ

TCAN5102-Q1 には I<sup>2</sup>C コントローラが搭載されており、標準、高速モード、高速モード プラスをサポートしています。標準のコントローラに加えて、スタック バス回復機能も内蔵されており、SDA ラインが Low に固着したことを検出し、バスを解除しようとする SCL ラインをトグルできます。また、クロック ストレッチ時間が  $t_{\text{STUCKBUS\_I2C}}$  時間より短い限り、クロック ストレッチもサポートされます。

メッセージはメッセージ RAM (MRAM) に格納され、IP が機能するには MRAM 内の領域を割り当てる必要があります。詳細については「[セクション 7.5.3](#)」を参照。

I<sup>2</sup>C コントローラの構成と使用方法の詳細については、[セクション 7.5.6](#) を参照してください。

#### 7.3.8.1 I2C スタック バス回復

I2C コントローラには、スタック バス回復 (SBR) と呼ばれるオプション機能があります。スタック バス回復は、SDA または SCL ラインが Low に固着した場合にバスを修正しようとします。これは、バス上にノイズがあり、I2C ペリフェラルが誤ったクロック数を認識するようになる場合に発生する可能性があります。スタックバス回復により、SCL ラインが最大 16 回トグルされ、問題の原因となるデバイスをバスの解放に試みます。詳細については、[セクション 7.5.6.2](#) を参照してください。

### 7.3.9 PWM コントローラ

TCAN5102-Q1 は、ランプ機能を内蔵した PWM コントローラを備えています。PWM コントローラは、固定デューティ サイクルと周波数による静的 / 一般的な PWM 出力をサポートするだけでなく、デューティ サイクルまたは周波数をランプさせる高度な PWM 機能もサポートしており、基本的なモーター制御に対応できます。このコントローラはステッピング モーター制御を念頭に開発されており、ユーザーが加速および減速プロファイルをプログラムできるようになっています。コントローラがランプを開始するように指示されると、演算を行い、指定されたランプ プロファイルを PWM 出力で実行します。

PWM コントローラは、PWM パルス カウントなどの追加機能もサポートしており、ユーザーは PWM 出力を設定して加速および減速プロファイルを生成し、所定のパルス数 / ステップ数の後に自動的に停止させることができます。シーケンスはレジスタにプログラムされ、ランプを開始するために指示される以外、アップストリームコントローラからの介入は必要ありません。

追加機能として、GPIO 入力を使用して、減速ランプを開始するか、PWM 出力を直ちにオフにすることができます。これはリミット スイッチやモーターの故障検出に役立ちます。

PWM コントローラの構成と使用方法の詳細については、[セクション 7.5.7](#) を参照してください。

#### 7.3.10 CAN トランシーバの制御ピン

TCAN5102-Q1 には CAN トランシーバが内蔵されておらず、外部トランシーバが必要です。この結果、ユーザーはサポートする CAN トランシーバを選択できます。このデバイスには、CAN TXD、RXD、nSLP ピンがあります。nSLP ピンを使用して、スリープ モードへの移行時に外部トランシーバのスリープ モードまたはスタンバイ モードを制御します。

#### 7.3.11 低電圧ロックアウトおよび電源オフのデバイス

TCAN5102-Q1 は、デバイス (VDD) の電源レールを監視します。VDD の低電圧が監視されます。UVDD 故障が検出されると、デバイスに対応する状態変化があります。

表 7-2. VDD 故障とデバイス モード

VDD	デバイス モード
> UVDD	正常
< UVDD	パワーオン オン リセット

## 7.4 デバイスの機能モード

TCAN5102-Q1 には、電源オフ、初期化、スリープ、通常モードの 4 つの動作モードがあります。各モードでの各種デバイスブロックの動作については、表 7-3 を参照してください。全体の動作モードがどのように相互作用するかについては、図 7-3 を参照してください。

表 7-3. モードの概要

ブロック	init	正常	スリープ
CAN トランシーバ制御	オフ (Hi-Z)	オン	オフ
CAN nSLP	オフ (Hi-Z)	High (VDD)	Low (GND)
SPI	オフ (Hi-Z)	プログラムされている場合はオン	オフ
I <sup>2</sup> C	オフ (Hi-Z)	プログラムされている場合はオン	オフ
UART	オフ (Hi-Z)	プログラムされている場合はオン	オフ
GPIOx	オフ (Hi-Z)	プログラムによる	プログラムによる

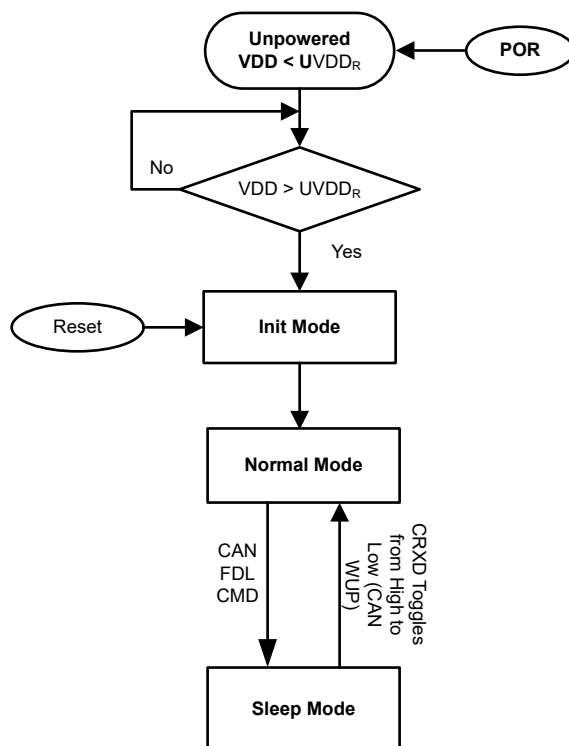


図 7-3. TCAN5102-Q1 動作モード

### 7.4.1 初期モード

これは、電源投入時の初期動作モードです。これは、VDD が  $UV_{DD}$  よりも大きくなり、保存されたデバイス構成がロードされると移行される遷移モードです。設定が読み込まれると、デバイスはノーマル モードに移行します。

init モードでは EEPROM が読み取られ、EEPROM が読み取られると、構成がレジスタにコピーされ、デバイスは通常モードに遷移します。保存する構成の詳細については、[セクション 7.3.5](#) を参照してください。

#### 7.4.2 スリープモード

スリープモードは、このデバイスにおける最も低消費電力の状態です。TCAN5102-Q1 は、CAN バスからスリープ移行コマンドを受信し、次の操作を実行します：

- 内部のハウス キーピング処理を実行します
- 外部 CAN トランシーバをスリープモードにします

CAN トランシーバがスリープモードに移行すると、VDD に電力は想定されません。デバイスは電源が供給されていない状態です。スリープモード中でも VDD に給電し続けることが可能です。CAN トランシーバからの WUP 信号を受信するまで、このデバイスは低消費電力状態のまま維持されます。

#### 7.4.3 通常モード

INIT モードを終了すると、デバイスは通常モードに移行します。ノーマルモードではトランシーバがアクティブになり、デバイスはデータの受信または送信準備が整います。

### 7.5 プログラミング

デバイスは、構成に CAN FD Light インターフェイスを使用します。構成は、デバイス上のレジスタに書き込むことで行われます。デバイス レジスタへのアクセスに使用する特定の CAN FD メッセージ構造があります。このデバイスがデバイス レジスタへのアクセスのために、複数のフレーム形式をサポートします。

#### 7.5.1 SPI ペリフェラル モードによるデバイスプログラミング

構成プログラミングには 2 つのオプションがあります：

1. CAN FD ライトを使用したプログラム (PROG ピンを使用してボーレートを強制的に 1Mbps に設定)
2. SPI を使用する場合 (PROG ピンと RXD ピンでトグルシーケンスを行い、ローカルの SPI ペリフェラルを有効にして、プログラマが EEPROM に書き込みできるようにします)

CAN インターフェイスが使用できない場合、EEPROM をプログラムする手段として SPI ペリフェラルモード方式を使用できます。この方法では、ローカル SPI ペリフェラルをイネーブルにするために、特定のイベントシーケンスが必要です。デバイスが SPI ペリフェラルモードの場合、CAN インターフェイスはデイスエーブルされ、電源を入れ直すまでデイスエーブルのままになります。

#### 7.5.2 CAN FD Light プロトコル

TCAN5102-Q1 は、論理リンク制御 (LLC) と FD ベース フレーム フォーマット (FBFF) を使用することにより、CiA 604-1 および CiA 604-3 通信規格をサポートします。[表 7-4](#) を参照してください。

**表 7-4. プロトコル CiA フォーマット**

フィールド	サイズ	機能
識別子：	11 ビット	ベース識別子 (ターゲット ID またはブロードキャスト ID)
DLC	4 ビット	ISO 11898-1:2015 の表 5 に指定されているデータ長コード
LLC データ	0~64 バイト	データ フレームのデータ コンテンツ
送受信されるすべてのフレームが FBFF であり、かつビット ESI = 0 および BRS = 0 であるため、フォーマット フィールド内のすべてのビットは固定されています。そのため、フォーマット フィールドは省略されます。		

デバイス プロトコルは要求と応答に分割されます。デバイス要求とは、このデバイスに対して特定のレジスタへの読み取りまたは書き込みを実行するよう求める要求のことです。要求には常に 3 つ以上のフィールドがあり、任意の操作をデコードするために必要です。これらのフィールドは常に、CAN フレーム データ セクションの開始位置にあります。

1. オペコード (Op Code)：これは、デバイスに対して、レジスタまたは FIFO に対する読み取りか書き込みが行われているのかを示します
2. 長さ：転送されるデータの合計サイズ (バイト単位) を示します (デバイスのヘッダの後)



### 3. アドレス登録書き込みまたは読み取り中のレジスタまたは FIFO の 16 ビットアドレス

デバイス レスポンスは、リクエストを受信したことを要求元のデバイスに知らせ、そのリクエストが処理中であるか、あるいは無効であることを伝えるために送信されます。応答には、少なくとも 3 つのフィールドが含まれています

1. オペコード (Op Code) : 受信したオペコードを繰り返します
2. 長さ : ヘッダの後に返されるデータの合計サイズ (バイト単位) を示します。リクエストが書き込みの場合は、書き込まれたバイト数を反映しますが、実際のデータは送信されません。長さフィールドは、書き込まれたバイト数を確認するために使用されます
3. ステータス バイト : リクエストのステータスを示します。これには、OK/ 成功、または無効なアドレスや長さなどの何らかのエラーがあります。

デバイスで使用する識別子には 2 種類あります。ターゲット識別子は、個々のレスポンスに一意的な特定のアドレスです。これにより、コマンドは個々のノードをアドレス指定 / 設定できます。ブロードキャスト識別子もサポートされており、ブロードキャスト ID およびマスクフィールドを使用して 1 つまたは複数の ID に一致します。ブロードキャスト ID からのリクエストは実行されますが、CAN FD Light ではアービトレーションがサポートされていないため、レスポンスからの応答はありません。ブロードキャストは、コマンドが一度に多数のノードを構成する場合や、複数のノードをスリープ状態にする場合に便利です。しかし、これはレスポンス フレームを破棄するため、レスポンスによって確認されることはありません。

#### 7.5.2.1 CAN フレーム形式

標準 CAN フレーム形式では、CAN ペイロードの 3 バイトをヘッダーとして使用し、オペコード (op code)、データ長、アドレスの 3 つのフィールドを含みます。

CAN ID		Byte 0	Byte 1	Byte 2	Byte 3	
Device ID[10:0]	CTRL	OP[7:6]/ LEN[5:0]	ADDR[15:8]	ADDR[7:0]	Data	...

図 7-4. 標準 CAN フレーム形式要求の例

表 7-5. 標準 CAN フレーム形式の要求ヘッダ フィールド

バイト	ビット	フィールド	説明
0	7:6	OPCODE	デバイスのオペコード 2'b00 = 書き込み 2'b01 = 読み出し 2'b10 = 予約済み 2'b11 = 予約済み
	5:0	LEN	このヘッダ有効値 (書き込み) 後に転送されるデータ バイト数 : 0 ~ 61 有効な値 (読み取り) : 0 ~ 62
1	7:0	ADDR[15:8]	アドレスの上位バイト
2	7:0	ADDR[7:0]	アドレスの下部バイト

#### 注

複数バイトの読み出しの場合、開始アドレスがシリアル通信ブロック (I2C、SPI、UART) の RX FIFO の前である場合、FIFO は実際には読み出されません。代わりに、RX FIFO に対応するバイトは 0 を返し、読み取りが続行されます。これは、RX FIFO でバースト読み出しが「スタック」しないようにするためです。読み取りの開始アドレスが RX FIFO から始まる場合、読み取られるすべてのバイトは RX FIFO からのみ得られます。

CAN ID		Byte 0	Byte 1	Byte 2	
Device ID[10:0]	CTRL	OP[7:6]/ LEN[5:0]	STATUS_BYTE	Data	...

図 7-5. 標準 CAN フレーム形式応答の例

**表 7-6. 標準 CAN フレーム形式応答のヘッダ フィールド**

バイト	ビット	フィールド	説明
0	7:6	OPCODE	デバイスのオペコード 2'b00 = 書き込み 2'b01 = 読み出し 2'b10 = 予約済み 2'b11 = 予約済み
	5:0	LEN	要求されたデータ バイト数が有効であれば、ヘッダの後にバイト数をエコー送信します。  <b>注</b> 応答が書き込み要求に対するものである場合、長さフィールドには要求されたバイト数が入りますが、ヘッダーの後にデータは送信されません。これにより、応答フレームが書き込み要求に対するものであることの確認が可能になります
1	7:5	RSVD	予約済み
1	4	FE	前の要求のフレーミング エラー。要求は無視されます。通常、DLC が 3 バイト未満の場合に発生し、ヘッダには最低要件があります
1	3	IO	無効な操作です。要求は無視されます
1	2	IL	無効な長さです。要求は無視されます
1	1	IA	アドレスが無効です。要求は無視されます
1	0	OK	OK リクエスト、リクエストの完了、または進行中です

### 7.5.3 メッセージ RAM (MRAM) および IP イネーブル

TCAN5102-Q1 には 2kB のメッセージ RAM (MRAM) があり、SPI、I2C、UART 通信のメールボックスを格納するために使用されます。この RAM は、エンド アプリケーションのニーズに基づき、2 つの IP ブロックに割り当てられたメモリを任意の組み合わせで 25% 単位に分割できます。RX と TX の間の分割は調整できず、常に 50/50% です。RX および TX FIFO の処理は TCAN5102-Q1 によって行われるため、各 IP ブロックに割り当てる MRAM の量を選択しないでユーザーの設定は不要です。

SPI と I2C はメモリ使用に関して単純な分割を行い、割り当てられたメモリの 50% は TX データの格納に使用され、残りの 50% は指定されていれば RX データの格納に使用されます。

UART では、受信した各バイトのステータスを格納するために追加のバイトが必要です。つまり、UART で受信する各データ バイトごとに、そのバイトのステータスを格納するために 1 バイトが追加で必要になります。その結果、UART に割り当てられたメモリの 3 分の 2 だけが実際のデータ バイトの格納に使用されます。

表 7-7 に、各 IP ブロックの TX バッファまたは RX バッファに割り当てられるデータのバイト数を示します。表示されるサイズは TX バッファと RX バッファの両方で同じ値ですが、両方で共有される値ではありません。たとえば、MRAM 領域の 100% が SPI に割り当てられている場合、SPI TX サイズは 1024 バイト、SPI RX バッファは 1024 バイトです。TX バッファと RX バッファを合わせると合計で 2048 バイトとなり、これがメモリ全体に相当します。

MRAM の 50 % が SPI と UART に割り当てられた例では、SPI の RX と TX のサイズはそれぞれ 512 で、合計 1 kB となります。UART IP は、バッファごとにわずか 340 バイトです。UART RX では、受信した各バイトのステータスを格納するために追加のバイトが必要となるため、利用可能な 1024 バイトのうちデータ格納に使用されるのは 3 分の 2 だけです。内部では、TX バッファと RX バッファに保存できるバイト数は同一です。このため、残りの 4 バイトは 3 つの機能 (TX、RX、RX ステータス) に均等に分割できないため、4 バイトのデータは未使用のままになります。

#### 注

MRAM のアロケーション設定は、SPI および I2C 用の IP モジュールを有効化するために使用されます。モジュールにメモリがアロケーションされると、モジュールは有効化されます。ただし、モジュールで使用する GPIO は、GPIO 構成レジスタで手動で特殊機能に設定する必要があります。す。

表 7-7. MRAM アロケーション

SPI へのアロケーション %	SPI TX または RX サイズ (バイト)	UART TX または RX サイズ (バイト)	使用された合計バイト数 (合計に対する %)
100%	1024	0	2048 (100%)
75%	768	168	2040 (99.6%)
50%	512	340	2044 (99.8%)
25%	256	512	2048 (100%)
0%	0	680	2040 (99.6%)

表 7-8. MRAM アロケーション

MRAM_IP_EN (16 進)	SPI へのアロケーション %	UART へのアロケーション %	I2C へのアロケーション %	SPI TX および RX サイズ (バイト)	UART TX および RX サイズ (バイト)	I2C TX および RX サイズ (バイト)	使用された合計バイト数 (合計に対する %)
0h	0%	0%	0%	0	0	0	0 (0%)
1h	0%	100%	0%	0	680	0	2040 (99.61%)
2h	25%	75%	0%	256	512	0	2048 (100%)
3h	50%	50%	0%	512	340	0	2044 (99.8%)
4h	75%	25%	0%	768	168	0	2040 (99.61%)
5h	100%	0%	0%	1024	0	0	2048 (100%)
6h	0%	0%	100%	0	0	1024	2048 (100%)
7h	0%	25%	75%	0	168	768	2040 (99.61%)
8h	0%	50%	50%	0	340	512	2044 (99.8%)
9h	0%	75%	25%	0	512	256	2048 (100%)
Ah	25%	0%	75%	256	0	768	2048 (100%)
Bh	50%	0%	50%	512	0	512	2048 (100%)
Ch	75%	0%	25%	768	0	256	2048 (100%)



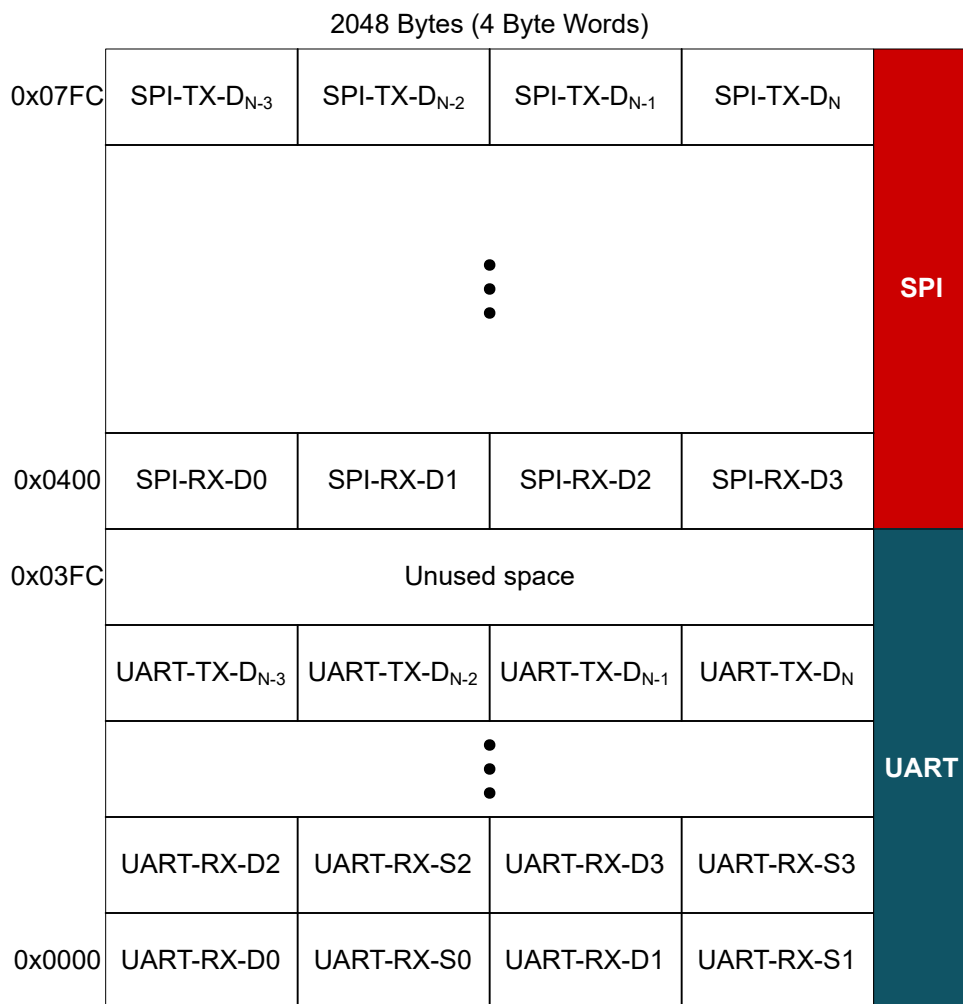


図 7-6. MRAM レイアウト例 (SPI と UART を有効化し、50 % で分割した場合)

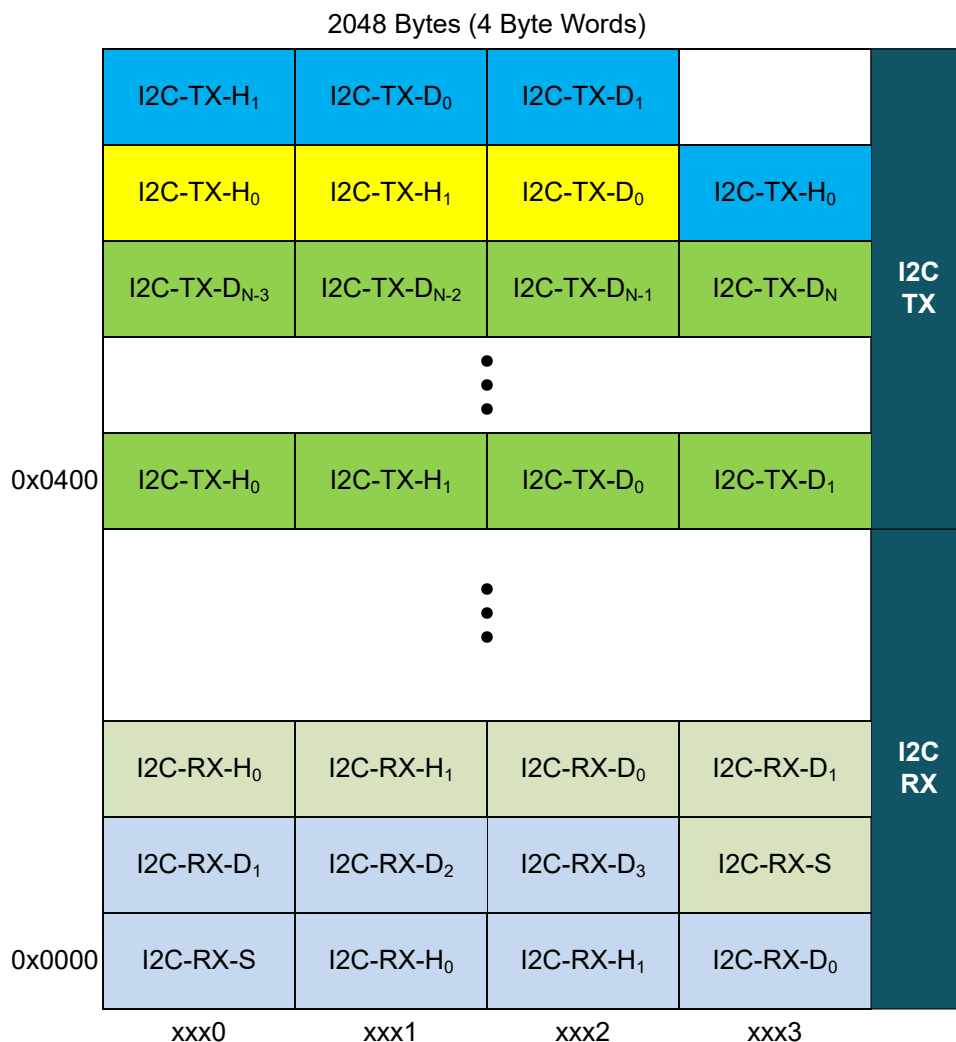


図 7-7. サンプル データ付き MRAM レイアウト例 (MRAM の 100 % を I2C に割り当てた場合)

上の図には、MRAM が I2C のデータを保持する方法の例が示されています。各 I2C フレーム / メッセージは異なる色で示され、各 RX または TX FIFO 内でメッセージが密にグループ化される様子を表しています。ユーザーは、FIFO 内でデータがどのように格納されるかを意識する必要はなく、FIFO の合計サイズと、ヘッダやステータスバイトもメモリに格納されるという事実だけを把握していれば十分です。

#### 7.5.4 SPI コントローラ

##### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があり (セクション 7.3.4 およびセクション 7.6.1.18 を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります (セクション 7.5.3 およびセクション 7.6.1.17 を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な SPI 構成レジスタを、セクション 7.6.2 に示します。

##### 7.5.4.1 SPI ピン

SPI 通信では通常、他の SPI デバイスとの通信に 4 本のピンを使用します：クロック (SCLK)、データ入力 (PICO)、データ出力 (POCI)、およびチップ セレクト (CS または nCS)。

#### 7.5.4.1.1 SPI クロック (SCLK)

SPI クロック出力ピンは、ターゲット ペリフェラルとの間でデータの入出力を同期させるためのクロックを生成します。このピンは、4 つの SPI モード (2 種類のクロック極性) のいずれにも設定でき、下流ペリフェラルとの最大限の柔軟性と互換性を提供します。

#### 7.5.4.1.2 ペリフェラル入力とコントローラ出力 (PICO)

この出力ピンは、ペリフェラル デバイス (ダウンストリーム デバイス) にデータをシフトするために使用されます。チップ セレクト信号がアサートされると (設定により High または Low のいずれか)、PICO はターゲット / ペリフェラル デバイスにデータをシフト出力します。このピンは、ペリフェラル / ターゲット デバイスのシリアル データ入力 (SDI) ピンに接続されます。

#### 7.5.4.1.3 ペリフェラル出力、コントローラ入力 (POCI)

この入力ピンは、ペリフェラル / 下流デバイスからデータをシフト出力し、このデバイス / コントローラに入力するために使用されます。POCI ピンは、通常ペリフェラル / ターゲット デバイスのシリアルデータ出力 (SDO) ピンに接続します。

#### 7.5.4.1.4 チップ セレクト (CS または nCS)

チップ セレクト出力ピンは、ターゲット ペリフェラルとの通信時にそのことを通知するために使用されます。このデバイスは、個別の極性選択機能を備えた最大 4 つの独立したチップ セレクト チャネルをサポートしています。

#### 7.5.4.2 SPI クロック ジェネレータ

TCAN5102-Q1 SPI モジュールには、8 ビットの除数を使用するプログラマブル クロック ジェネレータが内蔵されています。SPI\_DR レジスタは、SPI\_DR レジスタ内の値よりも 1 つ大きいと解釈され、半周期で使用されます。除数の式は次のとおりです：

$$\text{Divisor} = (\text{Clock Frequency} / (\text{Desired SPI clock frequency} \times 2)) - 1 \quad (1)$$

クロック周波数は 40MHz です。この式は、サポートされる最大 SPI 周波数が 20MHz で、最小周波数が 78,125Hz であることを示しています。

表 7-9. 共通 SPI クロック

必要な SPI クロック	クロックの生成に使用される除数	SPI_DR 値 (HEX)
80kHz	249	0xF9
100kHz	199	0xC7
200kHz	99	0x63
250kHz	79	0x4F
500kHz	39	0x27
1MHz	19	0x13
2MHz	9	0x09
4MHz	4	0x04
5MHz	3	0x03
6.67MHz	2	0x02
10MHz	1	0x01
20MHz	0	0x00

#### 7.5.4.3 SPI 制御プロトコル

SPI の TX または RX FIFO を読み書きする際には、対象ペリフェラルへの読み書きをより堅牢にするために、追加の 2 バイトのヘッダが付加されます。2 つの形式があり、いずれも類似していますが、SPI の TX または RX FIFO への読み取りが書き込みかによって異なります。

RX FIFO を読み出す際、データ フィールドの先頭には常に 2 バイトのヘッダが付加されます。このヘッダには、残りのバイト数や、読み出しが前回の読み出しの継続であるかどうかの情報が含まれています (単一の CAN フレームに収まりきれない大きな SPI メッセージに必要です)。

TX FIFO に書き込む場合、メッセージの最初の書き込みのみに 2 バイトのヘッダが含まれます。

ヘッダ形式の詳細については、[SPI 受信 FIFO \(アドレス = h1010\)](#) および [SPI 送信 FIFO \(アドレス = h1010\)](#) を参照してください。

#### 7.5.4.3.1 SPI 書き込み例 1

これは、1 つの CAN フレームに収まる SPI 経由のバイト ストリーム送信の基本的な例です。デバイスが SPI フレームのすべてのバイト (SPI ヘッダで指定されたバイト数によって決定) を受信すると、SPI フレームの送信を開始します。

#### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があり ([セクション 7.3.4](#) および [セクション 7.6.1.18](#) を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります ([セクション 7.5.3](#) および [セクション 7.6.1.17](#) を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な SPI 構成レジスタを、[セクション 7.6.2](#) に示します。

表 7-10. 設計パラメータと前提条件

パラメータ	値
CAN フレーム フォーマット	標準フレーム フォーマット
転送するバイト数	10'B
SPI チャネル	チャネル 1
受信データを RX FIFO に保存	なし
SPI バイト ストリーム	0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA

次の表は、手順と CAN バス上で送信されるデータを示しています。バスの中央ゲートウェイ / コントローラは ECU と呼ばれ、TCAN デバイスは「レスポнда」と呼ばれます。

- 太字で示された 16 進値は、CAN フレーム ヘッダのバイトを表します (この例では標準フレーム形式です)
- イタリック体で示された 16 進値は、SPI ヘッダ バイトを表します
- 書式なしで示された 16 進値は、SPI データ バイトです

表 7-11. CAN シーケンスの例 (シングル CAN フレーム)

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 16B)	<b>0x0C</b> , <b>0x10</b> , <b>0x10</b> , 0x01, 0x0A, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00	ECU は、レジスタ 0x1010 (SPI TX FIFO) に 10 バイトの書き込みを要求します。SPI ヘッダには、store = 0、SPI チャネル = 1、および 10 バイトの SPI フレームが含まれます。その後、10 バイトのデータはデバイスにシフト インされます。これら 15 バイトのデータを取るため、CAN DLC は 16 バイトでなければならないため、末尾に 0x00 がパディングされます。
2	レスポнда	<b>0x0C</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	SPI	PICO: 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA	この時点で、レスポндаは SPI チャネル 1 で 10 バイトの SPI 転送を実行します。

**表 7-12. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム)**

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 8B)	<b>0x05</b> , <b>0x10</b> , <b>0x10</b> , 0x01, 0x0A, 0x11, 0x22, 0x33	ECU は、レジスタ 0x1010 (SPI TX FIFO) に 5 バイトの書き込みを要求します。SPI ヘッダには、store = 0、SPI チャンネル = 1、および 10 バイトの SPI フレームが含まれます。SPI フレームの最初の 3 バイトのデータだけが転送されました。
2	レスポнда	<b>0x05</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	ECU (DLC = 3B)	<b>0x42</b> , <b>0x10</b> , <b>0x12</b>	ECU は、TX FIFO ステータスおよび TX s ステータス レジスタを読み取るために、2 バイトの読み出しを要求します。この手順はオプションですが、部分的な書き込みに役立つ情報がいくつか表示されます。
4	レスポнда (DLC = 4B)	<b>0x42</b> , <b>0x01</b> , 0x0C, 0x87	SPI_TXFS によると、TX FIFO には 12 バイト分の空き領域があるため、オーバーフローの危険はありません。SPI_TXES は、予想どおり TXEIP (TX 素子が進行中) フラグがセットされており、現在の SPI フレームがメモリ内で未完成であり、さらに 7 バイトのデータ (TXEBP フィールド) を必要としていることを示しています
5	ECU (DLC = 12B)	<b>0x07</b> , <b>0x10</b> , <b>0x10</b> , 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00, 0x00	ECU は、同じアドレスへの 7 バイトの書き込み要求を継続します。このフレームを格納できる CAN DLC は 12 バイトであるため、末尾に 2 バイトのパディングが追加されます。これらのパディング バイトの値は問題にならず、無視されます。
6	レスポнда	<b>0x07</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
7	SPI	PICO: 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA	この時点で、レスポндаは SPI チャンネル 1 で 10 バイトの SPI 転送を実行します。POCI は無視されます。

#### 7.5.4.3.2 SPI のバイト読み取り例 1

これは、SPI を介してバイト ストリームを送受信する基本的な例です。デバイスが SPI フレームのすべてのバイト (SPI ヘッダで指定されたバイト数によって決定) を受信すると、SPI フレームの送信を開始します。この例では、受信した POCI データを格納し、後で読み戻すことができます。

#### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があり ([セクション 7.3.4](#) および [セクション 7.6.1.18](#) を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります ([セクション 7.5.3](#) および [セクション 7.6.1.17](#) を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な SPI 構成レジスタを、[セクション 7.6.2](#) に示します。

**表 7-13. 設計パラメータと前提条件**

パラメータ	値
CAN フレーム フォーマット	標準フレーム フォーマット
転送するバイト数	10B
SPI チャンネル	チャンネル 2
受信データを RX FIFO に保存	あり
SPI バイト ストリーム	0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA

次の表は、手順と CAN バス上で送信されるデータを示しています。バスの中央ゲートウェイ / コントローラは ECU と呼ばれ、TCAN デバイスは「レスポнда」と呼ばれます。

- 太字で示された 16 進値は、CAN フレーム ヘッダのバイトを表します (この例では標準フレーム形式です)
- イタリック体で示された 16 進値は、SPI ヘッダ バイトを表します

- 書式なしで示された 16 進値は、SPI データ バイトです

表 7-14. CAN シーケンスの例 (シングル CAN フレーム)

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 16B)	<b>0x0C</b> , <b>0x10</b> , <b>0x10</b> , 0x82, 0x0A, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00	ECU は、レジスタ 0x1010 (SPI TX FIFO) に 10 バイトの書き込みを要求します。SPI ヘッダには、store = 1、SPI チャネル = 2、および 10 バイトの SPI フレームが含まれます。その後、10 バイトのデータはデバイスにシフト インされます。これら 15 バイトのデータを取るため、CAN DLC は 16 バイトでなければならないため、末尾に 0x00 がパディングされます。
2	レスポнда	<b>0x0C</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	SPI	PICO: 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA POCI: 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A	この時点で、レスポндаは SPI チャネル 2 で 10 バイトの SPI 転送を実行します。POCI データは RX FIFO に保存されます
4	ECU	<b>0x43</b> , <b>0x10</b> , <b>0x0F</b>	ECU は、レジスタ 0x100F~0x1011 (SPI_STATUS および SPI_RXFS) から 3 バイトの読み取りを要求します。バースト読み取りが 0x1010 (SPI RX FIFO) からではなく開始されたため、そのレジスタに対応するバイトはパディングとして 0 になります。FIFO から開始されないバースト読み取りでは、FIFO に対応するバイトは「0」として返されます。
5	レスポнда	<b>0x43</b> , <b>0x01</b> , 0xCA, 0x00, 0x01	レスポндаは、レジスタ 0x100F のデータ、0x1010 に対しては 0、0x1011 のデータを返します。結果のデータは、TX FIFO が空であり、RX FIFO に 1 つの SPI フレームが含まれ、次の RX FIFO エLEMENT に 10 バイトのデータがあることを示しています。この情報により、ECU はデバイスから 10 + 2 (SPI ヘッダ) バイトの読み取りを要求すべきことを認識します。
6	ECU	<b>0x4C</b> , <b>0x10</b> , <b>0x10</b>	ECU は、RX FIFO から 12 バイト (データ 10 バイト + ヘッダ 2 バイト) の読み取りを要求します
7	レスポнда (DLC = 16B)	<b>0x4C</b> , <b>0x01</b> , 0x02, 0x0A, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x00, 0x00	レスポндаは要求された 12 バイトのデータを返します。SPI ヘッダ バイトは、これがフレーム読み出しの開始であること、SPI チャネル 2 が使用されたこと、そして 10 バイトの SPI データを読み出す必要があることを示しています (これは CAN フレーム全体に収まります)。このデータを格納できる最も近い CAN DLC サイズは 16 バイトであったため、末尾に 2 バイトのパディングが追加されました。

表 7-15. CAN シーケンス例 (パイロードを転送する複数の CAN フレーム)

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 8B)	<b>0x05</b> , <b>0x10</b> , <b>0x10</b> , 0x02, 0x0A, 0x11, 0x22, 0x33	ECU は、レジスタ 0x1010 (SPI TX FIFO) に 5 バイトの書き込みを要求します。SPI ヘッダには、store = 1、SPI チャネル = 2、および 10 バイトの SPI フレームが含まれます。SPI フレームの最初の 3 バイトのデータだけが転送されました。
2	レスポнда (DLC = 2B)	<b>0x05</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	ECU (DLC = 12B)	<b>0x07</b> , <b>0x10</b> , <b>0x10</b> , 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00, 0x00	ECU は、同じアドレスへの 7 バイトの書き込み要求を続けます。このフレームを格納できる CAN DLC は 12 バイトであるため、末尾に 2 バイトのパディングが追加されます。これらのパディング バイトの値は問題にならず、無視されます。
4	レスポнда (DLC = 2B)	<b>0x07</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	SPI	PICO: 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA POCI: 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A	この時点で、レスポндаは SPI チャネル 2 で 10 バイトの SPI 転送を実行します。POCI データは RX FIFO に保存されます
4	ECU (DLC = 3B)	<b>0x43</b> , <b>0x10</b> , <b>0x0F</b>	ECU は、レジスタ 0x100F~0x1011 (SPI_STATUS および SPI_RXFS) から 3 バイトの読み取りを要求します。バースト読み取りが 0x1010 (SPI RX FIFO) からではなく開始されたため、そのレジスタに対応するバイトはパディングとして 0 になります。FIFO から開始されないバースト読み取りでは、FIFO に対応するバイトは「0」として返されます。



表 7-15. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム) (続き)

ステップ	トランスミッタ	データ	説明
5	レスポнда (DLC = 5B)	0x43, 0x01, 0xCA, 0x00, 0x01	レスポндаは、レジスタ 0x100F のデータ、0x1010 に対しては 0、0x1011 のデータを返します。結果のデータは、TX FIFO が空であり、RX FIFO に 1 つの SPI フレームが含まれ、次の RX FIFO エlement に 10 バイトのデータがあることを示しています。この情報により、ECU はデバイスから 10 + 2 (SPI ヘッダ) バイトの読み取りを要求すべきことを認識します。
6	ECU (DLC = 3B)	0x48, 0x10, 0x10	ECU は、RX FIFO から 8 バイトの読み取りを要求します (ヘッダ 2 バイト + データ 6 バイトを含みます)
7	レスポнда (DLC = 12B)	0x48, 0x01, 0x02, 0x0A, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x00, 0x00	レスポндаは要求された 8 バイトのデータを返します (DLC は 12 ですが、CAN レスponse ヘッダの後の 8 バイトのデータのみが有効であり、0 はパディングであることが分かります)。SPI ヘッダ バイトは、これがフレーム読み出しの開始であること、SPI チャネル 2 が使用されたこと、そして残り 10 バイトの SPI データを読み出す必要があることを示しています。このデータを格納できる最も近い CAN DLC サイズは 12 バイトであったため、末尾に 2 バイトのパディングが追加されました。ECU には 10 バイトの SPI データを読み出す必要があると伝えられましたが、要求したのは 6 バイトだけでした。これは、読み取るべき SPI データが 4 バイト残っていることを意味します。
8	ECU (DLC = 3B)	0x46, 0x10, 0x10	ECU は、読み取りを完了するために RX FIFO から 6 バイト (データ 4 バイト + ヘッダ 2 バイト) の読み取りを要求します。
9	レスポнда (DLC = 8B)	0x46, 0x01, 0x82, 0x04, 0x07, 0x08, 0x09, 0x0A	レスポндаは要求された 6 バイトのデータを返します。SPI ヘッダ バイトは、これがフレーム読み出しの継続であること、SPI チャネル 2 が使用されたこと、そして残り 4 バイトの SPI データを読み出す必要があることを示しています (これはこのフレームに収まります)。送信データが CAN フレーム サイズに一致したため、パディング バイトは不要でした。

## 7.5.5 UART コントローラ

### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があり (セクション 7.3.4 および セクション 7.6.1.18 を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります (セクション 7.5.3 および セクション 7.6.1.17 を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な UART 構成レジスタを、セクション 7.6.4 に示します。

### 7.5.5.1 UART ボーレート生成と分数除数

TCAN5102-Q1 UART にはプログラマブル ボー ジェネレータが搭載されており、リファレンス クロックを  $1 \sim (2^{16} - 1)$  の範囲の除数で分周し、10 進分解能は 1/64 です。ボーレート ジェネレータの出力周波数は、ボーレートの 16 倍です。ボーレート ジェネレータへの入力クロックは常に 40MHz です。除数の式は次のとおりです：

$$\text{Divisor} = (40\text{E6} / (\text{Desired baud rate} \times 16)) \quad (2)$$

レジスタに入力するために必要な値を計算するには、以下の関数が必要です：

- TRUNC(X) : X を切り捨て、実数の整数部分だけを返します。例：TRUNC (3.14) = 3
- ROUND(X) : X を最も近い整数に丸めます。例：ROUND(3.1) = 3 および ROUND(3.6) = 4
- >> : 右方向へのビット シフト操作。例：0x1000 >> 8 = 0x0010。または 0b0001 0000 0000 0000 >> 8 = 0b0000 0000 0001 0000
- & : ビット単位の AND 関数。ビットをマスクするために使用されます。例：0x1234 & 0x00FF = 0x0034 および 0x8765 & 0xFF00 = 0x8700

必要な除数が見つかったら、以下からレジスタ値を計算できます：

$$\text{UART\_BR\_MSB} = \text{TRUNC}(\text{Divisor}) \gg 8 \quad (3)$$

$$\text{UART\_BR\_LSB} = \text{TRUNC}(\text{Divisor}) \& 0xFF \quad (4)$$

$$\text{UART\_BR\_FRAC} = \text{ROUND}([ \text{Divisor} - \text{TRUNC}(\text{Divisor}) ] \times 64) \quad (5)$$

表 7-16. 16× ボーレート分周器 (40 MHz クロック) を使用したボーレート

希望するボーレート	16× CLOCK の生成に使用される除数	取得可能な最も近い除数	UART_BR_MSB 値 (HEX)	UART_BR_LSB 値 (HEX)	UART_BR_FRAC 値 (HEX)	希望値と実測値のパーセント誤差 (%)
400	6250.0000	6250	0x18	0x6A	0x00	0
2400	1041.6667	1041 43/64	0x04	0x11	0x2B	0
4800	520.8333	520 53/64	0x02	0x08	0x35	0.001
9600	260.4167	260 27/64	0x01	0x04	0x1B	0.002
10000	250.0000	250	0x00	0xFA	0x00	0
19200	130.2083	130 13/64	0x00	0x82	0x0D	0.004
25000	100.0000	100	0x00	0x64	0x00	0
28800	86.8056	86 52/64	0x00	0x56	0x34	0.008
38400	65.1042	65 7/64	0x00	0x41	0x07	0.008
50000	50.0000	50	0x00	0x32	0x00	0
57600	43.4028	43 26/64	0x00	0x2B	0x1A	0.008
75000	33.3333	33 21/64	0x00	0x21	0x15	0.016
100000	25.0000	25	0x00	0x19	0x00	0
115200	21.7014	21 45/64	0x00	0x15	0x2D	0.008
128000	19.5313	19 34/64	0x00	0x13	0x22	0
153600	16.2760	16 18/64	0x00	0x10	0x12	0.032
200000	12.5000	12 32/64	0x00	0x0C	0x20	0
225000	11.1111	11 7/64	0x00	0x0B	0x07	0.016
230400	10.8507	10 54/64	0x00	0x0A	0x36	0.064
250000	10.0000	10	0x00	0x0A	0x00	0
256000	9.7656	9 49/64	0x00	0x09	0x31	0
300000	8.3333	8 21/64	0x00	0x08	0x15	0.063
400000	6.2500	6 16/64	0x00	0x06	0x10	0
460800	5.4253	5 27/64	0x00	0x05	0x1B	0.064
500000	5.0000	5	0x00	0x05	0x00	0
750000	3.3333	3 21/64	0x00	0x03	0x15	0.156
921600	2.7127	2 46/64	0x00	0x02	0x2E	0.224
1000000	2.5000	2 32/64	0x00	0x02	0x20	0

### 7.5.5.2 UART 制御プロトコル

UART はバイト ベースの転送であり、上位のプロトコルではないため、フレームの概念はありません。UART モジュールは、各バイトのエラーをチェックして、バイトをシフトイン / シフトアウトします。このため、受信される各バイトにはステータス バイトが関連付けられ、エラーが発生した場合に MCU にアラートを送信します。エラーが共通することは想定されないため、エラーが発生しない限り、各バイトのステータス バイトを読み取る必要はありません。スループットを節約するため、各 UART RX FIFO 読み取りの終了には、グローバル ステータス バイトが追加されます。このグローバル ステータス バイトは、読み出されたすべてのバイトのステータス バイトを論理和したものです。これにより、MCU は転送されたバイト ブロックに対してエラーが報告されたかどうかを判断できます。その場合は、読み取りを要求できます。どのバイトが故障しており、どのエラーを受信したかを判別するために、RX エラー ステータス レジスタを読み出します。

データの送受信時に使用される追加の UART 固有のヘッダはなく、RX FIFO の読み取り時のグローバル ステータス バイトのみです。



レジスタおよびデータ形式の詳細については、[UART 送信 FIFO \(アドレス = h2010\)](#)、[UART 送信 FIFO \(アドレス = h2010\)](#)、[UART 受信エラー ステータス \(アドレス = h2011\)](#) を参照してください。

#### 7.5.5.2.1 UART 書き込み例 1

これは、UART を介してバイト ストリームを送信する基本的な例です。デバイスが TX FIFO にバイトを受信すると、デバイスは UART バイトの送信を開始します。

#### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があります ([セクション 7.3.4](#) および [セクション 7.6.1.18](#) を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります ([セクション 7.5.3](#) および [セクション 7.6.1.17](#) を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な UART 構成レジスタを、[セクション 7.6.4](#) に示します。

**表 7-17. 設計パラメータと前提条件**

パラメータ	値
CAN フレーム フォーマット	標準フレーム フォーマット
転送するバイト数	6B
TX への UART バイト ストリーム	0x11, 0x22, 0x33, 0x44, 0x55, 0x66

次の表は、手順と CAN バス上で送信されるデータを示しています。バスの中央ゲートウェイ / コントローラは ECU と呼ばれ、TCAN デバイスは「レスポнда」と呼ばれます。

- 太字で示された 16 進値は、CAN フレーム ヘッダのバイトを表します (この例では標準フレーム形式です)
- 書式なしで示された 16 進値は、UART データ バイトです

**表 7-18. CAN シーケンスの例 (シングル CAN フレーム)**

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 12B)	<b>0x06, 0x20, 0x10</b> , 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x00, 0x00, 0x00	ECU はレジスタ 0x2010 (UART TX FIFO) への 6 バイトの書き込みを要求します。この 6 バイトのデータを収めるために CAN DLC は 12 バイトでなければならないため、末尾に 0x00 がパディングされます。
2	レスポнда	<b>0x06, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	UART	TXD : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66	この時点で、レスポндаは 6 バイトの UART 転送を行います。

**表 7-19. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム)**

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 6B)	<b>0x03, 0x20, 0x10</b> , 0x11, 0x22, 0x33	ECU はレジスタ 0x2010 (UART TX FIFO) への 3 バイトの書き込みを要求します。
2	レスポнда	<b>0x03, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。レスポндаもここで UART の転送を開始しますが、簡潔にするため、この表ではすべてのバイトが後で送られる形で示しています。
3	ECU (DLC = 6B)	<b>0x03, 0x20, 0x10</b> , 0x44, 0x55, 0x66	ECU は残りのデータを完了するために、レジスタ 0x2010 (UART TX FIFO) へ 3 バイトの書き込みを要求します
4	レスポнда	<b>0x03, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
5	UART	TXD : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66	この時点で、レスポндаは 6 バイトの UART 転送を行います。

#### 7.5.5.2.2 UART のバイト読み取り例 1

この例は、UART を介してバイト ストリームを送受信する基本的な例です。デバイスが TX FIFO にバイトを受信すると、UART バイトの送信を開始します。

## 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があります (セクション 7.3.4 およびセクション 7.6.1.18 を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります (セクション 7.5.3 およびセクション 7.6.1.17 を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な UART 構成レジスタを、セクション 7.6.4 に示します。

表 7-20. 設計パラメータと前提条件

パラメータ	値
CAN フレーム フォーマット	標準フレーム フォーマット
転送するバイト数	6B
UART バイト ストリーム RX	0x11, 0x22, 0x33, 0x44, 0x55, 0x66
エラー	バイト 0x33 でパリティ エラー、バイト 0x44 でフレーミング エラー

次の表は、手順と CAN バス上で送信されるデータを示しています。バスの中央ゲートウェイ / コントローラは ECU と呼び、TCAN デバイスは「レスポнда」と呼びます。

- 太字で示された 16 進値は、CAN フレーム ヘッダのバイトを表します (この例では標準フレーム形式です)
- 書式なしで示された 16 進値は、UART データ バイトです

表 7-21. CAN シーケンスの例 (シングル CAN フレーム)

ステップ	トランスミッタ	データ	説明
1	UART	RXD : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66	UART RXD は 6 バイトを受信します
2	ECU (DLC = 16B)	<b>0x41, 0x20, 0x0B</b>	ECU は 0x200B (UART RX FIFO ステータス) の読み出しを要求します
3	レスポнда	<b>0x41, 0x01, 0x06</b>	レスポндаはリクエストを受信したことを確認するために OK を返し、レジスタ データを送信します。これにより、現在 RX FIFO に 6 バイトのデータが存在することが示されます
4	ECU	<b>0x47, 0x20, 0x10</b>	ECU は 0x2010 (UART RX FIFO) から 7 バイト (6 バイトのデータ + グローバル ステータス バイト) の読み出しを要求します
5	レスポнда	<b>0x47, 0x01, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x0C, 0x00, 0x00, 0x00</b>	レスポндаは 7 バイトのデータ (6 バイトの UART データと 1 バイトのグローバル ステータス) を返します。グローバル ステータス バイトは、直前に読み出された 6 バイトのデータのうち少なくとも 1 バイトにパリティ エラーとフレーム エラーの両方があることを示しています。
6	ECU	<b>0x46, 0x20, 0x11</b>	ECU は、グローバル ステータス バイトが非正常であることを検出したため、0x2011 (UART RX エラー ステータス) から 6 バイト (グローバル ステータス バイトなし) の読み出しを要求します。
7	レスポнда	<b>0x46, 0x01, 0x01, 0x01, 0x04, 0x08, 0x01, 0x01</b>	レスポндаは 6 バイトのステータスを返し、その内容は 0x33 がパリティ エラー、0x44 がフレーミング エラーであったことを示しています。ECU はエラーの処理方法を決定できます。

表 7-22. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム)

ステップ	トランスミッタ	データ	説明
1	UART	RXD : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66	UART RXD は 6 バイトを受信します
2	ECU	<b>0x41, 0x20, 0x0B</b>	ECU は 0x200B (UART RX FIFO ステータス) の読み出しを要求します

**表 7-22. CAN シーケンス例 (パイロードを転送する複数の CAN フレーム) (続き)**

ステップ	トランスミッタ	データ	説明
3	レスポнда	0x41, 0x01, 0x06	レスポндаはリクエストを受信したことを確認するために OK を返し、レジスタ データを送信します。これにより、現在 RX FIFO に 6 バイトのデータが存在することが示されます
4	ECU	0x43, 0x20, 0x10	ECU は 0x2010 (UART RX FIFO) から 3 バイト (2 バイトのデータ + グローバル ステータス バイト) の読み出しを要求します
5	レスポнда	0x43, 0x01, 0x11, 0x22, 0x01	レスポндаは 3 バイトのデータ (2 バイトの UART データと 1 バイトのグローバル ステータス) を返します。グローバル ステータス バイトは、エラーが報告されていない (正常ステータス) ことを示しています。
6	ECU	0x41, 0x20, 0x0B	ECU は 0x200B (UART RX FIFO ステータス) の読み出しを要求します
7	レスポнда	0x41, 0x01, 0x04	レスポндаはリクエストを受信したことを確認するために OK を返し、レジスタ データを送信します。これにより、現在 RX FIFO に 4 バイトのデータが存在することが示されます
8	ECU	0x45, 0x20, 0x10	ECU は 0x2010 (UART RX FIFO) から 5 バイト (4 バイトのデータ + グローバル ステータス バイト) の読み出しを要求します
9	レスポнда	0x45, 0x01, 0x33, 0x44, 0x55, 0x66, 0x0C	レスポндаは 3 バイトのデータ (2 バイトの UART データと 1 バイトのグローバル ステータス) を返します。グローバル ステータス バイトは、直前に読み出された 4 バイトのデータのうち少なくとも 1 バイトにパリティ エラーとフレーム エラーの両方があることを示しています。
10	ECU	0x44, 0x20, 0x11	ECU は、グローバル ステータス バイトが非正常であることを検出したため、0x2011 (UART RX エラー ステータス) から 4 バイト (グローバル ステータス バイトなし) の読み出しを要求します。
11	レスポнда	0x44, 0x01, 0x04, 0x08, 0x01, 0x01	レスポндаは 4 バイトのステータスを返し、その内容は 0x33 がパリティ エラー、0x44 がフレーミング エラーであったことを示しています。ECU はエラーの処理方法を決定できます。

### 7.5.6 I2C コントローラ

#### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があり (セクション 7.3.4 およびセクション 7.6.1.18 を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります (セクション 7.5.3 およびセクション 7.6.1.17 を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な I2C 構成レジスタを、セクション 7.6.6 に示します。

#### 7.5.6.1 I2C ボーレート生成

TCAN5102-Q1 I2C モジュールには、リファレンス クロックを 1 ~ 255 の範囲の除数で分周するプログラマブル ボーレート ジェネレータが内蔵されています。分周器への入力クロックは 10MHz と 2.5MHz のどちらかにでき、I2C\_CTRL.LSM (低速度モード) ビットによって設定されます。除数の式は次のとおりです：

$$\text{Divisor} = (\text{Input Clock} / \text{Desired I2C Speed}) - 1 \quad (6)$$

**表 7-23. 高速クロックを使用する I2C 速度 (10MHz クロック、I2C\_CTRL.LSM = 0)**

目標の I2C 速度 (kHz)	取得可能な最も近い除数	I2C_BR 値 (HEX)	実際の I2C 速度 (kHz)	希望値と実測値のパーセント誤差 (%)
39	255	0xFF	39	0.16
50	199	0xC7	50	0
60	166	0xA6	60	0.2

表 7-23. 高速クロックを使用する I2C 速度 (10MHz クロック、I2C\_CTRL.LSM = 0) (続き)

目標の I2C 速度 (kHz)	取得可能な最も近い除数	I2C_BR 値 (HEX)	実際の I2C 速度 (kHz)	希望値と実測値のパーセント誤差 (%)
70	142	0x8E	70	0.1
80	124	0x7C	80	0
100	99	0x63	100	0
200	49	0x31	200	0
250	39	0x27	250	0
400	24	0x18	400	0
500	19	0x13	500	0
600	16	0x10	588	2
800	12	0x0C	769	4
1000	9	0x09	1000	0

表 7-24. 低速クロックを使用した I2C 速度 (2.5MHz クロック、I2C\_CTRL.LSM = 1)

目標の I2C 速度 (kHz)	取得可能な最も近い除数	I2C_BR 値 (HEX)	実際の I2C 速度 (kHz)	希望値と実測値のパーセント誤差 (%)
10	249	0xF9	10	0
50	49	0x31	50	0
60	41	0x29	60	0.8
70	35	0x23	69	0.8
80	30	0x1E	81	0.8
100	24	0x18	100	0
200	12	0x0C	192	4
250	9	0x09	250	0
400	5	0x05	417	4
500	4	0x04	500	0
600	3	0x03	625	4
800	2	0x02	833	4

#### 7.5.6.2 I2C スタック バス回復

I2C コントローラには、スタック バス回復 (SBR) と呼ばれるオプション機能があります。イネーブルの場合、スタック状態が検出されると、IP は自動的にバスのスティック解除を試みます。スタック バスの検出は、SCL または SDA ラインが  $t_{\text{STUCKBUS\_I2C}}$  時間にわたって low に保持されている場合です。スタック条件が検出され、自動 SBR がイネーブルになると、I2C コントローラは最大 16 個のクロック パルスの生成を試みます。コントローラはクロック パルスを生成するときに、スタック状態が解消されたかどうかを監視します。条件が解消すると、クロック生成が直ちに停止し、STOP 条件が生成されてバスをリセットします。スタック バス回復シーケンスが完了すると、SBRC (スタックバス回復完了) 割り込みがセットされます。I2C バスの状態は、I2C\_STATUS レジスタを介してポーリングできます。I2C\_CTRL レジスタの SBR\_START ビットに書き込むことで、スタックバスの回復シーケンスを手動で要求できます。

#### 注

スタック バス検出用のタイマは、I2C IP がイネーブルになり、GPIO に接続されると開始します。I2C IP を有効にする前に、I2C バスにプル アップがあることを確認してください。そうしないと、誤ったバススタック状態が発生する可能性があります。

#### 7.5.6.3 I2C 制御プロトコル

I2C 送信 (TX) FIFO に書き込むとき、I2C アドレスとデータ バイト数には 2 バイトのヘッダが必要です。I2C 受信 (RX) FIFO から読み取るとき、3 つの追加のヘッダ バイトがあるため、ターゲット ペリフェラルから読み取

るときの堅牢性を高めることができます。2つのヘッダ形式があり、これらは類似していますが、アクションが TX FIFO と RX FIFO のどちらに対する読み取りであるか書き込みであるかによって異なります。

TX FIFO に書き込むとき、2バイトのヘッダは、FIFO に書き込まれる最初の 2 バイトです。

RX FIFO を読み出す際、データ フィールドの先頭には常に 2 バイトのヘッダが付加されます。このヘッダーには、残りのバイト数や、読み出しが前回の読み出しの継続であるかどうかの情報が含まれています (単一の CAN フレームに収まりきらない大きな SPI メッセージに必要です)。

TX FIFO に複数の I2C フレームがキューに登録されている場合、IP はストップとスタートの代わりにリピートスタートを使用します。

ヘッダ形式の詳細については、[セクション 7.6.7.1](#) および [セクション 7.6.7.2](#) を参照してください。

#### 7.5.6.3.1 I2C 書き込みの例 1

この例は、I2C を介してバイト ストリームを送信する基本的な例です。

#### 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があります ([セクション 7.3.4](#) および [セクション 7.6.1.18](#) を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります ([セクション 7.5.3](#) および [セクション 7.6.1.17](#) を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な I2C 構成レジスタを、[セクション 7.6.6](#) に示します。

**表 7-25. 設計パラメータと前提条件**

パラメータ	値
CAN フレーム フォーマット	標準フレーム フォーマット
I2C ターゲット アドレス (7 ビット)	0x12
I2C フレーム	書き込み
転送するバイト数	8B
書き込む I2C データ	0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08

次の表は、手順と CAN バス上で送信されるデータを示しています。バスの中央ゲートウェイ / コントローラは ECU と呼ばれ、TCAN デバイスは「レスポнда」と呼ばれます。

- **太字**で示された 16 進値は、CAN フレーム ヘッダのバイトを表します (この例では標準フレーム形式です)
- イタリック体で示された 16 進値は、I2C ヘッダ バイトを表します
- フォーマットされていない 16 進値は、I2C データ バイトです

**表 7-26. CAN シーケンスの例 (シングル CAN フレーム)**

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 16B)	<b>0x0A</b> , <b>0x30</b> , <b>0x10</b> , 0x25, 0x88, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x00, 0x00, 0x00	ECU はレジスタ 0x3010 (I2C TX FIFO) に 10 バイトの書き込みを要求します。I2C ヘッダーには store = 1、I2C ターゲット アドレス = 0x12、write = 1、I2C データ サイズ = 8 バイトが設定されています。その後、10 バイトのデータはデバイスにシフト インされます。この 13 バイトのデータを収めるために CAN DLC は 16 バイトでなければならないため、末尾に 0x00 バイトがパディングされます。
2	レスポнда	<b>0x0A</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	I2C	I2C フレーム : 0x25, 0x88, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08	この時点で、レスポндаは 10 バイトの I2C 転送を開始します。



表 7-27. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム)

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 7B)	<b>0x04</b> , <b>0x30</b> , <b>0x10</b> , 0x25, 0x88, 0x01, 0x02	ECU はレジスタ 0x3010 (I2C TX FIFO) に 4 バイトの書き込みを要求します。I2C ヘッダーには store = 1、I2C ターゲット アドレス = 0x12、write = 1、I2C データ サイズ = 8 バイトが設定されています。その後、10 バイトのデータはデバイスにシフト インされます。
2	レスポнда	<b>0x04</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	ECU (DLC = 12B)	<b>0x06</b> , <b>0x30</b> , <b>0x10</b> , 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x00, 0x00, 0x00	ECU は、同じアドレスへの 6 バイトの書き込み要求を継続します。このフレームを格納できる CAN DLC は 12 バイトであるため、末尾に 3 バイトのパディングが追加されます。これらのパディング バイトの値は問題にならず、無視されます。
4	レスポнда	<b>0x06</b> , <b>0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
5	I2C	I2C フレーム : 0x25, 0x88, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08	この時点で、レスポндаは 10 バイトの I2C 転送を開始します。

## 7.5.6.3.2 I2C 読み取り例 1

これは、I2C を介してバイト ストリームを送受信する基本的な例です。デバイスが TX FIFO にバイトを受け取ると、I2C バイトの送信を開始します。

## 注

モジュールを使用する前に、適切な GPIO を特殊機能に設定する必要があり ([セクション 7.3.4](#) および [セクション 7.6.1.18](#) を参照)、MRAM はアロケーション メモリによってモジュールをイネーブルにする必要があります ([セクション 7.5.3](#) および [セクション 7.6.1.17](#) を参照)。GPIO が特殊機能に設定され、メモリを割り当てる MRAM 構成を選択してモジュールが有効化されてはじめて、そのモジュールを使用できます。適切な I2C 構成レジスタを、[セクション 7.6.6](#) に示します。

表 7-28. 設計パラメータと前提条件

パラメータ	値
CAN フレーム フォーマット	標準フレーム フォーマット
I2C ターゲット アドレス (7 ビット)	0x12
I2C フレーム	読み出し
I2C ターゲット レジスタ アドレス	0x10
転送するバイト数	8B

次の表は、手順と CAN バス上で送信されるデータを示しています。バスの中央ゲートウェイ / コントローラは ECU と呼ばれ、TCAN デバイスは「レスポнда」と呼ばれます。

- **太字**で示された 16 進値は、CAN フレーム ヘッダのバイトを表します (この例では標準フレーム形式です)
- イタリック体で示された 16 進値は、I2C ヘッダ バイトを表します
- フォーマットされていない 16 進値は、I2C データ バイトです

I2C 読み取りを行う際には、読み取りの前にレジスタ アドレスを書き込むのが一般的です。これにより、ターゲット デバイスはどのレジスタを読み取るべきかを認識できます。つまり、I2C 読み取りは通常、2 つの独立した I2C フレームで構成され、I2C 書き込みの後に I2C 読み取りが行われます。この例は次のように示されています。

**表 7-29. CAN シーケンスの例 (シングル CAN フレーム)**

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 6B)	<b>0x03, 0x30, 0x10, 0x25, 0x81, 0x10</b>	ECU は、レジスタ 0x3010 (I2C TX FIFO) に 3 バイトの書き込みを要求します。I2C ヘッダには、store = 1、I2C ターゲット アドレス = 0x12、write = 1、I2C データ サイズ = 1 バイト (レジスタ アドレス 0x10) が含まれます
2	レスポнда	<b>0x03, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	I2C	I2C フレーム : 0x25, 0x10	この時点で、レスポндаは 1 バイトの I2C データ転送を開始します。
4	ECU (DLC = 3B)	<b>0x43, 0x30, 0x10</b>	ECU は、ターゲット アドレスを設定するための最初の I2C 書き込みのステータスを確認するために、RX FIFO からの読み取りを要求します。これは、ターゲットが承認済みであることを確認するために行われます。
5	レスポнда	<b>0x43, 0x01, 0x25, 0x01, 0x01</b>	レスポндаは I2C ステータス フラグ 0x01 を返し、「成功」ステータスを示します。これは、ターゲットがアドレスとデータ (レジスタ アドレス) の両方を確認応答したことを意味します。
6	ECU (DLC = 5B)	<b>0x02, 0x30, 0x10, 0x24, 0x08</b>	ECU は、レジスタ 0x3010 (I2C TX FIFO) に 2 バイトの書き込みを要求します。I2C ヘッダには、store = 1、I2C ターゲット アドレス = 0x12、write = 1、および I2C リード サイズ = 8 バイトが含まれます
7	レスポнда	<b>0x02, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
8	I2C	I2C フレーム : 0x24, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08	この時点で、レスポндаは 8 バイト長の I2C 読み取りを開始します
9	ECU (DLC = 3B)	<b>0x4B, 0x30, 0x10</b>	ECU は、I2C ターゲットから読み取られたデータを取得するために、11 バイト (2 バイトのヘッダ + 8 バイトのデータ + 1 バイトのステータス) を RX FIFO から読み取るよう要求します
10	レスポнда	<b>0x4B, 0x01, 0x24, 0x09, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x01, 0x00, 0x00, 0x00</b>	レスポндаは、I2C ヘッダ情報と共に 8 バイトのデータを返します。データの後に I2C ステータス フラグ 0x01 が続き、「成功」ステータスを示します。これは、ターゲットがアドレスをアクノリッジしたことを意味します。このデータを格納できる最も近い CAN FD ペイロード サイズは 16 バイトであるため、末尾に 0x00 のパディング バイトが追加されます。

**表 7-30. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム)**

ステップ	トランスミッタ	データ	説明
1	ECU (DLC = 6B)	<b>0x03, 0x30, 0x10, 0x25, 0x81, 0x10</b>	ECU は、レジスタ 0x3010 (I2C TX FIFO) に 3 バイトの書き込みを要求します。I2C ヘッダには、store = 1、I2C ターゲット アドレス = 0x12、write = 1、I2C データ サイズ = 1 バイト (レジスタ アドレス 0x10) が含まれます
2	レスポнда	<b>0x0A, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
3	I2C	I2C フレーム : 0x25, 0x10	この時点で、レスポндаは 1 バイトの I2C データ転送を開始します。
4	ECU (DLC = 3B)	<b>0x43, 0x30, 0x10</b>	ECU は、ターゲット アドレスを設定するための最初の I2C 書き込みのステータスを確認するために、RX FIFO からの読み取りを要求します。これは、ターゲットが承認済みであることを確認するために行われます。
5	レスポнда	<b>0x43, 0x01, 0x25, 0x01, 0x01</b>	レスポндаは I2C ステータス フラグ 0x01 を返し、「成功」ステータスを示します。これは、ターゲットがアドレスとデータ (レジスタ アドレス) の両方をアクノリッジしたことを意味します。
6	ECU (DLC = 5B)	<b>0x02, 0x30, 0x10, 0x24, 0x08</b>	ECU は、レジスタ 0x3010 (I2C TX FIFO) に 2 バイトの書き込みを要求します。I2C ヘッダには、store = 1、I2C ターゲット アドレス = 0x12、write = 1、および I2C リード サイズ = 8 バイトが含まれます
7	レスポнда	<b>0x02, 0x01</b>	レスポндаは OK を返信し、要求を受信したことをアクノリッジします。
8	I2C	I2C フレーム : 0x24, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08	この時点で、レスポндаは 8 バイト長の I2C 読み取りを開始します
9	ECU (DLC = 3B)	<b>0x46, 0x30, 0x10</b>	ECU は、I2C ターゲットから読み取られたデータを取得するために、RX FIFO から 6 バイトの読み取りを要求します
10	レスポнда	<b>0x46, 0x01, 0x24, 0x09, 0x01, 0x02, 0x03, 0x04</b>	レスポндаは、I2C ヘッダ情報と共に 8 バイトのデータを返します。

表 7-30. CAN シーケンス例 (ペイロードを転送する複数の CAN フレーム) (続き)

ステップ	トランスミッタ	データ	説明
11	ECU (DLC = 3B)	0x47, 0x30, 0x10	ECU は、I2C ターゲットから読み取られたデータを取得するために、RX FIFO から 7 バイトの読み取りを要求します (ヘッダ バイト 2 + データ バイト 8 + ステータス バイト 1 – すでに読み取った 4 バイト)
12	レスポンス	0x47, 0x01, 0x24, 0x85, 0x05, 0x06, 0x07, 0x08, 0x01, 0x00, 0x00, 0x00	レスポンスは、I2C ヘッダ情報と共に 8 バイトのデータを返します。2 番目の I2C ヘッダ バイトには CONT フラグが設定されており、これは部分的に読み取られた I2C フレームの継続であることを意味します。データの後に I2C ステータス フラグ 0x01 が続き、「成功」ステータスを示します。これは、ターゲットがアドレスをアックノリッジしたことを意味します。このデータを格納できる最も近い CAN FD ペイロード サイズは 12 バイトであるため、末尾に 0x00 のパディング バイトが追加されます

### 7.5.7 PWM および矩形波 PWM ランププロファイル

TCAN5102-Q1 は、高度に設定可能な 2 つの PWM 出力 (PWM0 および PWM1) を備え、20Hz ~ 100KHz の広い周波数範囲で 8 ビットまたは 10 ビットの出力分解能をサポートしています。動作モードには 2 つの主要な方式があり、静的 PWM 出力と台形 PWM ランプ制御です。ランプ制御により、デューティ サイクルまたは PWM 出力の周波数の 2 つのパラメータをランプすることができます。

#### 注

PWM モジュールは、IO\_CFG\_1 内で該当する GPIO を特殊機能モードに設定することで、GPIO に多重化する必要があります。これを行わないと、PWM 信号はピンに多重化されません。

PWM モードが静的出力に設定されている場合、ランプ動作が発生しないため、一部のレジスタだけが必要です。

- 周波数は、PWM\_END\_VAL レジスタ (0x4019 ~ 0x401B および 0x4119 ~ 0x411B) によって制御されます
- デューティ サイクルは、PWM\_CONST レジスタ (0x400B ~ 0x400C および 0x410B ~ 0x410C) で制御されます

設定可能なランプには、次の 2 つがあります：オンランプとオフランプ。オンランプは、次のオンランプを要求する前にランプ値を更新することで、さまざまなランプ プロファイルを作成するために使用できます。オフランプは、予測可能で制御されたランプダウン / オフ遷移を実現するために使用されます。PWM 出力を直ちにディセーブルにするためのハードシャットオフが使用できます。オンランプが完了すると、PWM 出力は、変更するように指示されるまで終了値に維持されます。オフランプは指定された終了ポイントまで上昇してから、PWM 出力はディセーブルになります。

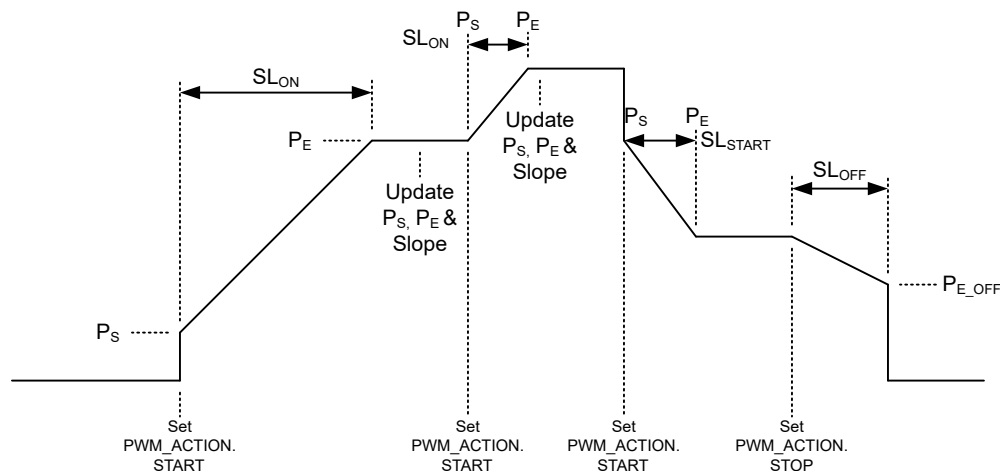


図 7-8. PWM ランプの例

次の点は、以下によって定義されます：



- $P_S$ : (オンランプの場合の) 開始点は、オンランプの開始点を表します。
- $P_E$ : 終了ポイント (オンランプの場合) は、オンランプの終了ポイントを表します。
- $SL_{ON}$ : ランプ スロープ (オンランプの場合) は、オンランプ中の変化率を表します。
- $P_{E\_OFF}$ : 終了ポイント (オフランプの場合) は、オフランプの終了ポイントを表します。
- $SL_{OFF}$ : ランプ スロープ (オフランプの場合) は、オフランプ中の変化率を表します。

PWM ランプを構成する値は次の 3 つあります：開始ポイント、終了ポイント、およびランプ スロープです。オンランプの開始ポイントは、現在の PWM 値または指定した開始点のいずれかを指定できます。これにより、必要に応じてステップ関数タイプの動作を柔軟に作成したり、既存のランプにスムーズに変更したりできます。ランプ スロープは、時間の経過に伴う変動パラメータの変化率を表すために使用されます。たとえば、ランプ スロープを使用して、1 秒あたり 5% のレートで出力デューティ サイクルを変更するようにデバイスに通知できます。

変化させる可能性のある 2 つの PWM パラメータ (PWM 周波数およびデューティサイクル) があるため、無変化パラメータはユーザーが設定できる静的値です。たとえば、ランプの PWM 周波数を変化させている場合、デューティ サイクルは任意の値 (30% など) に設定できますが、ランプ中に変更することはできません。

#### 注

該当するランプ中にオンランプまたはオフランプ プロファイルの設定値を更新しても、ランプ値はリアルタイムに変更されません。アクション コマンド (開始アクションや停止アクションなど) が設定されると、ランプ設定がバッファにロードされます。

#### 7.5.7.1 矩形波ランプ制御

TCAN5102-Q1 は、ハードウェア制御の台形波ランプをサポートしています。ランプ処理には、次の 2 つのパラメータがサポートされています：

- デューティ サイクルのランプ
- スwitching 周波数のランプ

さまざまなパラメータ (デューティ サイクルまたはスitching 周波数) をランプするための開始ポイントと終了ポイントの値が 1 組あります。このペアは、デューティ サイクルまたは PWM 出力のスitching 周波数の開始ポイントと終了ポイントを表します。変化させていないパラメータ (non-varied-parameter) には設定可能な値が 1 つだけあり、この値はランプ中に変化しません。

オフランプは、オフに命令された場合に定義された動作です。これは、ランプで到達する定義済みのストップ値と、指定された変化率で構成されます。オフランプが完了すると、再びイネーブルになるまで、PWM 出力はデイスエーブルになります。

##### 7.5.7.1.1 デューティ サイクルのランプ

TCAN5102-Q1 は、デューティ サイクルを変化させることによるランプをサポートしています。デューティ サイクルがランプするとき、スitching 周波数は固定値になります。

上昇中、デューティサイクルは各周期の開始時に更新されます。もう 1 つの言い方は、スitching 周波数レートで出力デューティ サイクル値が更新されることです。

図 7-9 において、デューティサイクル値は、各期間の開始時に赤い矢印で表示されています。

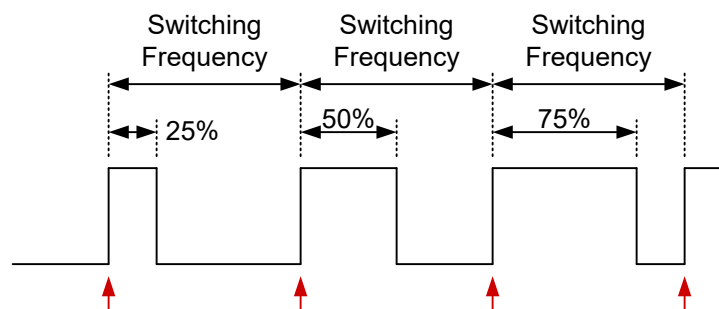


図 7-9. デューティ サイクル上昇の例

## 注

ランプが比較的遅い場合は、異なる値がロードされる前に数回の期間が発生する可能性があります。これは、8 ビットまたは 10 ビット出力の最小ステップサイズのために発生します。

## 7.5.7.1.2 スイッチング周波数のランプ

TCAN5102-Q1 はスイッチング周波数を変化させることでランプをサポートしています。スイッチング周波数が上昇すると、デューティ サイクルは固定されます。

ランプ中、スイッチング周波数の値は  $10\mu\text{s}$  ごとに計算されますが、出力スイッチング周波数は期間の開始時にのみ更新されます。これは、1 つの周波数で少なくとも 1 つの全周期が確実に発生するようにするためです。

図 7-10 に示すように、ランプの周波数値は  $10\mu\text{s}$  ごとに計算されますが、新しい周期の開始時に最後に生成された値が使用されます。100kHz を超える速度では、値が更新される前に複数の周期が発生します。

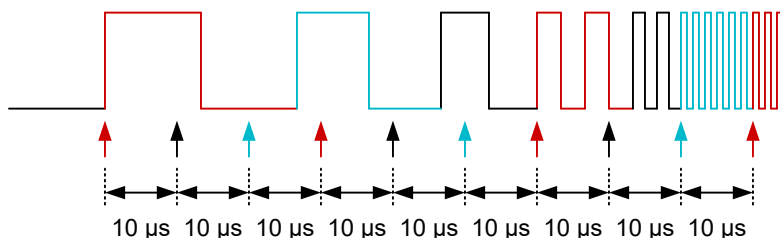


図 7-10. スイッチング周波数ランプの更新

## 注

ランプが遅い場合、異なるスイッチング周波数がロードされる前に数周期が発生する可能性があります。これは最小ステップサイズが原因で発生します。

## 7.5.7.1.3 オフランプ

TCAN5102-Q1 にはオフランプに使用する別々の値セットがあり、指定されたシャットオフ動作を実現できます。これらの値は、PWM ACTION レジスタの STOP ビットへの書き込みによってオフランプが要求されたときに使用されます。オフランプが要求されるタイミングは関係なく、既存のオンランプの途中でも、PWM 出力が現在ランプしていないときでも発生する可能性があります。オフランプは、オフランプレジスタで指定されたオフランプスロープおよび終了ポイントに従います。終了ポイントがゼロ以外の場合、可変パラメータが指定された終了ポイントに到達すると PWM 出力は停止します。これにより、モーターの駆動に便利なステップ機能を作成できます。オフランプの開始ポイントは、常に可変パラメータの現在の値です。

図 7-11 に、オフランプとして 2 つの例があります。赤い線は、オンランプ中にストップ ビットが設定された場合の出力を示します。黒い線は、オンランプが完了し、PWM 出力がアイドルの後に設定されている s ストップ ビットを示します。

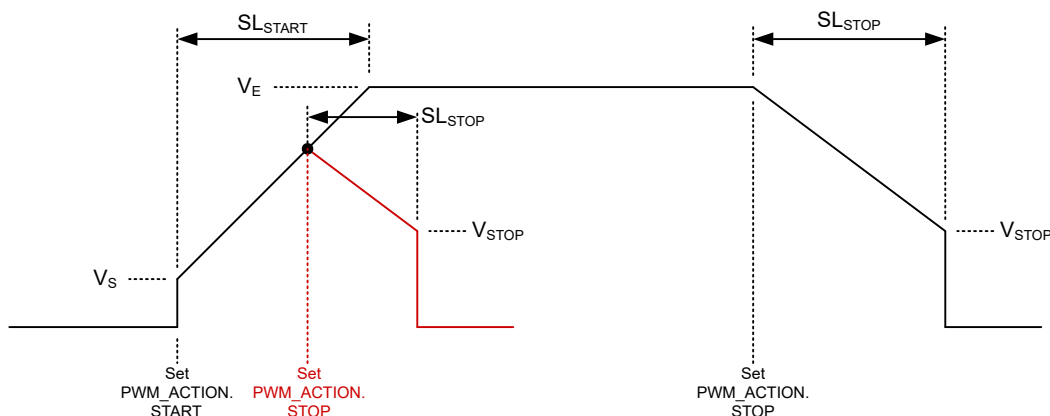


図 7-11. PWM オフランプの例

### 7.5.7.2 PWM クロック ジェネレータ

TCAN5102-Q1 には、分数値をサポートする PWM クロック ジェネレータが搭載されています。周波数値は常に、11 ビットの整数除数と 7 ビットの分数除数で構成されます。PWM クロック ジェネレータへの入力クロックは 40 MHz であり、8 ビット (256 ステップ) または 10 ビット (1024 ステップ) の出力 (デューティ サイクル分解能) を生成するように設定できます。可能な因子は 1 から  $(2^{11}-1)$  の範囲で、小数点の分解能は 1/128 です。

除数の計算式は次のとおりです：除数 =  $(40E6 / (\text{希望するスイッチング周波数} \times 2^{\text{RES}}))$

ここで、「RES」は PWM 出力の分解能のビット数 (8 または 10) です。

必要なレジスタ値を計算するには、以下の関数が必要です：

- TRUNC(x) : x を切り捨てます。数値の整数部分だけを返します。(例：TRUNC(3.14) = 3)
- ROUND(x) : x を最も近い整数に丸めます。(例：ROUND(3.1) = 3 および ROUND(3.6) = 4)
- >> : 右方向へのビット シフト操作。例：0x1000 >> 8 = 0x0010。または 0b0001 0000 0000 0000 >> 8 = 0b0000 0000 0001 0000
- & : ビット単位の AND 関数。ビットをマスクするために使用されます。例：0x1234 & 0x00FF = 0x0034 および 0x8765 & 0xFF00 = 0x8700

除数が計算されると、次の 3 つの等式を使用してレジスタ値を計算できます：

$$\text{FREQ MSB} = \text{TRUNC}(\text{Divisor}) \gg 8 \quad (7)$$

$$\text{FREQ LSB} = \text{TRUNC}(\text{Divisor}) \& 0x00FF \quad (8)$$

$$\text{FREQ FRAC} = \text{ROUND}([\text{Divisor} - \text{TRUNC}(\text{Divisor})] \times 128) \quad (9)$$

表 7-31. スwitchング周波数 (8 ビット出力)

希望スitchング周波数 (Hz)	周波数を生成する除数	取得可能な最も近い除数	周波数 MSB 値 (HEX)	FREQ LSB 値 (HEX)	周波数 FRAC 値 (HEX)	希望値と実測値のパーセント誤差 (%)
100	1562.5000	1562 64/128	0x06	0x1A	0x40	0
250	625.0000	625	0x02	0x71	0x00	0
500	312.5000	312 64/128	0x01	0x38	0x40	0
1,000	156.2500	156 32/128	0x00	0x9C	0x20	0
2,500	62.5000	62 64/128	0x00	0x3E	0x40	0
5,000	31.2500	31 32/128	0x00	0x1F	0x20	0
7,500	20.8333	20 107/128	0x00	0x14	0x6B	0.013
10,000	15.6250	15 80/128	0x00	0x0F	0x50	0
25,000	6.2500	6 32/128	0x00	0x06	0x20	0
50,000	3.1250	3 16/128	0x00	0x03	0x10	0
75,000	2.0833	2 11/128	0x00	0x02	0x0B	0.125
100,000	1.5625	1 72/128	0x00	0x01	0x48	0

表 7-32. スwitchング周波数 (10 ビット出力)

希望スitchング周波数 (Hz)	周波数を生成する除数	取得可能な最も近い除数	周波数 MSB 値 (HEX)	FREQ LSB 値 (HEX)	周波数 FRAC 値 (HEX)	希望値と実測値のパーセント誤差 (%)
50	781.2500	781 32/128	0x03	0x0D	0x20	0
100	390.6250	390 80/128	0x01	0x86	0x50	0
250	156.2500	156 32/128	0x00	0x9C	0x20	0
500	78.1250	78 16/128	0x00	0x4E	0x10	0
1,000	39.0625	39 8/128	0x00	0x27	0x08	0
2,500	15.6250	15 80/128	0x00	0x0F	0x50	0
5,000	7.8125	7 104/128	0x00	0x07	0x68	0

表 7-32. スイッチング周波数 (10 ビット出力) (続き)

希望スイッチング周波数 (Hz)	周波数を生成する除数	取得可能な最も近い除数	周波数 MSB 値 (HEX)	FREQ LSB 値 (HEX)	周波数 FRAC 値 (HEX)	希望値と実測値のパーセント誤差 (%)
7,500	5.2083	5 27/128	0x00	0x05	0x1B	0.05
10,000	3.9063	3 116/128	0x00	0x03	0x74	0
25,000	1.5625	1 72/128	0x00	0x01	0x48	0

## 注

8 ビット出力モードで構成されている場合に、100kHz を超える値を生成できます。周波数ランプを動作させる際は、特別な考慮事項があります。詳細については、[スイッチング周波数ランプ](#)を参照してください。

ランプするパラメータによって変化する値を書き込むための適切なレジスタ。たとえば、ランプがスイッチング周波数ランプである場合、レジスタ値は PWM\_START、PWM\_END、PWM\_OFF レジスタ グループに入ります。ランプがデューティ サイクル ランプの場合、周波数レジスタ値は PWM\_CONST と PWM\_END\_VAL\_CONST\_FRAC\_F に入ります。

## 7.5.7.3 PWM のデューティ サイクル

TCAN5102-Q1 には PWM モジュールが搭載されており、8 ビット (256 ステップ) または 10 ビット (1024 ステップ) の出力 (デューティ サイクル分解能) を生成するように設定できます。

デューティ サイクル カウント値は、PWM デューティ サイクルが 8 ビット モードと 10 ビット モードのどちらに設定されているかに基づいて、単純に計算して変化します。

必要なレジスタ値を計算するには、次の関数が必要です：

- $\text{ROUND}(x)$  :  $x$  を最も近い整数に丸めます。(例 :  $\text{ROUND}(3.1) = 3$  および  $\text{ROUND}(3.6) = 4$ )

$\text{DC}_{\text{TARGET}}$  は、求められるデューティ サイクルのパーセンテージを浮動小数点で表します (例 :  $40\% = 0.40$ 、 $25\% = 0.25$ )

$$\text{DC\_COUNT}_{8\text{-BIT}} = \text{ROUND}(\text{DC}_{\text{TARGET}} \times 256) \quad (10)$$

$$\text{DC\_COUNT}_{10\text{-BIT}} = \text{ROUND}(\text{DC}_{\text{TARGET}} \times 1024) \quad (11)$$

表 7-33. デューティ サイクル

目標デューティ サイクル (%)	PWM_CTRL.DC_8B = 1 8 ビット				PWM_CTRL.DC_8B = 0 10 ビット			
	デューティ サイクル カウント	入手可能な最も近いカウント	デューティ サイクル カウント (HEX)	実際のデューティ サイクル (%)	デューティ サイクル カウント	入手可能な最も近いカウント	デューティ サイクル カウント (HEX)	実際のデューティ サイクル (%)
5	12.800	13	0x0D	5.08%	51.200	51	0x033	4.98%
10	25.600	26	0x1A	10.16%	102.400	102	0x066	9.96%
15	38.400	38	0x26	14.84%	153.600	154	0x099	15.04%
20	51.200	51	0x33	19.92%	204.800	205	0x0CC	20.02%
25	64.000	64	0x40	25.00%	256.000	256	0x100	25.00%
30	76.800	77	0x4D	30.08%	307.200	307	0x133	29.98%
35	89.600	90	0x5A	35.16%	358.400	358	0x166	34.96%
40	102.400	102	0x66	39.84%	409.600	410	0x199	40.04%
45	115.200	115	0x73	44.92%	460.800	461	0x1CC	45.02%
50	128.000	128	0x80	50.00%	512.000	512	0x200	50.00%
55	140.800	141	0x8D	55.08%	563.200	563	0x233	54.98%
60	153.600	154	0x9A	60.16%	614.400	614	0x266	59.96%

**表 7-33. デューティ サイクル (続き)**

目標デューティ サイクル (%)	PWM_CTRL.DC_8B = 1 8 ビット				PWM_CTRL.DC_8B = 0 10 ビット			
	デューティ サ イクル カウン ト	入手可能な最 も近いカウン ト	デューティ サ イクル カウン ト (HEX)	実際のデュー ティ サイクル (%)	デューティ サ イクル カウン ト	入手可能な最 も近いカウン ト	デューティ サ イクル カウン ト (HEX)	実際のデュー ティ サイクル (%)
65	166.400	166	0xA6	64.84%	665.600	666	0x299	65.04%
70	179.200	179	0xB3	69.92%	716.800	717	0x2CC	70.02%
75	192.000	192	0xC0	75.00%	768.000	768	0x300	75.00%
80	204.800	205	0xCD	80.08%	819.200	819	0x333	79.98%
85	217.600	218	0xDA	85.16%	870.400	870	0x366	84.96%
90	230.400	230	0xE6	89.84%	921.600	922	0x399	90.04%
95	243.200	243	0xF3	94.92%	972.800	973	0x3CC	95.02%

#### 7.5.7.4 ランプ スロープとスケール係数

スロープ制御は台形 PWM ランプの一部であり、デューティ サイクルやスイッチング周波数などの可変パラメータを時間とともに線形に変化させるために使用されます。このデバイスでは、スロープはスイッチング周波数ランプの場合は Hz/10μs、デューティ サイクル ランプの場合は %/周期 で指定されます。

スロープ値は 21 ビットで構成されており、そのビットは整数フィールドと小数フィールドに分割されます。整数と分数の分割は、PWM\_CTRL.SLOPE\_SCALE ビットによって決定されます。調整可能なので、ユーザーは低速のランプと高速のランプを選択できます。必要な範囲を確認するには、以下の該当するセクションを参照してください。

#### 注

整数ビットと分数ビットの分割は、デューティ サイクルと周波数のどちらが上昇するかによって、同じスロープ スケールで異なります。特定の分割については、以下の各セクションを参照してください。

##### 7.5.7.4.1 デューティ サイクル ランプスロープ

デューティ サイクル ランプの場合、スロープ制御は「%/周期」単位で行われます。つまり、スイッチング周期ごとのデューティ サイクルの変化 (スイッチング周波数) です。

デューティ サイクル値は各周期の開始時に更新されるため、最も遅いランプと最も速いランプは複数の要因に依存します：

- 変更の %。0% ~ 100% は 0% ~ 50% よりも長くかかります。
- 出力分解能：同じ % に対してステップが多いため、10 ビットは 8 ビットより低速です。
- スwitchング周波数：新しい周期の開始時に更新が行われるため、デューティ サイクルの更新頻度が決定されます。

スロープの望ましい値は、所望の変化率 (%) を時間に対する変化として、次の式を用いて計算できます：

$$\text{SLOPE\_CHANGE\_STEPS} = \text{ABS}(\text{SLOPE\_CHANGE\_DEC}) \times (2^{\text{NUM\_DC\_BITS}}) \quad (12)$$

$$\text{SLOPE\_VAL} = \text{SLOPE\_CHANGE\_STEPS} / (t \times f) \quad (13)$$

$$\text{SLOPE\_INT} = \text{TRUNCATE}(\text{SLOPE\_VAL}) \quad (14)$$

$$\text{SLOPE\_FRAC} = \text{ROUND}([\text{SLOPE\_VAL} - \text{TRUNCATE}(\text{SLOPE\_VAL})] \times 2^{\text{NUM\_FRAC\_BITS}}) \quad (15)$$

ここで「SLOPE\_CHANGE\_DECIMAL」は所望の変化率を表します (例：100% から開始して 40% で終了する場合 = 60%、すなわち小数で 0.6 となります)。ただし、この式は SLOPE\_CHANGE\_STEPS を算出するためにのみ使用されます。SLOPE\_CHANGE\_STEPS は、変更されたステップの総数をリストするために使用されます。



たとえば、開始デューティ サイクルが 102/1024 (約 10%) から 768/1024 (約 75%) までの場合、合計の変化は 768 - 102 = 666 ステップです。

「NUM\_DC\_BITS」は、デューティ サイクルの分解能のビット数 (8 または 10 ビット) です。「t」は、変更が発生する時間を秒単位で表したものです。「f」はスイッチング周波数 (Hz)、または 1 秒あたりのパルス数です。

簡単な例として、ユーザーが 2 秒で 25% から 75% までランプさせたい場合、10 ビット解像度モードでスイッチング周波数が 5KHz なら、スロープの変更 = 75% - 25% = 50% (または 0.5)、 $t = 2$ 、 $f = 5000$ 、NUM\_DC\_BITS = 10 (10-bit デューティ サイクル モード) となります。この結果、SLOPE\_VAL は  $(0.50 \times 2^{10}) / (2 \times 5000) = 0.2048$  となります。このスロープ値が小さいため、ランプの分解能を最大化する最良の選択は SLOPE\_SCALE の最大値 3b000 (1.999999) です。したがって SLOPE\_SCALE は 3b000 を選択します。これにより、整数部 1 ビット、固定小数部 20 ビットとなります。SLOPE\_INT = 0、SLOPE\_FRAC =  $0.2048 \times 2^{20} = 214,748.3648$ 。これを四捨五入して 214,748 (10 進) または 0x346DC にします。

スロープ スケール設定は、21 ビットを整数フィールドと小数フィールドの間でどのように分割するかを決定します。より低速のランプを使用するには、より大きな分数フィールドが必要です。

表 7-34. スロープ スケール オプション (デューティ サイクル)

SLOPE_SCALE	NUM_INT_BITS	NUM_FRAC_BITS	最小スロープ値	最大スロープ値
3b000	1 - [20]	20 - [19:0]	1/1,048,576	1.999999
3b001	2 - [20:19]	19 - [18:0]	1/524,288	3.999998
3b010	3 - [20:18]	18 - [17:0]	1/262,144	7.999996
3b011	4 - [20:17]	17 - [16:0]	1/131,072	15.999992
3b100	5 - [20:16]	16 - [15:0]	1/65536	31.999985

表 7-35. デューティ サイクル勾配速度の制限値 (0% ~ 100%、SLOPE\_SCALE = 3b000)

スイッチング周波数 (Hz)	8 ビット デューティ サイクル		10 ビット デューティ サイクル	
	最低速ランプ (秒)	最高速ランプ (秒)	最低速ランプ (秒)	最高速ランプ (秒)
19 (10 ビット モードのみ)	14,128,182	6.7E+0	56,512,728	26.9E+0
50 (10 ビット モードのみ)	5,368,709	2.6E+0	21,474,836	10.2E+0
76	3,532,045	1.7E+0	14,128,182	6.7E+0
100	2,684,355	1.3E+0	10,737,418	5.1E+0
250	1,073,742	512.0E-3	4,294,967	2.0E+0
500	536,871	256.0E-3	2,147,484	1.0E+0
1,000	268,435	128.0E-3	1,073,742	512.0E-3
2,500	107,374	51.2E-3	429,497	204.8E-3
5,000	53,687	25.6E-3	214,748	102.4E-3
7,500	35,791	17.1E-3	143,166	68.3E-3
10,000	26,844	12.8E-3	107,374	51.2E-3
25,000	10,737	5.1E-3	42,950	20.5E-3
39,000	6,883	3.3E-3	27,532	13.1E-3
50,000 (8 ビット モードのみ)	5,369	2.6E-3	21,475	10.2E-3
75,000 (8 ビット モードのみ)	3,579	1.7E-3	14,317	6.8E-3
100,000 (8 ビット モードのみ)	2,684	1.3E-3	10,737	5.1E-3

**表 7-36. デューティ サイクル勾配速度の制限値 (0% ~ 100%、SLOPE\_SCALE = 3b001)**

スイッチング周波数 (Hz)	8 ビット デューティ サイクル		10 ビット デューティ サイクル	
	最低速ランプ (秒)	最高速ランプ (秒)	最低速ランプ (秒)	最高速ランプ (秒)
19 (10 ビット モードのみ)	7,064,091	3.4E+0	28,256,364	13.5E+0
50 (10 ビット モードのみ)	2,684,355	1.3E+0	10,737,418	5.1E+0
76	1,766,023	842.1E-3	7,064,091	3.4E+0
100	1,342,177	640.0E-3	5,368,709	2.6E+0
250	536,871	256.0E-3	2,147,484	1.0E+0
500	268,435	128.0E-3	1,073,742	512.0E-3
1,000	134,218	64.0E-3	536,871	256.0E-3
2,500	53,687	25.6E-3	214,748	102.4E-3
5,000	26,844	12.8E-3	107,374	51.2E-3
7,500	17,896	8.5E-3	71,583	34.1E-3
10,000	13,422	6.4E-3	53,687	25.6E-3
25,000	5,369	2.6E-3	21,475	10.2E-3
39,000	3,441	1.6E-3	13,766	6.6E-3
50,000 (8 ビット モードのみ)	2,684	1.3E-3	10,737	5.1E-3
75,000 (8 ビット モードのみ)	1,790	853.3E-6	7,158	3.4E-3
100,000 (8 ビット モードのみ)	1,342	640.0E-6	5,369	2.6E-3

**表 7-37. デューティ サイクル勾配速度の制限値 (0% ~ 100%、SLOPE\_SCALE = 3b010)**

スイッチング周波数 (Hz)	8 ビット デューティ サイクル		10 ビット デューティ サイクル	
	最低速ランプ (秒)	最高速ランプ (秒)	最低速ランプ (秒)	最高速ランプ (秒)
19 (10 ビット モードのみ)	3,532,045	1.7E+0	14,128,182	6.7E+0
50 (10 ビット モードのみ)	1,342,177	640.0E-3	5,368,709	2.6E+0
76	883,011	421.1E-3	3,532,045	1.7E+0
100	671,089	320.0E-3	2,684,355	1.3E+0
250	268,435	128.0E-3	1,073,742	512.0E-3
500	134,218	64.0E-3	536,871	256.0E-3
1,000	67,109	32.0E-3	268,435	128.0E-3
2,500	26,844	12.8E-3	107,374	51.2E-3
5,000	13,422	6.4E-3	53,687	25.6E-3
7,500	8,948	4.3E-3	35,791	17.1E-3
10,000	6,711	3.2E-3	26,844	12.8E-3
25,000	2,684	1.3E-3	10,737	5.1E-3
39,000	1,721	820.5E-6	6,883	3.3E-3
50,000 (8 ビット モードのみ)	1,342	640.0E-6	5,369	2.6E-3
75,000 (8 ビット モードのみ)	895	426.7E-6	3,579	1.7E-3
100,000 (8 ビット モードのみ)	671	320.0E-6	2,684	1.3E-3



表 7-38. デューティ サイクル勾配速度の制限値 (0% ~ 100%、SLOPE\_SCALE = 3b011)

スイッチング周波数 (Hz)	8 ビット デューティ サイクル		10 ビット デューティ サイクル	
	最低速ランプ (秒)	最高速ランプ (秒)	最低速ランプ (秒)	最高速ランプ (秒)
19 (10 ビット モードのみ)	1,766,023	842.1E-3	7,064,091	3.4E+0
50 (10 ビット モードのみ)	671,089	320.0E-3	2,684,355	1.3E+0
76	441,506	210.5E-3	1,766,023	842.1E-3
100	335,544	160.0E-3	1,342,177	640.0E-3
250	134,218	64.0E-3	536,871	256.0E-3
500	67,109	32.0E-3	268,435	128.0E-3
1,000	33,554	16.0E-3	134,218	64.0E-3
2,500	13,422	6.4E-3	53,687	25.6E-3
5,000	6,711	3.2E-3	26,844	12.8E-3
7,500	4,474	2.1E-3	17,896	8.5E-3
10,000	3,355	1.6E-3	13,422	6.4E-3
25,000	1,342	640.0E-6	5,369	2.6E-3
39,000	860	410.3E-6	3,441	1.6E-3
50,000 (8 ビット モードのみ)	671	320.0E-6	2,684	1.3E-3
75,000 (8 ビット モードのみ)	447	213.3E-6	1,790	853.3E-6
100,000 (8 ビット モードのみ)	336	160.0E-6	1,342	640.0E-6

表 7-39. デューティ サイクル勾配速度の制限値 (0% ~ 100%、SLOPE\_SCALE = 3b100)

スイッチング周波数 (Hz)	8 ビット デューティ サイクル		10 ビット デューティ サイクル	
	最低速ランプ (秒)	最高速ランプ (秒)	最低速ランプ (秒)	最高速ランプ (秒)
19 (10 ビット モードのみ)	883,011	421.1E-3	3,532,045	1.7E+0
50 (10 ビット モードのみ)	335,544	160.0E-3	1,342,177	640.0E-3
76	220,753	105.3E-3	883,011	421.1E-3
100	167,772	80.0E-3	671,089	320.0E-3
250	67,109	32.0E-3	268,435	128.0E-3
500	33,554	16.0E-3	134,218	64.0E-3
1,000	16,777	8.0E-3	67,109	32.0E-3
2,500	6,711	3.2E-3	26,844	12.8E-3
5,000	3,355	1.6E-3	13,422	6.4E-3
7,500	2,237	1.1E-3	8,948	4.3E-3
10,000	1,678	800.0E-6	6,711	3.2E-3
25,000	671	320.0E-6	2,684	1.3E-3
39,000	430	205.1E-6	1,721	820.5E-6
50,000 (8 ビット モードのみ)	336	160.0E-6	1,342	640.0E-6
75,000 (8 ビット モードのみ)	224	106.7E-6	895	426.7E-6
100,000 (8 ビット モードのみ)	168	80.0E-6	671	320.0E-6

#### 7.5.7.4.2 スイッチング周波数のランプ スロープ

スイッチング周波数のランプ制御では、スロープは「Hz/10μs」の単位で行われます。

更新が行われる周期は固定の 10μs であるため、ランプに必要な最小時間と最大時間は、周波数の変化量の大きさのみに依存します。

スロープの希望値は、時間あたりの希望する Hz/ 秒の変化量から、次の式を用いて計算できます：

$$\text{SLOPE\_VAL} = (\text{ABS}(\text{Freq Change}) \times 10\text{E-6}) / t$$

$$\text{SLOPE\_INT} = \text{TRUNC}(\text{SLOPE\_VAL})$$

$$\text{SLOPE\_FRAC} = \text{ROUND}([\text{SLOPE\_VAL} - \text{TRUNC}(\text{SLOPE\_VAL})] \times 2^{\text{NUM\_FRAC\_BITS}})$$

ここで、「周波数変更」は周波数の変化です (例：200Hz で開始し、2kHz ~ 1,800kHz で終了)。「t」は、変化が生じる時間 (秒) を表します。

簡単な例として、ユーザーが 250Hz から 2KHz までを 4 秒で上昇させたい場合、SLOPE\_SCALE = 0b000 で動作しているとする、周波数変化量 = 2000 - 250 = 1750、t = 4、NUM\_FRAC\_BITS = 16 となります。これにより、SLOPE\_VAL は  $(1750 \times 10\text{E-6}) / 4 = 0.07$  となります。SLOPE\_INT = 0、SLOPE\_FRAC = ROUND  $(0.07 \times 2^{16}) = 4588$ 。

表 7-40. スロープ スケール オプション (スイッチング周波数)

SLOPE_SCALE	NUM_INT_BITS	NUM_FRAC_BITS	最小スロープ値 (Hz/10μs)	最大スロープ値 (Hz/10μs)	最小スロープ値 (Hz/s)	最大スロープ値 (Hz/s)
3b000	5 - [20:16]	16 - [15:0]	1/65536	31.99998	1.526	3.2E+6
3b001	9 - [20:12]	12 - [11:0]	1/4096	511.99976	24.414	51.2E+6
3b010	13 - [20:8]	8 - [7:0]	1/256	8,191.99609	390.625	819.2E+6
3b011	17 - [20:4]	4 - [3:0]	1/16	131,071.93750	6250	13.1E+9
3b100	21 - [20:0]	0 - なし	1/1	2,097,151.00000	100000	209.7E+9

#### 7.5.7.5 自動減速および停止条件

TCAN5102-Q1 は、アップまたはダウン ランプのサポートに加えて、アップ ストリーム コントローラからの「開始」信号のみで加速および減速ランプ全体を実行する機能もサポートしています。これはパルス数をカウントするカウンタを設定することで構成され、『ストップ / オフ』ランプを開始するタイミングを決定するために使用されます。許容される最大パルス数のための別のカウンタもあります。これは、PWM ランプ中に所定の合計パルス数が生成された時点で PWM 出力を停止するためのハード カットオフとして機能します。ストップ機能は、すべての PWM モード (デューティサイクル、周波数、静的) でサポートされています。

ユーザーが選択可能な GPIO を使って、ストップ ランプまたは即時オフ状態をトリガできます。これは、リミット スイッチや何らかの故障が発生して PWM 出力を直ちにディセーブルにするのに便利です。

##### 7.5.7.5.1 パルス カウントによる自動減速オフランプ

TCAN5102-Q1 では、パルス カウンタを使用してオフランプを開始するタイミングを計測できます。PWM\_PULSE\_STOP\_RAMP レジスタは、32 ビット カウンタの比較値を設定するために使用されます。新しいランプが開始されると、内部パルス カウンタがリセットされ (PWM\_CUR\_PULSE を読み取ることで確認可能)、各パルスの開始時にカウントアップを開始します。START ビットがセットされたときに PWM.AUTO\_STOP ビットが 1 に設定されていた場合、PWM\_CUR\_PULSE が PWM\_PULSE\_STOP\_RAMP の値に一致すると、PWM\_IP は PWM\_ACTION.STOP ビットに 1 を書き込み、自動的に OFF\_RAMP を開始します。

次に、PWM\_PULSE\_MAX レジスタを使用して、PWM 出力をオフにするタイミングを決定します。PWM\_PULSE\_STOP\_RAMP レジスタと同様に、この値は PWM\_CUR\_PULSE レジスタと比較されます。PWM\_PULSE\_MAX が PWM\_CUR\_PULSE レジスタと一致すると、PWM 出力はオフになります。PWM\_PULSE\_MAX レジスタは、PWM\_PULSE\_STOP\_RAMP よりも優先されるため、PWM\_PULSE\_MAX の値が PWM\_PULSE\_STOP\_RAMP の値よりも小さい場合、オフランプは発生しません。

図 7-12 に、標準的な自動オフランプを示します。P<sub>E\_OFF</sub> 値に達すると、PWM\_PULSE\_MAX 値に達して出力がオフになるまで、出力は継続されます。図 7-13 では、PWM\_PULSE\_MAX の値が

PWM\_PULSE\_STOP\_RAMP に非常に近く、オフランプが完了するのに十分なパルスを許容していないため、オフランプの途中で出力がシャットオフされます。

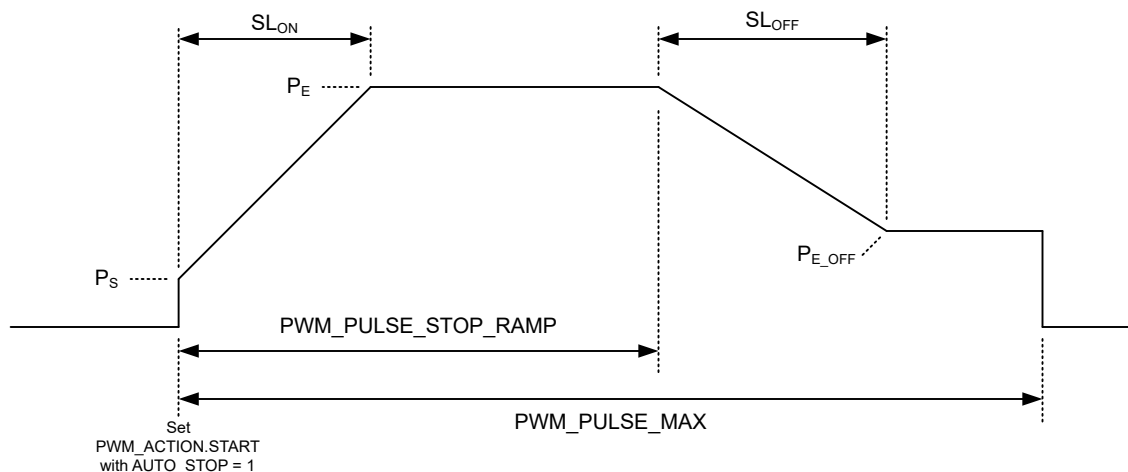


図 7-12. 自動オフ例 1

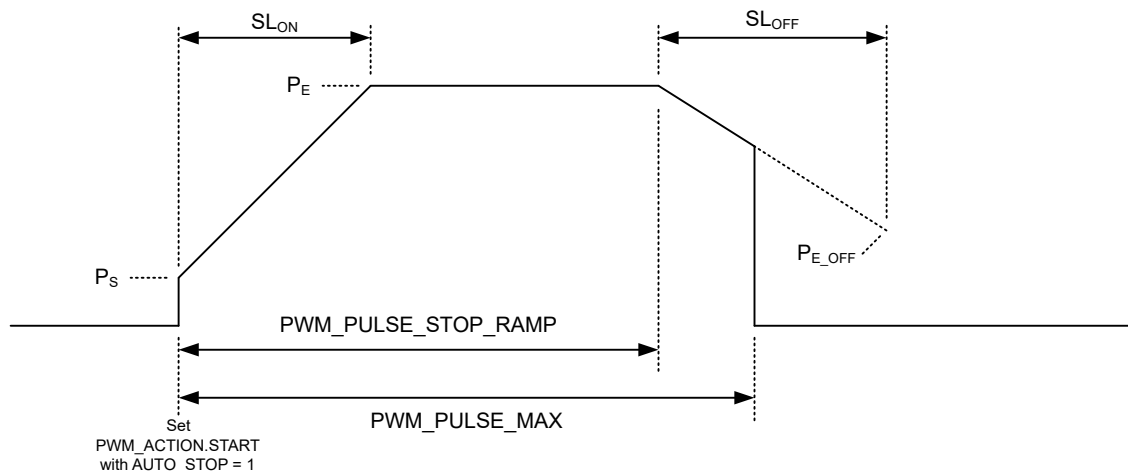


図 7-13. 自動オフ例 2

#### 7.5.7.5.2 GPIO 入力による自動減速 / オフランプ

TCAN5102-Q1 は GPIO 入力を使用して、オフランプをトリガできます。この機能の構成は PWM\_IAS\_CTRL レジスタにあり、トリガとして使用する GPIO を極性とともを選択できます。GPIO が選択された極性に切り替わった後の動作は、PWM\_IAS\_CTRL.STOP\_MODE の設定によって決まります。STOP\_MODE が 0 に設定されている場合、PWM\_ACTION.STOP ビットがセットされ、オフランプを開始します。STOP\_MODE が 1 に設定されている場合、GPIO がアクティブ極性にトグルされた時点で PWM 出力は即座に停止します。

図 7-14 に、GPIO 入力がアクティブ high 極性にトグルすると、標準的な自動オフランプが示されています。P\_E\_OFF 値に達すると、出力はオフになります。図 7-15 STOP\_MODE が 1 に設定されている場合、オフランプなしで、出力は直ちにオフになります。

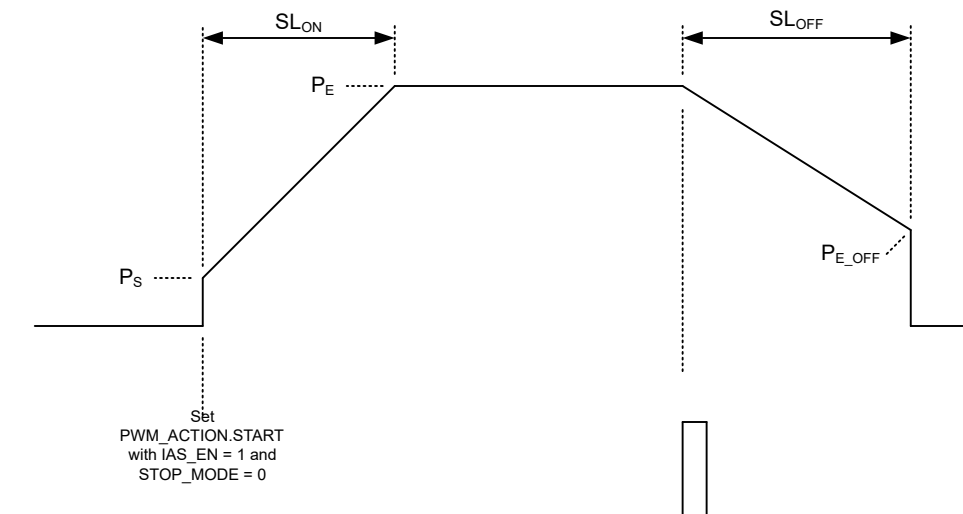


図 7-14. GPIO トリガ オフの例 1

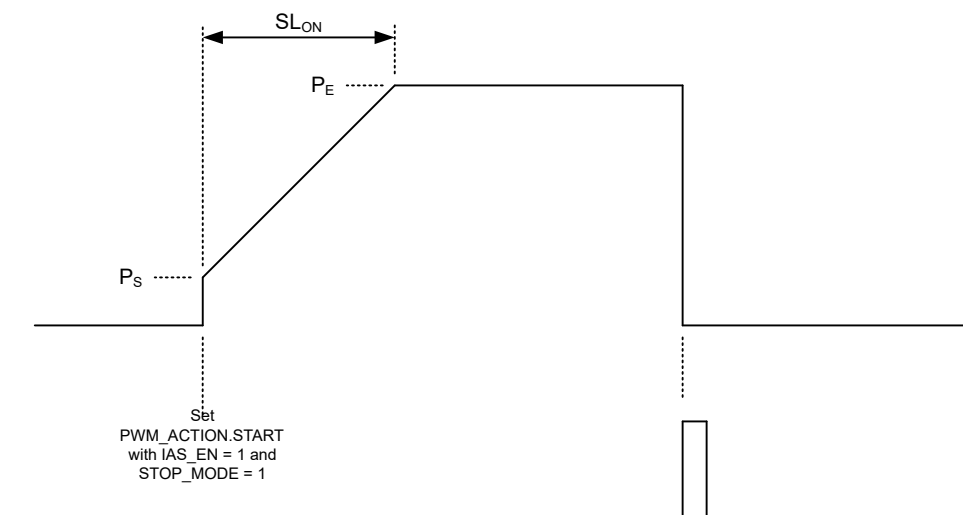


図 7-15. GPIO トリガ オフの例 2

#### 7.5.7.6 デューティ サイクル ランプの例

この例は、以下のパラメータでデューティ サイクルのランプを設定する基本的な例です。

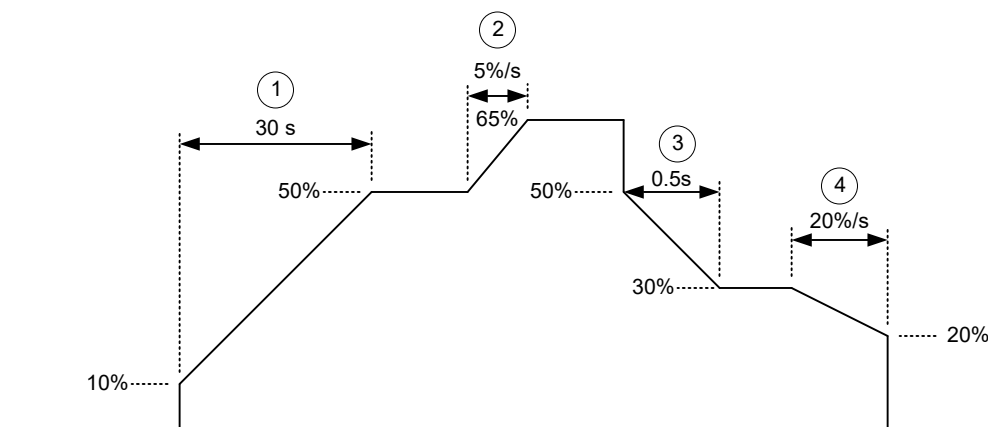


図 7-16. PWM DC ランプの例

表 7-41. 設計パラメータと前提条件

パラメータ	値
スイッチング周波数	20kHz
出力分解能	10 ビット
スロープ スケール	0b000 (1/1,048,576)

上記のランプ プロファイルは、4 つの個別の台形波ランプに分類できます。

1. オンランプ : 10% から 50% までを 30 秒
2. オンランプ : 50% から 65% までを 1 秒あたり 5% の割合
3. オンランプ : ステップ関数: 50% から 30% までを 500ms
4. オフランプ : 20% までを 1 秒あたり 20% の割合でランプし、その後出力をオフにします。

スロープ値を変更する目的は、スロープ値を絶対時間から計算することも、変化率の目標から計算することもできることを示すためです。この例は 4 つの個別のランプで構成されているため、それぞれのランプを分解して説明します。これら 4 つの例は、同じ結果に到達するためのいくつかの異なる方法 (異なる入力単位や一部のステップの組み合わせ) を示しています。

表 7-42. ランプ 1 : 30 秒で 10% から 50%

ステップ	パラメータ	例	説明	値
1	開始値	$10\% \times 1024 = 102.4 \Rightarrow 102$	開始デューティ サイクルを 10 ビット値に変換します	d102 または 0x066
2	停止値	$50\% \times 1024 = 512$	終了デューティ サイクルを 10 ビット値に変換します	d512 または 0x200
3	スロープ計算	$(512-102) = 410$	終了値と開始値の差を計算します	
		$410 / 30s = 13.6666667$	ランプする時間の差を計算します	
		$13.6666667 / 20kHz = 0.00068333 \text{ カウント / サイクル}$	デューティ サイクル (SLOPE_VAL) ごとのデューティ サイクル カウント値 (10 ビット値) の変化を計算します	
		$0.00068333 \times 1,048,576 = 716.52 \Rightarrow 717$	現在のスロープ スケール ファクタに基づいて分数値を計算します。最も近い整数値に丸めます	d717 または 0x2CD

表 7-43. ランプ 2 : 50% から 65% までを 1 秒あたり 5% の割合

ステップ	パラメータ	例	説明	値
1	開始値	-	「現在の PWM 値を使用」フラグを設定するため不要	-
2	停止値	$65\% \times 1024 = 665.6 \Rightarrow 666$	終了デューティ サイクルを 10 ビット値に変換します	d666 または 0x29A

**表 7-43. ランプ 2 : 50% から 65% までを 1 秒あたり 5% の割合 (続き)**

ステップ	パラメータ	例	説明	値
3	スロープ計算	$(5\% \times 1024) / 1s = 51.2 \text{ カウント/秒}$	%/s のスロープを カウント/秒 に変換します	
		$51.2 / 20\text{kHz} = 0.00256 \text{ カウント / サイクル}$	スイッチング周波数で割ることで、1 サイクルあたりのカウント変化量を求めます	
		$0.00256 \times 1,048,576 = 2684.35 \Rightarrow 2684$	現在のスロープ スケール ファクタに基づいて分数値を計算します。最も近い整数値に丸めます	d2684 または 0xA7C

**表 7-44. ランプ 3 : 0.5 秒で 50% から 30%**

ステップ	パラメータ	例	説明	値
1	開始値	$50\% \times 1024 = 512$	開始デューティ サイクルを 10 ビット値に変換します	d512 または 0x200
2	停止値	$30\% \times 1024 = 307.2 \Rightarrow 307$	終了デューティ サイクルを 10 ビット値に変換します	d307 または 0x133
3	スロープ計算	$512 - 307 = 205$	終了値と開始値の差を計算します	
		$205 / 0.5 \text{ 秒} = 410 \text{ カウント / 秒}$	ランプする時間の差を除算します	
		$410 / 20\text{kHz} = 0.0205 \text{ カウント / サイクル}$	カウント/秒をカウント / サイクル に変換します	
		$0.0205 \times 1,048,576 = 21,498.8 \Rightarrow 21,499$	現在のスロープ スケール ファクタに基づいて分数値を計算します。最も近い整数値に丸めます	d21499 または 0x53FB

**表 7-45. ランプ 4 : x% から 20% までを 1 秒あたり 20% の割合**

ステップ	パラメータ	例	説明	値
1	開始値	-	停止ランプでは使用しません。電流値が常に使用されます。	-
2	停止値	$20\% \times 1024 = 204.8$	終了デューティ サイクルを 10 ビット値に変換します	d205 または 0x0CD
3	スロープ計算	$20\%/s \times 1024 = 204.8 \text{ カウント/秒}$	スロープを 10 ビット値に変換します	
		$204.8 / 20\text{kHz} = 0.01024 \text{ カウント / サイクル}$	スロープをサイクルあたりのカウントに変換します	
		$0.01024 \times 1,048,576 = 10,737.4 \Rightarrow 10,737$	現在のスロープ スケール ファクタに基づいて分数値を計算します。最も近い整数値に丸めます	d10737 または 0x29F1

### 7.5.7.7 周波数ランプの例

この例は、以下のパラメータでスイッチング周波数ランプを設定する基本的な例です。

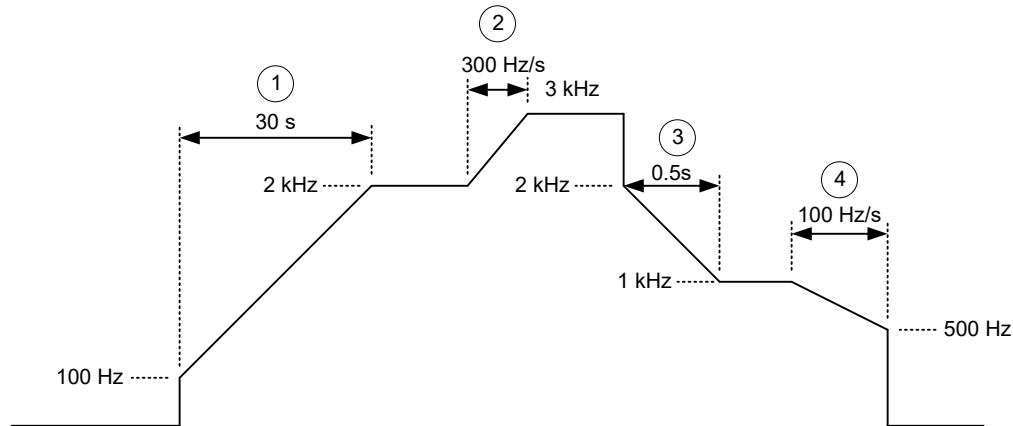


図 7-17. PWM スwitching 周波数ランプの例

表 7-46. 設計パラメータと前提条件

パラメータ	値
スイッチング周波数	20kHz
出力分解能	8 ビット
スロープ スケール	0b000 (1/65536)

上記のランプ プロファイルは、4 つの個別の台形波ランプに分類できます。

1. オンランプ : 30 秒で 100Hz ~ 2kHz
2. オンランプ : 3kHz までを 1 秒あたり 300Hz の割合で上昇させます
3. オンランプ : 2kHz ~ 1kHz の範囲で 0.5 秒のステップ機能
4. オフランプ : 500Hz までを 1 秒あたり 100 Hz の割合で上昇させます

スロープ値を変更する目的は、スロープ値を絶対時間から計算すること、変化率の目標から計算することでもできることを示すためです。この例は 4 つの個別のランプで構成されているため、それぞれのランプを分解して説明します。

表 7-47. ランプ 1 : 30 秒で 100Hz ~ 2kHz

ステップ	パラメータ	例	説明	値
1	開始値	$40\text{MHz} / (100\text{Hz} \times 2^8) = 1562.5$	開始周波数除数を計算します	
		$\text{TRUNC}(1562.5) = 1562$	除数の整数部分を計算します	d1562 または 0x61A
		$\text{ROUND}(0.5 \times 128) = 64$	除数の分数部分を計算します	d64 または 0x40
2	停止値	$40\text{MHz} / (2\text{kHz} \times 2^8) = 78.125$	停止周波数除数を計算します	
		$\text{TRUNC}(78.125) = 78$	除数の整数部分を計算します	d78 または 0x4E
		$\text{ROUND}(0.125 \times 128) = 16$	除数の分数部分を計算します	d16 または 0x10



**表 7-47. ランプ 1 : 30 秒で 100Hz ~ 2kHz (続き)**

ステップ	パラメータ	例	説明	値
3	スロープ計算	$(2\text{kHz} - 100\text{Hz}) / 30\text{s} = 63.33\text{Hz/s}$	スロープを Hz/s で計算します	
		$63.33 \times 10\mu\text{s} = 0.06333\text{Hz}/10\mu\text{s}$	スロープを 10 $\mu\text{s}$ に変換します	
		$\text{TRUNC}(0.06333) = 0$	スロープの整数値を計算します	d0 または 0x00
		$\text{ROUND}(0.06333 \times 65536) = 42$	スロープ スケール係数に基づいて分数値を計算します。	d42 または 0x02A

**表 7-48. ランプ 2 : 3kHz までを 1 秒あたり 300Hz の割合で上昇させます**

ステップ	パラメータ	例	説明	値
1	開始値	-	「現在の PWM 値を使用」フラグを設定するため不要	-
2	停止値	$40\text{MHz} / (3\text{kHz} \times 2^8) = 52.083$	停止周波数除数を計算します	
		$\text{TRUNC}(52.083) = 52$	除数の整数部分を計算します	d52 または 0x34
		$\text{ROUND}(0.083 \times 128) = 12$	除数の分数部分を計算します	d12 または 0x0C
3	スロープ計算	300Hz/s (指定値)	スロープを Hz/s で計算します	
		$300 \times 10\mu\text{s} = 0.003\text{Hz}/10\mu\text{s}$	スロープを 10 $\mu\text{s}$ に変換します	
		$\text{TRUNC}(0.003) = 0$	スロープの整数値を計算します	d0 または 0x00
		$\text{ROUND}(0.003 \times 65536) = 197$	スロープ スケール係数に基づいて分数値を計算します。	d197 または 0x0C5

**表 7-49. ランプ 3 : 0.5 秒で 2kHz を 1kHz に上昇させます**

ステップ	パラメータ	例	説明	値
1	開始値	$40\text{MHz} / (2\text{kHz} \times 2^8) = 78.125$	開始周波数除数を計算します	
		$\text{TRUNC}(78.125) = 78$	除数の整数部分を計算します	d78 または 0x4E
		$\text{ROUND}(0.125 \times 128) = 16$	除数の分数部分を計算します	d16 または 0x10
2	停止値	$40\text{MHz} / (1\text{kHz} \times 2^8) = 156.25$	停止周波数除数を計算します	
		$\text{TRUNC}(156.25) = 156$	除数の整数部分を計算します	d156 または 0x9C
		$\text{ROUND}(0.25 \times 128) = 32$	除数の分数部分を計算します	d32 または 0x20
3	スロープ計算	$(2\text{kHz} \sim 1\text{kHz}) / 0.5\text{s} = 2000\text{Hz/s}$	スロープを Hz/s で計算します	
		$2000 \times 10\mu\text{s} = 0.02\text{Hz}/10\mu\text{s}$	スロープを 10 $\mu\text{s}$ に変換します	
		$\text{TRUNC}(0.02) = 0$	スロープの整数値を計算します	d0 または 0x00
		$\text{ROUND}(0.02 \times 65536) = 1311$	スロープ スケール係数に基づいて分数値を計算します。	d1311 または 0x51F

**表 7-50. ランプ 4 : 500Hz までを 1 秒あたり 100 Hz の割合で上昇させます**

ステップ	パラメータ	例	説明	値
1	開始値	-	停止ランプでは使用しません。電流値が常に使用されます。	-

表 7-50. ランプ 4 : 500Hz までは 1 秒あたり 100 Hz の割合で上昇させます (続き)

ステップ	パラメータ	例	説明	値
2	停止値	$40\text{MHz} / (500\text{Hz} \times 2^8) = 312.5$	停止周波数除数を計算します	
		$\text{TRUNC}(312.5) = 312$	除数の整数部分を計算します	d312 または 0x138
		$\text{ROUND}(0.5 \times 128) = 64$	除数の分数部分を計算します	d64 または 0x40
3	スロープ計算	100Hz/s (指定値)	スロープを Hz/s で計算します	
		$100 \times 10\mu\text{s} = 0.001\text{Hz}/10\mu\text{s}$	スロープを 10 $\mu\text{s}$ に変換します	
		$\text{TRUNC}(0.001) = 0$	スロープの整数値を計算します	d0 または 0x00
		$\text{ROUND}(0.001 \times 65536) = 66$	スロープ スケール係数に基づいて分数値を計算します。	d66 または 0x042

## 7.5.7.8 静的オンの例

この例は、単純な静的オン PWM 波形を構成する基本的な例です。

表 7-51. 設計パラメータと前提条件

パラメータ	値
スイッチング周波数	10kHz
出力分解能	8 ビット
デューティ サイクル	20%

単純な静的 PWM 波形のために構成する必要があるレジスタはわずかです。必要な一般的なプロセスとレジスタは次のとおりです：

1. 構成レジスタでは、静的オンにモードを設定し、分解能を 8 ビット モードに設定する必要があります
2. 周波数用には、PWM\_END\_VAL\_CONST\_FRAC\_F、PWM\_END\_VAL\_MSB、および PWM\_END\_VAL\_LSB レジスタを使用します
3. デューティ サイクル用には、PWM\_CONST\_MSB および PWM\_CONST\_LSB レジスタを使用します
4. PWM\_ACTION レジスタのスタートビットに書き込みます

表 7-52. 静的 PWM の計算

ステップ	パラメータ	例	説明	値
1	スイッチング周波数	$40\text{MHz} / (10\text{kHz} \times 2^8) = 15.625$	スイッチング周波数除数を計算します	
		$\text{TRUNC}(15.625) = 15$	除数の整数部分を計算します	d15 または 0x00F
		$\text{ROUND}(0.625 \times 128) = 80$	除数の分数部分を計算します	d80 または 0x50
2	デューティ サイクル	$\text{ROUND}(20\% \times 256) = 51.2$	最も近い整数のデューティ サイクルのカウント値を算出して取得します。	d51 または 0x33

パラメータの計算後、以下の例に示すように、パラメータは適切なレジスタに書き込まれます。

表 7-53. レジスタ書き込み

ステップ	登録	データ (Hex)	説明
1	PWM_CTRL	0x0F	INIT をセットし、PWM モードを静的オンに設定し、8 ビットのデューティ サイクル分解能を有効にします
2	PWM_END_VAL_CONST_FRAC_F	0x50	スイッチング周波数の分数除数を書き込みます
3	PWM_END_VAL_MSB	0x00	スイッチング周波数の除数の MSB を書き込みます
4	PWM_END_VAL_LSB	0x0F	スイッチング周波数の除数の LSB を書き込みます

**表 7-53. レジスタ書き込み (続き)**

ステップ	登録	データ (Hex)	説明
5	PWM_CONST_MSB	0x00	デューティ サイクル カウントの MSB を書き込みます。この例は 8 ビット モードであるため、これは実際には重要ではありません
6	PWM_CONST_LSB	0x40	デューティ サイクル カウントの LSB を書き込みます
7	PWM_CTRL	0x0E	INIT ビットを無効にして IP を有効にします
8	PWM_ACTION	0x02	PWM 出力をオンにします

**注**

PWM モジュールの出力は、IO\_CFG\_0 レジスタの GPIO に多重化する必要があります。それ以外の場合、PWM 出力はピンにありません。

## 7.6 レジスタ マップ

TCAN5102-Q1 には、16 ビット アドレッシングを持つ包括的なレジスタ セットがあります。レジスタは、いくつかのセクションに分かれています：

- デバイスのレジスタ ([セクション 7.6.1](#))
- SPI レジスタ ([セクション 7.6.2](#) および [セクション 7.6.3](#))
- UART レジスタ ([セクション 7.6.4](#) および [セクション 7.6.5](#))
- I2C レジスタ ([セクション 7.6.6](#) および [セクション 7.6.7](#))
- PWM0 レジスタ ([セクション 7.6.8](#))
- PWM1 レジスタ ([セクション 7.6.9](#))

### 7.6.1 DEVICE レジスタ

[セクション 7.6.1](#) に、デバイスのレジスタ用のメモリ マップト レジスタを示します。[セクション 7.6.1](#) がないレジスタ オフセット アドレスはすべて予約済みと見なして、レジスタの内容は変更しないでください。

**表 7-54. DEVICE レジスタ**

オフセット	略称	レジスタ名	セクション
0h + 式	DEV_ID[y]	デバイス ID	<a href="#">セクション 7.6.1.1</a>
8h	DEV_ID_REV	デバイス リビジョン	<a href="#">セクション 7.6.1.2</a>
9h	DEV_CMD	デバイスのコマンド	<a href="#">セクション 7.6.1.3</a>
Ah	DEV_CFG_EN	デバイス設定イネーブル	<a href="#">セクション 7.6.1.4</a>
Bh	DEV_CFG_BR	デバイス CAN ボーレート	<a href="#">セクション 7.6.1.5</a>
Ch	DEV_CFG_ID_0	デバイスの CAN ID	<a href="#">セクション 7.6.1.6</a>
Dh	DEV_CFG_ID_1	デバイスの CAN ID	<a href="#">セクション 7.6.1.7</a>
Eh	DEV_CFG_ID_BCST_0	デバイス CAN ID (ブロードキャスト)	<a href="#">セクション 7.6.1.8</a>
Fh	DEV_CFG_ID_BCST_1	デバイス CAN ID (ブロードキャスト)	<a href="#">セクション 7.6.1.9</a>
10h	DEV_CFG_ID_BCST_MASK_0	デバイス CAN ID (ブロードキャスト) マスク	<a href="#">セクション 7.6.1.10</a>
11h	DEV_CFG_ID_BCST_MASK_1	デバイス CAN ID (ブロードキャスト) マスク	<a href="#">セクション 7.6.1.11</a>
12h	DEV_CFG_NVM_PROG_CODE	デバイス NVM プログラミング キー	<a href="#">セクション 7.6.1.12</a>
13h	DEV_CFG_NVM_PROG	デバイス NVM プログラミング	<a href="#">セクション 7.6.1.13</a>
1Dh	INT_CFG	GPIO 割り込み出力構成	<a href="#">セクション 7.6.1.14</a>
1Eh	DEV_IE_0	デバイス割り込みイネーブル 0	<a href="#">セクション 7.6.1.15</a>
1Fh	DEV_IE_1	デバイス割り込みイネーブル 1	<a href="#">セクション 7.6.1.16</a>
20h	MRAM_IP_CFG	MRAM と IP の構成	<a href="#">セクション 7.6.1.17</a>
21h	IO_CFG_0	GPIO ピン モード構成	<a href="#">セクション 7.6.1.18</a>
22h	IO_CFG_1	GPIO ピン モード構成	<a href="#">セクション 7.6.1.19</a>
23h	IO_OE_0	GPIO 出力または入力の構成	<a href="#">セクション 7.6.1.20</a>
24h	IO_OE_1	GPIO 出力または入力の構成	<a href="#">セクション 7.6.1.21</a>
25h	IO_OD_0	GPIO オープン ドレインの構成	<a href="#">セクション 7.6.1.22</a>
26h	IO_OD_1	GPIO オープン ドレインの構成	<a href="#">セクション 7.6.1.23</a>
27h	IO_RE_0	GPIO 抵抗のイネーブル	<a href="#">セクション 7.6.1.24</a>
28h	IO_RE_1	GPIO 抵抗のイネーブル	<a href="#">セクション 7.6.1.25</a>
29h	IO_PU_0	GPIO プルアップ抵抗のイネーブル	<a href="#">セクション 7.6.1.26</a>
2Ah	IO_PU_1	GPIO プルアップ抵抗のイネーブル	<a href="#">セクション 7.6.1.27</a>
2Bh	IO_POL_0	GPIO 極性反転有効	<a href="#">セクション 7.6.1.28</a>
2Ch	IO_POL_1	GPIO 極性反転有効	<a href="#">セクション 7.6.1.29</a>
2Dh	IO_OUTPUT_0	GPIO 出力値	<a href="#">セクション 7.6.1.30</a>
2Eh	IO_OUTPUT_1	GPIO 出力値	<a href="#">セクション 7.6.1.31</a>
2Fh	IO_INPUT_0	GPIO 入力値	<a href="#">セクション 7.6.1.32</a>

**表 7-54. DEVICE レジスタ (続き)**

オフセット	略称	レジスタ名	セクション
30h	IO_INPUT_1	GPIO 入力値	<a href="#">セクション 7.6.1.33</a>
31h	IR_STATUS	割り込みステータス	<a href="#">セクション 7.6.1.34</a>
32h	DEV_IR	デバイス割り込みレジスタ	<a href="#">セクション 7.6.1.35</a>
33h	SPI_IR	SPI 割り込みレジスタ	<a href="#">セクション 7.6.1.36</a>
34h	UART_IR	UART 割り込みレジスタ	<a href="#">セクション 7.6.1.37</a>
35h	I2C_IR	I2C 割り込みソーレジスタ	<a href="#">セクション 7.6.1.38</a>
36h	PWM0_IR	PWM0 割り込みレジスタ	<a href="#">セクション 7.6.1.39</a>
37h	PWM1_IR	PWM1 割り込みソーレジスタ	<a href="#">セクション 7.6.1.40</a>

表の小さなセルに収まるように、複雑なビット アクセス タイプを記号で表記しています。[セクション 7.6.1](#) に、このセクションでアクセス タイプに使用しているコードを示します。

**表 7-55. デバイスのアクセス タイプ コード**

アクセス タイプ	コード	説明
読み取りタイプ		
R	R	読み出し
RH	R H	ハードウェアによってセットまたはクリアされる の読み取り
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	1 を書き込むことで セット
WP	W P	書き込み 特権アクセスが必要
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ アレイ変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ グループの一部であるレジスタ アレイの値を示します。レジスタ グループは階層構造を形成し、アレイは式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ アレイの値を示します。

### 7.6.1.1 DEV\_ID[y] レジスタ (オフセット = 0h + 式) [リセット = 0032303135414354h]

図 7-18 に、DEV\_ID[y] を示し、表 7-56 に、その説明を示します。

概略表に戻ります。

ASCII で型番を含むデバイス ID レジスタ

オフセット = 0h + (y \* 8h)、ここで y = 0h ~ 7h

図 7-18. DEV\_ID[y] レジスタ

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
DEV_ID_63:56								DEV_ID_55:48							
R-0h								R-32h							
47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DEV_ID_40:47								DEV_ID_39:32							
R-30h								R-31h							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEV_ID_31:24								DEV_ID_23:16							
R-35h								R-41h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_ID_15:8								DEV_ID_7:0							
R-43h								R-54h							

表 7-56. DEV\_ID[y] レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
63-56	DEV_ID_63:56	R	0h	ヌル文字
55-48	DEV_ID_55:48	R	32h	2 の場合は ASCII
47-40	DEV_ID_40:47	R	30h	0 の場合は ASCII
39-32	DEV_ID_39:32	R	31h	1 の場合は ASCII
31-24	DEV_ID_31:24	R	35h	5 の場合は ASCII
23-16	DEV_ID_23:16	R	41h	A の場合は ASCII
15-8	DEV_ID_15:8	R	43h	C の場合は ASCII
7-0	DEV_ID_7:0	R	54h	T の場合は ASCII

### 7.6.1.2 DEV\_ID\_REV レジスタ (オフセット = 8h) [リセット= 10h]

図 7-19 に、DEV\_ID\_REV を示し、表 7-57 に、その説明を示します。

概略表に戻ります。

**図 7-19. DEV\_ID\_REV レジスタ**

7	6	5	4	3	2	1	0
DEV_ID_MAJOR				DEV_ID_MINOR			
R-1h				R-0h			

**表 7-57. DEV\_ID\_REV レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-4	DEV_ID_MAJOR	R	1h	デバイスのメジャー リビジョン
3-0	DEV_ID_MINOR	R	0h	デバイスのマイナー リビジョン



## 7.6.1.3 DEV\_CMD レジスタ (オフセット = 9h) [リセット= 00h]

図 7-20 に、DEV\_CMD を示し、表 7-58 に、その説明を示します。

概略表に戻ります。

図 7-20. DEV\_CMD レジスタ

7	6	5	4	3	2	1	0
RSVD							GO_TO_SLEEP
R-0h							R/W1S-0h

表 7-58. DEV\_CMD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-1	RSVD	R	0h	予約済み
0	GO_TO_SLEEP	R/W1S	0h	このビットをセットしてスリープモードに移行 0h = 通常モードにとどまる 1h = スリープ状態に移行する

#### 7.6.1.4 DEV\_CFG\_EN レジスタ (オフセット = Ah) [リセット = 00h]

図 7-21 に、DEV\_CFG\_EN を示し、表 7-59 に、その説明を示します。

概略表に戻ります。

デバイス構成レジスタのロックを解除します

図 7-21. DEV\_CFG\_EN レジスタ

7	6	5	4	3	2	1	0
RSVD					CAN_ID_BCST_CCE	CAN_ID_CCE	CAN_BR_CCE
R-0h					R/W-0h	R/W-0h	R/W-0h

表 7-59. DEV\_CFG\_EN レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	CAN_ID_BCST_CCE	R/W	0h	0h = DEV_CFG_ID_BCST と DEV_CFG_ID_BCST_MASK への変更は無効化 1h = DEV_CFG_ID_BCST と DEV_CFG_ID_BCST_MASK への変更は有効化
1	CAN_ID_CCE	R/W	0h	0h = DEV_CFG_ID の変更は無効化 1h = DEV_CFG_ID の変更は有効化
0	CAN_BR_CCE	R/W	0h	0h = DEV_CFG_BR の変更は無効化 1h = DEV_CFG_BR の変更は有効化

### 7.6.1.5 DEV\_CFG\_BR レジスタ (オフセット = Bh) [リセット = 10h]

図 7-22 に、DEV\_CFG\_BR を示し、表 7-60 に、その説明を示します。

概略表に戻ります。

デバイスは、CAN ボーレートできます

#### 注

変更を行うには、DEV\_CFG\_EN で「変更可能」を有効にする必要があります

図 7-22. DEV\_CFG\_BR レジスタ

7	6	5	4	3	2	1	0
RSVD			CAN_ACK_EN	RSVD	CAN_BR		
R-0h			R/WP-1h	R-0h	R/WP-0h		

表 7-60. DEV\_CFG\_BR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	予約済み
4	CAN_ACK_EN	R/WP	1h	メッセージ (ブロードキャストメッセージを含む) を受信するときに ACK ビットを有効にするかどうかを選択します。 2M または 5M では有効化を推奨しませんが、1M 以下の速度ではオプションです。 0h = ACK ビットは無効化 1h = ACK ビットは有効化
3	RSVD	R	0h	予約済み
2-0	CAN_BR	R/WP	0h	CAN のボーレートを選択します。 レスポンスが確認応答フレームを送信した直後に変更が発生します。 0h = 250k 1h = 500k 2h = 1M 3h = 2M (ACK ビットは無効) 4h = 5M (ACK ビットは無効)

### 7.6.1.6 DEV\_CFG\_ID\_0 レジスタ (オフセット = Ch) [リセット = 00h]

図 7-23 に、DEV\_CFG\_ID\_0 を示し、表 7-61 に、その説明を示します。

概略表に戻ります。

レスポンドが待機するデバイスの CAN ID

**注**  
変更を行うには、DEV\_CFG\_EN で「変更可能」を有効にする必要があります

**図 7-23. DEV\_CFG\_ID\_0 レジスタ**

7	6	5	4	3	2	1	0
RSVD					CAN_ID_10:8		
R-0h					R/WP-0h		

**表 7-61. DEV\_CFG\_ID\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2-0	CAN_ID_10:8	R/WP	0h	この特定のレスポンド宛のリクエストに使用される CAN ID[10:8]。レスポンドが応答するには、完全に一致する必要があります。LSB は無視され、リクエストでは常に 0 です。レスポンドは応答時にこのビットを 1 に設定します。

## 7.6.1.7 DEV\_CFG\_ID\_1 レジスタ (オフセット = Dh) [リセット = 00h]

図 7-24 に、DEV\_CFG\_ID\_1 を示し、表 7-62 に、その説明を示します。

概略表に戻ります。

レスポンドが待機するデバイスの CAN ID

## 注

変更を行うには、DEV\_CFG\_EN で「変更可能」を有効にする必要があります

図 7-24. DEV\_CFG\_ID\_1 レジスタ

7	6	5	4	3	2	1	0
CAN_ID_7:1							RSVD
R/WP-0h							R-0h

表 7-62. DEV\_CFG\_ID\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-1	CAN_ID_7:1	R/WP	0h	この特定のレスポンド宛のリクエストに使用される CAN ID。 レスポンドが応答するには、完全に一致する必要があります。 LSB は無視され、リクエストでは常に 0 です。レスポンドは応答時にこのビットを 1 に設定します。
0	RSVD	R	0h	CAN ID の LSB は無視されます。リクエストには 0 が使用され、応答フレームには 1 が使用されるためです

### 7.6.1.8 DEV\_CFG\_ID\_BCST\_0 レジスタ (オフセット = Eh) [リセット = 00h]

図 7-25 に、DEV\_CFG\_ID\_BCST\_0 を示し、表 7-63 に、その説明を示します。

概略表に戻ります。

レスポンドが待機するデバイスがブロードキャストする CAN ID。デバイスはブロードキャスト メッセージに応答フレームを送信しません

#### 注

変更を行うには、DEV\_CFG\_EN で Change Enable を有効にする必要があります

**図 7-25. DEV\_CFG\_ID\_BCST\_0 レジスタ**

7	6	5	4	3	2	1	0
RSVD					CAN_ID_BCST_10:8		
R-0h					R/WP-0h		

**表 7-63. DEV\_CFG\_ID\_BCST\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2-0	CAN_ID_BCST_10:8	R/WP	0h	この特定のレスポンド宛のリクエストに使用される CAN ID。 この ID は、どのビットが一致する必要があるかを決定するために、マスクレジスタと組み合わせて使用されます

#### 7.6.1.9 DEV\_CFG\_ID\_BCST\_1 レジスタ (オフセット= Fh) [リセット= 00h]

図 7-26 に、DEV\_CFG\_ID\_BCST\_1 を示し、表 7-64 に、その説明を示します。

概略表に戻ります。

レスポンドが待機するデバイスがブロードキャストする CAN ID。このデバイスはブロードキャストメッセージに対して応答フレームを送信しません

#### 注

変更を行うには、DEV\_CFG\_EN で「変更可能」を有効にする必要があります

図 7-26. DEV\_CFG\_ID\_BCST\_1 レジスタ

7	6	5	4	3	2	1	0
CAN_ID_BCST_7:0							
R/WP-0h							

表 7-64. DEV\_CFG\_ID\_BCST\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	CAN_ID_BCST_7:0	R/WP	0h	この特定のレスポンド宛のリクエストに使用される CAN ID。 この ID は、どのビットが一致する必要があるかを決定するために、 マスクレジスタと組み合わせて使用されます



#### 7.6.1.10 DEV\_CFG\_ID\_BCST\_MASK\_0 レジスタ (オフセット = 10h) [リセット = 00h]

図 7-27 に、DEV\_CFG\_ID\_BCST\_MASK\_0 を示し、表 7-65 に、その説明を示します。

概略表に戻ります。

レスポンドが待機するデバイスがブロードキャストする CAN ID マスク。このデバイスはブロードキャストメッセージに対して応答フレームを送信しません

**注**  
変更を行うには、DEV\_CFG\_EN で「変更可能」を有効にする必要があります

**図 7-27. DEV\_CFG\_ID\_BCST\_MASK\_0 レジスタ**

7	6	5	4	3	2	1	0
RSVD					CAN_ID_BCST_MSK_10:8		
R-0h					R/WP-0h		

**表 7-65. DEV\_CFG\_ID\_BCST\_MASK\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2-0	CAN_ID_BCST_MSK_10:8	R/WP	0h	このマスクは、CAN_ID_BCST とともに使用され、ブロードキャスト ID が一致する複数の ID を作成します。 「0」は、このビットが CAN_ID_BCST で指定された値と一致している必要があることを意味します。 値が「1」の場合、そのビットは「任意」を意味し、CAN ID 内の当該ビットに対して 0 または 1 のいずれでも受け入れられます。 すべて「0」の場合、ブロードキャスト ID は CAN_ID_BCST フィールドに入力された値と完全に一致する必要があります。

**7.6.1.11 DEV\_CFG\_ID\_BCST\_MASK\_1 レジスタ (オフセット = 11h) [リセット = 00h]**

図 7-28 に、DEV\_CFG\_ID\_BCST\_MASK\_1 を示し、表 7-66 に、その説明を示します。

概略表に戻ります。

レスポンドがリッスンするデバイス ブロードキャスト CAN ID マスク。デバイスはブロードキャスト メッセージに応答フレームを送信しません

**注**

変更を行うには、DEV\_CFG\_EN で Change Enable を有効にする必要があります

**図 7-28. DEV\_CFG\_ID\_BCST\_MASK\_1 レジスタ**

7	6	5	4	3	2	1	0
CAN_ID_BCST_MSK_7:0							
R/WP-0h							

**表 7-66. DEV\_CFG\_ID\_BCST\_MASK\_1 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	CAN_ID_BCST_MSK_7:0	R/WP	0h	このマスクは、CAN_ID_BCST とともに使用され、ブロードキャスト ID が一致する複数の ID を作成します。 「0」は、このビットが CAN_ID_BCST で指定された値と一致している必要があることを意味します。 値が「1」の場合、そのビットは「任意」を意味し、CAN ID 内の当該ビットに対して 0 または 1 のいずれでも受け入れられます。 すべて「0」の場合、ブロードキャスト ID は CAN_ID_BCST フィールドに入力された値と完全に一致する必要があります。

#### 7.6.1.12 DEV\_CFG\_NVM\_PROG\_CODE レジスタ (オフセット = 12h) [リセット = 00h]

図 7-29 に、DEV\_CFG\_NVM\_PROG\_CODE を示し、表 7-67 に、その説明を示します。

概略表に戻ります。

プロセスのロックを解除するには、EEPROM をプログラムする前に、このレジスタに書き込む必要があります。

図 7-29. DEV\_CFG\_NVM\_PROG\_CODE レジスタ

7	6	5	4	3	2	1	0
PROG_UNLOCK							
R/W-0h							

表 7-67. レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	PROG_UNLOCK	R/W	0h	EEPROM をフラッシュするには、このフィールドを 0x78 として書き込む必要があります。 それ以外の値を指定すると、フラッシュ要求は無視されます。

### 7.6.1.13 DEV\_CFG\_NVM\_PROG レジスタ (オフセット = 13h) [リセット = 00h]

表 7-68 に、DEV\_CFG\_NVM\_PROG を示します。

概略表に戻ります。

既存のデバイス構成を、デバイスのワンタイム プログラマブル メモリに保存します。

表 7-68. DEV\_CFG\_NVM\_PROG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
6	PROG_FAIL	RH	0h	EEPROM プログラム プロセスに故障が発生したかどうかを報告するステータス フラグ。 このフラグは、EEPROM をプログラムしようとした後で確認する必要があります。 何らかの種類のエラーが発生した場合、このビットはセットされ、ユーザーは EEPROM を再度プログラムしようと試みる必要があります。 0h = EEPROM には故障は発生していません 1h = EEPROM のプログラムが失敗しました
5	RSVD	R	0h	予約済み
4	EEPROM_FLASHED	RH	0h	EEPROM がすでにフラッシュされているかどうかを問い合わせるためのステータス フラグ 0h = EEPROM はプログラムされていません 1h = EEPROM はプログラム済みです
3	RSVD	R	0h	予約済み
2	LOCK_ID	RH/W1S	0h	この値が設定されている場合、デバイスは CAN_ID_CCE ビットを永続的に無効化し、CAN ID およびマスクを変更できないようにします。 0h = CAN ID とマスク レジスタは更新可能です (レジスタと EEPROM の両方) 1h = CAN ID とマスク レジスタは更新できません (レジスタと EEPROM の両方)
1	LOCK_BR	RH/W1S	0h	この値が設定されている場合、デバイスは CAN_BR_CCE ビットを永続的にディセーブルにし、CAN ボーレートを変更できないようにします。 0h = CAN データ レート レジスタは更新可能です (レジスタと EEPROM の両方) 1h = CAN データ レート レジスタは更新できません (レジスタと EEPROM の両方)
0	PROG	RH/W1S	0h	フラッシュ コードを正しく設定して設定すると、EEPROM がプログラムされます。 このビットは、フラッシュの手順が完了するまで 1 を読み戻します。 0h = EEPROM / プログラムのアクティブなプログラミングは完了していません 1h = EEPROM をアクティブにプログラミングしています

### 7.6.1.14 INT\_CFG レジスタ (オフセット = 1Dh) [リセット= 00h]

図 7-30 に、INT\_CFG を示し、表 7-69 に、その説明を示します。

概略表に戻ります。

GPIO 割り込み出力

**図 7-30. INT\_CFG レジスタ**

7	6	5	4	3	2	1	0
RSVD	INT1_SEL		INT1_EN	RSVD	INT0_SEL		INT0_EN
R-0h	R/W-0h		R/W-0h	R-0h	R/W-0h		R/W-0h

**表 7-69. INT\_CFG レジスタ フィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	予約済み
6-5	INT1_SEL	R/W	0h	割り込み出力として使用する GPIO ピンを選択 0h = GPIO1 をアクティブ Low の INT1 として使用 1h = GPIO8 をアクティブ Low の INT1 として使用 2h = GPIO1 をアクティブ High の INT1 として使用 3h = GPIO8 をアクティブ High の INT1 として使用
4	INT1_EN	R/W	0h	選択した GPIO を割り込み出力としてイネーブルにするために使用します。 0h = 通常機能 (構成されたとおり) 1h = 割り込み機能 (IO_CFG を GPIO 機能に設定する必要があります)
3	RSVD	R	0h	予約済み
2-1	INT0_SEL	R/W	0h	割り込み出力として使用する GPIO ピンを選択 0h = GPIO0 をアクティブ Low の INT0 として使用 1h = GPIO7 をアクティブ Low の INT0 として使用 2h = GPIO0 をアクティブ High の INT0 として使用 3h = GPIO7 をアクティブ High の INT0 として使用
0	INT0_EN	R/W	0h	選択した GPIO を割り込み出力としてイネーブルにするために使用します。 0h = 通常機能 (構成されたとおり) 1h = 割り込み機能 (IO_CFG を GPIO 機能に設定する必要があります)

### 7.6.1.15 DEV\_IE\_0 レジスタ (オフセット = 1Eh) [リセット = 00h]

図 7-31 に、DEV\_IE\_0 を示し、表 7-70 に、その説明を示します。

概略表に戻ります。

INT0 ピンの特定の割り込みをイネーブルするための割り込みイネーブル ビット。有効化された割り込みは、INT0 ピンで通知されます。このピンの出力を確認するには、INT\_CFG レジスタで INT0 機能をイネーブルにする必要があることに注意してください。

図 7-31. DEV\_IE\_0 レジスタ

7	6	5	4	3	2	1	0
RSVD						CANWK	RSVD
R-0h						R/W-0h	R-0h

表 7-70. DEV\_IE\_0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-2	RSVD	R	0h	予約済み
1	CANWK	R/W	0h	CAN ウェークアップ CAN バスにより、デバイスはスリープ モードから復帰しました 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RSVD	R	0h	予約済み

### 7.6.1.16 DEV\_IE\_1 レジスタ (オフセット = 1Fh) [リセット = 00h]

図 7-32 に、DEV\_IE\_1 を示し、表 7-71 に、その説明を示します。

概略表に戻ります。

INT0 ピンの特定の割り込みをイネーブルするための割り込みイネーブル ビット。有効化された割り込みは、INT1 ピンで通知されます。このピンの出力を確認するには、INT\_CFG レジスタで INT0 機能をイネーブルにする必要があることに注意してください。

**図 7-32. DEV\_IE\_1 レジスタ**

7	6	5	4	3	2	1	0
RSVD						CANWK	RSVD
R-0h						R/W-0h	R-0h

**表 7-71. DEV\_IE\_1 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-2	RSVD	R	0h	予約済み
1	CANWK	R/W	0h	CAN ウェークアップ CAN バスにより、デバイスはスリープ モードから復帰しました 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RSVD	R	0h	予約済み



### 7.6.1.17 MRAM\_IP\_CFG レジスタ (オフセット = 20h) [リセット = 00h]

図 7-33 に、MRAM\_IP\_CFG を示し、表 7-72 に、その説明を示します。

概略表に戻ります。

MRAM および IP 設定。このレジスタは、各種シリアル インターフェイスをイネーブルにし、MRAM の一部をそれらに割り当てるために使用されます。

図 7-33. MRAM\_IP\_CFG レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO_OUT_SYNC	MRAM_IP_EN			
R-0h			R/W-0h	R/W-0h			

表 7-72. MRAM\_IP\_CFG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO_OUT_SYNC	R/W	0h	GPIO 出力値の更新を同期するため、2 番目の GPIO レジスタが書き込まれるまで更新を保留します。 これには、0x002D および 0x002E に書き込むすべての出力の変更が必要です。0x002E の書き込みが完了すると、すべての GPIO が更新されます。 0h = バイトが書き込まれると GPIO は更新されます 1h = 0x2E が書き込まれると GPIO は更新されます
3-0	MRAM_IP_EN	R/W	0h	MRAM および IP イネーブル。 シリアル インターフェイスにメモリ (>0%) が割り当てられている場合、これらのビットによって選択された IP が有効になります。 有効な IP がアクティブな間はこの値を変更しないでください。未定義の動作が発生する可能性があります (今後のバージョンで変更されます)  <div style="text-align: center;"><b>注</b></div> <div style="text-align: center;">目的の IP の GPIO を特殊機能として設定する必要があります。設定しないと、IP をイネーブルにできません</div> 0h = すべてディスエーブル / 0% 1h = SPI 0%、UART 100%、I2C 0% 2h = SPI 25%、UART 75%、I2C 0% 3h = SPI 50%、UART 50%、I2C 0% 4h = SPI 75%、UART 25%、I2C 0% 5h = SPI 100%、UART 0%、I2C 0% 6h = SPI 0%、UART 0%、I2C 100% 7h = SPI 0%、UART 25%、I2C 75% 8h = SPI 0%、UART 50%、I2C 50% 9h = SPI 0%、UART 75%、I2C 25% Ah = SPI 25%、UART 0%、I2C 75% Bh = SPI 50%、UART 0%、I2C 50% Ch = SPI 75%、UART 0%、I2C 25%

### 7.6.1.18 IO\_CFG\_0 レジスタ (オフセット = 21h) [リセット = 00h]

図 7-34 に、IO\_CFG\_0 を示し、表 7-73 に、その説明を示します。

概略表に戻ります。

GPIO ピン モード構成

**図 7-34. IO\_CFG\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_CFG	GPIO6_CFG	GPIO5_CFG	GPIO4_CFG	GPIO3_CFG	GPIO2_CFG	GPIO1_CFG	GPIO0_CFG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-73. IO\_CFG\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (UART TXD)
6	GPIO6_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI PICO)
5	GPIO5_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI POCI)
4	GPIO4_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI SCK)
3	GPIO3_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI CS0)
2	GPIO2_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI CS1)
1	GPIO1_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI CS2)
0	GPIO0_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (SPI CS3)

### 7.6.1.19 IO\_CFG\_1 レジスタ (オフセット = 22h) [リセット = 00h]

図 7-35 に、IO\_CFG\_1 を示し、表 7-74 に、その説明を示します。

概略表に戻ります。

GPIO ピン モード構成

図 7-35. IO\_CFG\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_CFG	GPIO11_CFG	GPIO10_CFG	GPIO9_CFG	GPIO8_CFG
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-74. IO\_CFG\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (PWM1 がイネーブルの場合は SPI CS7 または PWM1)
3	GPIO11_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (PWM0 がイネーブルの場合は SPI CS6 または PWM0)
2	GPIO10_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (I2C が有効な場合は SPI CS5 または I2C SDA)
1	GPIO9_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (I2C が有効な場合は SPI CS4 または I2C SCL)
0	GPIO8_CFG	R/W	0h	0h = GPIO 機能 1h = 特殊機能 (UART RXD)

### 7.6.1.20 IO\_OE\_0 レジスタ (オフセット = 23h) [リセット = 00h]

図 7-36 に、IO\_OE\_0 を示し、表 7-75 に、その説明を示します。

概略表に戻ります。

GPIO 出力または入力の構成

**図 7-36. IO\_OE\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_OE	GPIO6_OE	GPIO5_OE	GPIO4_OE	GPIO3_OE	GPIO2_OE	GPIO1_OE	GPIO0_OE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-75. IO\_OE\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_OE	R/W	0h	0h = 入力 1h = 出力
6	GPIO6_OE	R/W	0h	0h = 入力 1h = 出力
5	GPIO5_OE	R/W	0h	0h = 入力 1h = 出力
4	GPIO4_OE	R/W	0h	0h = 入力 1h = 出力
3	GPIO3_OE	R/W	0h	0h = 入力 1h = 出力
2	GPIO2_OE	R/W	0h	0h = 入力 1h = 出力
1	GPIO1_OE	R/W	0h	0h = 入力 1h = 出力
0	GPIO0_OE	R/W	0h	0h = 入力 1h = 出力

### 7.6.1.21 IO\_OE\_1 レジスタ (オフセット = 24h) [リセット = 00h]

図 7-37 に、IO\_OE\_1 を示し、表 7-76 に、その説明を示します。

概略表に戻ります。

GPIO 出力または入力の構成

図 7-37. IO\_OE\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_OE	GPIO11_OE	GPIO10_OE	GPIO9_OE	GPIO8_OE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-76. IO\_OE\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_OE	R/W	0h	0h = 入力 1h = 出力
3	GPIO11_OE	R/W	0h	0h = 入力 1h = 出力
2	GPIO10_OE	R/W	0h	0h = 入力 1h = 出力
1	GPIO9_OE	R/W	0h	0h = 入力 1h = 出力
0	GPIO8_OE	R/W	0h	0h = 入力 1h = 出力

### 7.6.1.22 IO\_OD\_0 レジスタ (オフセット = 25h) [リセット = 00h]

図 7-38 に、IO\_OD\_0 を示し、表 7-77 に、その説明を示します。

概略表に戻ります。

GPIO オープン ドレインの構成

**図 7-38. IO\_OD\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_OD	GPIO6_OD	GPIO5_OD	GPIO4_OD	GPIO3_OD	GPIO2_OD	GPIO1_OD	GPIO0_OD
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-77. IO\_OD\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
6	GPIO6_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
5	GPIO5_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
4	GPIO4_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
3	GPIO3_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
2	GPIO2_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
1	GPIO1_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
0	GPIO0_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード

### 7.6.1.23 IO\_OD\_1 レジスタ (オフセット = 26h) [リセット = 00h]

図 7-39 に、IO\_OD\_1 を示し、表 7-78 に、その説明を示します。

概略表に戻ります。

GPIO オープン ドレインの構成

図 7-39. IO\_OD\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_OD	GPIO11_OD	GPIO10_OD	GPIO9_OD	GPIO8_OD
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-78. IO\_OD\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
3	GPIO11_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
2	GPIO10_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
1	GPIO9_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード
0	GPIO8_OD	R/W	0h	0h = プッシュプル モード 1h = オープン ドレイン モード



### 7.6.1.24 IO\_RE\_0 レジスタ (オフセット = 27h) [リセット = 00h]

図 7-40 に、IO\_RE\_0 を示し、表 7-79 に、その説明を示します。

概略表に戻ります。

GPIO 抵抗のイネーブル

**図 7-40. IO\_RE\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_RE	GPIO6_RE	GPIO5_RE	GPIO4_RE	GPIO3_RE	GPIO2_RE	GPIO1_RE	GPIO0_RE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-79. IO\_RE\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
6	GPIO6_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
5	GPIO5_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
4	GPIO4_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
3	GPIO3_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
2	GPIO2_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
1	GPIO1_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
0	GPIO0_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています

### 7.6.1.25 IO\_RE\_1 レジスタ (オフセット = 28h) [リセット = 00h]

図 7-41 に、IO\_RE\_1 を示し、表 7-80 に、その説明を示します。

概略表に戻ります。

GPIO 抵抗のイネーブル

図 7-41. IO\_RE\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_RE	GPIO11_RE	GPIO10_RE	GPIO9_RE	GPIO8_RE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-80. IO\_RE\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
3	GPIO11_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
2	GPIO10_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
1	GPIO9_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています
0	GPIO8_RE	R/W	0h	0h = ピンにバイアス抵抗を接続していません 1h = IO_PU に従ってピンにバイアス抵抗を接続しています

### 7.6.1.26 IO\_PU\_0 レジスタ (オフセット = 29h) [リセット = 00h]

図 7-42 に、IO\_PU\_0 を示し、表 7-81 に、その説明を示します。

概略表に戻ります。

GPIO プルアップ抵抗のイネーブル

**図 7-42. IO\_PU\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_PU	GPIO6_PU	GPIO5_PU	GPIO4_PU	GPIO3_PU	GPIO2_PU	GPIO1_PU	GPIO0_PU
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-81. IO\_PU\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
6	GPIO6_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
5	GPIO5_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
4	GPIO4_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
3	GPIO3_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
2	GPIO2_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
1	GPIO1_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
0	GPIO0_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます

### 7.6.1.27 IO\_PU\_1 レジスタ (オフセット = 2Ah) [リセット = 00h]

図 7-43 に、IO\_PU\_1 を示し、表 7-82 に、その説明を示します。

概略表に戻ります。

GPIO プルアップ抵抗のイネーブル

図 7-43. IO\_PU\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_PU	GPIO11_PU	GPIO10_PU	GPIO9_PU	GPIO8_PU
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-82. IO\_PU\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
3	GPIO11_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
2	GPIO10_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
1	GPIO9_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます
0	GPIO8_PU	R/W	0h	0h = IO_RE が設定されている場合、このピンにはプルダウン抵抗が接続されます 1h = IO_RE が設定されている場合、このピンにはプルアップ抵抗が接続されます

### 7.6.1.28 IO\_POL\_0 レジスタ (オフセット = 2Bh) [リセット = 00h]

図 7-44 に、IO\_POL\_0 を示し、表 7-83 に、その説明を示します。

概略表に戻ります。

GPIO 極性反転イネーブル。これらのビットは、ピンが入力として設定されている場合にのみ適用されます。IO\_INPUT レジスタに表示される信号入力を反転します

**図 7-44. IO\_POL\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_PIE	GPIO6_PIE	GPIO5_PIE	GPIO4_PIE	GPIO3_PIE	GPIO2_PIE	GPIO1_PIE	GPIO0_PIE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-83. IO\_POL\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
6	GPIO6_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
5	GPIO5_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
4	GPIO4_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
3	GPIO3_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
2	GPIO2_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
1	GPIO1_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
0	GPIO0_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています

### 7.6.1.29 IO\_POL\_1 レジスタ (オフセット = 2Ch) [リセット = 00h]

図 7-45 に、IO\_POL\_1 を示し、表 7-84 に、その説明を示します。

概略表に戻ります。

GPIO 極性反転イネーブル。これらのビットは、ピンが入力として設定されている場合にのみ適用されます。IO\_INPUT レジスタに表示される信号入力を反転します

図 7-45. IO\_POL\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_PIE	GPIO11_PIE	GPIO10_PIE	GPIO9_PIE	GPIO8_PIE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-84. IO\_POL\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
3	GPIO11_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
2	GPIO10_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
1	GPIO9_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています
0	GPIO8_PIE	R/W	0h	0h = 入力極性の反転は無効になっています 1h = 入力極性の反転は有効になっています

### 7.6.1.30 IO\_OUTPUT\_0 レジスタ (オフセット = 2Dh) [リセット = 00h]

図 7-46 に、IO\_OUTPUT\_0 を示し、表 7-85 に、その説明を示します。

概略表に戻ります。

**図 7-46. IO\_OUTPUT\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_OUTPUT T	GPIO6_OUTPUT T	GPIO5_OUTPUT T	GPIO4_OUTPUT T	GPIO3_OUTPUT T	GPIO2_OUTPUT T	GPIO1_OUTPUT T	GPIO0_OUTPUT T
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-85. IO\_OUTPUT\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
6	GPIO6_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
5	GPIO5_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
4	GPIO4_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
3	GPIO3_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
2	GPIO2_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
1	GPIO1_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
0	GPIO0_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)



### 7.6.1.31 IO\_OUTPUT\_1 レジスタ (オフセット = 2Eh) [リセット = 00h]

図 7-47 に、IO\_OUTPUT\_1 を示し、表 7-86 に、その説明を示します。

概略表に戻ります。

図 7-47. IO\_OUTPUT\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_OUTP UT	GPIO11_OUTP UT	GPIO10_OUTP UT	GPIO9_OUTPU T	GPIO8_OUTPU T
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-86. IO\_OUTPUT\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
3	GPIO11_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
2	GPIO10_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
1	GPIO9_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)
0	GPIO8_OUTPUT	R/W	0h	0h = ピンが出力として構成されている場合、出力は Low に駆動されます。 1h = このピンが出力として構成されている場合、出力は High またはハイ インピーダンスに駆動されます (IO_OD に依存)

### 7.6.1.32 IO\_INPUT\_0 レジスタ (オフセット = 2Fh) [リセット = 00h]

図 7-48 に、IO\_INPUT\_0 を示し、表 7-87 に、その説明を示します。

概略表に戻ります。

**図 7-48. IO\_INPUT\_0 レジスタ**

7	6	5	4	3	2	1	0
GPIO7_INPUT	GPIO6_INPUT	GPIO5_INPUT	GPIO4_INPUT	GPIO3_INPUT	GPIO2_INPUT	GPIO1_INPUT	GPIO0_INPUT
RH-0h	RH-0h	RH-0h	RH-0h	RH-0h	RH-0h	RH-0h	RH-0h

**表 7-87. IO\_INPUT\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	GPIO7_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
6	GPIO6_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
5	GPIO5_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
4	GPIO4_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
3	GPIO3_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
2	GPIO2_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
1	GPIO1_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
0	GPIO0_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High

### 7.6.1.33 IO\_INPUT\_1 レジスタ (オフセット = 30h) [リセット = 00h]

図 7-49 に、IO\_INPUT\_1 を示し、表 7-88 に、その説明を示します。

概略表に戻ります。

図 7-49. IO\_INPUT\_1 レジスタ

7	6	5	4	3	2	1	0
RSVD			GPIO12_INPUT	GPIO11_INPUT	GPIO10_INPUT	GPIO9_INPUT	GPIO8_INPUT
R-0h			RH-0h	RH-0h	RH-0h	RH-0h	RH-0h

表 7-88. IO\_INPUT\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	GPIO12_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
3	GPIO11_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
2	GPIO10_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
1	GPIO9_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High
0	GPIO8_INPUT	RH	0h	0h = ピンの現在の状態は Low 1h = ピンの現在の状態は High

### 7.6.1.34 IR\_STATUS レジスタ (オフセット = 31h) [リセット = 00h]

表 7-89 に、IR\_STATUS を示します。

概略表に戻ります。

割り込みステータス レジスタは、どの IP ブロックが保留中の割り込みを持ち、どちらの割り込みラインで有効化され、かつセットされているかを示します。割り込みが有効化されていない場合、割り込みは表示されません

**表 7-89. IR\_STATUS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	予約済み
5	PWM1_IR	R	0h	PWM1 割り込みレジスタに割り込みがセットされます 0h = PWM1 割り込みレジスタに割り込みはセットされていません 1h = PWM1 割り込みレジスタに割り込みがセットされています
4	PWM0_IR	R	0h	PWM0 割り込みレジスタに割り込みがセットされます 0h = PWM0 割り込みレジスタに割り込みはセットされていません 1h = PWM0 割り込みレジスタに割り込みがセットされています
3	I2C_IR	R	0h	I2C 割り込みレジスタに割り込みがセットされます 0h = I2C 割り込みレジスタに割り込みはセットされていません 1h = I2C 割り込みレジスタに割り込みがセットされています
2	UART_IR	R	0h	UART 割り込みレジスタに割り込みがセットされます 0h = UART 割り込みレジスタに割り込みはセットされていません 1h = UART 割り込みレジスタに割り込みがセットされています
1	SPI_IR	R	0h	SPI 割り込みレジスタに割り込みがセットされます 0h = SPI 割り込みレジスタに割り込みはセットされていません 1h = SPI 割り込みレジスタに割り込みがセットされています
0	DEVICE_IR	R	0h	デバイス割り込みレジスタに割り込みがセットされます 0h = デバイス割り込みレジスタに割り込みはセットされていません 1h = デバイス割り込みレジスタに割り込みがセットされています

**7.6.1.35 DEV\_IR レジスタ (オフセット = 32h) [リセット= 00h]**

図 7-50 に、DEV\_IR を示し、表 7-90 に、その説明を示します。

概略表に戻ります。

デバイス割り込みステータス レジスタ

**図 7-50. DEV\_IR レジスタ**

7	6	5	4	3	2	1	0
RSVD						CANWK	RSVD
R-0h						R/W1C-0h	R-0h

**表 7-90. DEV\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-2	RSVD	R	0h	予約済み
1	CANWK	R/W1C	0h	CAN ウェークアップ CAN バスによって、デバイスがスリープ モードから復帰しました 0h = 割り込みは発生していません 1h = 割り込みは発生していません
0	RSVD	R	0h	予約済み

### 7.6.1.36 SPI\_IR レジスタ (オフセット = 33h) [リセット = 00h]

図 7-51 に、SPI\_IR を示し、表 7-91 に、その説明を示します。

概略表に戻ります。

これは、割り込みの読み取りとクリアを簡単にするための SPI\_IR レジスタの書き込み可能なミラーです

**図 7-51. SPI\_IR レジスタ**

7	6	5	4	3	2	1	0
TXL	TXA	RSVD			RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-91. SPI\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXL	R/W1C	0h	TX メッセージ喪失 (FIFO フルによる割り込み) 0h = TX FIFO メッセージは失われていません 1h = 書き込みが試みられた際に FIFO がフルであったため、TX FIFO がメッセージを喪失しました (オーバーラン)
6	TXA	R/W1C	0h	TX 領域使用可能割り込み  <div style="text-align: center;"><b>注</b></div> <div style="text-align: center;">SPI IP がイネーブルの場合、FIFO は空になるため、この割り込みがセットされます</div> 0h = TX FIFO 領域は利用可能な割り込みがありません 1h = TX FIFO がトリガ レベル スレッシュホールドに達しました
5-3	RSVD	R	0h	
2	RXL	R/W1C	0h	RX オーバーラン / 喪失メッセージ割り込み 0h = RX FIFO メッセージ喪失 (オーバーランなし) 割り込みはありませんでした 1h = 全 RX FIFO (オーバーラン割り込み) のために少なくとも 1 個のメッセージが失われました
1	RXFL	R/W1C	0h	RX フィル レベル割り込み。 レベルは SPI_CTRL.RX_LVL_INT によって設定されます 0h = RX FIFO フィル レベル割り込みがありません 1h = RX FIFO がフィル レベルに到達しました
0	RXN	R/W1C	0h	RX 新しいメッセージ割り込み 0h = 新しいメッセージが受信されませんでした 1h = 新しいメッセージが受信されました

## 7.6.1.37 UART\_IR レジスタ (オフセット = 34h) [リセット= 00h]

図 7-52 に、UART\_IR を示し、表 7-92 に、その説明を示します。

概略表に戻ります。

これは UART\_IR レジスタの書き込み可能なミラーであり、割り込みを容易に読み取りとクリアができるようにします

図 7-52. UART\_IR レジスタ

7	6	5	4	3	2	1	0
TXL	TXA	RXBR	RXFE	RXPE	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 7-92. UART\_IR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXL	R/W1C	0h	フルのために TX に失われたバイト 0h = TX FIFO バイトは失われませんでした 1h = 書き込みの際に FIFO が満杯になったために TX FIFO がバイト (オーバーラン) を失いました
6	TXA	R/W1C	0h	TX スペースが利用可能  <b>注</b> UART IP がイネーブルの場合、FIFO は空になるため、この割り込みがセットされます  0h = TX FIFO 領域は利用可能な割り込みがありません 1h = TX FIFO がトリガ レベル スレッショルドに達しました
5	RXBR	R/W1C	0h	RX ブレークを受信しました 0h = ブレークを受信しませんでした 1h = ブレークを受信しました
4	RXFE	R/W1C	0h	RX フレーミング エラー 0h = 受信フレーミング エラーは発生していません 1h = RX フレーミング エラーが発生しました
3	RXPE	R/W1C	0h	RX パリティ エラー 0h = 受信パリティ エラーは発生していません 1h = RX パリティ エラーが発生しました
2	RXL	R/W1C	0h	RX オーバーラン / 喪失バイト 0h = RX FIFO バイト喪失 (オーバーランなし) 割り込みはありません 1h = 全 RX FIFO (オーバーラン割り込み) により少なくとも 1 バイトが喪失されました
1	RXFL	R/W1C	0h	RX フィル レベル 0h = RX FIFO フィル レベルの割り込みはありません 1h = RX FIFO がフィル レベルに達しました
0	RXN	R/W1C	0h	RX の新規バイト 0h = 新しいバイトは割り込みを受信していません 1h = 新しいバイトが受信されました



### 7.6.1.38 I2C\_IR レジスタ (オフセット = 35h) [リセット = 00h]

図 7-53 に、I2C\_IR を示し、表 7-93 に、その説明を示します。

概略表に戻ります。

これは I2C\_IR レジスタの書き込み可能なミラーであり、割り込みの読み取りとクリアを容易にするためのものです

**図 7-53. I2C\_IR レジスタ**

7	6	5	4	3	2	1	0
TXL	TXA	DNACK	ANACK	SBRC	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-93. I2C\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXL	R/W1C	0h	フル FIFO 割り込みにより TX 消失メッセージ 0h = TX FIFO バイトは失われていません 1h = 書き込みが試みられた際に FIFO がフルであったため、TX FIFO がバイトを喪失しました (オーバーラン)
6	TXA	R/W1C	0h	TX 領域使用可能割り込み。 トリガ レベルは、I2C_FIFO_CTRL.TX_TRG 設定によって設定されます  <div style="text-align: center;"><b>注</b></div> I2C IP が有効な場合、FIFO は空になるため、この割り込みがセットされます  0h = TX FIFO 領域は利用可能な割り込みがありません 1h = TX FIFO がトリガ レベル スレッシュホールドに達しました
5	DNACK	R/W1C	0h	データ NACK 割り込み 0h = NACK が検出されませんでした 1h = データ相中に NACK が検出されました
4	ANACK	R/W1C	0h	アドレス NACK 割り込み 0h = NACK は検出されませんでした 1h = アドレス フェーズ中に NACK が検出されました
3	SBRC	R/W1C	0h	スタックバス回復完了割り込み 0h = スタック バスの回復は発生していません 1h = スタック バスの回復は完了しました。I2C_STATUS レジスタを確認して、バスがスタックしているか、停止しているかを確認します。
2	RXL	R/W1C	0h	RX オーバーラン / 喪失メッセージ割り込み 0h = RX FIFO バイト消失 (オーバーランなし) 割り込みはありませんでした 1h = 全 RX FIFO (オーバーラン割り込み) のために少なくとも 1 バイトが失われました
1	RXFL	R/W1C	0h	RX フィル レベル割り込み 0h = RX FIFO フィル レベルの割り込みはありません 1h = RX FIFO がフィル レベルに達しました
0	RXN	R/W1C	0h	RX 新しいメッセージ割り込み 0h = 新しいバイトは受信されませんでした 1h = 新しいバイトが受信されました

### 7.6.1.39 PWM0\_IR レジスタ (オフセット = 36h) [リセット = 00h]

図 7-54 に、PWM0\_IR を示し、表 7-94 に、その説明を示します。

概略表に戻ります。

これは PWM0\_IR レジスタの書き込み可能なミラーであり、割り込みの読み取りとクリアを容易にするためのものです

図 7-54. PWM0\_IR レジスタ

7	6	5	4	3	2	1	0
RSVD				PULSE_OVF	IAS	RC	
R-0h				R/W1C-0h	R/W1C-0h	R/W1C-0h	

表 7-94. PWM0\_IR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF	R/W1C	0h	電流パルスオーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定される 0h=電流パルスカウンタがオーバーフローしていない 1h=電流パルスカウンタがオーバーフローした
1	IAS	R/W1C	0h	入力自動停止割り込み 入力自動停止条件が発生した際に設定されます。PWM が即時停止に設定されている場合、出力も停止されます(RC ビットに反映)。PWM が IAS トリガで停止ランプを開始するように構成されている場合、入力条件が発生すると、この割り込みが直ちに設定され、出力がオフになった時点で RC ビットが設定されます。 0h = IAS の入力はトリガされていません 1h = IAS 入力トリガされました
0	RC	R/W1C	0h	PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。 AS_EN = 0 の場合、PWM が終了値に達すると設定されます。 IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止された時点でこの値が設定されます 0h = PWM チャネルランプは完了していません 1h = PWM チャネルランプは完了しています

### 7.6.1.40 PWM1\_IR レジスタ (オフセット = 37h) [リセット = 00h]

図 7-55 に、PWM1\_IR を示し、表 7-95 に、その説明を示します。

概略表に戻ります。

これは PWM1\_IR レジスタの書き込み可能なミラーであり、割り込みの読み取りとクリアを容易にするためのものです

**図 7-55. PWM1\_IR レジスタ**

7	6	5	4	3	2	1	0
		RSVD			PULSE_OVF	IAS	RC
		R-0h			R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-95. PWM1\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF	R/W1C	0h	電流パルスオーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定される 0h=電流パルスカウンタがオーバーフローしていない 1h=電流パルスカウンタがオーバーフローした
1	IAS	R/W1C	0h	入力自動停止割り込み 入力自動停止条件が発生した際に設定されます。PWM が即時停止に設定されている場合、出力も停止されます(RC ビットに反映)。PWM が IAS トリガで停止ランプを開始するように構成されている場合、入力条件が発生すると、この割り込みが直ちに設定され、出力がオフになった時点で RC ビットが設定されます。 0h = IAS の入力はトリガされていません 1h = IAS 入力トリガされました
0	RC	R/W1C	0h	PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。 AS_EN = 0 の場合、PWM が終了値に達すると設定されます。 IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止された時点でこの値が設定されます 0h = PWM チャネルランプは完了していません 1h = PWM チャネルランプは完了しています

### 7.6.2 SPI レジスタ

セクション 7.6.2 に、SPI レジスタに対してメモリマップされたレジスタを一覧表示します。セクション 7.6.2 にないレジスタ オフセット アドレスはすべて予約済みと見なして、レジスタの内容は変更しないでください。

**表 7-96. SPI レジスタ**

オフセット	略称	レジスタ名	セクション
1000h	SPI_CREL	SPI コア リリース	セクション 7.6.2.1
1001h	SPI_SCRATCH	SPI スクラッチパッド	セクション 7.6.2.2
1002h	SPI_CTRL	SPI 制御	セクション 7.6.2.3
1003h	SPI_CFG_0	SPI チャネル構成 0	セクション 7.6.2.4
1004h	SPI_CFG_1	SPI チャネル構成 1	セクション 7.6.2.5
1005h	SPI_DR_0	SPI データ レート 0	セクション 7.6.2.6
1006h	SPI_DR_1	SPI データ レート 1	セクション 7.6.2.7
1007h	SPI_DR_2	SPI データ レート 2	セクション 7.6.2.8
1008h	SPI_DR_3	SPI データ レート 3	セクション 7.6.2.9
1009h	SPI_CHAN_EN	SPI チャネル イネーブル	セクション 7.6.2.10
100Ah	SPI_CS_POL	SPI チップ セレクト極性	セクション 7.6.2.11
100Bh	SPI_FIFO_CTRL	SPI FIFO 制御	セクション 7.6.2.12

表 7-96. SPI レジスタ (続き)

オフセット	略称	レジスタ名	セクション
100Ch	SPI_IE_0	SPI 割り込みイネーブル 0	<a href="#">セクション 7.6.2.13</a>
100Dh	SPI_IE_1	SPI 割り込みイネーブル 1	<a href="#">セクション 7.6.2.14</a>
100Eh	SPI_IR	SPI 割り込みレジスタ	<a href="#">セクション 7.6.2.15</a>
100Fh	SPI_FS	SPI FIFO ステータス	<a href="#">セクション 7.6.2.16</a>
1010h	SPI_RX_FIFO	SPI 受信 FIFO	<a href="#">セクション 7.6.2.18</a>
1010h	SPI_TX_FIFO	SPI 送信 FIFO	<a href="#">セクション 7.6.2.17</a>
1011h	SPI_RXFS	SPI RX FIFO ステータス	<a href="#">セクション 7.6.2.19</a>
1012h	SPI_TXFS	SPI TX FIFO ステータス	<a href="#">セクション 7.6.2.20</a>
1013h	SPI_TXES	SPI TX 素子ステータス	<a href="#">セクション 7.6.2.21</a>

表の小さなセルに収まるように、複雑なビット アクセス タイプを記号で表記しています。[セクション 7.6.2](#) に、このセクションでアクセス タイプに使用しているコードを示します。

表 7-97. SPI のアクセス タイプ コード

アクセス タイプ	コード	説明
読み取りタイプ		
R	R	読み出し
RH	R H	ハードウェアによってセットまたはクリアされる の読み取り
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	1 を書き込むことで セット
WP	W P	書き込み 特権アクセスが必要
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト 値

### 7.6.2.1 SPI\_CREL レジスタ (オフセット = 1000h) [リセット= 87h]

図 7-56 に、SPI\_CREL を示し、表 7-98 に、その説明を示します。

概略表に戻ります。

SPI IP リリース バージョン

図 7-56. SPI\_CREL レジスタ

7	6	5	4	3	2	1	0
CREL							
R-87h							

表 7-98. SPI\_CREL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	CREL	R	87h	SPI コア リリース バージョン

### 7.6.2.2 SPI\_SCRATCH レジスタ (オフセット = 1001h) [リセット= 00h]

図 7-57 に、SPI\_SCRATCH を示し、表 7-99 に、その説明を示します。

概略表に戻ります。

顧客スクラッチパッド レジスタ

図 7-57. SPI\_SCRATCH レジスタ

7	6	5	4	3	2	1	0
SCRATCH							
R/W-0h							

表 7-99. SPI\_SCRATCH レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	SCRATCH	R/W	0h	読み書きテスト用の、ユーザー書き込み可能なレジスタです。 このレジスタは、読み取り / 書き込みテスト以外の機能はありません。

### 7.6.2.3 SPI\_CTRL レジスタ (オフセット = 1002h) [リセット= 00h]

図 7-58 に、SPI\_CTRL を示し、表 7-100 に、その説明を示します。

概略表に戻ります。

このレジスタは SPI IP を制御します

**図 7-58. SPI\_CTRL レジスタ**

7	6	5	4	3	2	1	0
TX_TRG		RX_TRG		予約済み		SPI_EN	SPI_CCE
R/W-0h		R/W-0h		R-0h		RH-0h	R/W-0h

**表 7-100. SPI\_CTRL レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-6	TX_TRG	R/W	0h	<p>TX FIFO 内の空きバイト数が指定した値まで増加したときにトリガーするレベルを設定します。</p> <p>FIFO に一定数の空き領域ができたときに割り込みを発生させ、プロセッサが 1 バイトずつではなく複数バイトをまとめて FIFO にロードできるようにするために使用されます。</p> <hr/> <p style="text-align: center;"><b>注</b></p> <p>これにより、エッジベースの割り込みと同様にトリガされます。</p> <p>ユーザーが 8 つの空き領域を決して下回ることがない場合</p> <hr/> <p>0h = 16 バイトの空き  1h = 32 バイトの空き  2h = [(最大サイズ) - 128] バイトの空き  3h = [(最大サイズ) - 64] バイトの空き</p>
5-4	RX_TRG	R/W	0h	<p>RX FIFO が FIFO に格納されている指定されたバイト数に達したときの RX FIFO のトリガ レベルを設定します。</p> <p>これにより、CPU は一度に 1 バイトではなくバイトを一括でシフトアウトでき、割り込みの数を保存できます</p> <p>0h = RX FIFO で 4 バイト  1h = RX FIFO で 8 バイト  2h = [(最大サイズ) - 128] RX FIFO のバイト  3h = [(最大サイズ) - 64] RX FIFO のバイト</p>
3-2	予約済み	R	0h	予約済み
1	SPI_EN	RH	0h	<p>SPI IP イネーブル ステータス フラグ。</p> <p>このフラグは書き込み可能ではありませんが、MRAM から IP へのメモリ割り当てによって SPI IP がイネーブルになっている場合に設定されます。</p> <p>これは、<a href="#">MRAM_IP_CFG</a> レジスタで行われます</p> <p>0h = SPI IP が無効の場合、すべての SPI 機能が無効になります  1h = SPI IP が有効です</p>
0	SPI_CCE	R/W	0h	<p>SPI IP 変更制御イネーブル ビット。</p> <p>SPI_EN が 0 の場合のみ設定できます</p> <p>0h = SPI 構成レジスタは書き込み保護されています  1h = SPI 構成レジスタを変更できます</p>

#### 7.6.2.4 SPI\_CFG\_0 レジスタ (オフセット = 1003h) [リセット = 00h]

図 7-59 に、SPI\_CFG\_0 を示し、表 7-101 に、その説明を示します。

概略表に戻ります。

SPI チャンネルの構成。SPI モード、その他のチャンネル固有の設定を設定します。

図 7-59. SPI\_CFG\_0 レジスタ

7	6	5	4	3	2	1	0
RSVD	LSB_1	CLKPL_1	CLKPH_1	RSVD	LSB_0	CLKPL_0	CLKPH_0
R-0h	R/WP-0h	R/WP-0h	R/WP-0h	R-0h	R/WP-0h	R/WP-0h	R/WP-0h

表 7-101. SPI\_CFG\_0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	予約済み
6	LSB_1	R/WP	0h	LSB の最初の選択。 受信および送信シフトレジスタの方向を制御します 0h = MSB ファースト 1h = LSB ファースト
5	CLKPL_1	R/WP	0h	クロック極性の選択 0h = SCK の非アクティブ状態は Low です 1h = SCK の非アクティブ状態は High です
4	CLKPH_1	R/WP	0h	クロック位相の選択 0h = データは最初の CLK エッジで変更され、次のエッジでキャプチャされます 1h = データは最初の CLK エッジでキャプチャされ、以下のタイミングで変更されます
3	RSVD	R	0h	予約済み
2	LSB_0	R/WP	0h	LSB の最初の選択。 受信および送信シフトレジスタの方向を制御します 0h = MSB ファースト 1h = LSB ファースト
1	CLKPL_0	R/WP	0h	クロック極性の選択 0h = SCK の非アクティブ状態は Low です 1h = SCK の非アクティブ状態は High です
0	CLKPH_0	R/WP	0h	クロック位相の選択 0h = データは最初の CLK エッジで変更され、次のエッジでキャプチャされます 1h = データは最初の CLK エッジでキャプチャされ、以下のタイミングで変更されます



### 7.6.2.5 SPI\_CFG\_1 レジスタ (オフセット = 1004h) [リセット = 00h]

図 7-60 に、SPI\_CFG\_1 を示し、表 7-102 に、その説明を示します。

概略表に戻ります。

SPI チャンネルの構成。SPI モード、その他のチャンネル固有の設定を設定します。

**図 7-60. SPI\_CFG\_1 レジスタ**

7	6	5	4	3	2	1	0
RSVD	LSB_3	CLKPL_3	CLKPH_3	RSVD	LSB_2	CLKPL_2	CLKPH_2
R-0h	R/WP-0h	R/WP-0h	R/WP-0h	R-0h	R/WP-0h	R/WP-0h	R/WP-0h

**表 7-102. SPI\_CFG\_1 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	予約済み
6	LSB_3	R/WP	0h	LSB の最初の選択。 受信および送信シフトレジスタの方向を制御します 0h = MSB ファースト 1h = LSB ファースト
5	CLKPL_3	R/WP	0h	クロック極性の選択 0h = SCK の非アクティブ状態は Low です 1h = SCK の非アクティブ状態は High です
4	CLKPH_3	R/WP	0h	クロック位相の選択 0h = データは最初の CLK エッジで変更され、次のエッジでキャプチャされます 1h = データは最初の CLK エッジでキャプチャされ、以下のタイミングで変更されます
3	RSVD	R	0h	予約済み
2	LSB_2	R/WP	0h	LSB の最初の選択。 受信および送信シフトレジスタの方向を制御します 0h = MSB ファースト 1h = LSB ファースト
1	CLKPL_2	R/WP	0h	クロック極性の選択 0h = SCK の非アクティブ状態は Low です 1h = SCK の非アクティブ状態は High です
0	CLKPH_2	R/WP	0h	クロック極性の選択 0h = SCK の非アクティブ状態は Low です 1h = SCK の非アクティブ状態は High です

### 7.6.2.6 SPI\_DR\_0 レジスタ (オフセット = 1005h) [リセット = 00h]

図 7-61 に、SPI\_DR\_0 を示し、表 7-103 に、その説明を示します。

概略表に戻ります。

SPI チャンネル 0 の SPI データ レート レジスタ。このレジスタの値は、入力された値より 1 つ多いとして解釈されます。この値は、SPI クロックの半周期として解釈されます。周波数は、式  $f = 20\text{MHz} / (x + 1)$  で求められます。ここで、 $x$  は SPI\_DR 内の値です。最も近いレジスタ値は  $x = (20\text{MHz} / f) - 1$  で求められます

図 7-61. SPI\_DR\_0 レジスタ

7	6	5	4	3	2	1	0
プリスケアラ							
R/WP-0h							

表 7-103. SPI\_DR\_0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	プリスケアラ	R/WP	0h	SPI クロック プリスケアラの下位バイト。 プリスケアラ値は、入力した値より 1 つ大きい値として解釈されます。 この値は、EEPROM にプログラムされた値のデフォルト値です。

### 7.6.2.7 SPI\_DR\_1 レジスタ (オフセット = 1006h) [リセット = 00h]

図 7-62 に、SPI\_DR\_1 を示し、表 7-104 に、その説明を示します。

概略表に戻ります。

SPI チャンネル 1 の SPI データ レート レジスタ。このレジスタの値は、入力された値より 1 つ多いとして解釈されます。この値は、SPI クロックの半周期として解釈されます。周波数は、式  $f = 20\text{MHz} / (x + 1)$  で求められます。ここで、x は SPI\_DR 内の値です。最も近いレジスタ値は  $x = (20\text{MHz} / f) - 1$  で求められます

図 7-62. SPI\_DR\_1 レジスタ

7	6	5	4	3	2	1	0
プリスケアラ							
R/WP-0h							

表 7-104. SPI\_DR\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	プリスケアラ	R/WP	0h	SPI クロック プリスケアラの下位バイト。 プリスケアラ値は、入力した値より 1 つ大きい値として解釈されます。 この値は、EEPROM にプログラムされた値のデフォルト値です。

### 7.6.2.8 SPI\_DR\_2 レジスタ (オフセット = 1007h) [リセット = 00h]

図 7-63 に、SPI\_DR\_2 を示し、表 7-105 に、その説明を示します。

概略表に戻ります。

SPI チャンネル 2 の SPI データ レート レジスタ。このレジスタの値は、入力された値より 1 つ多いとして解釈されます。この値は、SPI クロックの半周期として解釈されます。周波数は、式  $f = 20\text{MHz} / (x + 1)$  で求められます。ここで、 $x$  は SPI\_DR 内の値です。最も近いレジスタ値は  $x = (20\text{MHz} / f) - 1$  で求められます

図 7-63. SPI\_DR\_2 レジスタ

7	6	5	4	3	2	1	0
プリスケアラ							
R/WP-0h							

表 7-105. SPI\_DR\_2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	プリスケアラ	R/WP	0h	SPI クロック プリスケアラの下位バイト。 プリスケアラ値は、入力した値より 1 つ大きい値として解釈されます。 この値は、EEPROM にプログラムされた値のデフォルト値です。

### 7.6.2.9 SPI\_DR\_3 レジスタ (オフセット = 1008h) [リセット = 00h]

図 7-64 に、SPI\_DR\_3 を示し、表 7-106 に、その説明を示します。

概略表に戻ります。

SPI チャンネル 3 の SPI データ レート レジスタ。このレジスタの値は、入力された値より 1 つ多いとして解釈されます。この値は、SPI クロックの半周期として解釈されます。周波数は、式  $f = 20\text{MHz} / (x + 1)$  で求められます。ここで、 $x$  は SPI\_DR 内の値です。最も近いレジスタ値は  $x = (20\text{MHz} / f) - 1$  で求められます

図 7-64. SPI\_DR\_3 レジスタ

7	6	5	4	3	2	1	0
プリスケアラ							
R/WP-0h							

表 7-106. SPI\_DR\_3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	プリスケアラ	R/WP	0h	<p>SPI クロック プリスケアラの下位バイト。 プリスケアラ値は、入力した値より 1 つ大きい値として解釈されます。 この値は、EEPROM にプログラムされた値のデフォルト値です。</p> <p style="text-align: center;">注</p> <p>SPI 構成は、チャンネル 4 ~ 7 と共有されます</p>

## 7.6.2.10 SPI\_CHAN\_EN レジスタ (オフセット = 1009h) [リセット = 00h]

図 7-65 に、SPI\_CHAN\_EN を示し、表 7-107 に、その説明を示します。

概略表に戻ります。

SPI チャンネル イネーブル レジスタ。SPI チャンネルは、個別のチップ セレクト チャンネルに割り当てられています。

## 注

チップ セレクト チャンネルの GPIO は、IO\_CFG\_0 および IO\_CFG\_1 レジスタで特殊機能に設定する必要があります

図 7-65. SPI\_CHAN\_EN レジスタ

7	6	5	4	3	2	1	0
C7_EN	C6_EN	C5_EN	C4_EN	C3_EN	C2_EN	C1_EN	C0_EN
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h

表 7-107. SPI\_CHAN\_EN レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	C7_EN	R/WP	0h	CS チャンネル 7 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
6	C6_EN	R/WP	0h	CS チャンネル 6 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
5	C5_EN	R/WP	0h	CS チャンネル 5 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
4	C4_EN	R/WP	0h	CS チャンネル 4 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
3	C3_EN	R/WP	0h	CS チャンネル 3 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
2	C2_EN	R/WP	0h	CS チャンネル 2 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
1	C1_EN	R/WP	0h	CS チャンネル 1 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています
0	C0_EN	R/WP	0h	CS チャンネル 0 イネーブル 0h = SPI チャンネルは無効化されているか、使用されていません 1h = SPI チャンネルが有効化されています

### 7.6.2.11 SPI\_CS\_POL レジスタ (オフセット = 100Ah) [リセット= 00h]

図 7-66 に、SPI\_CS\_POL を示し、表 7-108 に、その説明を示します。

概略表に戻ります。

**図 7-66. SPI\_CS\_POL レジスタ**

7	6	5	4	3	2	1	0
RSVD				CS3_AH	CS2_AH	CS1_AH	CS0_AH
R-0h				R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h

**表 7-108. SPI\_CS\_POL レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-4	RSVD	R	0h	
3	CS3_AH	R/WP	0h	チャンネルの CS ピンのアクティブ状態を設定します <div style="text-align: center;"> <b>注</b>              CS3 チップ セレクト構成は、CS4-7 と共有されます           </div> 0h = チップ セレクト ピンの機能はアクティブ Low (nCS または CS) 1h = チップ セレクト ピンの機能はアクティブ High (CS)
2	CS2_AH	R/WP	0h	チャンネルの CS ピンのアクティブ状態を設定します 0h = チップ セレクト ピンの機能はアクティブ Low (nCS または CS) 1h = チップ セレクト ピンの機能はアクティブ High (CS)
1	CS1_AH	R/WP	0h	チャンネルの CS ピンのアクティブ状態を設定します 0h = チップ セレクト ピンの機能はアクティブ Low (nCS または CS) 1h = チップ セレクト ピンの機能はアクティブ High (CS)
0	CS0_AH	R/WP	0h	チャンネルの CS ピンのアクティブ状態を設定します 0h = チップ セレクト ピンの機能はアクティブ Low (nCS または CS) 1h = チップ セレクト ピンの機能はアクティブ High (CS)

## 7.6.2.12 SPI\_FIFO\_CTRL レジスタ (オフセット = 100Bh) [リセット= 0h]

表 7-109 に、SPI\_FIFO\_CTRL を示します。

概略表に戻ります。

表 7-109. SPI\_FIFO\_CTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
2	CLR_TX	RH/W1S	0h	送信 FIFO の内容をクリアします。 クリアが完了すると、ハードウェアはビットを 0 に戻します。 0h = 何もしない 1h = 送信 FIFO をクリア
1	CLR_RX	RH/W1S	0h	受信 FIFO の内容をクリアします。 クリアが完了すると、ハードウェアはビットを 0 に戻します。 0h = 何もしない 1h = 受信 FIFO をクリア
0	RSVD	R	0h	



### 7.6.2.13 SPI\_IE\_0 レジスタ (オフセット = 100Ch) [リセット = 00h]

図 7-67 に、SPI\_IE\_0 を示し、表 7-110 に、その説明を示します。

概略表に戻ります。

INT0 ピンの特定の割り込みをイネーブルするための割り込みイネーブル ビット。有効化された割り込みは、INT0 ピンで通知されます。このピンの出力を確認するには、INT\_CFG レジスタで INT0 機能をイネーブルにする必要があることに注意してください。

**図 7-67. SPI\_IE\_0 レジスタ**

7	6	5	4	3	2	1	0
TXLIE0	TXAIE0	RSVD			RXLIE0	RXFLIE0	RXNIE0
R/W-0h	R/W-0h	R-0h			R/W-0h	R/W-0h	R/W-0h

**表 7-110. SPI\_IE\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXLIE0	R/W	0h	INT0 のフル FIFO 割り込みイネーブルによる TX 喪失メッセージ 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
6	TXAIE0	R/W	0h	INT0 での TX 領域使用可能割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
5-3	RSVD	R	0h	
2	RXLIE0	R/W	0h	INT0 の RX オーバーラン / 喪失メッセージ割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
1	RXFLIE0	R/W	0h	INT0 の RX フィル レベル割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RXNIE0	R/W	0h	INT0 の RX 新しいメッセージ割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。

#### 7.6.2.14 SPI\_IE\_1 レジスタ (オフセット = 100Dh) [リセット = 00h]

図 7-68 に、SPI\_IE\_1 を示し、表 7-111 に、その説明を示します。

概略表に戻ります。

INT0 ピンの特定の割り込みをイネーブルするための割り込みイネーブル ビット。有効化された割り込みは、INT1 ピンで通知されます。このピンの出力を確認するには、INT\_CFG レジスタで INT0 機能をイネーブルにする必要があることに注意してください。

図 7-68. SPI\_IE\_1 レジスタ

7	6	5	4	3	2	1	0
TXLIE1	TXAIE1	RSVD			RXLIE1	RXFLIE1	RXNIE1
R/W-0h	R/W-0h	R-0h			R/W-0h	R/W-0h	R/W-0h

表 7-111. SPI\_IE\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXLIE1	R/W	0h	INT1 のフル FIFO 割り込みイネーブルによる TX 喪失メッセージ 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
6	TXAIE1	R/W	0h	INT1 での TX 領域使用可能割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
5-3	RSVD	R	0h	
2	RXLIE1	R/W	0h	INT1 の RX オーバーラン / 喪失メッセージ割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
1	RXFLIE1	R/W	0h	INT1 の RX フィル レベル割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RXNIE1	R/W	0h	INT1 の RX 新しいメッセージ割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。

### 7.6.2.15 SPI\_IR レジスタ (オフセット = 100Eh) [リセット = 00h]

図 7-69 に、SPI\_IR を示し、表 7-112 に、その説明を示します。

概略表に戻ります。

現在の割り込みレジスタ。現在セットされているすべての割り込みビットが格納されています。W1C (1 を書き込むことでビットをクリア) でクリアされます。

**図 7-69. SPI\_IR レジスタ**

7	6	5	4	3	2	1	0
TXL	TXA	RSVD			RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-112. SPI\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXL	R/W1C	0h	フル FIFO 割り込みにより TX 喪失メッセージ 0h = TX FIFO メッセージは失われていません 1h = 書き込みが試みられた際に FIFO がフルであったため、TX FIFO がメッセージを喪失しました (オーバーラン)
6	TXA	R/W1C	0h	TX 領域使用可能割り込み  <div style="text-align: center;"><b>注</b></div> SPI IP がイネーブルの場合、FIFO は空になるため、この割り込みがセットされます  0h = TX FIFO 空き割り込みがありません 1h = TX FIFO がトリガ レベル スレッシュホールドに達しました
5-3	RSVD	R	0h	
2	RXL	R/W1C	0h	RX オーバーラン / メッセージ喪失割り込み 0h = RX FIFO メッセージ喪失なし (オーバーランなし) 割り込み 1h = RX FIFO 全体 (オーバーラン割り込み) のために少なくとも 1 つのメッセージが失われました
1	RXFL	R/W1C	0h	RX フィル レベル割り込み。 レベルは SPI_CTRL.RX_LVL_INT によって設定されます 0h = RX FIFO フィル レベルの割り込みがありません 1h = RX FIFO がフィル レベルに達しました
0	RXN	R/W1C	0h	RX 新規メッセージ割り込み 0h = 新規メッセージ受信割り込みはありません 1h = 新しいオートネゴシエーション ページを受信済みです

### 7.6.2.16 SPI\_FS レジスタ (オフセット = 100Fh) [リセット= 80h]

図 7-70 に、SPI\_FS を示し、表 7-113 に、その説明を示します。

概略表に戻ります。

SPI FIFO ステータス

図 7-70. SPI\_FS レジスタ

7	6	5	4	3	2	1	0
TXFSRE	RXFD					RXFB	
RH-1h	RH-0h					RH-0h	

表 7-113. SPI\_FS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXFSRE	RH	1h	<p>TX FIFO とシフトレジスタが空の状態かどうかを判断します。これは、キューに格納されたデータがすべて送信済みであり、コントローラがアイドル状態であることを意味します。FIFO にもシフトレジスタにもデータは残っていません。これを設定した場合、SPI コントローラ 構成または無効化を安全に行えます</p> <hr/> <p style="text-align: center;"><b>注</b></p> <p>ただし、この状態は、TX FIFO にデータが完全に書き込まれていない場合を考慮していません。部分的に書き込まれた TX FIFO 素子がある場合、その未送信素子は完全に書き込まれるまでアイドルフラグをクリアしません</p> <hr/> <p>0h = 少なくとも 1 つの進行中および/または保留中の転送があります。  1h = TX FIFO とシフトレジスタは空で、IP はアイドル状態です。コントローラの構成または無効化が安全です</p>
6	RXFD	RH	0h	<p>RX FIFO に 1 つ以上の未読 素子が含まれているかどうか (SPI 転送)  0h = 受信 FIFO が空です  1h = RX FIFO に少なくとも 1 つの素子が存在します</p>
5-0	RXFB	RH	0h	<p>次の RX FIFO 素子に格納される SPI フレームのサイズ。有効な値は 0~63 です。  63 より大きい値は 63 として反映され、残りの値はプロセッサがデータを読み取った時点で更新されます。</p>

### 7.6.2.17 SPI\_TX\_FIFO レジスタ (オフセット = 1010h) [リセット = 0000h]

図 7-71 に、SPI\_TX\_FIFO を示し、表 7-114 に、その説明を示します。

概略表に戻ります。

SPI 送信 FIFO。このアドレスへの書き込みを実行すると、データは SPI 送信 FIFO に書き込まれます。詳細については、[SPI 送信 FIFO](#) を参照してください。これには SPI フレームのヘッダーのみが表示されます

**図 7-71. SPI\_TX\_FIFO レジスタ**

15	14	13	12	11	10	9	8
STORE_RX	RSVD				CHAN		
W-0h	R-0h				W-0h		
7	6	5	4	3	2	1	0
NUM_BYTES							
W-0h							

**表 7-114. SPI\_TX\_FIFO レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
15	STORE_RX	W	0h	転送を RX FIFO に保存します。 不要な場合は、受信データを無視するために使用できます 詳細については、 <a href="#">SPI 送信 FIFO に関する説明をご覧ください</a> 。
14-11	RSVD	R	0h	予約済み
10-8	CHAN	W	0h	SPI チャネルは送信に使用されます SPI チャネルはチップセレクトピンを制御します 詳細は <a href="#">SPI 送信 FIFO</a> を参照してください。
7-0	NUM_BYTES	W	0h	SPI 転送のバイト数 詳細は <a href="#">SPI 送信 FIFO</a> を参照してください。

### 7.6.2.18 SPI\_RX\_FIFO レジスタ (オフセット = 1010h) [リセット = 0000h]

図 7-72 に、SPI\_RX\_FIFO を示し、表 7-115 に、その説明を示します。

概略表に戻ります。

このアドレスに対して読み出しを実行すると、データは SPI 受信 FIFO から読み取られます。詳細については、[SPI 受信 FIFO](#) を参照してください。。これには、SPI フレームのヘッダーのみが表示されます

図 7-72. SPI\_RX\_FIFO レジスタ

15	14	13	12	11	10	9	8
CONT	RSVD				CHAN		
R-0h	R-0h				R-0h		
7	6	5	4	3	2	1	0
NUM_BYTES_REMAINING							
R-0h							

表 7-115. SPI\_RX\_FIFO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	CONT	R	0h	これが部分的な RX FIFO 素子の読み取りであるかどうかを示すフラグ。 0 の場合、この FIFO 要素への最初の読み取りとなります。詳細は <a href="#">SPI 受信 FIFO</a> を参照してください。
14-11	RSVD	R	0h	予約済み
10-8	CHAN	R	0h	転送は <a href="#">SPI 受信 FIFO</a> 上で行われました。詳細についてはこちらをご覧ください。
7-0	NUM_BYTES_REMAINING	R	0h	FIFO 要素で読み取る残りのバイト数(本トランザクションを含む)詳細は <a href="#">SPI 受信 FIFO</a> を参照してください。

### 7.6.2.19 SPI\_RXFS レジスタ (オフセット = 1011h) [リセット= 00h]

図 7-73 に、SPI\_RXFS を示し、表 7-116 に、その説明を示します。

概略表に戻ります。

RX FIFO に格納されている未読メッセージの数のカウンタが含まれます。

**図 7-73. SPI\_RXFS レジスタ**

7	6	5	4	3	2	1	0
RX_FL							
R-0h							

**表 7-116. SPI\_RXFS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	RX_FL	R	0h	RX FIFO フィル レベルは、現在 RX FIFO 内に格納されているメッセージの数を示します。 値の範囲は 1~255 です。 表示される最大値は、デバイス メモリ の設定方法によって異なります。 255 が報告された場合、255 以上のフレームが FIFO にあります。

**7.6.2.20 SPI\_TXFS レジスタ (オフセット = 1012h) [リセット = 00h]**

図 7-74 に、SPI\_TXFS を示し、表 7-117 に、その説明を示します。

概略表に戻ります。

TX FIFO 内で利用可能な空き数を示すカウンタを含みます。

**図 7-74. SPI\_TXFS レジスタ**

7	6	5	4	3	2	1	0
TX_SA							
R-0h							

**表 7-117. SPI\_TXFS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	TX_SA	R	0h	TX FIFO フィル レベルは、現在 TX FIFO にあるフリーバイト数を示します。  <div>注</div> <div>利用可能なバイト数が 255 を超える場合、空き容量は 255 バイトと表示されます</div>



### 7.6.2.21 SPI\_TXES レジスタ (オフセット = 1013h) [リセット = 00h]

図 7-75 に、SPI\_TXES を示し、表 7-118 に、その説明を示します。

概略表に戻ります。

現在 / 次の送信 FIFO 素子に関する情報が含まれます

**図 7-75. SPI\_TXES レジスタ**

7	6	5	4	3	2	1	0
TXEIP	TXEBP						
R-0h	R-0h						

**表 7-118. SPI\_TXES レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXEIP	R	0h	TX FIFO 素子書き込み中 TX FIFO 素子が部分的に書き込まれていることを示すために使用されます。 (ヘッダーに従って) FIFO に書き込まれた十分なデータバイトがない場合、このフラグがセットされます。 クリアされている場合、TX FIFO への次の書き込みは新しい要素となり、SPI ヘッダフィールドから開始する必要があります 0h = TX 素子は進行中ではありません (次の TX FIFO への書き込みは新しい要素です) 1h = TX 要素は進行中で、予期される数のバイトが残っています
6-0	TXEBP	R	0h	TX 素子バイト保留 TX 素子の書き込みが開始された場合、FIFO 素子を完了するためにまだ何バイトが予想されるかを示します。 残りの予想バイト数が 127 バイトを超える場合でも、残り 127 バイトが表示されます (表示可能な最大数のため)  <div style="text-align: center;"><b>注</b></div> これにはヘッダーバイトが含まれており、転送中のバイト数がまだ書き込まれていない場合は、値を変更できることがあります。 例えば、SPI チャネルヘッダーバイトのみが書き込まれた場合、このフィールドは 1 と読み出されます。これは、転送するデータのバイト数を指定されるまで保留しているバイト数がわからないためです。 転送のバイト数が判明し、データが書き込まれると、この計算が再計算されます

### 7.6.3 SPI データ FIFO

先入れ先出しデータ構造 (FIFO) は、データをデバイスにシフトアウトするために使用されます。2 つの主な FIFO : 受信 (RX) FIFO と送信 (TX) FIFO です。アドレスはどちらにアクセスする場合も同じですが、書き込みを行うとデータは TX FIFO にコピーされ、読み取りを行うと RX FIFO からデータが読み出されます。

#### 7.6.3.1 SPI 送信 FIFO (アドレス = h1010)

送信 FIFO は書き込み専用で、アドレスに対して書き込みコマンドが発行されると書き込み専用になります。フレーム全体がバッファに書き込まれると、送信が開始されます。フレームの終端後に書き込まれたすべてのデータバイトは、新しい送信 FIFO 書き込みが開始されるまで無視されます。送信 FIFO へのすべての書き込みはメイン アドレスに対してのみ行う必要があります。書き込まれた各バイトは、自動的に FIFO にシフトされるためです。例えば、一つの CAN メッセージで転送できるより大きな SPI メッセージを書き込む場合、同じアドレスへの複数回の CAN 書き込みによって処理されます。これは、SPI ヘッダバイトがデバイスに受信すべきデータバイト数を知らせるためです。

TX バッファ内の各フレームのサイズは、メッセージ長によって決まります。各 SPI FIFO 素子の先頭には、2 バイトのヘッダが必要です。バッファ内の TX フレームの全長は、以下の合計です

- SPI ヘッダ用 2 バイト
- データ バイトのメッセージ長

詳細と例については、[SPI 制御プロトコル](#)を参照してください。

#### 注

送信 FIFO がフルの場合、書き込まれた新しいメッセージはすべて破棄され、割り込みがセットされます。

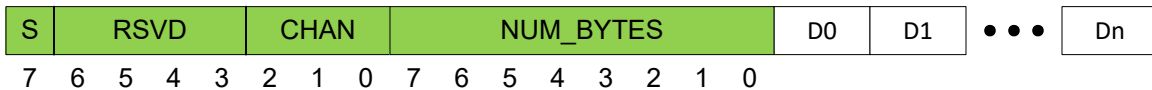


図 7-76. SPI TX FIFO 書き込みヘッダ (ヘッダおよびデータのみ)

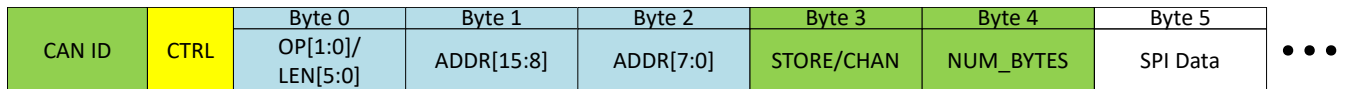


図 7-77. SPI TX FIFO ヘッダの例 (標準 CAN フレームフォーマット)

	7	6	5	4	3	2	1	0
B0	STORE	RSVD				CHAN		
B1	NUM_BYTES							
B2	DB0[7:0]							
B3	DB1[7:0]							
...	...							
Bn	DBn[7:0]							

図 7-78. SPI 送信 FIFO

表 7-119. SPI 送信 FIFO の説明

バイト	ビット	フィールド	タイプ	リセット	説明
0	7	STORE_RX	W	0	このメッセージの POCI データを受信 FIFO に保存するか どうか 0 = 受信 FIFO にデータを保存しません 1 = 受信した POCI データを受信 FIFO に保存します
	6:3	RSVD	R	0	予約済み
	2:0	CHAN	W	0x0	このメッセージに使用された SPI チャネル 有効な値 0 ~ 3 0 = SPI チャネル 0 1 = SPI チャネル 1 2 = SPI チャネル 2 3 = SPI チャネル 3 4 = SPI チャネル 4 5 = SPI チャネル 5 6 = SPI チャネル 6 7 = SPI チャネル 7  <div style="text-align: center;">注</div> SPI チャネルは特定の GPIO に対応しており、 GPIO が特殊機能として割り当てる必要があり ます
1	7:0	NUM_BYTES	W	0x0	SPI フレーム内のバイト数 (上記のヘッダ バイトは含まれ ません) 有効な値は 1 ~ 255 です。0 は無効です。

### 7.6.3.2 SPI 受信 FIFO (アドレス = h1010)

受信 FIFO は読み取り専用です。

RX バッファ内の各フレームのサイズは、メッセージのデータ長によって決まります。バッファ内の RX フレームの全長は、以下の合計です

- SPI ヘッダ用 2 バイト
- データ バイトのメッセージ長

RX FIFO からの読み取りは、FIFO 素子内のすべてのデータ バイトが読み出された時点で完了と見なされ、解放されます。読み取りの残りのバイト数を超えるバイトは、無効なバイトに対して 0x00 を返します。

RX FIFO からのフレームの部分読み出しは、自動的に次の SPI FIFO 読み出しで継続されます。その際、CONT ビットがセットされ、NUM\_BYTES\_REMAINING にこのメッセージで残りのデータ バイト数が反映されます。

空の RX バッファから読み出した場合、チャンネル 0 の 0 NUM\_BYTES\_REMAINING が返されます。フレームは常に受信された順序で読み取られます。

RX バッファは、SPI Clear RX FIFO ビット (SPI\_FIFO\_CTRL[1]) に 1 を書き込むことでクリアされ (すべてのフレームが破棄されます)、消去できます。

#### 注

受信 FIFO がフルの場合、すべての新しい受信メッセージは破棄され、メッセージが失われたことをプロセッサに通知する割り込みが設定されます。

詳細と例については、[SPI 制御プロトコル](#)を参照してください。

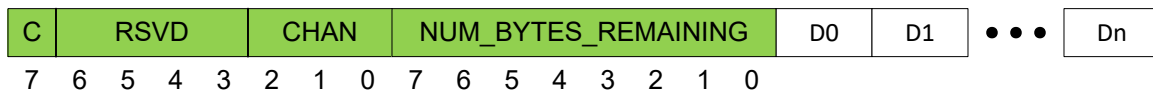


図 7-79. SPI RX FIFO 読み取りヘッダ (ヘッダおよびデータのみ)



図 7-80. SPI RX FIFO ヘッダの例 (標準 CAN フレームフォーマット)

	7	6	5	4	3	2	1	0
B0	CONT	RSVD				CHAN		
B1	NUM_BYTES_REMAINING							
B2	DB0[7:0]							
B3	DB1[7:0]							
...	...							
Bn	DBn[7:0]							

図 7-81. SPI 受信 FIFO

表 7-120. SPI 受信 FIFO の説明 (SPI ヘッダのスタート)

バイト	ビット	フィールド	タイプ	リセット	説明
0	7	CONT	R	0	これがメッセージの最初の部分か継続かを示すフラグ (複数の CAN 読み取りに分割されたメッセージの読み取りに使用) 0 = 新しいメッセージの最初の部分 1 = 前に読み取られたメッセージの継続かを示すフラグ
	6:3	RSVD	R	0	予約済み
	2:0	CHAN	R	0x0	このメッセージに使用された SPI チャネル 有効な値 0 ~ 3 0 = SPI チャネル 0 1 = SPI チャネル 1 2 = SPI チャネル 2 3 = SPI チャネル 3 4 = SPI チャネル 4 5 = SPI チャネル 5 6 = SPI チャネル 6 7 = SPI チャネル 7  <div style="text-align: center;">注</div> SPI チャネルは特定の GPIO に対応しており、GPIO が特殊機能として割り当てる必要があります
1	7:0	NUM_BYTES_REMAINING	R	0x0	このメッセージで読み取る必要がある残りのデータ バイト数 (現在のフレームで送信されるデータバイトを含みます)。このには、ヘッダ バイトは含まれません。これにより、プロセッサは、現在の CAN フレームのバイト数が有効であること、およびこのメッセージの FIFO からさらに多くのデータを取得するために別の読み取りが必要であるかどうかを知ることができます。

## 7.6.4 UART レジスタ

セクション 7.6.4 に、UART レジスタに対してメモリマップされたレジスタを一覧表示します。セクション 7.6.4 がないレジスタ オフセット アドレスはすべて予約済みと見なして、レジスタの内容は変更しないでください。

表 7-121. UART レジスタ

オフセット	略称	レジスタ名	セクション
2000h	UART_CREL	UART コア リリース	<a href="#">セクション 7.6.4.1</a>
2001h	UART_SCRATCH	UART スクラッチパッド	<a href="#">セクション 7.6.4.2</a>
2002h	UART_CTRL	UART 制御	<a href="#">セクション 7.6.4.3</a>
2003h	UART_BR_LSB	UART ボーレート (LSB)	<a href="#">セクション 7.6.4.4</a>
2004h	UART_BR_MSB	UART ボーレート (MSB)	<a href="#">セクション 7.6.4.5</a>
2005h	UART_BR_FRAC	UART ボーレート (分数)	<a href="#">セクション 7.6.4.6</a>
2006h	UART_FIFO_CTRL	UART FIFO 制御	<a href="#">セクション 7.6.4.7</a>
2007h	UART_IE_0	UART 割り込みイネーブル 0	<a href="#">セクション 7.6.4.8</a>
2008h	UART_IE_1	UART 割り込みイネーブル 1	<a href="#">セクション 7.6.4.9</a>
2009h	UART_IR	UART 割り込みレジスタ	<a href="#">セクション 7.6.4.10</a>
200Ah	UART_STATUS	UART のステータス	<a href="#">セクション 7.6.4.11</a>
200Bh	UART_RXFS	UART RX FIFO ステータス	<a href="#">セクション 7.6.4.12</a>
200Ch	UART_TXFS	UART TX FIFO ステータス	<a href="#">セクション 7.6.4.13</a>
2010h	UART_RX_FIFO	UART RX FIFO	<a href="#">セクション 7.6.4.14</a>
2010h	UART_TX_FIFO	UART TX FIFO	<a href="#">セクション 7.6.4.15</a>
2011h	UART_RX_ERR_STATUS	UART RX FIFO エラー ステータス	<a href="#">セクション 7.6.4.16</a>

表の小さなセルに収まるように、複雑なビット アクセス タイプを記号で表記しています。[セクション 7.6.4](#) に、このセクションでアクセス タイプに使用しているコードを示します。

表 7-122. UART のアクセス タイプ コード

アクセス タイプ	コード	説明
読み取りタイプ		
R	R	読み出し
RH	R H	ハードウェアによってセットまたはクリアされる の読み取り
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	1 を書き込むことで セット
WP	W P	書き込み 特権アクセスが必要
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト 値

#### 7.6.4.1 UART\_CREL レジスタ (オフセット = 2000h) [リセット= 10h]

図 7-82 に、UART\_CREL を示し、表 7-123 に、その説明を示します。

概略表に戻ります。

UART IP リリース バージョン

図 7-82. UART\_CREL レジスタ

7	6	5	4	3	2	1	0
MAJOR				MINOR			
R-1h				R-0h			

表 7-123. UART\_CREL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-4	MAJOR	R	1h	このリビジョンのメジャー バージョン番号
3-0	MINOR	R	0h	このリビジョンのマイナー バージョン番号

## 7.6.4.2 UART\_SCRATCH レジスタ (オフセット = 2001h) [リセット= 00h]

図 7-83 に、UART\_SCRATCH を示し、表 7-124 に、その説明を示します。

概略表に戻ります。

顧客スクラッチパッド レジスタ

図 7-83. UART\_SCRATCH レジスタ

7	6	5	4	3	2	1	0
スクラッチ パッド							
R/W-0h							

表 7-124. UART\_SCRATCH レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	スクラッチ パッド	R/W	0h	顧客スクラッチ パッド レジスタ。 これは、値の書き込みと読み取りによって通信を検証するために使用することも、一部のデータの保存に使用することもできます。 これは、リセット / 電源オン リセットによって保持されません。



### 7.6.4.3 UART\_CTRL レジスタ (オフセット = 2002h) [リセット = 04h]

図 7-84 に、UART\_CTRL を示し、表 7-125 に、その説明を示します。

概略表に戻ります。

このレジスタはデータ通信フォーマットを制御します。ワード長、ストップ ビット数、パリティ タイプ、および UART 構成を変更することも可能です。

**図 7-84. UART\_CTRL レジスタ**

7	6	5	4	3	2	1	0
BRKGEN	FPAREN	PAR	PEN	2BSTOP	8BIT	UART_EN	CCE
R/W-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-1h	RH-0h	R/W-0h

**表 7-125. UART\_CTRL レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	BRKGEN	R/W	0h	ブレーク制御ビット 0h = 通常の UART 動作 1h = TXD を Low に強制して、ブレークを生成
6	FPAREN	R/WP	0h	パリティ ビット (イネーブルの場合) を特定の値に強制します 0h = 設定されたとおりにパリティを生成 1h = 送信および受信データのパリティ ビット (PAR) 構成で定義されているパリティ ビットを強制します
5	PAR	R/WP	0h	奇数または偶数パリティ構成 0h = 奇数パリティ (1 ビット値) 1h = 偶数パリティ (0 ビット値)
4	PEN	R/WP	0h	各文字にパリティが使用されている場合にイネーブルにします 0h = パリティ ビットなし 1h = パリティ ビット ジェネレータ
3	2BSTOP	R/WP	0h	1 文字あたりに使用されるストップ ビット数を定義します 0h = 1 ストップ ビット 1h = 2 ストップ ビット
2	8BIT	R/WP	1h	文字のサイズを定義します 0h = 7 ビット データ 1h = 8 ビット データ
1	UART_EN	RH	0h	UART IP イネーブル ステータス フラグ。 このフラグは書き込み可能ではありませんが、MRAM から IP へのメモリ割り当てによって UART IP がイネーブルになっている場合に設定されます。 これは、 <a href="#">MRAM_IP_CFG</a> レジスタで行われます 0h = UART IP が無効の場合、すべての UART 機能が無効になります 1h = UART IP が有効です
0	CCE	R/W	0h	UART IP 変更制御イネーブル ビット。 UART_EN が 0 の場合のみ設定できます 0h = UART 構成レジスタは書き込み保護されています 1h = UART 構成レジスタを変更できます

#### 7.6.4.4 UART\_BR\_LSB レジスタ (オフセット = 2003h) [リセット = 00h]

図 7-85 に、UART\_BR\_LSB を示し、表 7-126 に、その説明を示します。

概略表に戻ります。

ボー クロックに使用される 16 ビット除数の下位バイト

図 7-85. UART\_BR\_LSB レジスタ

7	6	5	4	3	2	1	0
DLL							
R/WP-0h							

表 7-126. UART\_BR\_LSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	DLL	R/WP	0h	ボーレート生成に使用される 16 ビット除数の最下位バイト。

#### 7.6.4.5 UART\_BR\_MSB レジスタ (オフセット = 2004h) [リセット = 00h]

図 7-86 に、UART\_BR\_MSB を示し、表 7-127 に、その説明を示します。

概略表に戻ります。

ボー クロックに使用される 16 ビット除数の上位バイト

図 7-86. UART\_BR\_MSB レジスタ

7	6	5	4	3	2	1	0
DLH							
R/WP-0h							

表 7-127. UART\_BR\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	DLH	R/WP	0h	ボーレート生成に使用される 16 ビット除数の最上位バイト。

#### 7.6.4.6 UART\_BR\_FRAC レジスタ (オフセット = 2005h) [リセット= 00h]

図 7-87 に、UART\_BR\_FRAC を示し、表 7-128 に、その説明を示します。

概略表に戻ります。

ボー クロックに使用される 6 ビット分数除数 (x/64 値)

図 7-87. UART\_BR\_FRAC レジスタ

7	6	5	4	3	2	1	0
BD	RSVD	DLF					
R/WP-0h	R-0h	R/WP-0h					

表 7-128. UART\_BR\_FRAC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	BD	R/WP	0h	ボー生成に使用されるオーバー サンプリングを制御するためのボーデバイタ ビット。 0h = -16 分周器をイネーブル 1h = -8 分周器をイネーブル
6	RSVD	R	0h	
5-0	DLF	R/WP	0h	ボーレート生成に使用される分数除数 (x/64)。

#### 7.6.4.7 UART\_FIFO\_CTRL レジスタ (オフセット = 2006h) [リセット= 01h]

図 7-88 に、UART\_FIFO\_CTRL を示し、表 7-129 に、その説明を示します。

概略表に戻ります。

FIFO の有効化または FIFO の内容をクリアするための FIFO 制御フィールド。

**図 7-88. UART\_FIFO\_CTRL レジスタ**

7	6	5	4	3	2	1	0
RX_TRG		TX_TRG		RSVD	CLR_TX	CLR_RX	FIFO_EN
R/WP-0h		R/WP-0h		R-0h	RH/W1S-0h	RH/W1S-0h	R/WP-1h

**表 7-129. UART\_FIFO\_CTRL レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-6	RX_TRG	R/WP	0h	RX FIFO のトリガ レベルを設定します 0h = RX FIFO で 1 バイト 1h = RX FIFO で 4 バイト 2h = (最大サイズ) - RX FIFO で 16 バイト 3h = (最大サイズ) - RX FIFO で 8 バイト
5-4	TX_TRG	R/WP	0h	TX FIFO の空き領域数のトリガ レベルを設定します 0h = 16 バイトの空き 1h = 32 バイトの空き 2h = (最大サイズ) - 16 バイトの空き 3h = (最大サイズ) - 8 バイトの空き
3	RSVD	R	0h	
2	CLR_TX	RH/W1S	0h	送信 FIFO の内容をクリアします。 クリアが完了すると、ハードウェアはビットを 0 に戻します。 0h = 何もしない 1h = 送信 FIFO をクリア
1	CLR_RX	RH/W1S	0h	受信 FIFO の内容をクリアします。 クリアが完了すると、ハードウェアはビットを 0 に戻します。 0h = 何もしない 1h = 受信 FIFO をクリア
0	FIFO_EN	R/WP	1h	TX と RX の各 FIFO を有効にします 0h = 送信および受信 FIFO をデイスエーブル 1h = 送信および受信 FIFO をイネーブル

#### 7.6.4.8 UART\_IE\_0 レジスタ (オフセット = 2007h) [リセット = 00h]

図 7-89 に、UART\_IE\_0 を示し、表 7-130 に、その説明を示します。

概略表に戻ります。

INT0 ピンの特定の割り込みをイネーブルするための割り込みイネーブル ビット。ここでイネーブルになっている割り込みは、INT0 ピンで通知されます。INT0 機能を有効にする必要があることに注意してください。

図 7-89. UART\_IE\_0 レジスタ

7	6	5	4	3	2	1	0
TXLIE0	TXAIE0	RXBRIE0	RXFEIE0	RXPEIE0	RXLIE0	RXFLIE0	RXNIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-130. UART\_IE\_0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXLIE0	R/W	0h	TX ガフルのためにバイトを失った場合に、INT0 割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
6	TXAIE0	R/W	0h	INT0 での TX 領域使用可能割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
5	RXBRIE0	R/W	0h	RX ブレーク受信時の INT0 割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
4	RXFEIE0	R/W	0h	RX フレーミング エラー発生時の INT0 割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
3	RXPEIE0	R/W	0h	INT0 の RX パリティ エラー割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
2	RXLIE0	R/W	0h	INT0 の RX オーバーラン / 喪失バイト割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
1	RXFLIE0	R/W	0h	INT0 の RX フィル レベル割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RXNIE0	R/W	0h	INT0 の RX 新しいバイト割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。

#### 7.6.4.9 UART\_IE\_1 レジスタ (オフセット = 2008h) [リセット = 00h]

図 7-90 に、UART\_IE\_1 を示し、表 7-131 に、その説明を示します。

概略表に戻ります。

INT1 ピンの特定の割り込みをイネーブルするための割り込みイネーブルビット。ここでイネーブルになっている割り込みは、INT1 ピンで通知されます。INT1 機能を有効にする必要があることに注意してください。

**図 7-90. UART\_IE\_1 レジスタ**

7	6	5	4	3	2	1	0
TXLIE1	TXAIE1	RXBRIE1	RXFEIE1	RXPEIE1	RXLIE1	RXFLIE1	RXNIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-131. UART\_IE\_1 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXLIE1	R/W	0h	TX ガフルのためにバイトを失った場合に、INT1 割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
6	TXAIE1	R/W	0h	INT1 での TX 領域使用可能割り込みイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
5	RXBRIE1	R/W	0h	RX ブレーク受信時の INT1 割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
4	RXFEIE1	R/W	0h	RX フレーミング エラー発生時の INT1 割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
3	RXPEIE1	R/W	0h	INT1 の RX パリティ エラー割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
2	RXLIE1	R/W	0h	INT1 の RX オーバーラン / 喪失バイト割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
1	RXFLIE1	R/W	0h	INT1 の RX フィル レベル割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RXNIE0	R/W	0h	INT1 の RX 新しいバイト割り込みをイネーブル 0h = 割り込みはデイスエーブル。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。

#### 7.6.4.10 UART\_IR レジスタ (オフセット = 2009h) [リセット = 00h]

図 7-91 に、UART\_IR を示し、表 7-132 に、その説明を示します。

概略表に戻ります。

保留中の割り込みレジスタ。現在セットされているすべての割り込みビットが格納されます。W1C (1 を書き込むことでビットをクリア) でクリアされます。

図 7-91. UART\_IR レジスタ

7	6	5	4	3	2	1	0
TXL	TXA	RXBR	RXFE	RXPE	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 7-132. UART\_IR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXL	R/W1C	0h	フル状態のために TX バイトを喪失 0h = TX FIFO バイトは失われていません 1h = 書き込みの試行時に FIFO がフルになったために TX FIFO がバイト (オーバーラン) を失いました
6	TXA	R/W1C	0h	TX スペースが利用可能  <b>注</b> UART IP がイネーブルの場合、FIFO は空になるため、この割り込みがセットされます  0h = TX FIFO 空き割り込みがありません 1h = TX FIFO がトリガ レベル スレッシュホールドに達しました
5	RXBR	R/W1C	0h	RX ブレークを受信 0h = ブレークは受信されていません 1h = ブレークを受信しました
4	RXFE	R/W1C	0h	RX フレーミング エラー 0h = 受信フレーミング エラー イベントが発生していません 1h = RX フレーミング エラーが発生しました
3	RXPE	R/W1C	0h	RX パリティ エラー 0h = 受信エラー イベントが未発生です 1h = RX パリティ エラーが発生しました
2	RXL	R/W1C	0h	RX オーバーラン / 喪失バイト 0h = RX FIFO バイト喪失 (オーバーランなし) 割り込みはありません 1h = RX FIFO 全体 (オーバーラン割り込み) のために少なくとも 1 バイトが喪失されました
1	RXFL	R/W1C	0h	RX フィルレベル 0h = RX FIFO フィル レベルの割り込みがありません 1h = RX FIFO がフィル レベルに達しました
0	RXN	R/W1C	0h	RX 新規バイト 0h = 割り込み受信された新しいバイトはありません 1h = 新しいバイトを受信済みです



#### 7.6.4.11 UART\_STATUS レジスタ (オフセット = 200Ah) [リセット = 60h]

図 7-92 に、UART\_STATUS を示し、表 7-133 に、その説明を示します。

概略表に戻ります。

**図 7-92. UART\_STATUS レジスタ**

7	6	5	4	3	2	1	0
RXBS	TXFSRE	TXFE	RSVD				RXFB
RH-0h	RH-1h	RH-1h	R-0h				RH-0h

**表 7-133. UART\_STATUS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RXBS	RH	0h	RX FIFO 内の少なくとも 1 バイトにパリティ エラー、フレーミング エラー、またはブレークが発生しています 0h = RX FIFO 内のいずれのバイトにもパリティ エラー、フレーミング エラー、またはブレークはありません 1h = パリティ エラー、フレーミング エラー、または RX FIFO のブレークで少なくとも 1 バイトです
6	TXFSRE	RH	1h	TX FIFO とシフト レジスタが空であるかどうかです。 これは、すべてのキューに入っていたデータがシフトアウトされたことを意味します。 FIFO には何もなく、シフト レジスタには何もありません 0h = TX FIFO またはシフト レジスタに少なくとも 1 つの素子 1h = TX FIFO とシフト レジスタは空です
5	TXFE	RH	1h	TX FIFO が空かどうかで、必ずしも TX シフト レジスタではありません 0h = TX FIFO に少なくとも 1 つの素子が存在します 1h = TX FIFO は空です
4-1	RSVD	R	0h	
0	RXFB	RH	0h	RX FIFO に 1 つ以上の未読バイトが含まれているかどうかです 0h = RX FIFO は空です 1h = RX FIFO に少なくとも 1 つの素子が存在します

**7.6.4.12 UART\_RXFS レジスタ (オフセット = 200Bh) [リセット = 00h]**

図 7-93 に、UART\_RXFS を示し、表 7-134 に、その説明を示します。

概略表に戻ります。

RX FIFO に格納されている未読メッセージの数のカウンタが含まれます。

**図 7-93. UART\_RXFS レジスタ**

7	6	5	4	3	2	1	0
RX_FL							
R-0h							

**表 7-134. UART\_RXFS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	RX_FL	R	0h	RX FIFO フィル レベルは、現在 RX FIFO 内に格納されているバイト数を示します。 有効な値は 0~255 です。 表示される最大値は、デバイス メモリ の設定方法によって異なります。 255 が報告された場合、255 以上のバイトが FIFO にあります。

#### 7.6.4.13 UART\_TXFS レジスタ (オフセット = 200Ch) [リセット= 00h]

図 7-94 に、UART\_TXFS を示し、表 7-135 に、その説明を示します。

概略表に戻ります。

TX FIFO 内で利用可能な空き数を示すカウンタを含みます。

図 7-94. UART\_TXFS レジスタ

7	6	5	4	3	2	1	0
TX_SA							
R-0h							

表 7-135. UART\_TXFS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	TX_SA	R	0h	TX FIFO フィル レベルは、現在 TX FIFO にあるフリー スペース数を示します

#### 7.6.4.14 UART\_RX\_FIFO レジスタ (オフセット = 2010h) [リセット= 00h]

図 7-95 に、UART\_RX\_FIFO を示し、表 7-136 に、その説明を示します。

概略表に戻ります。

次の FIFO 素子を読み出し、CAN メッセージに収まる要求バイト数までのデータを返します。読み出しデータセクション内のいずれかのバイトに非正常ステータスがあるかどうかを示すステータス バイトが、データの末尾に追加されます。

図 7-95. UART\_RX\_FIFO レジスタ

7	6	5	4	3	2	1	0
データ							
R-0h							

表 7-136. UART\_RX\_FIFO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	データ	R	0h	RX FIFO の次のデータ バイトを読み出します。 バースト読み取りを行う前に、FIFO 内のバイト数を確認するために UART_RXFS レジスタを読むことが推奨されます。 このレジスタへのバースト読み取りでは、要求したバイト数までの複数バイトのデータが返されます。 データ バイトの最後に、グローバル ステータス バイトが付加され、読み出されたデータ バイトにエラーが記録されているかどうかを示します。 ステータス バイトが非正常ステータスを返した場合、同じ長さで UART_RX_ERR_STATUS レジスタを読み出すことで、読み出された各データ バイトに対応するステータス バイトが得られ、どのバイトが非正常ステータスであったかを判別できます。

#### 7.6.4.15 UART\_TX\_FIFO レジスタ (オフセット = 2010h) [リセット= 00h]

図 7-96 に、UART\_TX\_FIFO を示し、表 7-137 に、その説明を示します。

概略表に戻ります。

UART 送信 FIFO への書き込みは、バッファに書き込まれるとすぐに、送信するデータのバイトをキューに格納します。

図 7-96. UART\_TX\_FIFO レジスタ

7	6	5	4	3	2	1	0
データ							
W-0h							

表 7-137. UART\_TX\_FIFO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	データ	W	0h	この FIFO にデータを書き込むと、UART バスへのデータ送信が開始されます。 書き込みを行う前に、利用可能な空き数を確認するために UART_TXFS レジスタを読み取ることを推奨します。

#### 7.6.4.16 UART\_RX\_ERR\_STATUS レジスタ (オフセット = 2011h) [リセット = 00h]

図 7-97 に、UART\_RX\_ERR\_STATUS を示し、表 7-138 に、その説明を示します。

概略表に戻ります。

これは、UART\_RX\_FIFO 素子を読み出した後に読み取られます。ステータス バイトが、少なくとも 1 バイトに非正常ステータスがあったことを示した場合、このレジスタから同じ長さのデータを読み出すと、直前に RX FIFO から読み出された各バイトに対応するステータス バイトが返されます。

図 7-97. UART\_RX\_ERR\_STATUS レジスタ

7	6	5	4	3	2	1	0
RSVD		BRK		FE	PAR	NO_RX	NO_ERR
R-0h		R-0h		R-0h	R-0h	R-0h	R-0h

表 7-138. UART\_RX\_ERR\_STATUS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	BRK	R	0h	ブレーク条件が検出されたときに設定されます。 データ バイトは、ブレークの場合 0x00 として返されます 0h = ブレーク条件ではありません 1h = ブレークです
3	FE	R	0h	セットされている場合、そのバイトはフレーミング エラー (ストップ ビット数が無効) とともに受信されたことを示します 0h = バイト フレーミングが有効 1h = バイト フレーミングが無効
2	PAR	R	0h	設定されているとき、そのバイトに無効なパリティ ビットが送られ ました 0h = パリティ ビットが有効 1h = パリティ ビットが無効
1	NO_RX	R	0h	空の RX FIFO を読み取ると、0x00 が返され、このビットは、データ が無効 / 受信されないことを示します。 これは受信がないことを知らせます 0h = 空 / 受信バイトではありません 1h = このバイトは無効または受信されず、空です
0	NO_ERR	R	0h	設定されると、そのバイトがエラーなしで受信され、かつ有効なバ イトであることを示します。 他のビットがセットされていない場合、このビットがセットされます 0h = いくつかの非通常ステータスが発生しています 1h = バイトの受信が成功

#### 7.6.5 UART データ FIFO

FIFO は、UART のデータ転送に使用されます。インターフェイスにはいくつかの FIFO が使用されます

- UART 送信 FIFO (h2010) : UART インターフェイスに送信するデータのバイトをキューイングするために使用されます
- UART 受信 FIFO (h2010) : UART で受信したバイトをグローバル ステータス バイトで読み取るために使用されます。
- UART 受信エラー ステータス (h2011) : 前の読み出しで受信した各バイトのステータスをチェックするために使用されます

##### 7.6.5.1 UART 送信 FIFO (アドレス = h2010)

送信 FIFO は書き込み専用で、アドレスに対して書き込みコマンドが発行されると書き込み専用になります。モジュールがアイドル状態の場合、FIFO にバイトが書き込まれるとすぐに送信が開始されます。UART インターフェイスは、バイトごとにデータを処理します。

詳細と例については、UART 制御プロトコルを参照してください。

**注**

送信 FIFO がフルの場合、新たに書き込まれたバイトは破棄され、割り込みが設定されます。

**表 7-139. UART 送信 FIFO の説明**

ビット	フィールド	タイプ	リセット	説明
7:0	データ	W	0	この FIFO にデータを書き込むと、UART バスへのデータ送信が開始されます。書き込みを行う前に、利用可能な空き数を確認するために UART_TXFS レジスタを読み取ることを推奨します。

### 7.6.5.2 UART 受信 FIFO (アドレス = h2010)

受信 FIFO は読み取り専用で、RX FIFO の次のデータバイトを読み取ります。バースト読み取りを行う前に、FIFO 内のバイト数を確認するために UART\_RXFS レジスタを読むことが推奨されます。このレジスタへのバースト読み取りでは、要求したバイト数までの複数バイトのデータが返されます。

データ バイトの末尾に、すべてのバイトのステータスを論理的にビット単位で OR したグローバル ステータス バイトが付加されます。これは、読み込まれたバイトのいずれかに非正常ステータスがあるかどうかを示すために使用されます。ステータス バイトが非正常を返す場合、UART\_RX\_ERR\_STATUS レジスタを読み取ることで、直近に読み取った各データ バイトに対応するステータス バイトを取得できます。これにより、どのバイトが非正常ステータスであったかを確認できます。

通常のグローバル ステータス バイトは 0x01 を返します。0x00 は、無効または読み取りオーバーランが発生したことを示します。

空の RX バッファから読み出した場合、0 が返され、ステータス バイトは 0 です。フレームは常に受信された順に読み取られます。

RX バッファは、UART クリア RX FIFO ビット (UART\_FIFO\_CTRL[1]) に 1 を書き込むことによってクリアできます (すべてのフレームが破棄されます)。

詳細と例については、[UART 制御プロトコル](#)を参照してください。

#### 注

受信 FIFO がフルの場合、すべての新しい受信メッセージは破棄され、メッセージが失われたことをプロセッサに通知する割り込みが設定されます。

	7	6	5	4	3	2	1	0
B0	DB0[7:0]							
B1	DB1[7:0]							
...	...							
Bn	RSVD		BRK	FE	PAR	NO RX	NO ERR	Global Status Byte

図 7-98. UART 受信 FIFO バースト読み取り

表 7-140. UART 受信 FIFO の説明 (グローバル ステータス バイトあり)

バイト	ビット	フィールド	タイプ	リセット	説明
0 ~ N-1	7:0	データ	R	0	受信したデータのバイト



**表 7-140. UART 受信 FIFO の説明 (グローバル ステータス バイトあり) (続き)**

バイト	ビット	フィールド	タイプ	リセット	説明
N (最後)	7:5	RSVD	R	0x0	予約済み
	4	BRK	R	0	ブレーク条件が検出されたときに設定されます。ブレークの場合、データバイトは 0x00 として返されます 0 = バイトでブレークが検出されます 1 = 少なくとも 1 バイトはブレークです
	3	フレーム	R	0	設定されている場合、フレーミング エラー (無効なストップビット数) でバイトを受信しました 0 = バイト内でフレーミング エラーが検出されませんでした 1 = 少なくとも 1 バイトにフレーミング エラーがありました
	2	PAR	R	0	パリティ エラー。設定されている場合、無効なパリティ ビットがある状態で 1 バイト以上が受信されました。 0 = のすべてのパリティビットが有効 1 = 少なくとも 1 つのパリティ ビットが不正
	1	NO_RX	R	0	空バイト / RX なし。セットされている場合、1 バイト以上が無効なバイトであり、読み取りオーバーランが発生しました。 0 = すべてのバイトが有効 1 = 少なくとも 1 バイトが無効
	0	NO_ERR	R	0	エラーなし。設定されている場合、すべてのバイトが正常に受信されました。 0 = 少なくとも 1 バイトが非正常なステータスです 1 = すべてのバイトが正常に受信されました。

### 7.6.5.3 UART 受信エラー ステータス (アドレス = h2011)

UART 受信エラー ステータスレジスタは UART 受信 FIFO と同様に動作しますが、直近に読み取られた各 UART バイトに対応する個別のステータス バイトを返す点が異なります。

プロセッサが UART RX FIFO から読み取ると、対応するデータ バイトのステータス バイトが UART 受信エラー ステータス [FIFO] にロードされます。一般的な手順では、UART RX FIFO から読み取り、グローバル ステータス レジスタをチェックして、いずれかのバイトが通常以外のステータスを報告したかどうかを確認します。通常以外のステータス バイトが返される場合、プロセッサは UART RX FIFO から読み出された UART データ バイト数の UART 受信エラー ステータスを読み取る必要があります。これは、UART RX FIFO から最新に読み出された各 UART データ バイトのステータス バイトを返します。

このレジスタの内容は、UART RX FIFO を読み出すとクリアされ、このレジスタの内容が最新の UART データ バイトのステータス バイトで更新されます。

詳細と例については、[UART 制御プロトコル](#)を参照してください。ステータス バイトの値を以下に示します。

図 7-99. UART RX FIFO エラー ステータス レジスタ

7	6	5	4	3	2	1	0
RSVD		BRK		FE	PAR	NO_RX	NO_ERR
R-0h		R-0h		R-0h	R-0h	R-0h	R-0h

表 7-141. UART RX FIFO エラー ステータス レジスタのフィールドの説明

ビット	フィールド	タイプ	デフォルト	説明
7-5	RSVD	R	0h	
4	BRK	R	0h	ブレーク条件が検出されたときに設定されます。 ブレークの場合、データ バイトは 0x00 として返されます 0h = ブレーク条件ではありません 1h = ブレークです
3	FE	R	0h	このビットが設定されている場合、そのバイトはフレーミング エラー (ストップ ビット数が無効) とともに受信されたことを示します 0h = バイトのフレーミングが有効 1h = バイトのフレーミングが無効
2	PAR	R	0h	セットされている場合、そのバイトは無効なパリティ ビットで受信されたことを示します 0h = パリティ ビットが有効 1h = パリティ ビットが無効
1	NO_RX	R	0h	空の RX FIFO を読み取ると、0x00 が返され、このビットは、データが無効 / 受信されないことを示します。 これは、受信が存在しないことを示します 0h = 空 / 受信されていないバイトではありません 1h = このバイトが無効 / 受信されず、空です
0	NO_ERR	R	0h	設定されると、そのバイトがエラーなしで受信され、かつ有効なバイトであることを示します。 他のビットが設定されていない場合、このビットはセットされます 0h = 通常以外のステータスが存在します 1h = バイトの受信が成功しました

### 7.6.6 I2C のレジスタ

I2C レジスタのメモリマップされたレジスタを、[セクション 7.6.6](#) に示します。[セクション 7.6.6](#) にないレジスタ オフセット アドレスはすべて予約済みと見なして、レジスタの内容は変更しないでください。

表 7-142. I2C のレジスタ

オフセット	略称	レジスタ名	セクション
3000h	I2C_CREL	I2C コア リリース	<a href="#">セクション 7.6.6.1</a>
3001h	I2C_SCRATCH	I2C スクラッチパッド	<a href="#">セクション 7.6.6.2</a>
3002h	I2C_CTRL	I2C 制御	<a href="#">セクション 7.6.6.3</a>

**表 7-142. I2C のレジスタ (続き)**

オフセット	略称	レジスタ名	セクション
3003h	I2C_BR	I2C ボーレート	<a href="#">セクション 7.6.6.4</a>
3004h	I2C_FIFO_CTRL	I2C FIFO コントロール	<a href="#">セクション 7.6.6.5</a>
3005h	I2C_IE_0	I2C 割り込みイネーブル 0	<a href="#">セクション 7.6.6.6</a>
3006h	I2C_IE_1	I2C 割り込みイネーブル 1	<a href="#">セクション 7.6.6.7</a>
3007h	I2C_IR	I2C 割り込みステータス	<a href="#">セクション 7.6.6.8</a>
3008h	I2C_STATUS	I2C のステータス	<a href="#">セクション 7.6.6.9</a>
3009h	I2C_FS	I2C FIFO ステータス	<a href="#">セクション 7.6.6.10</a>
300Ah	I2C_RXFS	I2C RX FIFO ステータス	<a href="#">セクション 7.6.6.11</a>
300Bh	I2C_TXFS	I2C TX FIFO ステータス	<a href="#">セクション 7.6.6.12</a>
300Ch	I2C_TXES	I2C TX 素子ステータス	<a href="#">セクション 7.6.6.13</a>

表の小さなセルに収まるように、複雑なビット アクセス タイプを記号で表記しています。[セクション 7.6.6](#) に、このセクションでアクセス タイプに使用しているコードを示します。

**表 7-143. I2C のアクセス タイプ コード**

アクセス タイプ	コード	説明
読み取りタイプ		
R	R	読み出し
RH	R H	ハードウェアによってセットまたはクリアされる の読み取り
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	1 を書き込むことで セット
WP	W P	書き込み 特権アクセスが必要
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト 値

#### 7.6.6.1 I2C\_CREL レジスタ (オフセット = 3000h) [リセット = 10h]

図 7-100 に、I2C\_CREL を示し、表 7-144 に、その説明を示します。

概略表に戻ります。

I2C IP のリリースバージョン

図 7-100. I2C\_CREL レジスタ

7	6	5	4	3	2	1	0
MAJOR				MINOR			
R-1h				R-0h			

表 7-144. I2C\_CREL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-4	MAJOR	R	1h	このリリースのメジャー バージョン番号
3-0	MINOR	R	0h	このリリースのマイナー バージョン番号

#### 7.6.6.2 I2C\_SCRATCH レジスタ (オフセット = 3001h) [リセット = 00h]

図 7-101 に、I2C\_SCRATCH を示し、表 7-145 に、その説明を示します。

概略表に戻ります。

顧客スクラッチパッド レジスタ

図 7-101. I2C\_SCRATCH レジスタ

7	6	5	4	3	2	1	0
スクラッチ パッド							
R/W-0h							

表 7-145. I2C\_SCRATCH レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	スクラッチ パッド	R/W	0h	顧客スクラッチ パッド レジスタ。 これは、値の書き込みと読み取りによって通信を検証するために使用することも、一部のデータの保存に使用することもできます。 これは、リセット / 電源オン リセットによって保持されません。

### 7.6.6.3 I2C\_CTRL レジスタ (オフセット = 3002h) [リセット = 18h]

図 7-102 に、I2C\_CTRL を示し、表 7-146 に、その説明を示します。

概略表に戻ります。

このレジスタは、I2C IP を制御します

図 7-102. I2C\_CTRL レジスタ

7	6	5	4	3	2	1	0
SBR_START	AUTO_SBR_EN	RSVD	NACK_FAIL_D	NACK_FAIL_A	LSM	I2C_EN	CCE
RH/W1S-0h	R/WP-0h	R-0h	R/WP-1h	R/WP-1h	R/WP-0h	RH-0h	R/W-0h

表 7-146. I2C\_CTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	SBR_START	RH/W1S	0h	I2C スタック バス回復シーケンスを手動で生成します。 リセットが完了したときに IP によって自動的にクリア 0h = リセット パターンは生成されません 1h = I2C スタック バス回復シーケンスを開始します
6	AUTO_SBR_EN	R/WP	0h	デバイスがバス データラインのスタックを検出した場合、自動スタック バス回復シーケンスを有効にします。 0h = スタック バスが検出された場合は何もしません 1h = スタックバスが検出されると、スタック バス回復シーケンスを自動的に生成します
5	RSVD	R	0h	
4	NACK_FAIL_D	R/WP	1h	データ フィールドで NACK が受信された場合、メッセージは失敗したと考えます。 失敗したメッセージは終了処理を中断し、停止を通知します。 0h = NACK フラグに注意し、送信を続行します (メッセージはまだ成功と見なされます) 1h = 予期しないときに NACK を受信した場合、トランザクションを終了します
3	NACK_FAIL_A	R/WP	1h	アドレス (最初のバイト) の後に NACK を受信した場合、そのメッセージは失敗と見なします。 失敗したメッセージは終了処理を中断し、停止を通知します。 0h = NACK フラグに注意し、送信を続行します (メッセージはまだ成功と見なされます) 1h = 予期しないときに NACK を受信した場合、トランザクションを終了します
2	LSM	R/WP	0h	I2C 速度を遅くするため、入力クロックを 20MHz ではなく 5MHz クロックに変更 0h = 高速クロック、10MHz 1h = 低速クロック、2.5MHz (100KHz 未満)
1	I2C_EN	RH	0h	I2C IP イネーブル ステータス フラグ。 このフラグは書き込み可能ではありませんが、MRAM から IP へのメモリ割り当てによって I2C IP がイネーブルになっている場合に設定されます。 これは、 <a href="#">MRAM_IP_CFG</a> レジスタで行われます 0h = I2C IP がディセーブル、すべての I2C 機能を無効化します 1h = I2C IP を有効化
0	CCE	R/W	0h	I2C IP 変更制御イネーブル ビット。 このビットがセットされている場合、I2C IP がリセット状態に保持されます 0h = I2C 構成レジスタは書き込み保護されています 1h = I2C 構成レジスタは変更可能です

#### 7.6.6.4 I2C\_BR レジスタ (オフセット = 3003h) [リセット = 18h]

図 7-103 に、I2C\_BR を示し、表 7-147 に、その説明を示します。

概略表に戻ります。

図 7-103. I2C\_BR レジスタ

7	6	5	4	3	2	1	0
BAUD_RATE							
R/WP-18h							

表 7-147. I2C\_BR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	BAUD_RATE	R/WP	18h	I2C の 8 ビット分周器値。 入力された値より 1 つ多い値として解釈されます

## 7.6.6.5 I2C\_FIFO\_CTRL レジスタ (オフセット = 3004h) [リセット = 00h]

図 7-104 に、I2C\_FIFO\_CTRL を示し、表 7-148 に、その説明を示します。

概略表に戻ります。

図 7-104. I2C\_FIFO\_CTRL レジスタ

7	6	5	4	3	2	1	0
RX_TRG		TX_TRG		RSVD	CLR_TX	CLR_RX	RSVD
R/WP-0h		R/WP-0h		R-0h	RH/W1S-0h	RH/W1S-0h	R-0h

表 7-148. I2C\_FIFO\_CTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-6	RX_TRG	R/WP	0h	<p>RX FIFO が FIFO に格納されている指定されたバイト数に達したときの RX FIFO のトリガ レベルを設定します。</p> <p>これにより、CPU は一度に 1 バイトではなく一括でバイトをシフトアウトでき、割り込み数を保存できます</p> <p>0h = RX FIFO に 4 バイト  1h = RX FIFO に 8 バイト  2h = [(max size) - 128] バイトが RX FIFO に存在  3h = [(max size) - 64] バイトが RX FIFO に存在</p>
5-4	TX_TRG	R/WP	0h	<p>TX FIFO 内の空きバイト数が指定した値まで増加したときにトリガーするレベルを設定します。</p> <p>FIFO に一定数の空き領域ができたときに割り込みを発生させ、プロセッサが 1 バイトずつではなく複数バイトをまとめて FIFO にロードできるようにするために使用されます。</p> <p style="text-align: center;"><b>注</b></p> <p>これにより、エッジベースの割り込みと同様にトリガされます。</p> <p>ユーザーが 8 つの空き領域を決して下回ることがない場合</p> <p>0h = 16 バイトがフリー  1h = 32 バイトがフリー  2h = [(max size) - 128] バイトがフリー  3h = [(max size) - 64] バイトがフリー</p>
3	RSVD	R	0h	
2	CLR_TX	RH/W1S	0h	<p>送信 FIFO の内容をクリアします。</p> <p>クリアが完了すると、ハードウェアはビットを 0 に戻します。</p> <p>0h = 何もしません  1h = 送信 FIFO をクリアします</p>
1	CLR_RX	RH/W1S	0h	<p>受信 FIFO の内容をクリアします。</p> <p>クリアが完了すると、ハードウェアはビットを 0 に戻します。</p> <p>0h = 何もしません  1h = 受信 FIFO をクリアします</p>
0	RSVD	R	0h	



### 7.6.6.6 I2C\_IE\_0 レジスタ (オフセット = 3005h) [リセット = 00h]

図 7-105 に、I2C\_IE\_0 を示し、表 7-149 に、その説明を示します。

概略表に戻ります。

**図 7-105. I2C\_IE\_0 レジスタ**

7	6	5	4	3	2	1	0
TXLIE0	TXAIE0	DNACKIE0	ANACKIE0	SBRCIE0	RXLIE0	RXFLIE0	RXNIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-149. I2C\_IE\_0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXLIE0	R/W	0h	INT0 のフル FIFO 割り込みのイネーブルによる TX 喪失メッセージ 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
6	TXAIE0	R/W	0h	INT0 での TX 領域使用可能割り込みがイネーブルです。 トリガ レベルは、I2C_FIFO_CTRL.TX_TRG で設定されます 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
5	DNACKIE0	R/W	0h	INT0 のデータ NACK 割り込みイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
4	ANACKIE0	R/W	0h	INT0 のアドレス NACK 割り込みがイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
3	SBRCIE0	R/W	0h	スタック バス回復完了割り込み 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
2	RXLIE0	R/W	0h	INT0 の RX オーバーラン / 喪失メッセージ割り込みがイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
1	RXFLIE0	R/W	0h	INT0 の RX フィル レベル割り込みをイネーブル。 トリガ レベルは、I2C_FIFO_CTRL.RX_TRG で設定されます 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RXNIE0	R/W	0h	INT0 の RX 新規メッセージ割り込みのイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。

## 7.6.6.7 I2C\_IE\_1 レジスタ (オフセット = 3006h) [リセット = 00h]

図 7-106 に、I2C\_IE\_1 を示し、表 7-150 に、その説明を示します。

概略表に戻ります。

図 7-106. I2C\_IE\_1 レジスタ

7	6	5	4	3	2	1	0
TXLIE1	TXAIE1	DNACKIE1	ANACKIE1	SBRCIE1	RXLIE1	RXFLIE1	RXNIE1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-150. I2C\_IE\_1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXLIE1	R/W	0h	INT1 のフル FIFO 割り込みのイネーブルによる TX 喪失メッセージ 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
6	TXAIE1	R/W	0h	INT1 での TX 領域使用可能割り込みがイネーブルです。 トリガ レベルは、I2C_FIFO_CTRL.TX_TRG で設定されます 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはイネーブル。設定されるフラグにより、割り込み出力がアサートされます。
5	DNACKIE1	R/W	0h	INT1 のデータ NACK 割り込みイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
4	ANACKIE1	R/W	0h	INT1 のアドレス NACK 割り込みイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
3	SBRCIE1	R/W	0h	スタック バス回復完了割り込み 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
2	RXLIE1	R/W	0h	INT1 の RX オーバーラン / 喪失メッセージ割り込みイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
1	RXFLIE1	R/W	0h	INT1 の RX フィル レベル割り込みをイネーブル。 トリガ レベルは、I2C_FIFO_CTRL.RX_TRG で設定されます 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。
0	RXNIE1	R/W	0h	INT1 の RX 新規メッセージ割り込みのイネーブル 0h = 割り込みはデイスエーブルです。設定されているフラグは、割り込み出力をアサートしません。 1h = 割り込みはデイスエーブル。設定されるフラグにより、割り込み出力がアサートされます。

### 7.6.6.8 I2C\_IR レジスタ (オフセット = 3007h) [リセット = 00h]

図 7-107 に、I2C\_IR を示し、表 7-151 に、その説明を示します。

概略表に戻ります。

**図 7-107. I2C\_IR レジスタ**

7	6	5	4	3	2	1	0
TXL	TXA	DNACK	ANACK	SBRC	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-151. I2C\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXL	R/W1C	0h	フル FIFO 割り込みにより TX 消失メッセージ 0h = TX FIFO バイトは消失しませんでした 1h = 書き込みが試みられたときに FIFO がフルになったために TX FIFO が少なくとも1バイト (オーバーラン) を失いました
6	TXA	R/W1C	0h	TX 領域使用可能割り込み。 トリガ レベルは、I2C_FIFO_CTRL.TX_TRG 設定によって設定されます  <div style="text-align: center;"><b>注</b></div> <div style="text-align: center;">I2C IP が有効な場合、FIFO は空になるため、この割り込みがセットされます</div> 0h = TX FIFO 領域は利用可能な割り込みがありません 1h = TX FIFO がトリガ レベル スレッショルドに達しました
5	DNACK	R/W1C	0h	データ NACK 割り込み 0h = NACK が検出されませんでした 1h = データ相中に NACK が検出されました
4	ANACK	R/W1C	0h	アドレス NACK 割り込み 0h = NACK は検出されませんでした 1h = アドレス フェーズ中に NACK が検出されました
3	SBRC	R/W1C	0h	スタックバス回復完了割り込み 0h = スタック バスの回復は発生していません 1h = スタック バスの回復は完了しました。I2C_STATUS レジスタを確認して、バスがまだスタックしているかどうかを確認します。
2	RXL	R/W1C	0h	RX オーバーラン / 喪失メッセージ割り込み 0h = RX FIFO バイト消失 (オーバーランなし) 割り込みはありませんでした 1h = 全 RX FIFO (オーバーラン割り込み) のために少なくとも 1 バイトが失われました
1	RXFL	R/W1C	0h	RX フィル レベル割り込み。 トリガ レベルは I2C_FIFO_CTRL.RX_TRG に設定されました 0h = RX FIFO フィル レベル割り込みなし 1h = RX FIFO がフィル レベルに到達しました
0	RXN	R/W1C	0h	RX 新しいメッセージ割り込み 0h = 新しいバイトは受信されませんでした 1h = 新しいバイトが受信されました

### 7.6.6.9 I2C\_STATUS レジスタ (オフセット = 3008h) [リセット = XXh]

図 7-108 に、I2C\_STATUS を示し、表 7-152 に、その説明を示します。

概略表に戻ります。

図 7-108. I2C\_STATUS レジスタ

7	6	5	4	3	2	1	0
SDA_S	SCL_S	SDA_V	SCL_V	RSVD		IDLE	
RH-0h	RH-0h	RH-Xh	RH-Xh	R-0h		RH-Xh	

表 7-152. I2C\_STATUS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	SDA_S	RH	0h	SDA スタック 0h = SDA はスタックしません 1h = SDA は現在 Low にスタックしています
6	SCL_S	RH	0h	SCL スタック 0h = SCL はスタックしません 1h = SCL は現在 Low にスタックしています
5	SDA_V	RH	X	SDA 値 0h = SDA が現在 Low 状態です 1h = SDA が現在 High 状態です
4	SCL_V	RH	X	SCL 値 0h = SCL が現在 Low 状態です 1h = SCL が現在 High 状態です
3-1	RSVD	R	0h	
0	IDLE	RH	X	I2C IP がアイドルかどうか 0h = I2C IP がビジーで、トランザクションは進行中 1h = I2C IP はアイドルです

### 7.6.6.10 I2C\_FS レジスタ (オフセット = 3009h) [リセット = 00h]

図 7-109 に、I2C\_FS を示し、表 7-153 に、その説明を示します。

概略表に戻ります。

**図 7-109. I2C\_FS レジスタ**

7	6	5	4	3	2	1	0
TXFSRE	RXFD	RXFB					
RH-0h	RH-0h	RH-0h					

**表 7-153. I2C\_FS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	TXFSRE	RH	0h	<p>TX FIFO とシフト レジスタが空であるかどうかです。これは、すべてのキューに入れられたデータが送信され、コントローラがアイドル状態であることを意味します。FIFO には何もなく、シフト レジスタには何もありません。これを設定した場合、I2C コントローラを構成または無効にしても安全です</p> <hr/> <p style="text-align: center;"><b>注</b></p> <p>この場合、完全に書き込まれていない TX FIFO 素子は考慮されていません。部分的に書き込まれた TX FIFO 素子がある場合、未送信の素子は、素子が完全に書き込まれるまでアイドル フラグをクリアしません</p> <hr/> <p>0h = 少なくとも 1 つの送信中または保留中の送信が存在します。 1h = TX FIFO とシフト レジスタは空で、IP はアイドル状態です。コントローラを設定またはオフにしても安全です</p>
6	RXFD	RH	0h	<p>RX FIFO に 1 つ以上の未読バイトが含まれているかどうか 0h = RX FIFO は空です 1h = RX FIFO の少なくとも 1 つの素子</p>
5-0	RXFB	RH	0h	<p>次の RX FIFO 素子に格納された I2C フレームから読み取るべき、残りのデータバイトの数です。 有効な値は 0~63 です。 63 より大きい値は 63 として反映され、残りの値はプロセッサがデータを読み取った時点で更新されます。</p> <hr/> <p style="text-align: center;"><b>注</b></p> <p>これは、I2C フレーム内のデータ バイト数のみを指します。 RX FIFO の読み取りごとに、I2C フレーム ヘッダの 3 バイトを各 RX FIFO の読み取りに追加する必要があることに注意してください</p> <hr/>

**7.6.6.11 I2C\_RXFS レジスタ (オフセット = 300Ah) [リセット = 00h]**

図 7-110 に、I2C\_RXFS を示し、表 7-154 に、その説明を示します。

概略表に戻ります。

RX FIFO に格納されている未読 I2C メッセージの数のカウンタが含まれます。

図 7-110. I2C\_RXFS レジスタ

7	6	5	4	3	2	1	0
RX_FL							
R-0h							

表 7-154. I2C\_RXFS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	RX_FL	R	0h	RX FIFO フィル レベルは、現在 RX FIFO 内に格納されているメッセージの数を示します。 有効な値は 0~255 です。 表示される最大値は、デバイス メモリ の設定方法によって異なります。 255 が報告された場合、255 以上のメッセージが FIFO に格納されます。

### 7.6.6.12 I2C\_TXFS レジスタ (オフセット = 300Bh) [リセット = 00h]

図 7-111 に、I2C\_TXFS を示し、表 7-155 に、その説明を示します。

概略表に戻ります。

TX FIFO で使用可能なバイト数のカウンタが含まれます。2 バイトの I2C パケット ヘッダーデータを含める必要があることに注意してください

**図 7-111. I2C\_TXFS レジスタ**

7	6	5	4	3	2	1	0
TX_SA							
R-0h							

**表 7-155. I2C\_TXFS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	TX_SA	R	0h	TX FIFO フィル レベルは、現在 TX FIFO にあるフリーバイト数を示します。 255 を超える数値は 255 と表示されます。  <div style="text-align: center;"><b>注</b></div> I2C フレームに必要なバイト数には、データ バイトに加えて 2 バイトの I2C ヘッダが含まれる必要があります

### 7.6.6.13 I2C\_TXES レジスタ (オフセット = 300Ch) [リセット = 00h]

図 7-112 に、I2C\_TXES を示し、表 7-156 に、その説明を示します。

概略表に戻ります。

現在 / 次の送信 FIFO 素子に関する情報が含まれます

図 7-112. I2C\_TXES レジスタ

7	6	5	4	3	2	1	0
TXEIP							TXEBP
R-0h							R-0h

表 7-156. I2C\_TXES レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	TXEIP	R	0h	TX FIFO 素子書き込み中 TX FIFO 素子が部分的に書き込まれていることを示すために使用されます。 (ヘッダーに従って) FIFO に書き込まれた十分なデータバイトがない場合、このフラグがセットされます。 クリアされた場合、TX FIFO への次の書き込みは新しい素子となり、I2C ヘッダーフィールドから開始する必要があります 0h = TX 素子は進行中ではありません (TX FIFO への次の書き込みは新しい素子です) 1h = TX 素子は進行中で、予想されるバイト数が残っています
6-0	TXEBP	R	0h	TX 素子バイト保留 TX 素子の書き込みが開始された場合、FIFO 素子を完了するためにまだ何バイトが予想されるかを示します。  <b>注</b> これにはヘッダーバイトが含まれており、転送中のバイト数がまだ書き込まれていない場合は、値を変更できることがあります。 例えば、I2C アドレスバイトのみが書き込まれた場合、このフィールドは 1 と読み出されます。これは、転送するデータのバイト数を指定されるまで保留しているバイト数がわからないためです。 転送のバイト数が判明し、データが書き込まれると、この計算が再計算されます

### 7.6.7 I2C データ FIFO

FIFO は、I2C のデータ転送に使用されます。インターフェイスには 2 つの FIFO が使用されます

- I2C 送信 FIFO (h3010) : I2C インターフェイスに送信するデータのバイトをキューイングするために使用されます
- I2C 受信 FIFO (h3010) : I2C 読み取りから受信したバイトを読み取るために使用されます

#### 7.6.7.1 I2C 送信 FIFO (アドレス = h3010)

送信 FIFO は書き込み専用で、アドレスに対して書き込みコマンドが発行されると書き込み専用になります。フレーム全体がバッファに書き込まれると、送信が開始されます。フレームの終了後に書き込まれた (最初のヘッダで指定された) すべてのデータバイトは、新しい送信 FIFO 書き込みが開始されるまで無視されます。送信 FIFO へのすべての書き込みは、指定された I2C 送信 FIFO アドレスのみに対して行う必要があります。書き込まれた各バイトは、自動的に FIFO にシフトされます。例えば、1 つの CAN メッセージで転送できるより大きな I2C フレームを書き込む場合、同じアドレスへの複数回の CAN 書き込みによって処理されます。これは、SPI ヘッダバイトがデバイスに受信すべきデータバイト数を知らせるためです。



TX バッファ内の各フレームのサイズは、目的の I2C フレームの長さによって決まります。各 I2C FIFO 素子の先頭には、必要な 2 バイトのヘッダがあります。TX FIFO 内のフレームの合計長は、以下の合計です

- I2C ヘッダ用 2 バイト
- データ バイトのメッセージ長 (フレームが I2C 書き込みの場合。I2C 読み取り専用では、ヘッダで読み取るバイト数を必要とし、追加のバイトは無視されます)

詳細と例については、[セクション 7.5.6.3](#) を参照してください。

注

送信 FIFO がフルの場合、書き込まれた新しいメッセージはすべて破棄され、割り込みがセットされます。



図 7-113. I2C 書き込み TX FIFO ヘッダ (ヘッダおよびデータのみ)

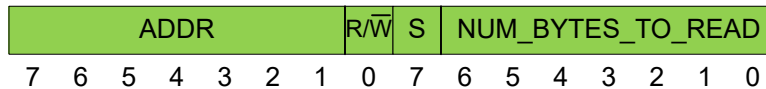


図 7-114. I2C 読み出し TX FIFO ヘッダ (ヘッダのみ)

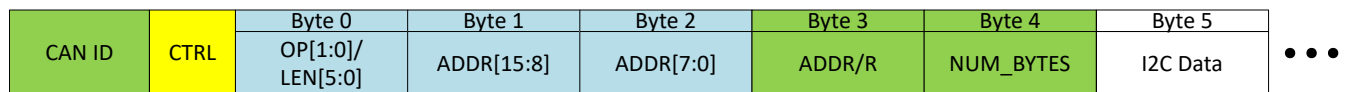


図 7-115. I2C 書き込み TX FIFO ヘッダの例 (標準 CAN フレーム フォーマット)

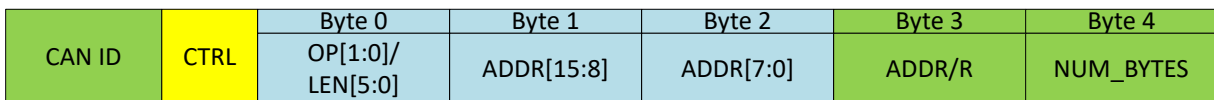


図 7-116. I2C 読み取り TX FIFO ヘッダの例 (標準 CAN フレーム フォーマット)

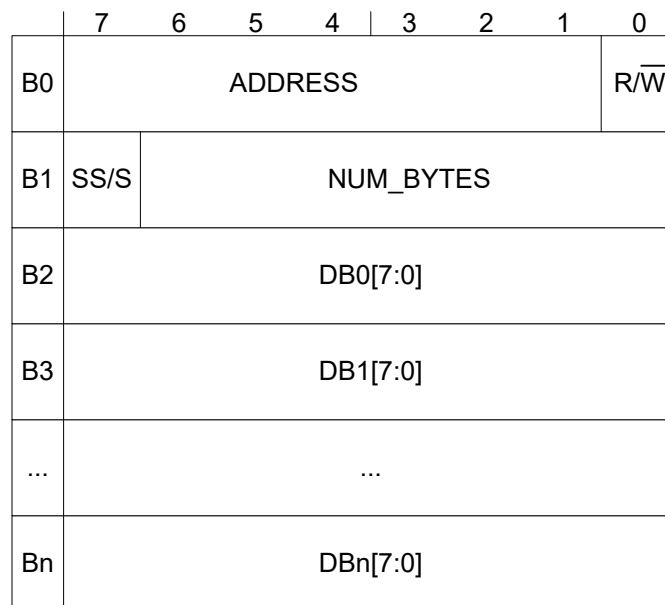


図 7-117. I2C 送信 FIFO

表 7-157. I2C 送信 FIFO の説明

バイト	ビット	フィールド	タイプ	リセット	説明
0	7:1	アドレス	W	0x0	I2C ターゲット デバイス アドレス
	0	R/W	W	0	I2C メッセージが I2C 読み取りまたは I2C 書き込み 0 = I2C 書き込み 1 = I2C 読み取り
1	7	S (I2C 読み取り) SS (I2C 書き込み)	W	0	格納 (I2C 読み取り) または格納ステータス (I2C 書き込み) I2C 読み取りの場合、受信したデータを受信 FIFO に格納します。I2C 書き込みの場合、ターゲット デバイスからデータが受信されないため、ユーザーは、I2C 書き込みのステータス フレームを保存するか、または成功したと仮定するかを選択できます (受信 FIFO にステータスを保存しない) 0 = このトランザクションのデータ / ステータスを受信 FIFO に格納しない 1 = 受信 FIFO にデータまたは書き込みステータスを保存するかを選択します
	6:0	NUM_BYTES_TO_WRITE/READ	W	0x0	I2C 書き込み (B0.R/W = 0) : アドレスの後に続く I2C フレーム内のバイト数 (上記ヘッダ バイトは含みません) 有効な値 0 ~ 127。0 はデータを送信せず、デバイス アドレスのみを送信します (ACK のチェックに便利) I2C 読み出し (B0.R/W = 1) : アドレス バイトの後に読み取るバイト数 (上記のヘッダ バイトは含みません)

### 7.6.7.2 I2C 受信 FIFO (アドレス = h3010)

受信 FIFO は読み取り専用です。

RX バッファ内の各フレームのサイズは、I2C フレームのデータ長によって決まります。バッファ内の RX フレームの全長は、以下の合計です

- I2C ヘッダー用 2 バイト
- I2C ステータス用 1 バイト
- データ バイトのメッセージ長

RX FIFO からの読み取りは、FIFO 素子内のすべてのデータ バイトが読み出された時点で完了と見なされ、解放されます。読み取りの残りのバイト数を超えるバイトは、無効なバイトに対して 0x00 を返します。

RX FIFO からのフレームの一部読み取りは、自動的に次回の I2C 受信 FIFO 読み取りに継続されます。その際、CONT ビットが 1 に設定され、この I2C フレーム内で残り読み取り可能なデータバイト数が NUM\_BYTES\_REMAINING に反映されることで示されます。

空の RX バッファから読み出した場合、チャンネル 0 の 0 NUM\_BYTES\_REMAINING が返されます。フレームは常に受信された順序で読み取られます。

RX バッファは、I2C クリア RX FIFO ビット (I2C\_FIFO\_CTRL[1]) に 1 を書き込むことによってクリアできます (すべてのフレームが破棄されます)。

#### 注

受信 FIFO がフルの場合、すべての新しい受信メッセージは破棄され、メッセージが失われたことをプロセッサに通知する割り込みが設定されます。

詳細と例については、[セクション 7.5.6.3](#) を参照してください。



図 7-118. I2C RX FIFO 読み取りヘッダー (ヘッダーおよびデータのみ)

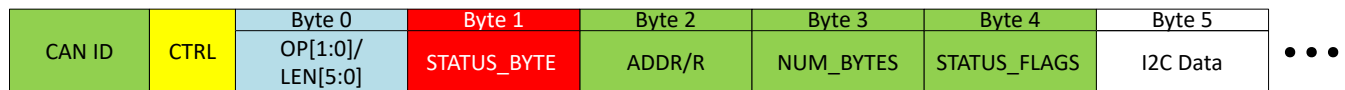


図 7-119. I2C RX FIFO ヘッダーの例 (標準 CAN フレーム フォーマット)

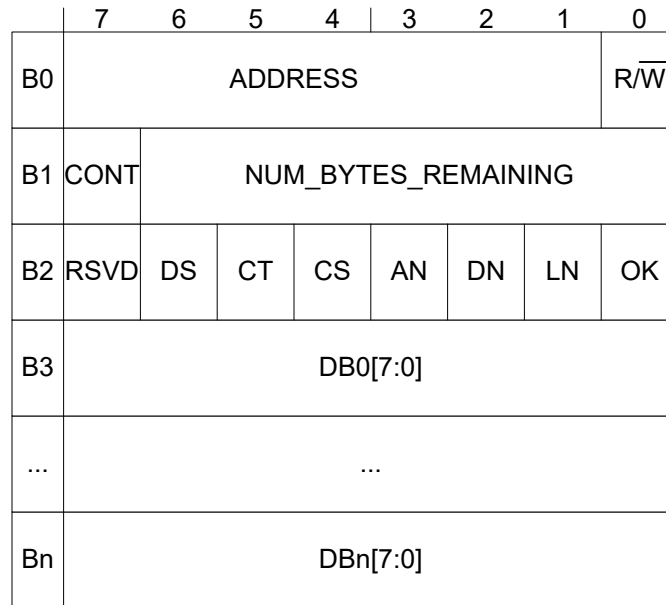


図 7-120. I2C 受信 FIFO フレーム フォーマット

表 7-158. I2C 受信 FIFO バイトの説明

バイト	ビット	フィールド	タイプ	リセット	説明
0	7:1	アドレス	R	0x0	I2C ターゲット デバイス アドレス
	0	W/R	R	0	I2C メッセージが I2C 読み取りまたは I2C 書き込みのどちらか 0 = I2C 読み取り 1 = I2C 書き込み
1	7	CONT	R	0	連続読み取り 0 = この読み取りはフレームの先頭から得られます 1 = この読み取りは前の読み取りから続行されます
	6:0	NUM_BYTES_REMAINING	R	0x0	I2C 読み取り (B0.W/R = 0): アドレスバイトの後に読み取るバイト数 (上記のヘッダ バイトは何も含まれません) I2C 書き込み (B0.W/R = 1) : アドレスの後に続く I2C フレーム内のバイト数 (上記ヘッダ バイトは含まれません) 有効な値 0 ~ 127。0 はデータを送信せず、デバイス アドレスのみを送信します (ACK の確認に便利)

**表 7-158. I2C 受信 FIFO バイトの説明 (続き)**

バイト	ビット	フィールド	タイプ	リセット	説明
2	7	RSVD	R	0	予約済み
	6	DS	R	0	データ スタック バスが停止している (クロックまたはデータが解放されて いません)
	5	CT	R	0	クロック タイムアウト SMBus クロック タイムアウトが検出されました
	4	CS	R	0	クロック ストレッチ このフレーム中にクロックが発生しました
	3	AN	R	0	アドレス NACK I2C ターゲット デバイスは、アドレス バイトに対して NACK を返します
	2	DN	R	0	データ NACK 書き込み中、I2C ターゲット デバイスは少なくとも 1 バイ トで NACK を返します。
	1	LN	R	0	最後のバイト NACK I2C ターゲット デバイスは、I2C フレームの最後のバイトに NACK を返しました
	0	OK	R	0	成功 I2C フレームが正常に完了しました

### 7.6.8 PWM0 のレジスタ

PWM0 レジスタのメモリマップされたレジスタを、[セクション 7.6.8](#) に示します。[セクション 7.6.8](#) がないレジ  
スタ オフセット アドレスはすべて予約済みと見なして、レジスタの内容は変更しないでください。

**表 7-159. PWM0 のレジスタ**

オフセット	略称	レジスタ名	セクション
4000h	PWM1_ACTION		<a href="#">セクション 7.6.8.1</a>
4000h	PWM0_CTRL	PWM0 制御	<a href="#">セクション 7.6.8.2</a>
4001h	PWM0_IE0	割り込みイネーブル	<a href="#">セクション 7.6.8.3</a>
4002h	PWM0_IE1	割り込みイネーブル	<a href="#">セクション 7.6.8.4</a>
4003h	PWM0_IR	割り込みレジスタ	<a href="#">セクション 7.6.8.5</a>
4004h	PWM0_STATUS	ステータス	<a href="#">セクション 7.6.8.6</a>
4005h + 式	PWM0_CUR_PULSE[y]	現在のパルス カウント	<a href="#">セクション 7.6.8.7</a>
4009h	PWM0_CUR_VAL_MSB	現在の値 MSB	<a href="#">セクション 7.6.8.8</a>
400Ah	PWM0_CUR_VAL_LSB	現在の値 LSB	<a href="#">セクション 7.6.8.9</a>
400Bh	PWM0_CONST_MSB	定数整数 MSB	<a href="#">セクション 7.6.8.10</a>
400Ch	PWM0_CONST_LSB	定数整数 LSB	<a href="#">セクション 7.6.8.11</a>
400Dh	PWM0_STOP_VAL_FRAC_F	停止分数値 (周波数のみ)	<a href="#">セクション 7.6.8.12</a>
400Eh	PWM0_STOP_VAL_MSB	ストップ値整数 MSB	<a href="#">セクション 7.6.8.13</a>
400Fh	PWM0_STOP_VAL_LSB	ストップ値整数 LSB	<a href="#">セクション 7.6.8.14</a>
4010h	PWM0_STOP_SL_MSB	ストップ スロープ MSB	<a href="#">セクション 7.6.8.15</a>
4011h	PWM0_STOP_SL_MID	ストップ スロープ 中間	<a href="#">セクション 7.6.8.16</a>
4012h	PWM0_STOP_SL_LSB	ストップ スロープ LSB	<a href="#">セクション 7.6.8.17</a>
4013h	PWM0_START_VAL_FRAC_F	開始分数値 (周波数のみ)	<a href="#">セクション 7.6.8.18</a>
4014h	PWM0_START_VAL_MSB	開始値 MSB	<a href="#">セクション 7.6.8.19</a>
4015h	PWM0_START_VAL_LSB	開始値 LSB	<a href="#">セクション 7.6.8.20</a>
4016h	PWM0_START_SL_MSB	開始スロープ MSB	<a href="#">セクション 7.6.8.21</a>
4017h	PWM0_START_SL_MID	開始スロープ 中間バイト	<a href="#">セクション 7.6.8.22</a>
4018h	PWM0_START_SL_LSB	開始スロープ LSB	<a href="#">セクション 7.6.8.23</a>

表 7-159. PWM0 のレジスタ (続き)

オフセット	略称	レジスタ名	セクション
4019h	PWM0_END_VAL_CONST_FRAC_F	終了 (Freq) または制御 (DC) 分数の値	<a href="#">セクション 7.6.8.24</a>
401Ah	PWM0_END_VAL_MSB	終了値 MSB	<a href="#">セクション 7.6.8.25</a>
401Bh	PWM0_END_VAL_LSB	終了値 LSB	<a href="#">セクション 7.6.8.26</a>
401Ch + 式	PWM0_PULSE_STOP_RAMP[y]	ストップ ランプを開始するためのパルス カウント	<a href="#">セクション 7.6.8.27</a>
4020h + 式	PWM0_PULSE_MAX[y]	最大パルス カウント	<a href="#">セクション 7.6.8.28</a>
4024h	PWM0_ACTION	開始アクション レジスタ	<a href="#">セクション 7.6.8.29</a>
4030h	PWM0_IAS_CTRL	入力自動停止制御	<a href="#">セクション 7.6.8.30</a>

表の小さなセルに収まるように、複雑なビット アクセス タイプを記号で表記しています。[セクション 7.6.8](#) に、このセクションでアクセス タイプに使用しているコードを示します。

表 7-160. PWM0 のアクセス タイプ コード

アクセス タイプ	コード	説明
読み取りタイプ		
R	R	読み出し
RH	R H	ハードウェアによってセットまたはクリアされる の読み取り
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	1 を書き込むことで セット
WP	W P	書き込み 特権アクセスが必要
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ アレイ変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタグループの一部であるレジスタアレイの値を示します。レジスタグループは階層構造を形成し、アレイは式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタアレイの値を示します。

### 7.6.8.1 PWM1\_ACTION レジスタ (オフセット = 4000h) [リセット = 00h]

PWM1\_ACTION を表 7-161 に示します。

[概略表](#)に戻ります。

**表 7-161. PWM1\_ACTION レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
4	COUNT_RST	R/W1S	0h	<p>パルスカウンタを 0 にリセットする このビットが設定されると、パルスカウンタはリセットされます。 これは通常、パルスの合計数を追跡するために、ランプの開始時に実行されます。 リセットが完了すると、ハードウェアによってクリアされます。</p> <hr/> <p style="text-align: center;"><b>注</b></p> <p>AUTO_STOP を使用する場合は、STOP および最大パルス カウントの値に注意してください。 これらのパルス カウンタは現在のパルス カウンタと比較され、最大パルスカウンタが現在のパルス カウンタ値よりも小さい場合、PWM 出力は生成されません</p>
3-0	予約済み	R	0h	

### 7.6.8.2 PWM0\_CTRL レジスタ (オフセット = 4000h) [リセット = 11h]

図 7-121 に、PWM0\_CTRL を示し、表 7-162 に、その説明を示します。

概略表に戻ります。

PWM0 モジュールを設定するための制御フィールド

図 7-121. PWM0\_CTRL レジスタ

7	6	5	4	3	2	1	0
RSVD	SLOPE_SCALE		モード		DC_8B		INIT
R-0h	R/WP-1h		R/WP-0h		R/WP-0h		R/W-1h

表 7-162. PWM0\_CTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-4	SLOPE_SCALE	R/WP	1h	<p>スロープ スケール スロープ レジスタの整数部分と分数部分に使用されるビット数を調整します。 周波数がデューティ サイクルのどちらをランプさせるかによって、列挙値の意味が変わります。 詳細については「<a href="#">セクション 7.5.7.4</a>」を参照してください  0h = Int[20 ~ 16], Frac[15-0] (周波数) - Int[20], Frac[19-0] (デューティ サイクル)  1h = Int[20 ~ 12], Frac[11-0] (周波数) - Int[20-19], Frac[18-0] (デューティ サイクル)  2h = Int[20 ~ 8], Frac[7-0] (周波数) - Int[20-18], Frac[17-0] (デューティ サイクル)  3h = Int[20 ~ 4], Frac[3-0] (周波数) - Int[20-17], Frac[16-0] (デューティ サイクル)  4h = Int[20-0], 分数なし (周波数) - Int[20-16], Frac[15-0] (デューティ サイクル)</p>
3-2	モード	R/WP	0h	<p>スロープ レジスタの整数部分と分数部分に使用されるビット数を調整します。 GPIO が特殊機能用に設定されている場合、モードを off 以外に設定すると、SPI チップセレクトよりも優先されることに注意してください。  0h = オフ  1h = デューティ サイクル ランプ  2h = 周波数ランプ  3h = 静的オン</p>
1	DC_8B	R/WP	0h	<p>8 ビット デューティ サイクル モード  0h = 10 ビットのデューティ サイクル分解能  1h = 8 ビットのデューティ サイクル分解能</p>
0	INIT	R/W	1h	<p>初期化モード PWM IP がリセット状態に保持され、この PWM0_CTRL レジスタの設定に使用されます  0h = 初期化モードはディスエーブルです。保護された PWM ビットへの書き込みは許可されません  1h = 初期化モードはイネーブルです。保護された PWM ビットへの書き込みが許可されます</p>



### 7.6.8.3 PWM0\_IE0 レジスタ (オフセット = 4001h) [リセット = 00h]

図 7-122 に、PWM0\_IE0 を示し、表 7-163 に、その説明を示します。

概略表に戻ります。

**図 7-122. PWM0\_IE0 レジスタ**

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RC_IE
R-0h					R/W-0h	R/W-0h	R/W-0h

**表 7-163. PWM0\_IE0 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF_IE	R/W	0h	電流パルス オーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定されます 0h = 現在のパルス カウンタはオーバーフローしていません 1h = 現在のパルス カウンタがオーバーフローしました
1	IAS_IE	R/W	0h	入力自動停止割り込み 入力自動停止条件が発生したときにセットされます。PWM が即時停止に設定されている場合、出力も停止します (RC ビットに反映されます)。 PWM が IAS トリガでストップ ランプを開始するように設定されている場合、この割り込みは入力条件が発生した時点で直ちにセットされ、出力がオフになった時点で RC ビットがセットされます。 0h = IAS の入力はトリガされていません 1h = IAS の入力がトリガされました
0	RC_IE	R/W	0h	PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。 AS_EN = 0 の場合、PWM が終了値に達すると設定されます。 IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止するとこれが設定されます。 ストップ ランプが手動でアクティブになった場合、ランプが完了し、出力がオフになると RC ビットがセットされます。  <div style="text-align: center;"><b>注</b></div> <div style="text-align: center;">オンランプの実行中で、STOP 条件が発生した場合、RC ビットはオンランプに設定されず、STOP 条件のために出力が停止するとのみ一度だけです</div> 0h = PWM チャネル ランプは完了していません 1h = PWM チャネル ランプが完了しています

## 7.6.8.4 PWM0\_IE1 レジスタ (オフセット = 4002h) [リセット = 00h]

図 7-123 に、PWM0\_IE1 を示し、表 7-164 に、その説明を示します。

概略表に戻ります。

図 7-123. PWM0\_IE1 レジスタ

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RC_IE
R-0h					R/W-0h	R/W-0h	R/W-0h

表 7-164. PWM0\_IE1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF_IE	R/W	0h	電流パルス オーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定されます 0h = 現在のパルス カウンタはオーバーフローしていません 1h = 現在のパルス カウンタがオーバーフローしました
1	IAS_IE	R/W	0h	入力自動停止割り込み 入力自動停止条件が発生したときにセットされます。PWM が即時停止に設定されている場合、出力も停止します (RC ビットに反映されます)。 PWM が IAS トリガでストップ ランプを開始するように設定されている場合、この割り込みは入力条件が発生した時点で直ちにセットされ、出力がオフになった時点で RC ビットがセットされます。 ストップ ランプが手動でアクティブになった場合、ランプが完了し、出力がオフになると RC ビットがセットされます。 0h = IAS の入力はトリガされていません 1h = IAS の入力がトリガされました
0	RC_IE	R/W	0h	PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。 AS_EN = 0 の場合、PWM が終了値に達すると設定されます。 IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止すると、設定されます。 ストップ ランプが手動でアクティブになると、ランプが完了して出力がオフになると、RC ビットがセットされます。  <b>注</b> オンランプの実行中で、STOP 条件が発生した場合、RC ビットはオンランプに設定されず、STOP 条件のために出力が停止するとのみ一度だけです  0h = PWM チャネル ランプは完了していません 1h = PWM チャネル ランプが完了しています

### 7.6.8.5 PWM0\_IR レジスタ (オフセット = 4003h) [リセット = 00h]

図 7-124 に、PWM0\_IR を示し、表 7-165 に、その説明を示します。

概略表に戻ります。

**図 7-124. PWM0\_IR レジスタ**

7	6	5	4	3	2	1	0
RSVD				PULSE_OVF		IAS	RC
R-0h				R/W1C-0h		R/W1C-0h	R/W1C-0h

**表 7-165. PWM0\_IR レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF	R/W1C	0h	電流パルス オーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定されます 0h = 現在のパルス カウンタはオーバーフローしていません 1h = 現在のパルス カウンタがオーバーフローしました
1	IAS	R/W1C	0h	入力自動停止割り込み 入力自動停止条件が発生したときにセットされます。PWM が即時停止に設定されている場合、出力も停止します (RC ビットに反映されます)。 PWM が IAS トリガでストップ ランプを開始するように設定されている場合、この割り込みは入力条件が発生した時点で直ちにセットされ、出力がオフになった時点で RC ビットがセットされます。 0h = IAS の入力はトリガされていません 1h = IAS の入力トリガされました
0	RC	R/W1C	0h	PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。 AS_EN = 0 の場合、PWM が終了値に達すると設定されます。 IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止すると、設定されます。 ストップ ランプが手動でアクティブになると、ランプが完了して出力がオフになると、RC ビットがセットされます。  <div style="text-align: center;"><b>注</b></div> <div style="text-align: center;">オンランプの実行中で、STOP 条件が発生した場合、RC ビットはオンランプに設定されず、STOP 条件のために出力が停止するとのみ一度だけです</div> 0h = PWM チャネル ランプは完了していません 1h = PWM チャネル ランプが完了しています

### 7.6.8.6 PWM0\_STATUS レジスタ (オフセット = 4004h) [リセット = 00h]

図 7-125 に、PWM0\_STATUS を示し、表 7-166 に、その説明を示します。

概略表に戻ります。

図 7-125. PWM0\_STATUS レジスタ

7	6	5	4	3	2	1	0
RSVD				IAS_ACT		BUSY	OEN
R-0h				RH-0h		RH-0h	RH-0h

表 7-166. PWM0\_STATUS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	IAS_ACT	RH	0h	入力自動停止アクティブ GPIO 入力のアサート状態にトグルすると設定され、PWM 出力が停止すると自動的にクリアされます。 ランプが完了すると、RC 割り込みビットをセットします 0h = 入力の自動ストップはアクティブではありません 1h = 入力自動停止がアクティブです
1	BUSY	RH	0h	ランプ IP がビジーです 0h = PWM ランプが完了しています 1h = PWM ランプは現在進行中です
0	OEN	RH	0h	出力イネーブル 0h = PWM 出力は無効です 1h = PWM 出力はイネーブルで、出力を生成しています

#### 7.6.8.7 PWM0\_CUR\_PULSE[y] レジスタ (オフセット = 4005h + 式) [リセット = 00000000h]

図 7-126 に PWM0\_CUR\_PULSE[y] の制御を示し、表 7-167 にその説明を示します。

概略表に戻ります。

現在のパルス カウントを示す符号なし 32 ビット カウンタ値です。最下位アドレスはカウンタの MSB に対応します。

オフセット = 4005h + (y \* 4h)、ここで y = 0h ~ 3h

図 7-126. PWM0\_CUR\_PULSE[y] レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_CUR_PULSE																															
R-0h																															

表 7-167. PWM0\_CUR\_PULSE[y] レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	PWM0_CUR_PULSE	R	0h	電流ランプの電流パルス数。 ランプが完了した場合、この値は、最後のランプに対して生成されたパルスの合計数になります。 この値は、新しいランプが開始されるたびにリセットされます。 最下位アドレスは MSB です。

**7.6.8.8 PWM0\_CUR\_VAL\_MSB レジスタ (オフセット = 4009h) [リセット = 00h]**

図 7-127 に、PWM0\_CUR\_VAL\_MSB を示し、表 7-168 に、その説明を示します。

概略表に戻ります。

**図 7-127. PWM0\_CUR\_VAL\_MSB レジスタ**

7	6	5	4	3	2	1	0
RSVD					CUR_VAL[10:8]		
R-0h					R-0h		

**表 7-168. PWM0\_CUR\_VAL\_MSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2-0	CUR_VAL[10:8]	R	0h	

#### 7.6.8.9 PWM0\_CUR\_VAL\_LSB レジスタ (オフセット = 400Ah) [リセット = 00h]

図 7-128 に、PWM0\_CUR\_VAL\_LSB を示し、表 7-169 に、その説明を示します。

概略表に戻ります。

**図 7-128. PWM0\_CUR\_VAL\_LSB レジスタ**

7	6	5	4	3	2	1	0
CUR_VAL[7:0]							
R-0h							

**表 7-169. PWM0\_CUR\_VAL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	CUR_VAL[7:0]	R	0h	

#### 7.6.8.10 PWM0\_CONST\_MSB レジスタ (オフセット = 400Bh) [リセット = 00h]

図 7-129 に、PWM0\_CONST\_MSB を示し、表 7-170 に、その説明を示します。

概略表に戻ります。

周波数を可変 / ランプさせるように設定されている場合、これは静的デューティ サイクルの MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合です)。デューティ サイクルを可変 / ランプさせるように設定されている場合、これはスイッチング周波数の固定整数除数の MSB として使用されます。静的 PWM 出力に設定されている場合、これはデューティ サイクルの MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合です)。

図 7-129. PWM0\_CONST\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD					CV_F[10]	CV[9:8]	
R-0h					R/W-0h	R/W-0h	

表 7-170. PWM0\_CONST\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	CV_F[10]	R/W	0h	一定値 (周波数のみ) これは多目的レジスタです。 PWM0_CTRL.MODE = デューティ サイクル。この場合、スイッチング周波数分周器 CONST_FREQ[10] の整数値になります。 PWM0_CTRL.MODE = 周波数または静的の場合、このビットは使用されません
1-0	CV[9:8]	R/W	0h	定数値 これは多目的レジスタです PWM 0_CTRL.MODE = デューティ サイクルの場合、スイッチング周波数除数の整数値 (CONST_FREQ[9:8]) となります。 PWM 0_CTRL.MODE = (周波数または静的) かつ PWM 0_CTRL.MODE = 0 (10 ビット) の場合、デューティ サイクル値 (CONST_DC[9:8]) (CONST_DC/1024 = デューティ サイクル) となります。 8 ビット分解能モードでは、これは使われません。



#### 7.6.8.11 PWM0\_CONST\_LSB レジスタ (オフセット = 400Ch) [リセット = 00h]

図 7-130 に、PWM0\_CONST\_LSB を示し、表 7-171 に、その説明を示します。

概略表に戻ります。

周波数を可変 / ランプさせるように設定されている場合、これは静的デューティ サイクルの LSB として使用されます。デューティ サイクルを可変 / ランプさせるように設定されている場合、これはスイッチング周波数の固定整数除数の LSB として使用されます。静的 PWM 出力に設定されている場合、これはデューティ サイクルの LSB として使用されます。

**図 7-130. PWM0\_CONST\_LSB レジスタ**

7	6	5	4	3	2	1	0
CV[7:0]							
R/W-0h							

**表 7-171. PWM0\_CONST\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	CV[7:0]	R/W	0h	<p>定数値 これは多目的レジスタです。</p> <p>PWM 0_CTRL.MODE = Duty Cycle の場合、スイッチング周波数除数の整数値 CONST_FREQ[7:0] となります。</p> <p>PWM 0_CTRL.MODE = 周波数または静的の場合、これはデューティサイクル除数 CONST_DC[7:0] です。</p> <p>8 ビットモードの場合 、CONST_DC/256 = デューティ サイクルです。</p> <p>10 ビット モードの場合、 CONST_DC/1024 = デューティ サイクルです</p>

#### 7.6.8.12 PWM0\_STOP\_VAL\_FRAC\_F レジスタ (オフセット = 400Dh) [リセット = 00h]

図 7-131 に、PWM0\_STOP\_VAL\_FRAC\_F を示し、表 7-172 に、その説明を示します。

概略表に戻ります。

周波数を可変 / ランプ設定にしている場合、これはストップまたはオフ ターゲット時の周波数分数除数として使用されます。これは、デューティ サイクルを上昇させるように構成されている場合や静的 PWM には使用されません。

図 7-131. PWM0\_STOP\_VAL\_FRAC\_F レジスタ

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

表 7-172. PWM0\_STOP\_VAL\_FRAC\_F レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	ストップ値分数値 (周波数のみ) PWM0_CTRL.MODE = 周波数の場合にのみ使用されます。 周波数除数の x/128 分数部分と解釈されます。 PWM0_CTRL.MODE = デューティ サイクルまたは静的の場合、これは使われません。

### 7.6.8.13 PWM0\_STOP\_VAL\_MSB レジスタ (オフセット = 400Eh) [リセット = 00h]

図 7-132 に、PWM0\_STOP\_VAL\_MSB を示し、表 7-173 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクル MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合のみ使用されます)。周波数を可変 / ランプさせるように設定されている場合、これは周波数のストップ / オフターゲット整数除数の MSB として使用されません。これは、静的 PWM として構成されている場合は使用されません

**図 7-132. PWM0\_STOP\_VAL\_MSB レジスタ**

7	6	5	4	3	2	1	0
RSVD					SPV_F[10]	SPV[9:8]	
R-0h					R/W-0h	R/W-0h	

**表 7-173. PWM0\_STOP\_VAL\_MSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	SPV_F[10]	R/W	0h	停止値 (周波数のみ) これは多目的レジスタです。 PWM0_CTRL.MODE = 周波数の場合、スイッチング周波数除数 STOP_VAL[10] の整数値です。 PWM0_CTRL.MODE = デューティ サイクルの場合、このビットは使用されません。 PWM0_CTRL.MODE = 静的の場合、これは使われません。
1-0	SPV[9:8]	R/W	0h	ストップ値 これは多目的レジスタです。 PWM0_CTRL.MODE = デューティ サイクルかつ PWM0_CTRL.MODE = 周波数の場合、これはスイッチング周波数除数 STOP_VAL[9:8] の整数値です。それ以外の場合は、この値は使用されません。 SPV[9:8] = デューティ サイクル値 SPV[9:8] / 1024 = デューティ サイクルで、8 ビット モードでは使用されません。 PWM0_CTRL.MODE = 周波数の場合、これはスイッチング周波数除数 STOP_VAL[9:8] の整数値です。 PWM0_CTRL.MODE = 静的の場合、これは使われません。

#### 7.6.8.14 PWM0\_STOP\_VAL\_LSB レジスタ (オフセット = 400Fh) [リセット = 00h]

図 7-133 に、PWM0\_STOP\_VAL\_LSB を示し、表 7-174 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクルの LSB として使用されます。周波数を可変 / ランプさせるように設定されている場合、これは周波数のストップ / オフターゲット整数除数の LSB として使用されます。これは、静的 PWM として構成されている場合は使用されません

図 7-133. PWM0\_STOP\_VAL\_LSB レジスタ

7	6	5	4	3	2	1	0
SPV[7:0]							
R/W-0h							

表 7-174. PWM0\_STOP\_VAL\_LSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	SPV[7:0]	R/W	0h	<p>ストップ値 これは多目的レジスタです。</p> <p>PWM 0_CTRL.MODE = デューティ サイクル、これはデューティ サイクル値 SPV[7:0] です。 8 ビット モードの場合、SPV/256 = デューティ サイクルです。 10 ビット モードの場合、SPV/1024 = デューティ サイクルです。</p> <p>PWM 0_CTRL.MODE = 周波数の場合、スイッチング周波数除数 STOP_VAL[7:0] の整数値です。 PWM 0_CTRL.MODE = 静的の場合、これは使われません。</p>

#### 7.6.8.15 PWM0\_STOP\_SL\_MSB レジスタ (オフセット = 4010h) [リセット = 00h]

図 7-134 に、PWM0\_STOP\_SL\_MSB を示し、表 7-175 に、その説明を示します。

概略表に戻ります。

ストップ ランプのスロープ スケールを設定するために使用します

図 7-134. PWM0\_STOP\_SL\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD				SP_SL[20:16]			
R-0h				R/W-0h			

表 7-175. PWM0\_STOP\_SL\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4-0	SP_SL[20:16]	R/W	0h	ストップ スロープ ストップ ランプのスロープです。 フィールドは、PWM0_CTRL.SLOPE_SCALE の値によって異なります。 PWM0_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

**7.6.8.16 PWM0\_STOP\_SL\_MID レジスタ (オフセット = 4011h) [リセット = 00h]**

図 7-135 に、PWM0\_STOP\_SL\_MID を示し、表 7-176 に、その説明を示します。

概略表に戻ります。

ストップ ランプのスロープ スケールを設定するために使用します

図 7-135. PWM0\_STOP\_SL\_MID レジスタ

7	6	5	4	3	2	1	0
SP_SL[15:8]							
R/W-0h							

表 7-176. PWM0\_STOP\_SL\_MID レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	SP_SL[15:8]	R/W	0h	ストップ スロープ ストップ ランプのスロープです。 フィールドは、PWM0_CTRL.SLOPE_SCALE の値によって異なります。 PWM0_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

#### 7.6.8.17 PWM0\_STOP\_SL\_LSB レジスタ (オフセット = 4012h) [リセット = 00h]

図 7-136 に、PWM0\_STOP\_SL\_LSB を示し、表 7-177 に、その説明を示します。

概略表に戻ります。

ストップ ランプのスロープ スケールを設定するために使用します

図 7-136. PWM0\_STOP\_SL\_LSB レジスタ

7	6	5	4	3	2	1	0
SP_SL[7:0]							
R/W-0h							

表 7-177. PWM0\_STOP\_SL\_LSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	SP_SL[7:0]	R/W	0h	ストップ スロープ ストップ ランプのスロープです。 フィールドは、PWM0_CTRL.SLOPE_SCALE の値によって異なります。 PWM0_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、セクション 7.5.7.4 を参照してください。

**7.6.8.18 PWM0\_START\_VAL\_FRAC\_F レジスタ (オフセット = 4013h) [リセット = 00h]**

図 7-137 に、PWM0\_START\_VAL\_FRAC\_F を示し、表 7-178 に、その説明を示します。

概略表に戻ります。

変動 / ランプ周波数に設定されている場合、周波数の開始分数除数に使用されます。これは、デューティ サイクルを上昇させるように構成されている場合や静的 PWM には使用されません。

**図 7-137. PWM0\_START\_VAL\_FRAC\_F レジスタ**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-178. PWM0\_START\_VAL\_FRAC\_F レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	開始値分数値 (周波数のみ) PWM0_CTRL.MODE = 周波数の場合にのみ使用されます。 周波数除数の x/128 分数部分と解釈されます。 PWM0_CTRL.MODE = デューティ サイクルまたは静的の場合、これは使われません。



#### 7.6.8.19 PWM0\_START\_VAL\_MSB レジスタ (オフセット = 4014h) [リセット = 00h]

図 7-138 に、PWM0\_START\_VAL\_MSB を示し、表 7-179 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクル MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合のみ使用されます)。周波数を変化 / ランプ動作させるように設定した場合、この値は周波数の開始整数分周値の MSB として使用されます。これは、静的 PWM として構成されている場合は使用されません

**図 7-138. PWM0\_START\_VAL\_MSB レジスタ**

7	6	5	4	3	2	1	0
RSVD					STV_F[10]	STV[9:8]	
R-0h					R/W-0h	R/W-0h	

**表 7-179. PWM0\_START\_VAL\_MSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	STV_F[10]	R/W	0h	開始値 (周波数のみ) これは多目的レジスタです。 PWM0_CTRL.MODE = 周波数の場合、スイッチング周波数除数 START_VAL[10] の整数値です。 PWM0_CTRL.MODE = デューティ サイクルの場合、このビットは使用されません。 PWM0_CTRL.MODE = 静的の場合、これは使用しません。
1-0	STV[9:8]	R/W	0h	開始値 これは多目的レジスタです。 PWM 0_CTRL.MODE = デューティ サイクルかつ PWM 0_CTRL の場合、 8B = 0 ( 10 ビット)。これはデューティ サイクル値 STV[9 : 8] です。それ以外の場合は、この値は使用されません。 STV/ 1024= デューティサイクルであり、 8 ビット モードでは使用されません。 PWM 0_CTRL.MODE = 周波数の場合、スイッチング周波数除数 START_VAL[ 9: 8] の整数値です。 PWM 0_CTRL.MODE = 静的の場合、これは使われません。

#### 7.6.8.20 PWM0\_START\_VAL\_LSB レジスタ (オフセット = 4015h) [リセット = 00h]

図 7-139 に、PWM0\_START\_VAL\_LSB を示し、表 7-180 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクルの LSB として使用されます。周波数を変化 / ランプ動作させるように設定した場合、この値は周波数の開始整数分周値の LSB として使用されます。これは、静的 PWM として構成されている場合は使用されません

図 7-139. PWM0\_START\_VAL\_LSB レジスタ

7	6	5	4	3	2	1	0
STV[7:0]							
R/W-0h							

表 7-180. PWM0\_START\_VAL\_LSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	STV[7:0]	R/W	0h	<p>開始値 これは多目的レジスタです。</p> <p>PWM 0_CTRL.MODE = デューティ サイクルの場合、これはデューティ サイクル値 STV [7:0] です。 8 ビット モードの場合、STV/256 = デューティ サイクルです。 10 ビット モードの場合、STV/1024 = デューティ サイクルです。</p> <p>PWM 0_CTRL.MODE = 周波数の場合、スイッチング周波数除数 START_VAL[7:0] の整数値です。 PWM 0_CTRL.MODE = 静的の場合、これは使われません。</p>

#### 7.6.8.21 PWM0\_START\_SL\_MSB レジスタ (オフセット = 4016h) [リセット = 00h]

図 7-140 に、PWM0\_START\_SL\_MSB を示し、表 7-181 に、その説明を示します。

概略表に戻ります。

スタート ランプのスロープ スケールを設定するために使用されます

図 7-140. PWM0\_START\_SL\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD				ST_SL[20:16]			
R-0h				R/W-0h			

表 7-181. PWM0\_START\_SL\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4-0	ST_SL[20:16]	R/W	0h	スタート スロープ ストップ ランプのスロープです。 フィールドは、PWM0_CTRL.SLOPE_SCALE の値によって異なります。 PWM0_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

**7.6.8.22 PWM0\_START\_SL\_MID レジスタ (オフセット = 4017h) [リセット = 00h]**

図 7-141 に、PWM0\_START\_SL\_MID を示し、表 7-182 に、その説明を示します。

概略表に戻ります。

スタート ランプのスロープ スケールを設定するために使用されます

**図 7-141. PWM0\_START\_SL\_MID レジスタ**

7	6	5	4	3	2	1	0
ST_SL[15:8]							
R/W-0h							

**表 7-182. PWM0\_START\_SL\_MID レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	ST_SL[15:8]	R/W	0h	スタート スロープ ストップ ランプのスロープです。 フィールドは、PWM0_CTRL.SLOPE_SCALE の値によって異なります。 PWM0_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

### 7.6.8.23 PWM0\_START\_SL\_LSB レジスタ (オフセット = 4018h) [リセット = 00h]

図 7-142 に、PWM0\_START\_SL\_LSB を示し、表 7-183 に、その説明を示します。

概略表に戻ります。

スタート ランプのスロープ スケールを設定するために使用されます

**図 7-142. PWM0\_START\_SL\_LSB レジスタ**

7	6	5	4	3	2	1	0
ST_SL[7:0]							
R/W-0h							

**表 7-183. PWM0\_START\_SL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	ST_SL[7:0]	R/W	0h	スタート スロープ ストップ ランプのスロープです。 フィールドは、PWM0_CTRL.SLOPE_SCALE の値によって異なります。 PWM0_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

**7.6.8.24 PWM0\_END\_VAL\_CONST\_FRAC\_F レジスタ (オフセット = 4019h) [リセット = 00h]**

図 7-143 に、PWM0\_END\_VAL\_CONST\_FRAC\_F を示し、表 7-184 に、その説明を示します。

概略表に戻ります。

周波数を変化させるように設定されている場合、周波数の終了値 (ターゲット値) 分数除数になります。デューティ サイクルを可変 / ランプさせるように設定されている場合、これはスイッチング周波数の分数除数となります。静的 PWM 出力として構成されている場合、このレジスタは周波数除数の分数部分に使用されます。

**図 7-143. PWM0\_END\_VAL\_CONST\_FRAC\_F レジスタ**

7	6	5	4	3	2	1	0
RSVD	EVCV_FR_F						
R-0h	R/W-0h						

**表 7-184. PWM0\_END\_VAL\_CONST\_FRAC\_F レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-0	EVCV_FR_F	R/W	0h	エンド値の小数部 (周波数モード) または定数値の小数部 (DC モード) エンド値小数部分。 これは常に使用されますが、フィールドは現在使用されているモードに基づいて変更されます。 周波数除数の x/128 分数部分と解釈されます。 PWM0_CTRL.MODE = デューティ サイクル。この場合、スイッチング周波数分周器 CONST_FREQ[10] の整数値になります。 PWM0_CTRL.MODE = 周波数または静的の場合、このビットは使用されません

### 7.6.8.25 PWM0\_END\_VAL\_MSB レジスタ (オフセット = 401Ah) [リセット = 00h]

図 7-144 に、PWM0\_END\_VAL\_MSB を示し、表 7-185 に、その説明を示します。

概略表に戻ります。

ランプに設定されている場合は、変化する PWM 終了値 (目標値) の設定。静的 PWM 出力として構成されている場合、このレジスタは周波数除数の整数 MSB 部分に使用されます。

**図 7-144. PWM0\_END\_VAL\_MSB レジスタ**

7	6	5	4	3	2	1	0
RSVD				EV_FR [10]		END_VAL [9:8]	
R-0h				R/W-0h		R/W-0h	

**表 7-185. PWM0\_END\_VAL\_MSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	EV_FR [10]	R/W	0h	終了値 (周波数のみ) これは多目的レジスタです。 PWM0_CTRL.MODE = デューティ サイクルまたは静的。これはスイッチング周波数の値 START_VAL[10] の整数値です。 PWM0_CTRL.MODE = 周波数の場合、このビットは使用されません
1-0	END_VAL [9:8]	R/W	0h	終了値 これは多目的レジスタです。 PWM 0_CTRL.MODE = デューティサイクルまたは静的である場合、これはスイッチング周波数除数 END_VAL[9:8] の整数値です。 PWM 0_CTRL.MODE = 周波数と PWM 0_CTRL の場合。 8B = 0 ( 10 ビット)。これはデューティ サイクル値 END_VAL[9 : 8] です。それ以外の場合は、この値は使用されません。 10 ビット モードの場合 、END_VAL/1024 = デューティ サイクルですが、 8 ビットモードではこのピンは使用されません

**7.6.8.26 PWM0\_END\_VAL\_LSB レジスタ (オフセット = 401Bh) [リセット = 00h]**

図 7-145 に、PWM0\_END\_VAL\_LSB を示し、表 7-186 に、その説明を示します。

概略表に戻ります。

ランプに設定されている場合は、変化する PWM 終了値 (目標値) の設定。静的 PWM 出力として構成されている場合、このレジスタは周波数除数の整数 LSB 部分に使用されます。

**図 7-145. PWM0\_END\_VAL\_LSB レジスタ**

7	6	5	4	3	2	1	0
END_VAL[7:0]							
R/W-0h							

**表 7-186. PWM0\_END\_VAL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	END_VAL[7:0]	R/W	0h	終了値 これは多目的レジスタです。 PWM 0_CTRL.MODE = デューティ サイクルまたは静的である場合、これはスイッチング周波数除数 END_VAL[7:0] の整数値です。 PWM 0_CTRL.MODE = 周波数の場合、これはデューティ サイクル値 END_VAL[7:0] です。 8 ビットモードの場合、END_VAL/256 = デューティ サイクルです。 10 ビットモードの場合、END_VAL/1024 = デューティ サイクル



#### 7.6.8.27 PWM0\_PULSE\_STOP\_RAMP[y] レジスタ (オフセット = 401Ch + 式) [リセット = 00000000h]

図 7-146 に PWM0\_PULSE\_STOP\_RAMP[y] の制御を示し、表 7-187 にその説明を示します。

概略表に戻ります。

自動ストップランプに設定されている場合、これは開始パルスから生成される PWM パルスの数であり、その時点でモジュールが自動的にストップランプ コマンドを設定します。最下位アドレスはカウンタの MSB に対応します。

オフセット = 401Ch + (y \* 4h)、ここで y = 0h ~ 3h

**図 7-146. PWM0\_PULSE\_STOP\_RAMP[y] レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_STOP_RAMP																															
R/W-0h																															

**表 7-187. PWM0\_PULSE\_STOP\_RAMP[y] レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
31-0	PWM0_PULSE_STOP_RAMP	R/W	0h	指定された数の PWM パルスが生成されると、自動的にストップランプが開始されます。 これは、PWM0_ACTION.AUTO_STOP ビットがセットされている場合にのみ使用されます。 最下位アドレスは MSB です。

### 7.6.8.28 PWM0\_PULSE\_MAX[y] レジスタ (オフセット = 4020h + 式) [リセット = 00000000h]

図 7-147 に PWM0\_PULSE\_MAX[y] の制御を示し、表 7-188 にその説明を示します。

概略表に戻ります。

自動ストップ ランプに設定されている場合、これは許可される PWM パルスの最大数となります。PWM パルス  
の数がこの制限値に達すると、出力は直ちに停止します。最下位アドレスはカウンタの MSB に対応します。

オフセット = 4020h + (y \* 4h)、ここで y = 0h ~ 3h

図 7-147. PWM0\_PULSE\_MAX[y] レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_MAX																															
R/W-0h																															

表 7-188. PWM0\_PULSE\_MAX[y] レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	PWM0_PULSE_MAX	R/W	0h	指定された数の PWM パルスが生成されると、PWM 出力は無効化されます。 これは、PWM0_ACTION.AUTO_STOP ビットがセットされている場合にのみ使用されます。 最下位アドレスは MSB です。

### 7.6.8.29 PWM0\_ACTION レジスタ (オフセット = 4024h) [リセット = 00h]

図 7-148 に、PWM0\_ACTION を示し、表 7-189 に、その説明を示します。

概略表に戻ります。

**図 7-148. PWM0\_ACTION レジスタ**

7	6	5	4	3	2	1	0
RSVD			COUNT_RST	AUTO_STOP	UDS	START	ストップ
R-0h			R/W1S-0h	R/W-0h	R/W-0h	RH/W1S-0h	RH/W1S-0h

**表 7-189. PWM0\_ACTION レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	COUNT_RST	R/W1S	0h	<p>パルスカウンタを 0 にリセットする このビットがセットされると、パルス カウンタはリセットされます。 これは通常、パルスの合計数を追跡するために、ランプの開始時に実行されます。 リセットが完了すると、ハードウェアによってクリアされます。</p> <p style="text-align: center;"><b>注</b></p> <p>AUTO_STOP を使用する場合は、STOP および最大パルス カウントの値に注意してください。 これらのパルス カウンタは現在のパルス カウンタと比較され、最大パルスカウンタが現在のパルス カウンタ値よりも小さい場合、PWM 出力は生成されません</p>
3	AUTO_STOP	R/W	0h	<p>自動ストップ ランプを使用 有効化されている場合、パルス カウントが指定値に達した場合です 0h = 自動ストップ ランプ動作は使用しません 1h = パルス カウント レジスタに基づいて、自動的にストップ ランプを開始します</p>
2	UDS	R/W	0h	<p>定義済み開始を使用 開始値レジスタで定義されている開始点を使用します。 開始点が既知であることを確認する場合、または現在の値 (中間ランプまたは既存の終点) からランプを実行する場合に使用します 0h = 既存の現在の値を開始点として使用します 1h = 定義された開始点から開始します</p>
1	START	RH/W1S	0h	<p>開始 構成済みの設定に基づいて、定義済みのランプ プロファイルを開始します。 STOP も同時に設定されている場合、START は無視されます。 ビットはハードウェアによってクリアされます。</p>
0	ストップ	RH/W1S	0h	<p>停止 指定された停止点まで上昇し、その後 OFF になります。 START と STOP を同時に設定すると、STOP が発生します。 ビットはハードウェアによってクリアされます</p>

### 7.6.8.30 PWM0\_IAS\_CTRL レジスタ (オフセット = 4030h) [リセット = 00h]

図 7-149 に、PWM0\_IAS\_CTRL を示し、表 7-190 に、その説明を示します。

概略表に戻ります。

図 7-149. PWM0\_IAS\_CTRL レジスタ

7	6	5	4	3	2	1	0
RSVD			IN_POL	GPIO_SEL		STOP_MODE	AS_EN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h

表 7-190. PWM0\_IAS\_CTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	予約済み
4	IN_POL	R/W	0h	入力極性 ストップ動作をトリガするために使用するレベルを選択します 0h = アクティブ Low 入力信号 1h = アクティブ High 入力信号
3-2	GPIO_SEL	R/W	0h	GPIO 選択 使用する GPIO を指定します。 動作するには、特殊機能の代わりに GPIO 機能として GPIO を構成する必要があります。 0h = GPIO0 1h = GPIO1 2h = GPIO7 3h = GPIO8
1	STOP_MODE	R/W	0h	ストップ モード GPIO によりトリガされたストップ イベントが発生すると、PWM 出力を直ちにディスエーブルにするか、ストップ ランプを要求するかを選択します。 0h = GPIO トリガ イベントはストップ ランプを要求しています 1h = GPIO トリガ イベントにより、PWM 出力は即座に無効化されます
0	AS_EN	R/W	0h	入力自動停止のイネーブル  <b>注</b> この場合、AUTO_STOP ビットを設定する必要はなく、GPIO 入力に基づいて出力を停止するための別の機能です  0h = GPIO 信号からの自動停止は有効ではありません 1h = GPIO によりトリガされた自動停止機能を有効にします

### 7.6.9 PWM1 のレジスタ

PWM1 レジスタのメモリマップされたレジスタを、セクション 7.6.9 に示します。セクション 7.6.9 がないレジスタ オフセット アドレスはすべて予約済みと見なして、レジスタの内容は変更しないでください。

表 7-191. PWM1 のレジスタ

オフセット	略称	レジスタ名	セクション
4100h	PWM1_CTRL	PWM1 制御	セクション 7.6.9.1
4101h	PWM1_IE0	割り込みイネーブル	セクション 7.6.9.2
4102h	PWM1_IE1	割り込みイネーブル	セクション 7.6.9.3
4103h	PWM1_IR	割り込みレジスタ	セクション 7.6.9.4
4104h	PWM1_STATUS	ステータス	セクション 7.6.9.5
4105h + 式	PWM1_CUR_PULSE[y]	現在のパルス カウント	セクション 7.6.9.6
4109h	PWM1_CUR_VAL_MSB	現在の値 MSB	セクション 7.6.9.7
410Ah	PWM1_CUR_VAL_LSB	現在の値 LSB	セクション 7.6.9.8

**表 7-191. PWM1 のレジスタ (続き)**

オフセット	略称	レジスタ名	セクション
410Bh	PWM1_CONST_MSB	定数整数 MSB	<a href="#">セクション 7.6.9.9</a>
410Ch	PWM1_CONST_LSB	定数整数 LSB	<a href="#">セクション 7.6.9.10</a>
410Dh	PWM1_STOP_VAL_FRAC_F	停止分数値 (周波数のみ)	<a href="#">セクション 7.6.9.11</a>
410Eh	PWM1_STOP_VAL_MSB	ストップ値整数 MSB	<a href="#">セクション 7.6.9.12</a>
410Fh	PWM1_STOP_VAL_LSB	ストップ値整数 LSB	<a href="#">セクション 7.6.9.13</a>
4110h	PWM1_STOP_SL_MSB	ストップ スロープ MSB	<a href="#">セクション 7.6.9.14</a>
4111h	PWM1_STOP_SL_MID	ストップ スロープ中間	<a href="#">セクション 7.6.9.15</a>
4112h	PWM1_STOP_SL_LSB	ストップ スロープ LSB	<a href="#">セクション 7.6.9.16</a>
4113h	PWM1_START_VAL_FRAC_F	開始分数値 (周波数のみ)	<a href="#">セクション 7.6.9.17</a>
4114h	PWM1_START_VAL_MSB	開始値 MSB	<a href="#">セクション 7.6.9.18</a>
4115h	PWM1_START_VAL_LSB	開始値 LSB	<a href="#">セクション 7.6.9.19</a>
4116h	PWM1_START_SL_MSB	開始スロープ MSB	<a href="#">セクション 7.6.9.20</a>
4117h	PWM1_START_SL_MID	開始スロープ中間バイト	<a href="#">セクション 7.6.9.21</a>
4118h	PWM1_START_SL_LSB	開始スロープ LSB	<a href="#">セクション 7.6.9.22</a>
4119h	PWM1_END_VAL_CONST_FRAC_F	終了 (Freq) または制御 (DC) 分数の値	<a href="#">セクション 7.6.9.23</a>
411Ah	PWM1_END_VAL_MSB	終了値 MSB	<a href="#">セクション 7.6.9.24</a>
411Bh	PWM1_END_VAL_LSB	終了値 LSB	<a href="#">セクション 7.6.9.25</a>
411Ch + 式	PWM1_PULSE_STOP_RAMP[y]	ストップ ランプを開始するためのパルス カウント	<a href="#">セクション 7.6.9.26</a>
4120h + 式	PWM1_PULSE_MAX[y]	最大パルス カウント	<a href="#">セクション 7.6.9.27</a>
4124h	PWM1_ACTION	開始アクション レジスタ。	<a href="#">セクション 7.6.9.28</a>
4130h	PWM1_IAS_CTRL	入力自動停止制御	<a href="#">セクション 7.6.9.29</a>

表の小さなセルに収まるように、複雑なビット アクセス タイプを記号で表記しています。[セクション 7.6.9](#) に、このセクションでアクセス タイプに使用しているコードを示します。

**表 7-192. PWM1 のアクセス タイプ コード**

アクセス タイプ	コード	説明
読み取りタイプ		
R	R	読み出し
RH	R H	ハードウェアによってセットまたはクリアされる の読み取り
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	1 を書き込むことで セット
WP	W P	書き込み 特権アクセスが必要
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ アレイ変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアド レスで使用されている場合、レジスタが反復レジ スタ グループの一部であるレジスタ アレイの値を 示します。レジスタ グループは階層構造を形成し、 アレイは式で表されます。

表 7-192. PWM1 のアクセス タイプ コード (続き)

アクセス タイプ	コード	説明
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ アレイの値を示します。

### 7.6.9.1 PWM1\_CTRL レジスタ (オフセット = 4100h) [リセット = 11h]

図 7-150 に、PWM1\_CTRL を示し、表 7-193 に、その説明を示します。

概略表に戻ります。

PWM1 モジュールを設定するための制御フィールド

**図 7-150. PWM1\_CTRL レジスタ**

7	6	5	4	3	2	1	0
RSVD	SLOPE_SCALE		モード		DC_8B	INIT	
R-0h	R/WP-1h		R/WP-0h		R/WP-0h	R/W-1h	

**表 7-193. PWM1\_CTRL レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-4	SLOPE_SCALE	R/WP	1h	<p>スロープ スケール スロープ レジスタの整数部分と分数部分に使用されるビット数を調整します。 周波数がデューティ サイクルのどちらをランプさせるかによって、列挙値の意味が変わります。 詳細については「<a href="#">セクション 7.5.7.4</a>」を参照してください  0h = Int[20 ~ 16], Frac[15-0] (周波数) - Int[20], Frac[19-0] (デューティ サイクル)  1h = Int[20 ~ 12], Frac[11-0] (周波数) - Int[20-19], Frac[18-0] (デューティ サイクル)  2h = Int[20 ~ 8], Frac[7-0] (周波数) - Int[20-18], Frac[17-0] (デューティ サイクル)  3h = Int[20 ~ 4], Frac[3-0] (周波数) - Int[20-17], Frac[16-0] (デューティ サイクル)  4h = Int[20-0], 分数なし (周波数) - Int[20-16], Frac[15-0] (デューティ サイクル)</p>
3-2	モード	R/WP	0h	<p>スロープ レジスタの整数部分と分数部分に使用されるビット数を調整します。 GPIO が特殊機能用に設定されている場合、モードを off 以外に設定すると、SPI チップセレクトよりも優先されることに注意してください。  0h = オフ  1h = デューティ サイクル ランプ  2h = 周波数ランプ  3h = 静的オン</p>
1	DC_8B	R/WP	0h	<p>8 ビット デューティ サイクル モード  0h = 10 ビットのデューティ サイクル分解能  1h = 8 ビットのデューティ サイクル分解能</p>
0	INIT	R/W	1h	<p>初期化モード PWM IP がリセット状態に保持され、この PWM1_CTRL レジスタの設定に使用されます  0h = 初期化モードはディスエーブルです。保護された PWM ビットへの書き込みは許可されません  1h = 初期化モードはイネーブルです。保護された PWM ビットへの書き込みが許可されます</p>

### 7.6.9.2 PWM1\_IE0 レジスタ (オフセット = 4101h) [リセット = 00h]

図 7-151 に、PWM1\_IE0 を示し、表 7-194 に、その説明を示します。

概略表に戻ります。

図 7-151. PWM1\_IE0 レジスタ

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	予約済み
R-0h					R/W-0h	R/W-0h	R-0h

表 7-194. PWM1\_IE0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF_IE	R/W	0h	電流パルス オーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定されます 0h = 現在のパルス カウンタはオーバーフローしていません 1h = 現在のパルス カウンタがオーバーフローしました
1	IAS_IE	R/W	0h	入力自動停止割り込み 入力自動停止条件が発生したときにセットされます。PWM が即時停止に設定されている場合、出力も停止します (RC ビットに反映されます)。 PWM が IAS トリガでストップ ランプを開始するように設定されている場合、この割り込みは入力条件が発生した時点で直ちにセットされ、出力がオフになった時点で RC ビットがセットされます。 0h = IAS の入力トリガされていません 1h = IAS の入力トリガされました
0	予約済み	R	0h	



### 7.6.9.3 PWM1\_IE1 レジスタ (オフセット = 4102h) [リセット = 00h]

図 7-152 に、PWM1\_IE1 を示し、表 7-195 に、その説明を示します。

概略表に戻ります。

**図 7-152. PWM1\_IE1 レジスタ**

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RC_IE
R-0h					R/W-0h	R/W-0h	R/W-0h

**表 7-195. PWM1\_IE1 レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF_IE	R/W	0h	電流パルス オーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定されます 0h = 現在のパルス カウンタはオーバーフローしていません 1h = 現在のパルス カウンタがオーバーフローしました
1	IAS_IE	R/W	0h	<p>入力自動停止割り込み 入力自動停止条件が発生したときにセットされます。PWM が即時停止に設定されている場合、出力も停止します (RC ビットに反映されます)。</p> <p>PWM が IAS トリガでストップ ランプを開始するように設定されている場合、この割り込みは入力条件が発生した時点で直ちにセットされ、出力がオフになった時点で RC ビットがセットされます。</p> <p style="text-align: center;"><b>注</b></p> <p>オンランプの実行中で、STOP 条件が発生した場合、RC ビットはオンランプに設定されず、STOP 条件のために出力が停止するとのみ一度だけです</p> <p>0h = IAS の入力はトリガされていません 1h = IAS の入力トリガされました</p>
0	RC_IE	R/W	0h	<p>PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。AS_EN = 0 の場合、PWM が終了値に達すると設定されます。IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止すると、設定されます。</p> <p>ストップ ランプが手動でアクティブになると、ランプが完了して出力がオフになると、RC ビットがセットされます。</p> <p style="text-align: center;"><b>注</b></p> <p>オンランプの実行中で、STOP 条件が発生した場合、RC ビットはオンランプに設定されず、STOP 条件のために出力が停止するとのみ一度だけです</p> <p>0h = PWM チャネル ランプは完了していません 1h = PWM チャネル ランプが完了しています</p>

## 7.6.9.4 PWM1\_IR レジスタ (オフセット = 4103h) [リセット = 00h]

図 7-153 に、PWM1\_IR を示し、表 7-196 に、その説明を示します。

概略表に戻ります。

図 7-153. PWM1\_IR レジスタ

7	6	5	4	3	2	1	0
RSVD				PULSE_OVF		IAS	RC
R-0h				R/W1C-0h		R/W1C-0h	R/W1C-0h

表 7-196. PWM1\_IR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	予約済み
2	PULSE_OVF	R/W1C	0h	電流パルス オーバーフロー CUR_PULSE カウンタがオーバーフローしたときに設定されます 0h = 現在のパルス カウンタはオーバーフローしていません 1h = 現在のパルス カウンタがオーバーフローしました
1	IAS	R/W1C	0h	入力自動停止割り込み 入力自動停止条件が発生したときにセットされます。PWM が即時停止に設定されている場合、出力も停止します (RC ビットに反映されます)。 PWM が IAS トリガでストップ ランプを開始するように設定されている場合、この割り込みは入力条件が発生した時点で直ちにセットされ、出力がオフになった時点で RC ビットがセットされます。 0h = IAS の入力はトリガされていません 1h = IAS の入力トリガされました
0	RC	R/W1C	0h	PWM ランプ完了 ランプが完了した時点で設定されます。AS_EN = 1 (自動停止) の場合、これは PWM チャネルがオフになったときに発生します。 AS_EN = 0 の場合、PWM が終了値に達すると設定されます。 IAS_EN = 1 (入力自動停止) の場合、PWM 出力が停止するとこれが設定されます。 ストップ ランプが手動でアクティブになった場合、ランプが完了し、出力がオフになると RC ビットがセットされます。  <b>注</b> オンランプの実行中で、STOP 条件が発生した場合、RC ビットはオンランプに設定されず、STOP 条件のために出力が停止するとのみ一度だけです  0h = PWM チャネル ランプは完了していません 1h = PWM チャネル ランプが完了しています

### 7.6.9.5 PWM1\_STATUS レジスタ (オフセット = 4104h) [リセット = 00h]

図 7-154 に、PWM1\_STATUS を示し、表 7-197 に、その説明を示します。

概略表に戻ります。

**図 7-154. PWM1\_STATUS レジスタ**

7	6	5	4	3	2	1	0
RSVD				IAS_ACT		BUSY	OEN
R-0h				RH-0h		RH-0h	RH-0h

**表 7-197. PWM1\_STATUS レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	IAS_ACT	RH	0h	入力自動停止アクティブ GPIO 入力のアサート状態にトグルすると設定され、PWM 出力が停止すると自動的にクリアされます。 ランプが完了すると、RC 割り込みビットをセットします 0h = 入力の自動ストップはアクティブではありません 1h = 入力自動停止がアクティブです
1	BUSY	RH	0h	ランプ IP がビジーです 0h = PWM ランプが完了しています 1h = PWM ランプは現在進行中です
0	OEN	RH	0h	出力イネーブル 0h = PWM 出力は無効です 1h = PWM 出力はイネーブルで、出力を生成しています

**7.6.9.6 PWM1\_CUR\_PULSE[y] レジスタ (オフセット = 4105h + 式) [リセット = 00000000h]**

図 7-155 に PWM1\_CUR\_PULSE[y] の制御を示し、表 7-198 にその説明を示します。

概略表に戻ります。

現在のパルス カウントを示す符号なし 32 ビット カウンタ値です。最下位アドレスはカウンタの MSB に対応します。

オフセット = 4105h + (y \* 4h)、ここで y = 0h ~ 3h

**図 7-155. PWM1\_CUR\_PULSE[y] レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_CUR_PULSE																															
R-0h																															

**表 7-198. PWM1\_CUR\_PULSE[y] レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
31-0	PWM0_CUR_PULSE	R	0h	電流ランプの電流パルス数。 ランプが完了した場合、この値は、最後のランプに対して生成されたパルスの合計数になります。 この値は、新しいランプが開始されるたびにリセットされます。 最下位アドレスは MSB です。

### 7.6.9.7 PWM1\_CUR\_VAL\_MSB レジスタ (オフセット = 4109h) [リセット = 00h]

図 7-156 に、PWM1\_CUR\_VAL\_MSB を示し、表 7-199 に、その説明を示します。

概略表に戻ります。

**図 7-156. PWM1\_CUR\_VAL\_MSB レジスタ**

7	6	5	4	3	2	1	0
RSVD					CUR_VAL[10:8]		
R-0h					R-0h		

**表 7-199. PWM1\_CUR\_VAL\_MSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2-0	CUR_VAL[10:8]	R	0h	

#### 7.6.9.8 PWM1\_CUR\_VAL\_LSB レジスタ (オフセット = 410Ah) [リセット = 00h]

図 7-157 に、PWM1\_CUR\_VAL\_LSB を示し、表 7-200 に、その説明を示します。

概略表に戻ります。

図 7-157. PWM1\_CUR\_VAL\_LSB レジスタ

7	6	5	4	3	2	1	0
CUR_VAL[7:0]							
R-0h							

表 7-200. PWM1\_CUR\_VAL\_LSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	CUR_VAL[7:0]	R	0h	

#### 7.6.9.9 PWM1\_CONST\_MSB レジスタ (オフセット = 410Bh) [リセット = 00h]

図 7-158 に、PWM1\_CONST\_MSB を示し、表 7-201 に、その説明を示します。

概略表に戻ります。

周波数を可変 / ランプさせるように設定されている場合、これは静的デューティ サイクルの MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合です)。デューティ サイクルを可変 / ランプさせるように設定されている場合、これはスイッチング周波数の固定整数除数の MSB として使用されます。静的 PWM 出力に設定されている場合、これはデューティ サイクルの MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合です)。

図 7-158. PWM1\_CONST\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD					CV_F[10]	CV[9:8]	
R-0h					R/W-0h	R/W-0h	

表 7-201. PWM1\_CONST\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	CV_F[10]	R/W	0h	一定値 (周波数のみ) これは多目的レジスタです。 PWM1_CTRL.MODE = デューティ サイクル。この場合、スイッチング周波数分周器 CONST_FREQ[10] の整数値になります。 PWM1_CTRL.MODE = 周波数または静的の場合、このビットは使用されません
1-0	CV[9:8]	R/W	0h	一定数 これは多目的レジスタです。 PWM 1_CTRL.MODE = デューティ サイクル。これはスイッチング周波数分周器です。CONST_FREQ[9:8] の整数値です。 PWM 1_CTRL.MODE = (周波数または静的) かつ PWM1_CTRL. 8B = 0 (10 ビット) の場合、これはデューティ サイクル値 CONST_DC[9:8] となります (CONST_DC/1024 = デューティ サイクル)。 8 ビット分解能モードでは、これは使われません。

#### 7.6.9.10 PWM1\_CONST\_LSB レジスタ (オフセット = 410Ch) [リセット = 00h]

図 7-159 に、PWM1\_CONST\_LSB を示し、表 7-202 に、その説明を示します。

概略表に戻ります。

周波数を可変 / ランプさせるように設定されている場合、これは静的デューティ サイクルの LSB として使用されます。デューティ サイクルを可変 / ランプさせるように設定されている場合、これはスイッチング周波数の固定整数除数の LSB として使用されます。静的 PWM 出力に設定されている場合、これはデューティ サイクルの LSB として使用されます。

図 7-159. PWM1\_CONST\_LSB レジスタ

7	6	5	4	3	2	1	0
CV[7:0]							
R/W-0h							

表 7-202. PWM1\_CONST\_LSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-0	CV[7:0]	R/W	0h	<p>定数値 これは多目的レジスタです。</p> <p>PWM 1_CTRL.MODE = Duty Cycle の場合、スイッチング周波数除数の整数値 CONST_FREQ[7:0] となります。</p> <p>PWM 1_CTRL.MODE = 周波数または静的の場合、これはデューティサイクル除数 CONST_DC[7:0] です。</p> <p>8 ビットモードの場合、 CONST_DC/256 = デューティ サイクルです。</p> <p>10 ビット モードの場合、 CONST_DC/1024 = デューティ サイクルです</p>



### 7.6.9.11 PWM1\_STOP\_VAL\_FRAC\_F レジスタ (オフセット = 410Dh) [リセット = 00h]

図 7-160 に、PWM1\_STOP\_VAL\_FRAC\_F を示し、表 7-203 に、その説明を示します。

概略表に戻ります。

周波数を可変 / ランプ設定にしている場合、これはストップまたはオフ ターゲット時の周波数分数除数として使用されます。これは、デューティ サイクルを上昇させるように構成されている場合や静的 PWM には使用されません。

**図 7-160. PWM1\_STOP\_VAL\_FRAC\_F レジスタ**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-203. PWM1\_STOP\_VAL\_FRAC\_F レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	ストップ値分数値 (周波数のみ) PWM1_CTRL.MODE = 周波数の場合にのみ使用されます。 周波数除数の x/128 分数部分と解釈されます。 PWM1_CTRL.MODE = デューティ サイクルまたは静的の場合、これは使われません。

### 7.6.9.12 PWM1\_STOP\_VAL\_MSB レジスタ (オフセット = 410Eh) [リセット = 00h]

図 7-161 に、PWM1\_STOP\_VAL\_MSB を示し、表 7-204 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクル MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合のみ使用されます)。周波数を可変 / ランプさせるように設定されている場合、これは周波数のストップ / オフターゲット整数除数の MSB として使用されません。これは、静的 PWM として構成されている場合は使用されません。

図 7-161. PWM1\_STOP\_VAL\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD				SPV_F[10]		SPV[9:8]	
R-0h				R/W-0h		R/W-0h	

表 7-204. PWM1\_STOP\_VAL\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	SPV_F[10]	R/W	0h	停止値 (周波数のみ) これは多目的レジスタです。 PWM1_CTRL.MODE = 周波数の場合、スイッチング周波数除数 STOP_VAL[10] の整数値です。 PWM1_CTRL.MODE = デューティ サイクルの場合、このビットは使用されません。 PWM1_CTRL.MODE = 静的の場合、これは使われません。
1-0	SPV[9:8]	R/W	0h	ストップ値 これは多目的レジスタです。 PWM 1_CTRL.MODE = デューティ サイクルかつ PWM 1_CTRL の場合、 8B = 0 (10 ビット)。これはデューティ サイクル値 SPV[9:8] です。それ以外の場合は、この値は使用されません。 SPV/ 1024 = デューティ サイクルで、8 ビット モードでは使用されません。 PWM 1_CTRL.MODE = 周波数の場合、これはスイッチング周波数除数 STOP_VAL[9:8] の整数値です。 PWM 1_CTRL.MODE = 静的の場合、これは使われません。

### 7.6.9.13 PWM1\_STOP\_VAL\_LSB レジスタ (オフセット = 410Fh) [リセット = 00h]

図 7-162 に、PWM1\_STOP\_VAL\_LSB を示し、表 7-205 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクルの LSB として使用されます。周波数を可変 / ランプさせるように設定されている場合、これは周波数のストップ / オフターゲット整数除数の LSB として使用されます。これは、静的 PWM として構成されている場合は使用されません

**図 7-162. PWM1\_STOP\_VAL\_LSB レジスタ**

7	6	5	4	3	2	1	0
SPV[7:0]							
R/W-0h							

**表 7-205. PWM1\_STOP\_VAL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	SPV[7:0]	R/W	0h	<p>ストップ値 これは多目的レジスタです。</p> <p>PWM 1_CTRL.MODE = デューティ サイクル、これはデューティ サイクル値 SPV[7:0] です。 8 ビット モードの場合、SPV/256 = デューティ サイクルです。 10 ビット モードの場合、SPV/1024 = デューティ サイクルです。</p> <p>PWM 1_CTRL.MODE = 周波数の場合、スイッチング周波数除数 STOP_VAL[7:0] の整数値です。 PWM 1_CTRL.MODE = 静的の場合、これは使われません。</p>

**7.6.9.14 PWM1\_STOP\_SL\_MSB レジスタ (オフセット = 4110h) [リセット = 00h]**

図 7-163 に、PWM1\_STOP\_SL\_MSB を示し、表 7-206 に、その説明を示します。

概略表に戻ります。

ストップランプのスロープスケールを設定するために使用します

**図 7-163. PWM1\_STOP\_SL\_MSB レジスタ**

7	6	5	4	3	2	1	0
RSVD			SP_SL[20:16]				
R-0h			R/W-0h				

**表 7-206. PWM1\_STOP\_SL\_MSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4-0	SP_SL[20:16]	R/W	0h	ストップ スロープ ストップ ランプのスロープです。 フィールドは、PWM1_CTRL.SLOPE_SCALE の値によって異なります。 PWM1_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

#### 7.6.9.15 PWM1\_STOP\_SL\_MID レジスタ (オフセット = 4111h) [リセット = 00h]

図 7-164 に、PWM1\_STOP\_SL\_MID を示し、表 7-207 に、その説明を示します。

概略表に戻ります。

ストップ ランプのスロープ スケールを設定するために使用します

**図 7-164. PWM1\_STOP\_SL\_MID レジスタ**

7	6	5	4	3	2	1	0
SP_SL[15:8]							
R/W-0h							

**表 7-207. PWM1\_STOP\_SL\_MID レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	SP_SL[15:8]	R/W	0h	ストップ スロープ ストップ ランプのスロープです。 フィールドは、PWM1_CTRL.SLOPE_SCALE の値によって異なります。 PWM1_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

**7.6.9.16 PWM1\_STOP\_SL\_LSB レジスタ (オフセット = 4112h) [リセット = 00h]**

図 7-165 に、PWM1\_STOP\_SL\_LSB を示し、表 7-208 に、その説明を示します。

概略表に戻ります。

ストップ ランプのスロープ スケールを設定するために使用します

**図 7-165. PWM1\_STOP\_SL\_LSB レジスタ**

7	6	5	4	3	2	1	0
SP_SL[7:0]							
R/W-0h							

**表 7-208. PWM1\_STOP\_SL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	SP_SL[7:0]	R/W	0h	ストップ スロープ ストップ ランプのスロープです。 フィールドは、PWM1_CTRL.SLOPE_SCALE の値によって異なります。 PWM1_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

### 7.6.9.17 PWM1\_START\_VAL\_FRAC\_F レジスタ (オフセット = 4113h) [リセット = 00h]

図 7-166 に、PWM1\_START\_VAL\_FRAC\_F を示し、表 7-209 に、その説明を示します。

概略表に戻ります。

変動 / ランプ周波数に設定されている場合、周波数の開始分数除数に使用されます。これは、デューティ サイクルを上昇させるように構成されている場合や静的 PWM には使用されません。

**図 7-166. PWM1\_START\_VAL\_FRAC\_F レジスタ**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-209. PWM1\_START\_VAL\_FRAC\_F レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	開始値分数値 (周波数のみ) PWM1_CTRL.MODE = 周波数の場合にのみ使用されます。 周波数除数の x/128 分数部分と解釈されます。 PWM1_CTRL.MODE = デューティ サイクルまたは静的の場合、これは使われません。

### 7.6.9.18 PWM1\_START\_VAL\_MSB レジスタ (オフセット = 4114h) [リセット = 00h]

図 7-167 に、PWM1\_START\_VAL\_MSB を示し、表 7-210 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクル MSB として使用されます (10 ビット デューティ サイクル分解能が有効な場合のみ使用されます)。周波数を変化 / ランプ動作させるように設定した場合、この値は周波数の開始整数分周値の MSB として使用されます。これは、静的 PWM として構成されている場合は使用されません

図 7-167. PWM1\_START\_VAL\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD					STV_F[10]	STV[9:8]	
R-0h					R/W-0h	R/W-0h	

表 7-210. PWM1\_START\_VAL\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	STV_F[10]	R/W	0h	開始値 (周波数のみ) これは多目的レジスタです。 PWM1_CTRL.MODE = 周波数の場合、スイッチング周波数除数 START_VAL[10] の整数値です。 PWM1_CTRL.MODE = デューティ サイクルの場合、このビットは使用されません。 PWM1_CTRL.MODE = 静的の場合、これは使用しません。
1-0	STV[9:8]	R/W	0h	開始値 これは多目的レジスタです。 PWM 1_CTRL.MODE = デューティ サイクルかつ PWM 1_CTRL の場合、 8B = 0 ( 10 ビット)。これはデューティ サイクル値 STV[9 : 8] です。それ以外の場合は、この値は使用されません。 STV/ 1024= デューティサイクルであり、 8 ビット モードでは使用されません。 PWM 1_CTRL.MODE = 周波数の場合、スイッチング周波数除数 START_VAL[ 9: 8] の整数値です。 PWM 1_CTRL.MODE = 静的の場合、これは使われません。



#### 7.6.9.19 PWM1\_START\_VAL\_LSB レジスタ (オフセット = 4115h) [リセット = 00h]

図 7-168 に、PWM1\_START\_VAL\_LSB を示し、表 7-211 に、その説明を示します。

概略表に戻ります。

デューティ サイクルを可変 / ランプさせるように設定されている場合、これはデューティ サイクルの LSB として使用されます。周波数を変化 / ランプ動作させるように設定した場合、この値は周波数の開始整数分周値の LSB として使用されます。これは、静的 PWM として構成されている場合は使用されません

**図 7-168. PWM1\_START\_VAL\_LSB レジスタ**

7	6	5	4	3	2	1	0
STV[7:0]							
R/W-0h							

**表 7-211. PWM1\_START\_VAL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	STV[7:0]	R/W	0h	<p>開始値 これは多目的レジスタです。</p> <p>PWM 1_CTRL.MODE = デューティ サイクルの場合、これはデューティサイクル値 STV [7:0] です。</p> <p>8 ビット モードの場合、STV/256 = デューティ サイクルです。</p> <p>10 ビット モードの場合、STV/1024 = デューティ サイクルです。</p> <p>PWM 1_CTRL.MODE = 周波数の場合、スイッチング周波数除数 START_VAL[7:0] の整数値です。</p> <p>PWM 1_CTRL.MODE = 静的の場合、これは使われません。</p>

#### 7.6.9.20 PWM1\_START\_SL\_MSB レジスタ (オフセット = 4116h) [リセット = 00h]

図 7-169 に、PWM1\_START\_SL\_MSB を示し、表 7-212 に、その説明を示します。

概略表に戻ります。

スタート ランプのスロープ スケールを設定するために使用されます

図 7-169. PWM1\_START\_SL\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD			ST_SL[20:16]				
R-0h			R/W-0h				

表 7-212. PWM1\_START\_SL\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4-0	ST_SL[20:16]	R/W	0h	スタート スロープ ストップ ランプのスロープです。 フィールドは、PWM1_CTRL.SLOPE_SCALE の値によって異なります。 PWM1_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

### 7.6.9.21 PWM1\_START\_SL\_MID レジスタ (オフセット = 4117h) [リセット = 00h]

図 7-170 に、PWM1\_START\_SL\_MID を示し、表 7-213 に、その説明を示します。

概略表に戻ります。

スタート ランプのスロープ スケールを設定するために使用されます

**図 7-170. PWM1\_START\_SL\_MID レジスタ**

7	6	5	4	3	2	1	0
ST_SL[15:8]							
R/W-0h							

**表 7-213. PWM1\_START\_SL\_MID レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	ST_SL[15:8]	R/W	0h	スタート スロープ ストップ ランプのスロープです。 フィールドは、PWM1_CTRL.SLOPE_SCALE の値によって異なります。 PWM1_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

**7.6.9.22 PWM1\_START\_SL\_LSB レジスタ (オフセット = 4118h) [リセット = 00h]**

図 7-171 に、PWM1\_START\_SL\_LSB を示し、表 7-214 に、その説明を示します。

概略表に戻ります。

スタート ランプのスロープ スケールを設定するために使用されます

**図 7-171. PWM1\_START\_SL\_LSB レジスタ**

7	6	5	4	3	2	1	0
ST_SL[7:0]							
R/W-0h							

**表 7-214. PWM1\_START\_SL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	ST_SL[7:0]	R/W	0h	スタート スロープ ストップ ランプのスロープです。 フィールドは、PWM1_CTRL.SLOPE_SCALE の値によって異なります。 PWM1_CTRL.MODE= 静的の場合、これは未使用です。 詳細については、 <a href="#">セクション 7.5.7.4</a> を参照してください。

### 7.6.9.23 PWM1\_END\_VAL\_CONST\_FRAC\_F レジスタ (オフセット = 4119h) [リセット = 00h]

図 7-172 に、PWM1\_END\_VAL\_CONST\_FRAC\_F を示し、表 7-215 に、その説明を示します。

概略表に戻ります。

周波数を変化させるように設定されている場合、周波数の終了値 (ターゲット値) 分数除数になります。デューティ サイクルを可変 / ランプさせるように設定されている場合、これはスイッチング周波数の分数除数となります。静的 PWM 出力として構成されている場合、このレジスタは周波数除数の分数部分に使用されます。

**図 7-172. PWM1\_END\_VAL\_CONST\_FRAC\_F レジスタ**

7	6	5	4	3	2	1	0
RSVD	EVCV_FR_F						
R-0h	R/W-0h						

**表 7-215. PWM1\_END\_VAL\_CONST\_FRAC\_F レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7	RSVD	R	0h	
6-0	EVCV_FR_F	R/W	0h	<p>エンド値の小数部 (周波数モード) または定数値の小数部 (DC モード)</p> <p>エンド値小数部分。</p> <p>これは常に使用されますが、フィールドは現在使用されているモードに基づいて変更されます。</p> <p>周波数除数の x/128 分数部分と解釈されます。</p> <p>PWM1_CTRL.MODE = デューティ サイクル。この場合、スイッチング周波数分周器 CONST_FREQ[10] の整数値になります。</p> <p>PWM1_CTRL.MODE = 周波数または静的の場合、このビットは使用されません</p>

#### 7.6.9.24 PWM1\_END\_VAL\_MSB レジスタ (オフセット = 411Ah) [リセット = 00h]

図 7-173 に、PWM1\_END\_VAL\_MSB を示し、表 7-216 に、その説明を示します。

概略表に戻ります。

ランプに設定されている場合は、変化する PWM 終了値 (目標値) の設定。静的 PWM 出力として構成されている場合、このレジスタは周波数除数の整数 MSB 部分に使用されます。

図 7-173. PWM1\_END\_VAL\_MSB レジスタ

7	6	5	4	3	2	1	0
RSVD				EV_FR [10]		END_VAL [9:8]	
R-0h				R/W-0h		R/W-0h	

表 7-216. PWM1\_END\_VAL\_MSB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-3	RSVD	R	0h	
2	EV_FR [10]	R/W	0h	終了値 (周波数のみ) これは多目的レジスタです。 PWM1_CTRL.MODE = デューティ サイクルまたは静的。これはスイッチング周波数の値 START_VAL[10] の整数値です。 PWM1_CTRL.MODE = 周波数の場合、このビットは使用されません
1-0	END_VAL [9:8]	R/W	0h	終了値 これは多目的レジスタです。 PWM 1_CTRL.MODE = デューティサイクルまたは静的である場合、これはスイッチング周波数除数 END_VAL[9:8] の整数値です。 PWM 1_CTRL.MODE = 周波数と PWM 1_CTRL の場合。 8B = 0 (10 ビット)。これはデューティ サイクル値 END_VAL[9:8] です。それ以外の場合は、この値は使用されません。 10 ビット モードの場合 、END_VAL/1024 = デューティ サイクルですが、 8 ビットモードではこのピンは使用されません

### 7.6.9.25 PWM1\_END\_VAL\_LSB レジスタ (オフセット = 411Bh) [リセット = 00h]

図 7-174 に、PWM1\_END\_VAL\_LSB を示し、表 7-217 に、その説明を示します。

概略表に戻ります。

ランプに設定されている場合は、変化する PWM 終了値 (目標値) の設定。静的 PWM 出力として構成されている場合、このレジスタは周波数除数の整数 LSB 部分に使用されます。

**図 7-174. PWM1\_END\_VAL\_LSB レジスタ**

7	6	5	4	3	2	1	0
END_VAL[7:0]							
R/W-0h							

**表 7-217. PWM1\_END\_VAL\_LSB レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-0	END_VAL[7:0]	R/W	0h	<p>終了値 これは多目的レジスタです。</p> <p>PWM 1_CTRL.MODE = デューティ サイクルまたは静的である場合、これはスイッチング周波数除数 END_VAL[7:0] の整数値です。</p> <p>PWM 1_CTRL.MODE = 周波数の場合、これはデューティ サイクル値 END_VAL[7:0] です。</p> <p>8 ビットモードの場合、END_VAL/256 = デューティ サイクルです。</p> <p>10 ビットモードの場合、END_VAL/1024 = デューティ サイクル</p>

### 7.6.9.26 PWM1\_PULSE\_STOP\_RAMP[y] レジスタ (オフセット = 411Ch + 式) [リセット = 00000000h]

図 7-175 に PWM1\_PULSE\_STOP\_RAMP[y] の制御を示し、表 7-218 にその説明を示します。

概略表に戻ります。

自動ストップランプに設定されている場合、これは開始パルスから生成される PWM パルスの数であり、その時点でモジュールが自動的にストップランプ コマンドを設定します。最下位アドレスはカウンタの MSB に対応します。

オフセット = 411Ch + (y \* 4h)、ここで y = 0h ~ 3h

図 7-175. PWM1\_PULSE\_STOP\_RAMP[y] レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_STOP_RAMP																															
R/W-0h																															

表 7-218. PWM1\_PULSE\_STOP\_RAMP[y] レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	PWM0_PULSE_STOP_RAMP	R/W	0h	指定された数の PWM パルスが生成されると、自動的にストップランプが開始されます。これは、PWM1_ACTION.AUTO_STOP ビットがセットされている場合にのみ使用されます。最下位アドレスは MSB です。



### 7.6.9.27 PWM1\_PULSE\_MAX[y] レジスタ (オフセット = 4120h + 式) [リセット = 00000000h]

図 7-176 に PWM1\_PULSE\_MAX[y] の制御を示し、表 7-219 にその説明を示します。

概略表に戻ります。

自動ストップ ランプに設定されている場合、これは許可される PWM パルスの最大数となります。PWM パルスの数がこの制限値に達すると、出力は直ちに停止します。最下位アドレスはカウンタの MSB に対応します。

オフセット = 4120h + (y \* 4h)、ここで y = 0h ~ 3h

**図 7-176. PWM1\_PULSE\_MAX[y] レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_MAX																															
R/W-0h																															

**表 7-219. PWM1\_PULSE\_MAX[y] レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
31-0	PWM0_PULSE_MAX	R/W	0h	指定された数の PWM パルスが生成されると、PWM 出力は無効化されます。 これは、PWM1_ACTION.AUTO_STOP ビットがセットされている場合にのみ使用されます。 最下位アドレスは MSB です。

## 7.6.9.28 PWM1\_ACTION レジスタ (オフセット = 4124h) [リセット = 00h]

図 7-177 に、PWM1\_ACTION を示し、表 7-220 に、その説明を示します。

概略表に戻ります。

図 7-177. PWM1\_ACTION レジスタ

7	6	5	4	3	2	1	0
RSVD			予約済み	AUTO_STOP	UDS	START	ストップ
R-0h			R-0h	R/W-0h	RH/W1S-0h	RH/W1S-0h	RH/W1S-0h

表 7-220. PWM1\_ACTION レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	
4	予約済み	R	0h	
3	AUTO_STOP	R/W	0h	自動ストップ ランプを使用 有効化されている場合、パルス カウントが指定値に達した場合です 0h = 自動ストップ ランプ動作は使用しません 1h = パルス カウント レジスタに基づいて、自動的にストップ ランプを開始します
2	UDS	RH/W1S	0h	定義済み開始を使用 開始値レジスタで定義されている開始点を使用します。 開始点が既知であることを確認する場合、または現在の値 (中間ランプまたは既存の終点) からランプを実行する場合に使用します。 START が設定されている状態でこのビットがセットされている場合、ランプは定義された開始点から始まります。 それ自体で設定された場合、ランプが開始されません 0h = 既存の現在の値を開始点として使用します 1h = 定義された開始点から開始します
1	START	RH/W1S	0h	開始 構成済みの設定に基づいて、定義済みのランプ プロファイルを開始します。 STOP も同時に設定されている場合、START は無視されます。 ビットはハードウェアによってクリアされます。
0	ストップ	RH/W1S	0h	停止 指定された停止点まで上昇し、その後 OFF になります。 START と STOP を同時に設定すると、STOP が発生します。 ビットはハードウェアによってクリアされます

### 7.6.9.29 PWM1\_IAS\_CTRL レジスタ (オフセット = 4130h) [リセット = 00h]

図 7-178 に、PWM1\_IAS\_CTRL を示し、表 7-221 に、その説明を示します。

概略表に戻ります。

**図 7-178. PWM1\_IAS\_CTRL レジスタ**

7	6	5	4	3	2	1	0
RSVD			IN_POL	GPIO_SEL		STOP_MODE	AS_EN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h

**表 7-221. PWM1\_IAS\_CTRL レジスタのフィールドの説明**

ビット	フィールド	タイプ	リセット	説明
7-5	RSVD	R	0h	予約済み
4	IN_POL	R/W	0h	入力極性 ストップ動作をトリガするために使用するレベルを選択します 0h = アクティブ Low 入力信号 1h = アクティブ High 入力信号
3-2	GPIO_SEL	R/W	0h	GPIO 選択 使用する GPIO を指定します。 動作するには、特殊機能の代わりに GPIO 機能として GPIO を構成する必要があります。 0h = GPIO0 1h = GPIO1 2h = GPIO7 3h = GPIO8
1	STOP_MODE	R/W	0h	ストップ モード GPIO によりトリガされたストップ イベントが発生すると、PWM 出力を直ちにデイスエーブルにするか、ストップ ランプを要求するかを選択します。 0h = GPIO トリガ イベントはストップ ランプを要求しています 1h = GPIO トリガ イベントにより、PWM 出力は即座に無効化されます
0	AS_EN	R/W	0h	入力自動停止のイネーブル  <div style="text-align: center;"><b>注</b></div> <div style="text-align: center;">この場合、AUTO_STOP ビットを設定する必要はなく、GPIO 入力に基づいて出力を停止するための別の機能です</div> 0h = GPIO 信号からの自動停止は有効ではありません 1h = GPIO によりトリガされた自動停止機能を有効にします

## 8 アプリケーションと実装

### 注

以下のアプリケーション情報は、テキサス・インスツルメンツの製品仕様に含まれるものではなく、テキサス・インスツルメンツはその正確性も完全性も保証いたしません。個々の目的に対する製品の適合性については、お客様の責任で判断していただくことになります。また、お客様は自身の設計実装を検証しテストすることで、システムの機能を確認する必要があります。

### 8.1 アプリケーション情報

TCAN5102-Q1 はコントロール エリア ネットワーク (CAN) フレキシブル データ (FD) ライト レスポンダ デバイスです。レスポンド ノードへの制御はすべて、コマンド ノードプロセッサからの CAN バスを介して行われるため、レスポンド ノードプロセッサやソフトウェアは不要です。このため、非常に柔軟性が高く、エンド ノード モーター制御など、多くのアプリケーションで使用できます。ここでは、TCAN5102 は TCAN11625 または TCAN11623 などの CAN FD トランシーバや DRV8889 などのステッピング モーター ドライバと接続します。ここでは、CANH と CANL を経由してモーターの制御をコマンド ノードから実行できます。TCAN5102-Q1 はこれらの指示と構成を理解し、ステッパ モーター ドライバを動作させることができます。この場合、リモート コントロール エッジ (RCE) でプロセッサ ソフトウェアのオーバーヘッドは不要です。

### 8.2 代表的なアプリケーション

RCE CAN FD ライト ステッパ モーター制御アプリケーション

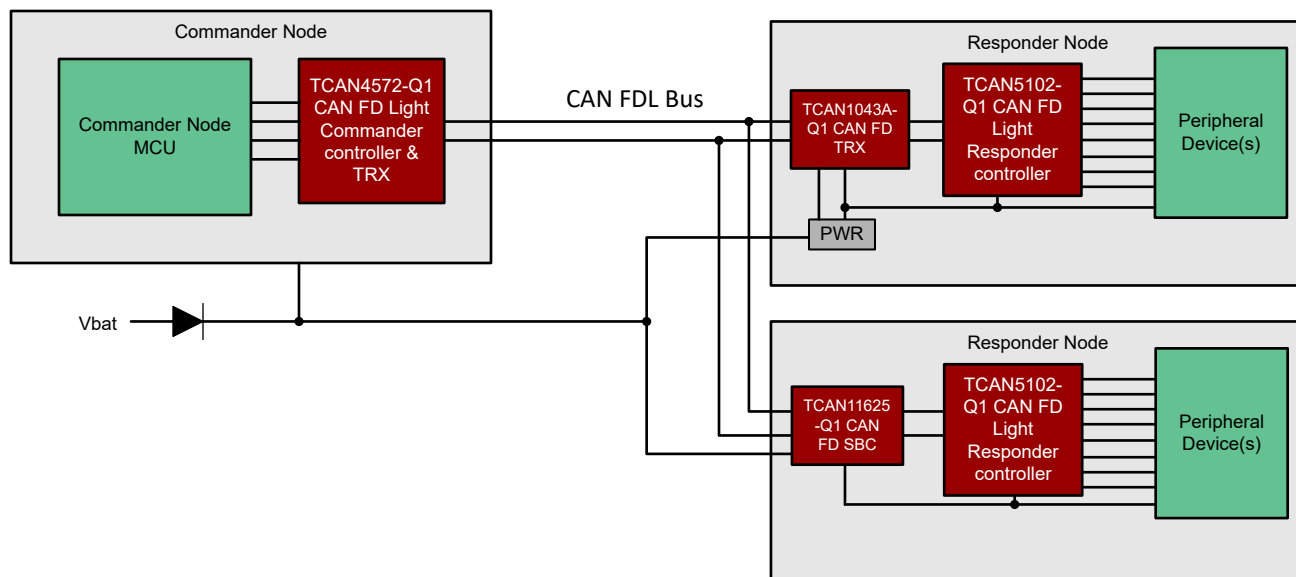


図 8-1. 代表的なコマンド レスポンダ バス アーキテクチャ

## 8.2.1 設計要件

表 8-1. 設計パラメータ

パラメータ	値
VDD	5V または 3V
SPI 構成	nSCS、SCLK、SDO、SDI
GPIO の構成	GPIO0 - 故障検出 GPIO1 - モーター ディスエーブル GPIO8 - モーター方向 GPIO11 (PWM) - 注入モーター 1 ステップ GPIO9 - スリープ イネーブル

## 8.2.2 詳細な設計手順

ステッパ モーター ドライバ RCE を実装するには、SPI モードと GPIO に合わせて TCAN5102-Q1 を適切に構成する必要があります。SPI コマンドおよび GPIO コマンドは、CAN からモーター ドライバに直接変換されます。[セクション 7.6.2](#) SPI データレート、極性、モードを対応するモーター ドライバ向けに構成するためのレジスタマップの概要を示します。GPIO ピンは、対応するレジスタを使用して構成することもできます。たとえば、ステッピング モーターの例では、GPIO11 を PWM として構成する必要があります。まず、GPIO11\_CFG を 1h (特殊機能 PWM0) に設定する必要があります。その後、ステッピング モーターの制御 / ステップ設定に応じて PWM0 を設定できます。レジスタ マップを[セクション 7.6.8](#) に示します。

(16)

## 8.3 電源に関する推奨事項

TCAN5102-Q1 は、単一の 5V または 3V 電源で動作するように設計されています。電源はおそらく Vbatt から供給され、TCAN11625 や TCAN11623 のような CAN FD SBC を使用して降圧できます。これらのデバイスは VIO/VCCOUT にそれぞれ 100mA または 70mA を供給でき、TCAN5102 (<2mA) を動作させるには十分です。その概要については、[セクション 8.2](#) を参照してください。

通常、1μF と 100nF コンデンサは、TCAN5102-Q1 の VDD のできるだけ近くに配置する必要があります。内部 デジタル コア レギュレータにフィルタ処理を行うため、DIGFLTR に 1μF コンデンサが必要です。これにより、スイッチ モード電源の出力に存在する電源電圧リップルを低減し、PCB 電源プレーンとトレースの抵抗とインダクタンスを補償するのに役立ちます。

## 8.4 レイアウト

### 8.4.1 レイアウトのガイドライン

実効ピア インダクタンスを最小化するため、バイパス コンデンサと保護デバイスの VCC およびグランド接続には少なくとも 2 つのピアを使用します。バイパス コンデンサとバルク コンデンサは、トランシーバの電源端子にできるだけ近づけて配置する必要があります。

高速パターンは、ピアとコーナーを最小限にして配線します。これにより、インピーダンスの変化量を減らすことができます。パターンを 90° 曲げる必要がある場合は、1 回の 90° カーブではなく、2 回の 45° カーブまたはアークを使用してください。高速信号パターンは、水晶発振子、オシレータ、外部クロック信号、スイッチングレギュレータ、取付穴、または磁気デバイスの近くに配置しないでください。信号ライン上のスタブは避けてください。

### 8.4.2 レイアウト例

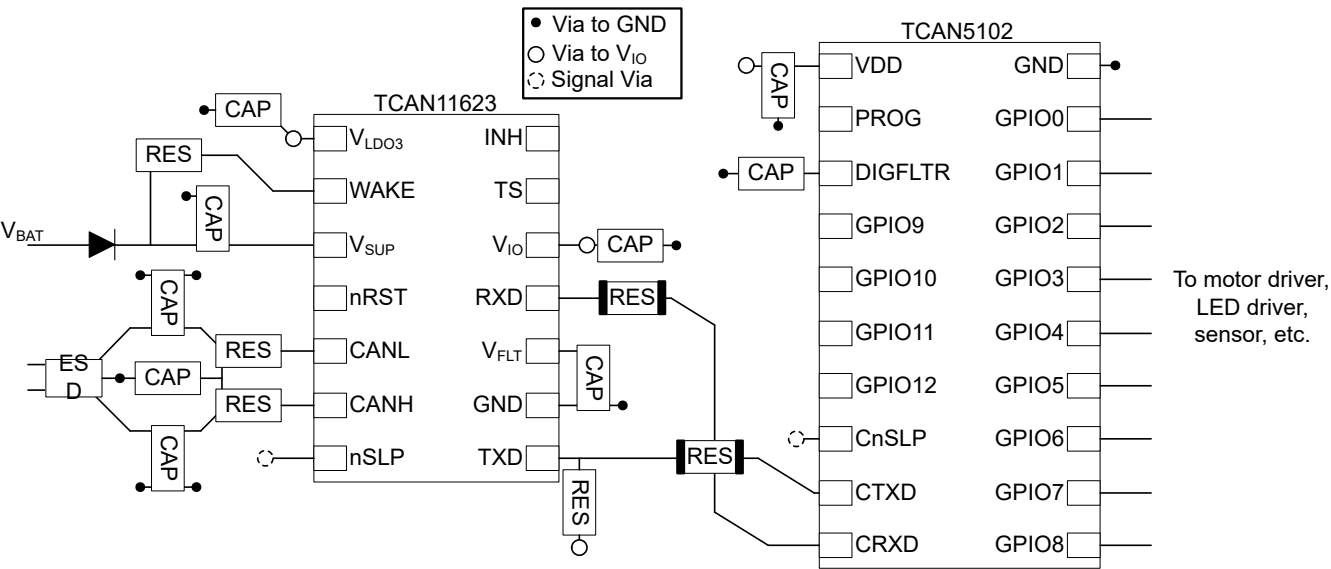


図 8-2. レイアウト例

## 9 デバイスおよびドキュメントのサポート

テキサス・インスツルメンツでは、幅広い開発ツールを提供しています。デバイスの性能の評価、コードの生成、ソリューションの開発を行うためのツールとソフトウェアを以下で紹介します。

### 9.1 ドキュメントのサポート

#### 9.1.1 関連資料

### 9.2 ドキュメントの更新通知を受け取る方法

ドキュメントの更新についての通知を受け取るには、[www.tij.co.jp](http://www.tij.co.jp) のデバイス製品フォルダを開いてください。[通知] をクリックして登録すると、変更されたすべての製品情報に関するダイジェストを毎週受け取ることができます。変更の詳細については、改訂されたドキュメントに含まれている改訂履歴をご覧ください。

### 9.3 サポート・リソース

**テキサス・インスツルメンツ E2E™ サポート・フォーラム**は、エンジニアが検証済みの回答と設計に関するヒントをエキスパートから迅速かつ直接得ることができる場所です。既存の回答を検索したり、独自の質問をしたりすることで、設計に必要な支援を迅速に得ることができます。

リンクされているコンテンツは、各寄稿者により「現状のまま」提供されるものです。これらはテキサス・インスツルメンツの仕様を構成するものではなく、必ずしもテキサス・インスツルメンツの見解を反映したものではありません。テキサス・インスツルメンツの[使用条件](#)を参照してください。

### 9.4 商標

テキサス・インスツルメンツ E2E™ is a trademark of Texas Instruments.  
すべての商標は、それぞれの所有者に帰属します。

### 9.5 静電気放電に関する注意事項



この IC は、ESD によって破損する可能性があります。テキサス・インスツルメンツは、IC を取り扱う際には常に適切な注意を払うことを推奨します。正しい取り扱いおよび設置手順に従わない場合、デバイスを破損するおそれがあります。ESD による破損は、わずかな性能低下からデバイスの完全な故障まで多岐にわたります。精密な IC の場合、パラメータがわずかに変化するだけで公表されている仕様から外れる可能性があるため、破損が発生しやすくなっています。

### 9.6 用語集

**テキサス・インスツルメンツ用語集** この用語集には、用語や略語の一覧および定義が記載されています。

## 10 改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

Changes from Revision A (October 2025) to Revision B (January 2026)	Page
・ タイトルを「TCAN5102-Q1 車載用 CAN FD ライトレ スポンダ、SPI、UART または I <sup>2</sup> C コントローラおよびプログラマブル PWM 搭載」に更新.....	1
・ DGS パッケージを削除.....	232
・ Added DGQ mechanical data.....	233

Changes from Revision * (September 2025) to Revision A (October 2025)	Page
・ シリコンに関するすべての記述を 4562 から 4572 に更新.....	1


## 11 メカニカル、パッケージ、および注文情報

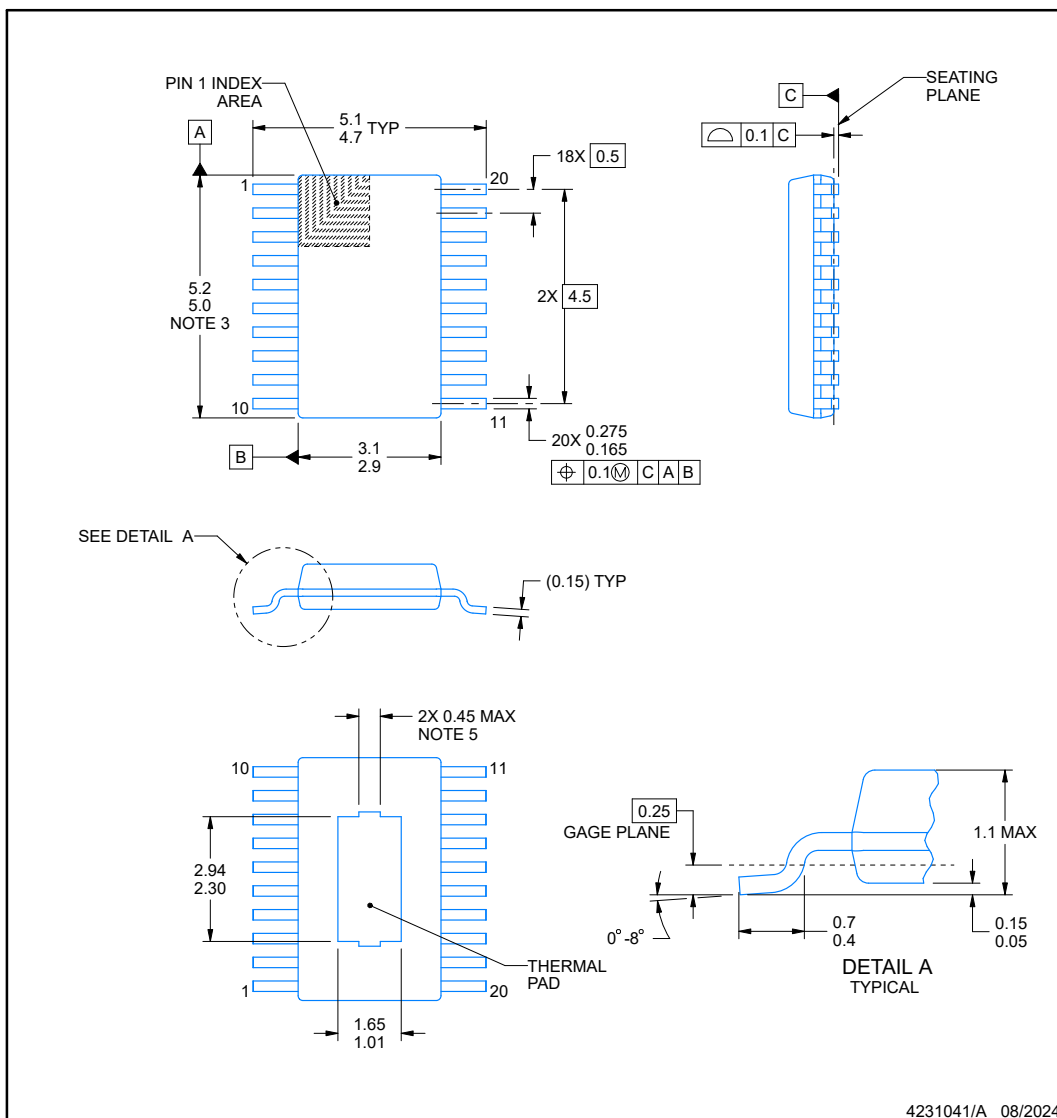
以降のページには、メカニカル、パッケージ、および注文に関する情報が記載されています。この情報は、指定のデバイスに使用できる最新のデータです。このデータは、予告なく、このドキュメントを改訂せずに変更される場合があります。本データシートのブラウザ版を使用されている場合は、画面左側の説明をご覧ください。



## 11.1 メカニカルデータ

### PACKAGE OUTLINE

**DGQ0020A**  **PowerPAD™ VSSOP - 1.1 mm max height**  
SMALL OUTLINE PACKAGE



#### NOTES:

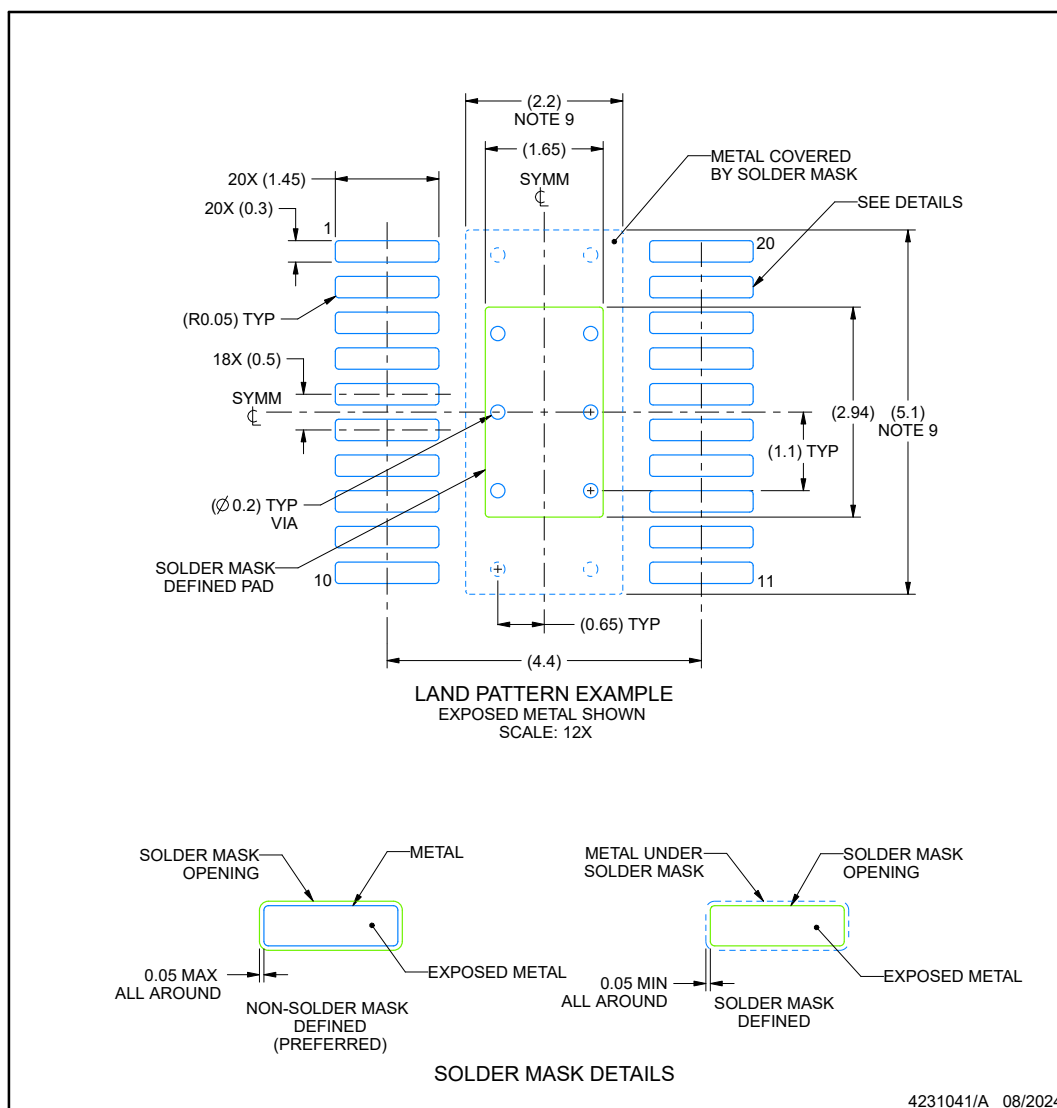
PowerPAD is a trademark of Texas Instruments.

1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.
3. This dimension does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.15 mm per side.
4. No JEDEC registration as of September 2020.
5. Features may differ or may not be present.

**EXAMPLE BOARD LAYOUT****DGQ0020A****PowerPAD™ VSSOP - 1.1 mm max height**

SMALL OUTLINE PACKAGE

ADVANCE INFORMATION



NOTES: (continued)

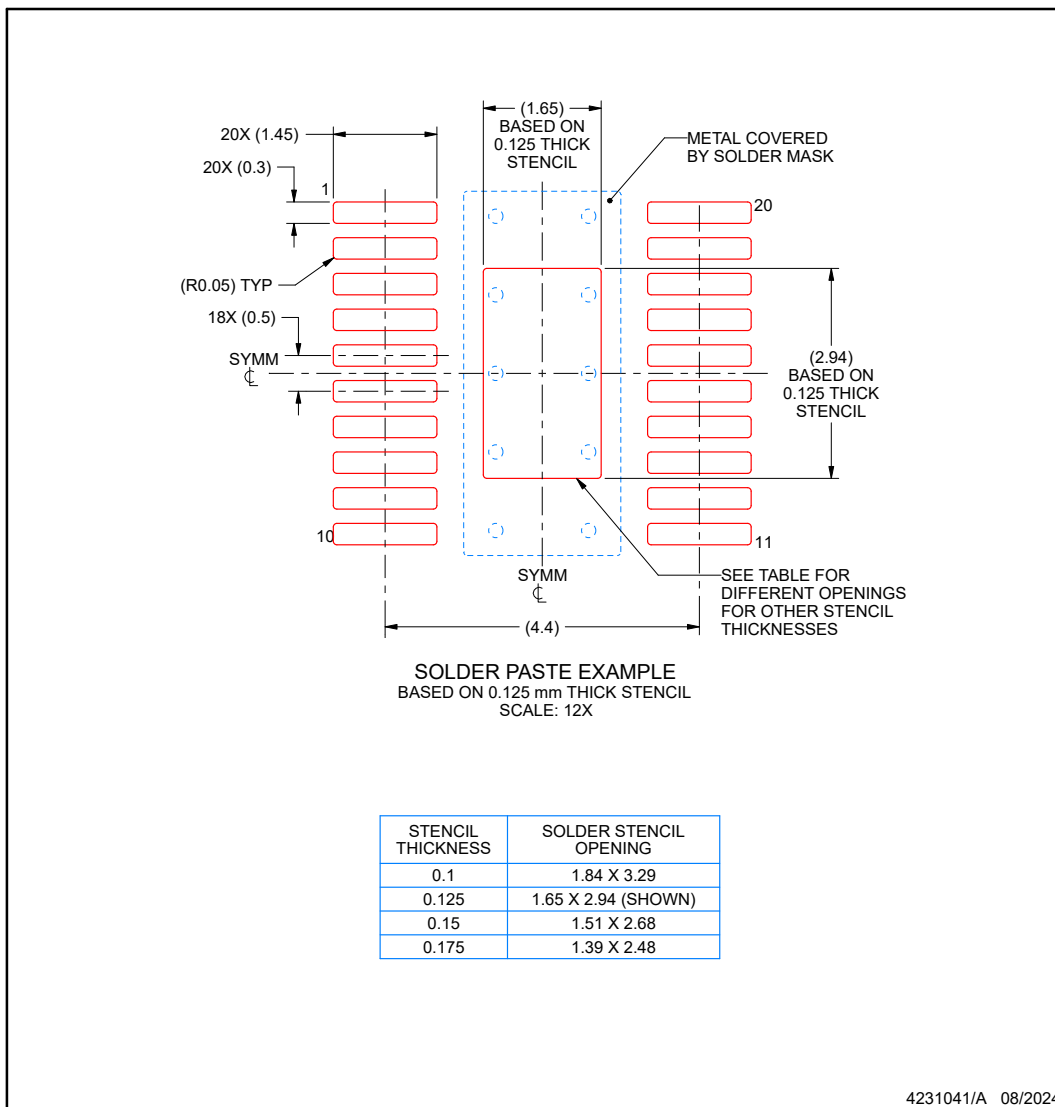
6. Publication IPC-7351 may have alternate designs.
7. Solder mask tolerances between and around signal pads can vary based on board fabrication site.
8. This package is designed to be soldered to a thermal pad on the board. For more information, see Texas Instruments literature numbers SLMA002 ([www.ti.com/lit/slma002](http://www.ti.com/lit/slma002)) and SLMA004 ([www.ti.com/lit/slma004](http://www.ti.com/lit/slma004)).
9. Size of metal pad may vary due to creepage requirement.
10. Vias are optional depending on application, refer to device data sheet. It is recommended that vias under paste be filled, plugged or tented.

## EXAMPLE STENCIL DESIGN

DGQ0020A

PowerPAD™ VSSOP - 1.1 mm max height

SMALL OUTLINE PACKAGE



NOTES: (continued)

11. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.
12. Board assembly site may have different recommendations for stencil design.

## PACKAGING INFORMATION

Orderable part number	Status (1)	Material type (2)	Package   Pins	Package qty   Carrier	RoHS (3)	Lead finish/ Ball material (4)	MSL rating/ Peak reflow (5)	Op temp (°C)	Part marking (6)
<a href="#">PTCAN5102DGQRQ1</a>	Active	Preproduction	HVSSOP (DGQ)   20	2500   LARGE T&R	-	Call TI	Call TI	-40 to 125	

(1) **Status:** For more details on status, see our [product life cycle](#).

(2) **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

(3) **RoHS values:** Yes, No, RoHS Exempt. See the [TI RoHS Statement](#) for additional information and value definition.

(4) **Lead finish/Ball material:** Parts may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

(5) **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

(6) **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

**Important Information and Disclaimer:**The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

## 重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含みいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、TI は一切の責任を拒否します。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月