

MAX17851

SPI-UART安全監視ブリッジ

概要

MAX17851 SPI-UART安全監視ブリッジは、SPIフォーマットから、アナログ・デバイセズのバッテリー管理データ・アクイジション・システムとのインターフェース用に特別に設計されたユニバーサル非同期レシーバ/トランスミッタ(UART)フォーマットに通信を変換します。安全監視ブリッジは、シングルおよびデュアル・デイジーチェーン・システム・アーキテクチャにおいて、最大4Mbpsのボー・レートで堅牢な通信を可能にします。高スループットのデュアルUARTシステムは、96セルのバッテリー電圧を1173us以内に読み取ることができます。

UARTは、組み込みUART通信ロックステップ安全対策ステート・マシン(LSSM)を介した、メッセージの破損、遅延、喪失、挿入の検出を行う自動車安全水準(ASIL)Dシステム向けのISO26262に適合しています。MAX17851は、ホスト・マイクロコントローラでのソフトウェア開発を簡素化するために、メッセージ・ペイロード内の全ての通信を自己検証します。

更に、MAX17851はホスト・マイクロコントローラの通信を監視し、あらゆる障害を検出します。ホストとの通信に障害が発生した場合、MAX17851は最初に自動的に通信の回復を試みます。失敗した場合、制御を行います。その場合、MAX17851は、オプションでデイジーチェーン・ネットワークを再設定し、バッテリーの状態を監視し続けることができます。これにより、安全なシステム運転を継続できます。最後に、MAX17851は、バッテリーの状態が事前に設定された範囲から外れた場合、コンタクタに信号を発するため、システム全体がフェールセーフ状態になります。

アプリケーション

- バッテリー管理システム(BMS)
- 電気自動車およびハイブリッド電気自動車(EV/HEV)
- エネルギー貯蔵システム(ESS)

利点と特長

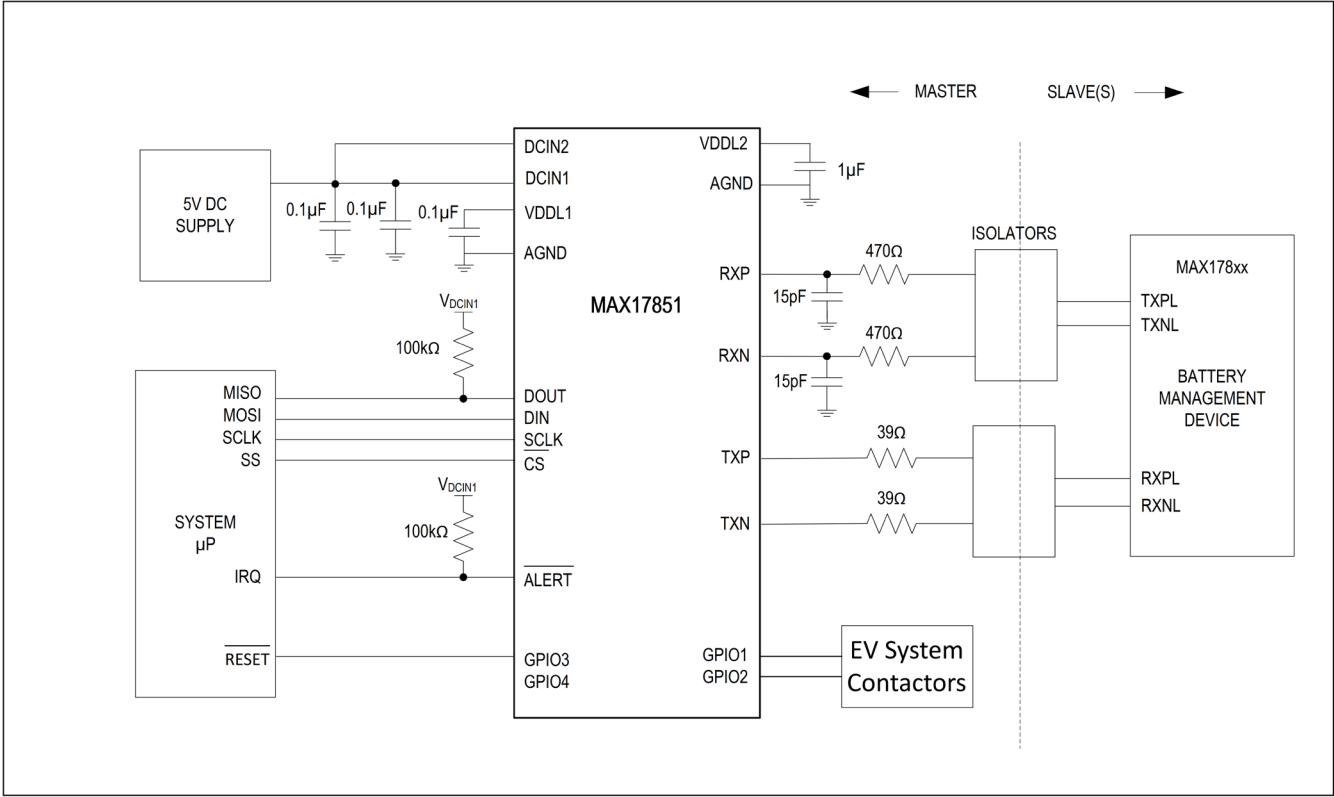
- アナログ・デバイセズのバッテリー管理UARTプロトコルに対応
- ASIL D機能安全要件に対応
 - ・ 簡素化されたホスト・レポート機能を備える、自動化されたオン・チップ通信検証
 - ・ フォルトの種類および位置に対する常時オンのデイジーチェーン・フォルト・ポーリング
- 冗長デイジーチェーン・コントローラを備える内蔵システム・ウォッチドッグ
 - ・ プログラマブルなシステム回復
 - ・ デイジーチェーン構成用のプログラマブルなメモリ
 - ・ プログラマブルなコンタクタ制御
- マイクロコントローラがスリープ・モード時のセル・バランシング機能を内蔵
 - ・ プログラマブルな割込み機能を備えた安全な長期バランシング・モニタリング
- 最大4Mbpsまで設定可能なUARTボー・レート
- 最大10MHzのSPIインターフェース
- 1.71V~5.5Vで動作
- 超低自己消費電流
- 動作温度範囲:-40°C~+125°C(AEC-Q100)

型番はデータシートの最後に記載されています。

※こちらのデータシートには正誤表が付属しています。当該資料の最終ページ以降をご参照ください。



簡略アプリケーション回路



目次

概要	1
アプリケーション	1
利点と特長	1
簡略アプリケーション回路図	2
絶対最大定格	8
パッケージ情報	8
TSSOP	8
電気的特性	8
標準動作特性	16
ピン配置	16
MAX17851A	16
端子説明	17
機能図	18
MAX17851の機能ブロック図	18
詳細説明	20
動作モード	20
コマンド動作モード	21
スリープ・モード	21
BMS安全モニタリング・モード	23
BMS安全モニタリング・モードおよび安全対策診断状態への移行	24
マイクロコントローラおよび電源の回復	25
BMS安全モニタリング動作	26
ALERTPACKETを用いたフォルト・ポーリング	27
BMS安全モニタリング・モードの終了	28
設定メモリ	28
データバスおよび設定メモリの検証	29
設定メモリの伝送	30
設定メモリのセットアップ	31
マスタ/スレーブのデバイス設定	32
BMS安全モニタリング・モードにおけるマスタ	32
BMS安全モニタリング・モードにおけるスレーブ	32
コマンド動作モードのスレーブ	32
スリープ・モードのスレーブ	33
診断および動作の検証	33
ウォッチドッグ	34
チャレンジ/応答モード	36
単純なウィンドウ・モード	36
LFSR/CRCコードの例	37
GPIO制御	37

目次(続き)

シリアル・ペリフェラル・インターフェース (SPI)	38
SPIトランザクション	38
SPIのタイミング	39
SPIでの読出しと書込みに対する制限	40
UARTインターフェース	40
バッテリー管理UARTプロトコル	41
UARTメッセージ	42
プリアンプル・キャラクタ	42
ストップ・キャラクタ	43
マンチェスター・エンコーディング	43
データ・タイプ	43
スレーブ・デバイス・アドレスの割り当て	43
UARTのメッセージ・データ・タイプ	44
一般的なUARTコマンド	44
UARTの動作	44
UARTの動作モード	45
送信バッファ	46
送信バッファのキュー	47
送信バッファのクリア	48
メッセージ長	48
ロード・キューの書込み	48
送信バッファのフィル	49
メッセージの送信	49
受信バッファ	49
受信バッファのクリア	50
メッセージの受信	51
メッセージの例外	51
メッセージの読出し	52
アラート・パケット・バッファ	52
自動およびユーザ指定アラート・カウンタ	53
データチェック・パーサ	54
ロックステップ安全対策の検証	55
デュアル・ロックステップ・データバス処理	56
アラートおよびフォルト	57
$\overline{\text{ALERT}}$ ピン、ステータス、アラート・フラグの動作	57
フォルト・タイマー	58
ハードウェア・フォルトの検出	58
レジスタ・フォルトの検出	58
MAX17851ユーザ・レジスタ・マップ	58

目次(続き)

レジスタの詳細.....	63
アプリケーション情報	120
システム構成.....	120
デュアルUART動作.....	120
デュアルUART MAX17851の初期化.....	121
デュアルUARTのBMSデータ・アクイジション・システムの初期化.....	124
最終的なMAX17851設定.....	126
シングルUART動作.....	126
シングルUARTのMAX17851の初期化.....	127
シングルUARTのBMSデータ・アクイジション・システムの初期化.....	129
最終的なMAX17851設定.....	130
デ이지チェーン設定用の設定メモリ・シーケンス.....	131
ウォッチドッグ設定	135
UART書込みおよび読出しのためのトランザクション・シーケンス.....	136
補助的なエラー・チェック機能.....	137
破損したプリアンブル・キャラクタ	137
破損メッセージの内容	138
ストップ・キャラクタの破損または喪失	138
意図しないプリアンブル.....	138
意図しないストップ・キャラクタ	138
フォルト処理のガイドライン	138
スリープ・モードの設定	142
TX_AUTO機能を用いたハードウェア・イン・ザ・ループ(HIL)テスト.....	142
UARTの物理レイヤ.....	145
UARTのシングルエンド・モード動作.....	145
UARTのトランス・カップリング	145
UARTの補足的ESD保護.....	146
電源に関する考慮事項	147
PCBレイアウトに関する推奨事項.....	148
レイアウト例(シルクスクリーン)	149
レイアウト例(金属)	150
標準アプリケーション回路.....	151
MAX17851のアプリケーション回路.....	151
型番	151
改訂履歴.....	152

図一覧

図1. SPIタイミング図	14
図2. 受信UARTのタイミング	14
図3. 送信UARTのタイミング	14
図4. TXバッファの書込みおよび読出し	15
図5. 動作モード	21
図6. スリープ・モードの状態遷移図	23
図7. BMS安全モニタリング・モードのフロー・チャート	24
図8. BMS安全モニタリング(NOMON)モードのフロー・チャート	27
図9. 設定メモリ	29
図10. 設定メモリ・データの伝送	31
図11. ウォッチドッグ動作	36
図12. システムのデータ・フロー	41
図13. UARTでのブリアンプルのタイミング	42
図14. UARTでのストップ・キャラクタのタイミング	43
図15. UARTでのマンチェスター・エンコーディング・データ・ニブル0hのタイミング	43
図16. UARTのデータ・フロー	45
図17. 送信バッファのメモリ・マップ	47
図18. 受信バッファのメモリ・マップ	50
図19. LSSMのデータパス	57
図20. デュアルUART	121
図21. シングルUART	127
図22. ハードウェア・イン・ザ・ループのテスト・セットアップ	143
図23. シングルエンド・モード	145
図24. UART信号のトランス・カップリング	146
図25. UARTトランスミットの補足的ESD保護	146
図26. UARTレシーバの補足的ESD保護(容量性カップリングと共に表示)	147
図27. 電源に関する考慮事項	148
図28. レイアウト例: 最上層のシルクスクリーン	149
図29. レイアウト例: 最下層のシルクスクリーン	149
図30. レイアウト例: 最上層の金属	150
図31. レイアウト例: 2層目の金属	150
図32. レイアウト例: 最下層の金属	150
図33. レイアウト例: 3層目の金属	150

 表一覧

表1. MAX17851の機能ブロックに対する電力分配.....	18
表2. データバスおよび設定メモリの実行時間.....	30
表3. 設定メモリからのWRITEALLコマンド・シーケンス.....	30
表4. デュアルUART下り経路でのREADALLコマンドのシーケンス(zはデバイス数).....	33
表5. GPIO制御.....	37
表6. SPIレジスタの一覧.....	39
表7. SPI通信の概要.....	40
表8. SPIでの読出しと書き込みに対する制限.....	40
表9. UARTのメッセージ・データ・タイプ.....	44
表10. 一般的なUARTコマンド.....	44
表11. UARTの動作モード.....	45
表12. キューのメモリ・マップ.....	48
表13. HELLOALLメッセージでロードされるキューの例.....	49
表14. ユーザ指定アライブ・カウンタを伴うコマンドの伝搬 (ALIVECOUNT_EN = 10).....	54
表15. 自動アライブ・カウンタを伴うコマンドの伝搬 (ALIVECOUNT_EN = 11).....	54
表16. LSSMステータス・バイトのエラー・マッピング.....	55
表17. マスタ/スレーブ・デバイスの設定シーケンス.....	122
表18. マスタ/スレーブのBMSデিজリーチェーンの初期化シーケンス.....	124
表19. マスタ/スレーブのUARTアラート・パケット設定.....	126
表20. シングルUARTデバイス設定シーケンス.....	128
表21. シングルUARTのBMSデিজリーチェーンの初期化シーケンス.....	129
表22. アラート・パケット設定.....	131
表23. 設定メモリのシーケンス.....	131
表24. ウォッチドッグ設定.....	135
表25. UART書き込みおよび読出しのためのトランザクション・シーケンス.....	136
表26. フォルト処理のガイドライン.....	138
表27. ハードウェア・イン・ザ・ループのテスト・シーケンス.....	143

Absolute Maximum Ratings

DCIN1 to AGND -0.3V to +6V
 DCIN2 to AGND -0.3V to +6V
 V_{DDL1} to AGND -0.3V to +2.2V
 V_{DDL2} to AGND -0.3V to +6V
 TXP, TXN to AGND -0.3V to $V_{DDL2} + 0.3V$
 DOUT to AGND -0.3V to $V_{DCIN1} + 0.3V$
 GPIO1, GPIO2, GPIO3, GPIO4 to AGND . -0.3V to $V_{DCIN1} + 0.3V$
 CS, DIN, SCLK, ALERT/ to AGND -0.3V to $V_{DCIN1} + 0.3V$

RXP, RXN to AGND..... -32V to +32V
 Maximum Continuous Current into Any Pin 20mA
 Continuous Power Dissipation (Multilayer Board) ($T_A = +70^\circ\text{C}$)
 20 TSSOP (derate 11.1mW/ $^\circ\text{C}$ above $+70^\circ\text{C}$)..... 887mW
 Operating Temperature Range..... -40°C to $+125^\circ\text{C}$
 Storage Temperature Range -55°C to $+150^\circ\text{C}$
 Junction Temperature (Continuous) $+150^\circ\text{C}$
 Soldering Lead Temperature (10s)..... $+300^\circ\text{C}$

Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Package Information

TSSOP

Package Code	U20+7C
Outline Number	21-0066
Land Pattern Number	90-0116
Thermal Resistance, Single-Layer Board:	
Junction to Ambient (θ_{JA})	91°C/W
Junction to Case (θ_{JC})	20°C/W
Thermal Resistance, Four-Layer Board:	
Junction to Ambient (θ_{JA})	73.8°C/W
Junction to Case (θ_{JC})	20°C/W

For the latest package outline information and land patterns (footprints), go to www.maximintegrated.com/jp/packages. Note that a “+”, “#”, or “-” in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/jp/thermal-tutorial.

Electrical Characteristics

($V_{DCIN1} > V_{DDL1}$, $V_{DCIN1} < 5.5V$, $V_{DCIN2} > V_{DDL2}$, $V_{DCIN2} < 5.5V$, $V_{DDL1} = 1.8V$, $V_{DDL2} = 3.3V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted, where $T_{MIN} = -40^\circ\text{C}$ and $T_{MAX} = +125^\circ\text{C}$. Typical values are at $T_A = +25^\circ\text{C}$. Operation is with the recommended application circuit.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER REQUIREMENTS						
DCIN1 Supply Voltage	V_{DCIN1}	$V_{DDL1} = 1.8V$	2.375		5.5	V
		$V_{DCIN1} = V_{DDL1}$	1.71	1.8	1.89	
DCIN2 Supply Voltage	V_{DCIN2}	$V_{DDL2} = 3.3V$ nominal	4.5		5.5	V
		$V_{DCIN2} = V_{DDL2}$	3.2		5.5	
DCIN1 Static Supply Current	I_{DCIN1_STATIC}	$f_{SCLK} = 0$, SDO not loaded		920		μA
Incremental DCIN1 Communication Supply Current	I_{DCIN1_COMM}	Continuous SPI Reads, $f_{SCLK} = 10\text{MHz}$, SDO loaded 20pF, Note: DCIN1 Current		310		μA
DCIN2 Static Supply Current	I_{DCIN2_STATIC}	UART Baud Rate = 2MHz, TXIDLEHZ, TX not active with 200pF load		10	20	μA

Electrical Characteristics (continued)

($V_{DCIN1} > V_{DDL1}$, $V_{DCIN1} < 5.5V$, $V_{DCIN2} > V_{DDL2}$, $V_{DCIN2} < 5.5V$, $V_{DDL1} = 1.8V$, $V_{DDL2} = 3.3V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted, where $T_{MIN} = -40^{\circ}C$ and $T_{MAX} = +125^{\circ}C$. Typical values are at $T_A = +25^{\circ}C$. Operation is with the recommended application circuit.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Incremental DCIN2 Communication Supply Current	I _{DCIN2_COMM}	UART Baud Rate = 2MHz, TXIDLEHIZ, TXL not active with 200pF		880	910	uA
V _{DDL1} REGULATOR						
Output Voltage	V _{DDL1}	0mA < I _{VDDL1} < 10mA, 2.375V < V _{DCIN1} < 5.5V	1.71	1.8	1.89	V
Short-Circuit Current	I _{DDL1_SC}	V _{DDL1} = AGND	5		50	mA
POR Threshold	V _{DDL1_PORRISE}	V _{DDL1} rising	1.3	1.5	1.65	V
POR Hysteresis	V _{DDL1_PORHYS}		20	50		mV
V _{DDL2} REGULATOR						
Output Voltage	V _{DDL2}	0mA < I _{VDDL2} < 10mA, 4.5V < V _{DCIN2} < 5.5V	3.2	3.3	3.4	V
Short-Circuit Current	I _{DDL2_SC}	V _{DDL2} = AGND	13			mA
LOGIC INPUTS ($\overline{CS}$, DIN, SCLK)						
Input High Voltage	V _{IH}	V _{DCIN1} > 2.375V	0.7 x V _{DCIN1}			V
		1.71V < V _{DCIN1} < 1.89V	0.8 x V _{DCIN1}			
Input Low Voltage	V _{IL}	V _{DCIN1} > 2.375V	0.3 x V _{DCIN1}			V
		1.71V < V _{DCIN1} < 1.89V	0.2 x V _{DCIN1}			
Input Leakage Current	I _{IN}	V _{in} = 0V or V _{DCIN1} (Note 12)	±1			µA
Internal Impedance	R _{PD}	DIN, SCLK pull down to GND (Notes 13, 14)	40	100	160	kΩ
	R _{PU}	$\overline{CS}$ pull up to V _{DCIN1} (Notes 13, 14)	40	100	160	
Input Capacitance	C _{IN}			10		pF
Hysteresis Voltage	V _H			250		mV
LOGIC OUTPUTS (DOUT, $\overline{ALERT}$)						
Output High Voltage	V _{OH}	V _{DCIN1} >2.375V, I _{SOURCE} = 4mA, Note 10	V _{DCIN1} -0.2			V
		1.71V < V _{DCIN1} < 1.89V, I _{SOURCE} = 1.5mA, Note 10	V _{DCIN1} -0.2			
Output Low Voltage	V _{OL}	V _{DCIN1} >2.375V, I _{SINK} = 4mA	0.2			V
		1.71V < V _{DCIN1} < 1.89V, I _{SINK} = 1.5mA	0.2			
Output Tristate Leakage	I _{OZ}	V _{DOUT} = 0 and 5V, V $\overline{ALERT}$ = 5V	±1			µA
Output Tristate Capacitance	C _{OZ}			10		pF

Electrical Characteristics (continued)

($V_{DCIN1} \geq V_{DDL1}$, $V_{DCIN1} < 5.5V$, $V_{DCIN2} \geq V_{DDL2}$, $V_{DCIN2} < 5.5V$, $V_{DDL1} = 1.8V$, $V_{DDL2} = 3.3V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted, where $T_{MIN} = -40^{\circ}C$ and $T_{MAX} = +125^{\circ}C$., Typical values are at $T_A = +25^{\circ}C$. Operation is with the recommended application circuit.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
GENERAL PURPOSE INPUTS/OUTPUTS (GPIO1, GPIO2, GPIO3, GPIO4)						
Output Leakage Current	I _{GPO_LKG}	V _{GPIO_n} = 0 and 5V, GPIO disabled	-1		1	μA
Output High Voltage	V _{OH}	V _{DCIN1} > 2.375V, I _{SOURCE} = 4mA	V _{DCIN1} - 0.2			V
		1.71V < V _{DCIN1} < 1.89V, I _{SOURCE} = 1.5mA	V _{DCIN1} - 0.2			
Output Low Voltage	V _{OL}	V _{DCIN1} > 2.375V, I _{SINK} = 4mA	0.2			V
		1.71V < V _{DCIN1} < 1.89V, I _{SINK} = 1.5mA	0.2			
Input High Voltage	V _{IH}	V _{DCIN1} > 2.375V	0.7 x V _{DCIN1}			V
		1.71V < V _{DCIN1} < 1.89V	0.8 x V _{DCIN1}			
Input Low Voltage	V _{IL}	V _{DCIN1} > 2.375V	0.3 x V _{DCIN1}			V
		1.71V < V _{DCIN1} < 1.89V	0.2 x V _{DCIN1}			
Pulldown Resistance	R _{GPIO}	GPIO _n _CFG configured as input	2M			Ω
Output Short Circuit Current	I _{OSS}	I _{SINK} , V _{GPIO_n} = V _{DCIN1} = 5V	50			mA
POWER AND GROUND FAULT DETECTION						
Open Detection Voltage (V _{DDL1})	V _{ALRTVDDL1}		1.62	1.65	1.68	V
Open Detection Voltage (V _{DDL2})	V _{ALRTVDDL2}		2.90	3.00	3.10	V
Over Voltage Detection (V _{DCIN1})	V _{ALRTOV_DCIN1}		5.6	5.7	5.8	V
Overvoltage Detection Hysteresis (V _{DCIN1})	V _{DCIN1OV_HYS}		100		150	mV
Over Voltage Detection (V _{DCIN2})	V _{ALRTOV_DCIN2}		5.6	5.7	5.8	V
Overvoltage Detection Hysteresis (V _{DCIN2})	V _{DCIN2OV_HYS}		100		150	mV
OSCILLATORS						
LF Oscillator Frequency	f _{LFOSC}		32.11	32.768	33.42	kHz
HF Oscillator Frequency	f _{HFOSC}		62.72	64	65.28	MHz
UART INPUTS (RXP, RXN)						
Differential Input High Threshold	V _{TH}	Note 1	V _{DDL1} /2 - 400mV	V _{DDL1} /2	V _{DDL1} /2 + 400mV	V
Differential Input Zero-Crossing Threshold	V _{ZC}	Note 1	-400	0	400	mV
Differential Input Low Threshold	V _{TL}	Note 1	-V _{DDL1} /2 - 400mV	-V _{DDL1} /2	-V _{DDL1} /2 + 400mV	V

Electrical Characteristics (continued)

($V_{DCIN1} > V_{DDL1}$, $V_{DCIN1} < 5.5V$, $V_{DCIN2} > V_{DDL2}$, $V_{DCIN2} < 5.5V$, $V_{DDL1} = 1.8V$, $V_{DDL2} = 3.3V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted, where $T_{MIN} = -40^{\circ}C$ and $T_{MAX} = +125^{\circ}C$. Typical values are at $T_A = +25^{\circ}C$. Operation is with the recommended application circuit.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Differential Input Hysteresis	V _{HYST}	Note 1	30	80	160	mV
Common-Mode Voltage Bias	V _{CM}		V _{DDL1} /2			V
Input Capacitance	C _{IN}		2			pF
Leakage Current	I _{LKG_RX}	V _{RX} = 0.9V	±1.0			μA
Input Resistance to Common Mode Voltage	R _{RXIN}		0.8			MΩ
UART OUTPUTS (TXP, TXN)						
Output Low Voltage	V _{OL}	I _{OL} = -20mA	V _{AGND} + 0.4			V
Output High Voltage	V _{OH}	I _{OH} = 20mA	V _{DDL2} - 0.4			V
UART TIMING						
Bit Period Except for Second STOP Bit	t _{BIT}	f _{UART} = 4Mbps, (Note 2, Note 3)	16			1/f _{HFOSC}
Bit Period Except for Second STOP Bit	t _{BIT}	f _{UART} = 2Mbps (Note 3, Note 4)	32			1/f _{HFOSC}
		f _{UART} = 1Mbps (Note 3, Note 4)	64			
		f _{UART} = 0.5Mbps (Note 3, Note 4)	128			
Second STOP Bit Period	t _{STOPBIT}	Note 2, Note 3	1.125			t _{BIT}
Tx Idle to START Setup Time	t _{TXSTSU}	Note 2, Note 3, Note 4	0.5			t _{BIT}
STOP Hold Time to Idle	t _{SPHD}	Note 2, Note 3	0.5			t _{BIT}
Rx Minimum Idle Time (STOP Bit to START Bit)	t _{RXIDLESPST}	(Note 2, Note 3)	1			t _{BIT}
Tx Idle Time	t _{TXIDLESPST}	(Note 1, Note 2)	0.5			t _{BIT}
Rx Fall Time	t _{FALL}	Note 2, Note 3, Note 4	0.5			t _{BIT}
Rx Rise Time	t _{RISE}	(Note 2, Note 3, Note 4)	0.5			t _{BIT}
UART MESSAGE TIMING						
SPI Command to Tx Valid Propagation Delay	t _{TX}	(Note 5)	125		250	ns
Tx Valid to Rx Valid Up Stack Delay	t _{RXUP}	(Note 6)			n x t _{PROP}	
Tx Valid to Rx Valid Down Stack Delay	t _{RXDN}	(Note 6)			n x t _{PROP}	
End of STOP Character to RX_STOP_ALRT Flag True	t _{ALERT}	(Note 7)			2	t _{BIT}
SPI START to UART Slave Device Register Write Delay	t _{REGWR}	(Note 8, Note 9)			8 / f _{SCLK} + 130 x t _{BIT} + n x t _{PROP}	

Electrical Characteristics (continued)

($V_{DCIN1} > V_{DDL1}$, $V_{DCIN1} < 5.5V$, $V_{DCIN2} > V_{DDL2}$, $V_{DCIN2} < 5.5V$, $V_{DDL1} = 1.8V$, $V_{DDL2} = 3.3V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted, where $T_{MIN} = -40^{\circ}C$ and $T_{MAX} = +125^{\circ}C$. Typical values are at $T_A = +25^{\circ}C$. Operation is with the recommended application circuit.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SPI TIMING						
SCLK Frequency	f_{SCLK}	$V_{DCIN1} > 2.375V$ (Note 15)			10	MHz
SCLK Period	t_{CP}		100			ns
SCLK Pulse Width High	t_{CH}		40			ns
SCLK Pulse Width Low	t_{CL}		40			ns
CSB Fall to SCLK Rise Setup Time	t_{CSS0}	To first SCLK rising edge (RE).	40			ns
CSB Fall to SCLK Rise Hold Time	t_{CSH0}	Applies to inactive RE preceding first RE.	25			ns
SCLK Rise to CSB Rise Hold Time	t_{CSH1}	Applies to 8+8*n RE.	25			ns
CSB Pulse Width High	t_{CSPW}		400			ns
CSB Pulse Width High After SWPOR	t_{CSPWSP}	Applies after an accepted/executed SWPOR command.	100			us
SDI to SCLK Rise Setup Time	t_{DS}		10			ns
SDI to SCLK Rise Hold Time	t_{DH}		10			ns
SCLK Fall to SDO Transition	t_{DOT}	$V_{DCIN1} > 2.375V$, $C_{LOAD} = 20pf$			40	ns
SCLK Fall to SDO Hold	t_{DOH}	$C_{LOAD} = 0pf$	2			ns
CSB Fall to SDO Transition	t_{DOE}	$V_{DCIN1} > 2.375V$, $C_{LOAD} = 20pf$			40	ns
CSB Rise to SDO Hi-Z	t_{DOZ}	$V_{DCIN1} > 2.375V$, Output disable time			40	ns

Note 1: Differential signal ($V_{RXP} - V_{RXN}$) where V_{RXP} , V_{RXN} do not exceed a common-mode voltage range of $\pm 25V$.

Note 2: All parameters measured based on differential signal.

Note 3: Guaranteed by design and not production tested.

Note 4: Fall time measured 90% to 10%, rise time measured 10% to 90%.

Note 5: Measured from falling edge of 8th SCLK cycle of the WR_NXT_LD_Q SPI command byte (B0h).

Note 6: t_{PROP} is the maximum propagation delay through a slave device in a given direction. Refer to the UART slave device data sheet for the actual delay. The number of UART slave devices is denoted by n.

Note 7: Measured from end of 10th bit of stop character.

Note 8: Parameter t_{REGWR} is the minimum amount of time needed to write a register in the nth slave device of the daisy-chain. It is measured from the start of the SPI transaction WR_NXT_LD_Q (B0h) that initiates transmission of a WRITEALL message to when the nth device receives a valid WRITEALL message. For example, for 4MHz SPI frequency, 2Mbps UART baud rate, $n = 10$ and $t_{PROP} = 3 \times t_{BIT}$, $t_{REGWR} = 2\mu s + 65\mu s + 15\mu s = 82\mu s$.

Note 9: Computation of t_{REGWR} consists of three terms: 1) duration of the SPI transaction, 2) partial duration of the UART message, and 3) propagation delay of the UART message. The first term equals the number of bits in the SPI transaction (8) $\times$ the SPI bit time ($1 / f_{SCLK}$). The second term equals the time from the start of the WRITEALL message to the first STOP bit of the last PEC nibble. The last PEC nibble is the 11th character in the message. With each character lasting 12 UART bit times, there are $11 \times 12 = 132$ bit times from the start of the message to the end of the last PEC nibble. Since the write occurs just before the two STOP bits of the 11th character, the term is actually $130 \times t_{BIT}$. The third term is the propagation delay required for the WRITEALL message to get to the nth device.

Note 10: V_{OH} specification for $\overline{ALERT}$ is determined by the external pullup resistor and leakage current of the network.

Note 11: Static logic inputs with $V_{IL} = AGND$ and $V_{IH} = V_{DCIN1}$. CSB = V_{IH} (if pullup is active).

Note 12: No internal safety pullup/pulldown impedances active. Input buffers only.

Note 13: Internal safety pullup/pulldown impedances available with enable function.

Note 14: If pullup is supported, note CSB connection and diode to V_{DDL1} . This diode is present regardless of enable mode.

Note 15: Applications must afford time for the device to drive data on the SDO bus and meet the μC setup time prior to the μC latching in the result on the following SCLK rising edge. In practice, this can be determined by loading and μC characteristics, and the relevant t_{DOT}/t_{DOE} .

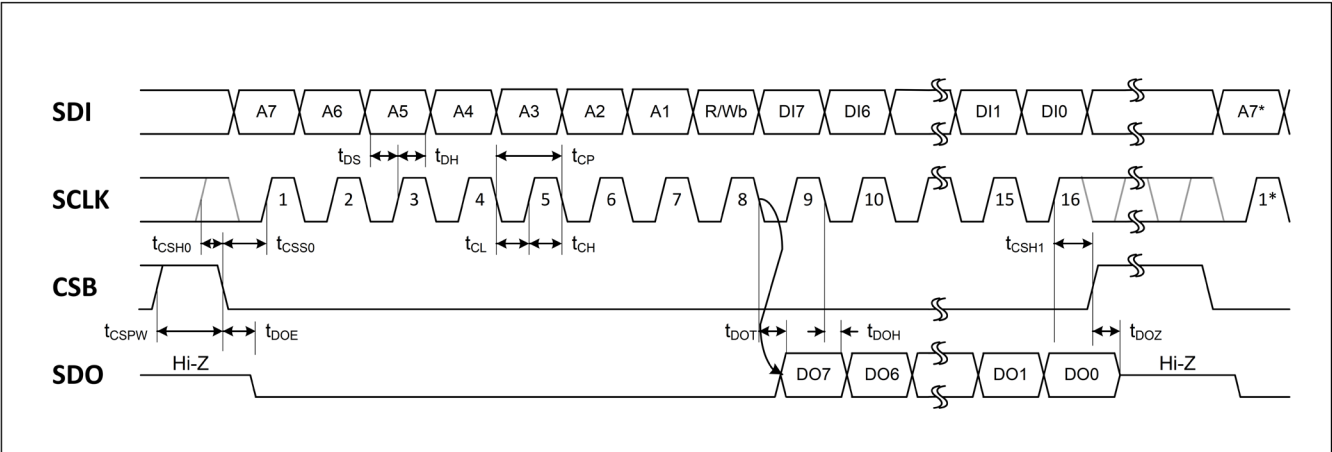


図1. SPIタイミング図

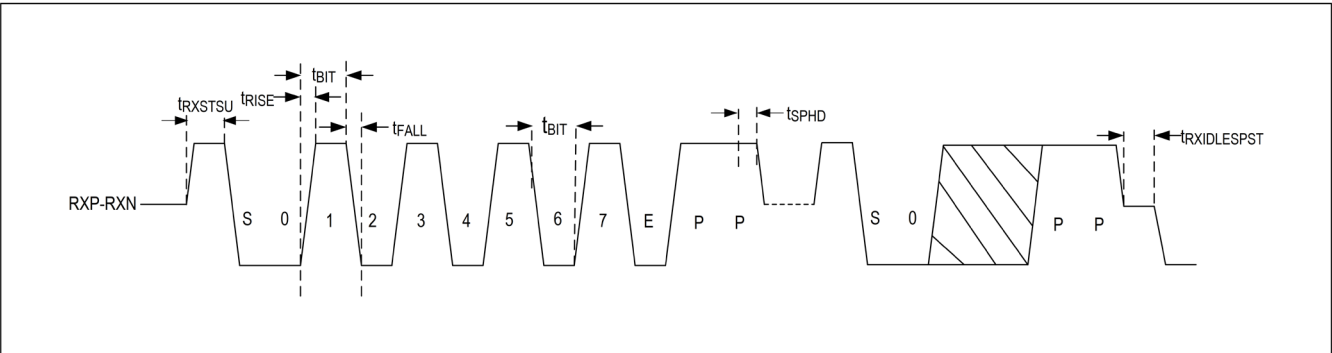


図2. 受信UARTのタイミング

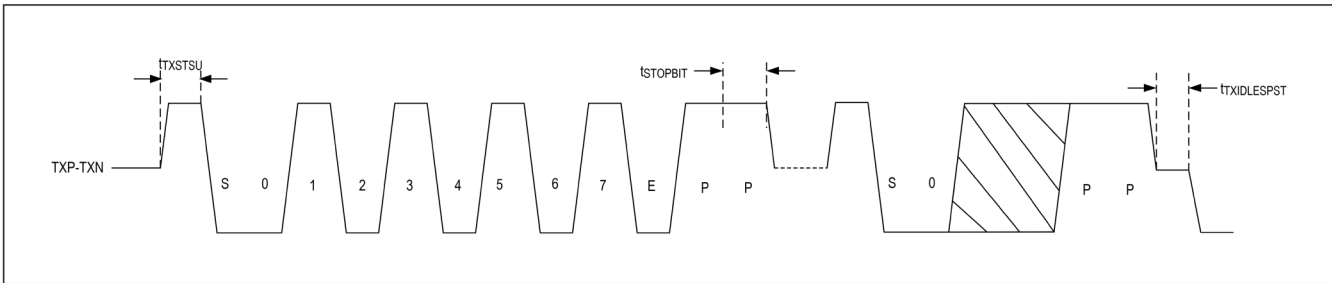


図3. 送信UARTのタイミング

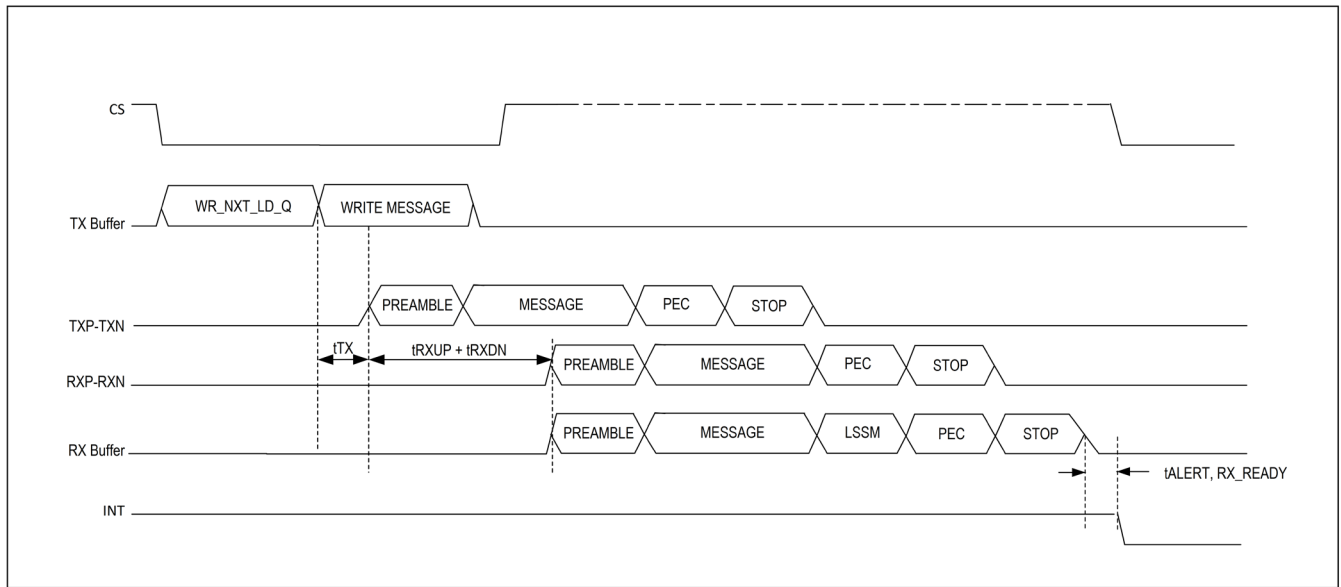


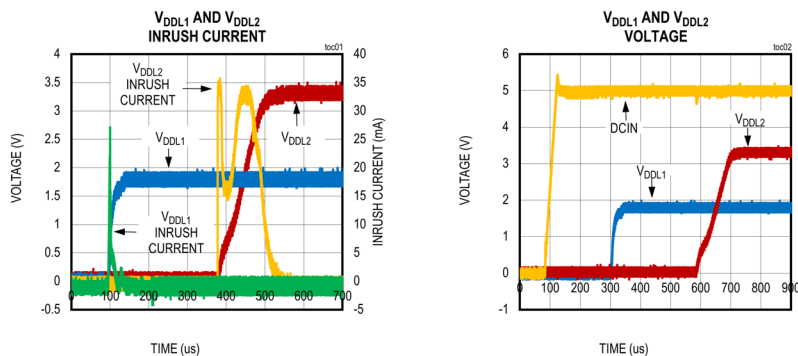
図4. TXバッファの書き込みおよび読出し

注:DCIN1の電流: DCIN1の総電流は、IDCIN1_STATICとIDCIN1_COMMの和です。IDCIN1_COMMは、SPIの動作周波数、SDOの出力データ、DCIN1の電圧、各通信ラインの合計容量に依存します。使用したデータ・パターンは、5VのDCIN1を用いた平均50%の遷移のデータ・パターンを仮定しています。

注:DCIN2の電流: DCIN2の総電流は、IDCIN2_STATICとIDCIN2_COMMの和です。IDCIN2_COMMは、UARTのボー・レート、UARTの出力データ、DCIN2の電圧、各通信ラインの合計容量に依存します。使用したデータ・パターンは、3.3VのDCIN2を用いた平均67%の遷移のデータ・パターンを仮定しています。

標準動作特性

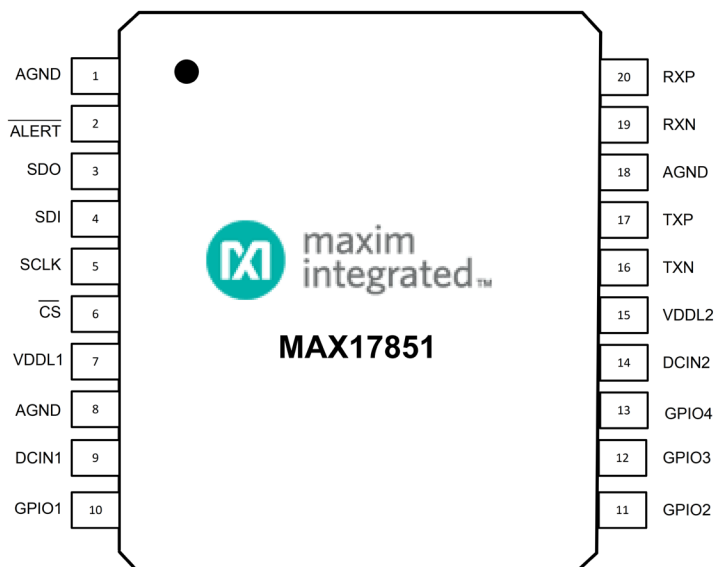
(特に指定にない限り、DCIN1 = 3.3V、DCIN2 = 5.0V、 $T_A = +25^{\circ}\text{C}_o$)



ピン配置

MAX17851A

Top View



端子説明

ピン	名称	説明	タイプ
1	AGND	アナログ・グランド。電源グランドに接続します。	Ground
2	$\overline{\text{ALERT}}$	アクティブ・ロー、オープン・ドレインの割込み出力。アプリケーションの要件に従ってこのピンにプルアップ抵抗を接続します。何らかの割込みフラグがセットされた場合、このピンがアサートされます。	Output
3	SDO	SPIデータ出力。SPIマスタのSDI/MISO入力に接続します。 $\overline{\text{CS}}$ がデアサートされるとこの出力はトライステートになります。 $\overline{\text{CS}}$ がアサートされると、このピンは、DCIN1 電源とAGND電源の間で駆動されます。	Output
4	SDI	SPIデータ入力。SPIマスタのSDO/MOSI出力に接続します。5Vに対応可能です。	Input
5	SCLK	SPIクロック入力。SPIマスタのSCLK出力に接続します。5Vに対応可能です。	Input
6	$\overline{\text{CS}}$	アクティブ・ローのSPIチップ・セレクト入力。SPIマスタのスレーブ・セレクト出力に接続します。これをアサートするとSPIポートが有効化されます。5Vに対応可能です。	Input
7	V_{DDL1}	1.8V LDOの電源出力および内部発振器とロジック用の電源出力。直接給電するには、DCIN1に接続します。アプリケーション回路に従ってデカップリングします。	Power
8	AGND	アナログ・グランド。電源グランドに接続します。	Ground
9	DCIN1	1.8V LDOレギュレータ用の電源入力。アプリケーション回路に従ってデカップリングします。	Power
10	GPIO1	汎用I/O。デフォルト設定は汎用入力です。ハードウェアの冗長性を確保するために、このピンはGPIO2と並列に用いることができます。 使用しない場合、ピンは無接続のままにすることができます。詳細については、 GPIO制御 のセクションを参照してください。	Input/Output
11	GPIO2	汎用I/O。デフォルト設定は汎用入力です。ハードウェアの冗長性を確保するために、このピンはGPIO1と並列に用いることができます。 使用しない場合、ピンは無接続のままにすることができます。詳細については、 GPIO制御 のセクションを参照してください。	Input/Output
12	GPIO3	汎用I/O。デフォルト設定は汎用入力です。 使用しない場合、ピンは無接続のままにすることができます。詳細については、 GPIO制御 のセクションを参照してください。	Input/Output
13	GPIO4	汎用I/O。デフォルト設定は汎用入力です。 使用しない場合、ピンは無接続のままにすることができます。詳細については、 GPIO制御 のセクションを参照してください。	Input/Output
14	DCIN2	3.3V LDOレギュレータ用の電源入力。アプリケーション回路に従ってデカップリングします。	Power
15	V_{DDL2}	3.3V LDOの電源出力およびUART用の電源出力。直接給電するには、DCIN2に接続します。アプリケーション回路に従ってデカップリングします。	Power
16	TXN	UARTトランスミッタ負出力。アプリケーション回路に従ってUARTスレーブ・デバイスのRXNポートに接続します。このピンは、 V_{DDL2} 電源とAGND電源の間で駆動されます。	Output
17	TXP	UARTトランスミッタ正出力。アプリケーション回路に従ってUARTスレーブ・デバイスのRXPポートに接続します。このピンは、 V_{DDL2} 電源とAGND電源の間で駆動されます。	Output
18	AGND	アナログ・グランド。電源グランドに接続します。	Ground
19	RXN	UARTレシーバ負入力。アプリケーション回路に従ってUARTスレーブ・デバイスのTXNポートに接続します。	Input
20	RXP	UARTレシーバ正入力。アプリケーション回路に従ってUARTスレーブ・デバイスのTXPポートに接続します。	Input

機能図

MAX17851の機能ブロック図

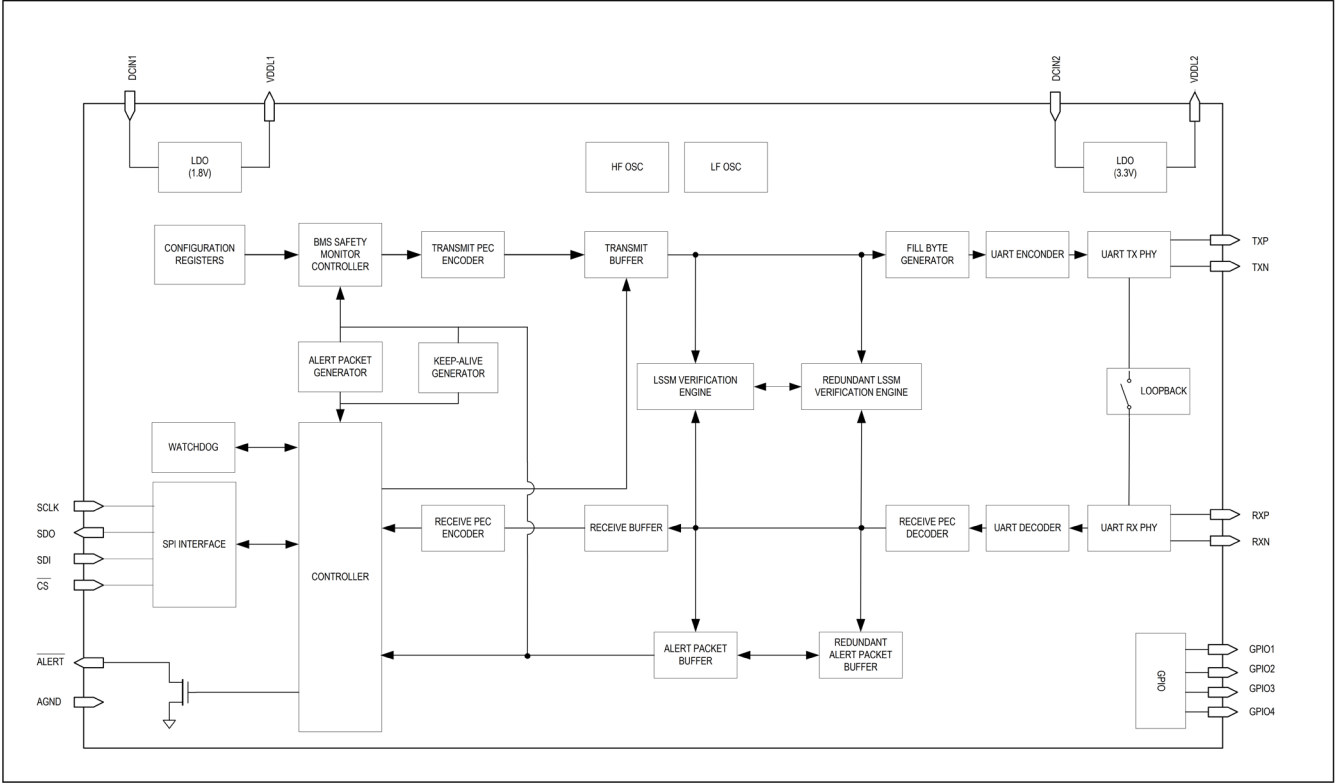


表1. MAX17851の機能ブロックに対する電力分配

BLOCK	SUPPLY
Oscillators	VDDL1
Configuration Registers	
BMS Safety Monitor Controller	
Transmit PEC Encoder	
Alert Packet Generator	
Keep Alive Generator	
Watchdog	
Digital Controller	
Transmit Buffer	
Fill Byte Generator	
UART Encoder	
LSSM/Redundant LSSM	
Alert Packet Buffer/Redundant Alert Packet Buffer	
UART Decoder	
Receive PEC Decoder	
Receive Buffer	

表1. MAX17851の機能ブロックに対する電力分配(続き)

BLOCK	SUPPLY
Receive PEC Encoder	
UART RX Phy	
UART TX Phy	
SPI Interface	V _{DDL2}
ALERT [¯] Output	DCIN1
GPIO	
V _{DDL1} LDO Regulator	
V _{DDL2} LDO Regulator	
	DCIN2

詳細説明

MAX17851は、標準的なSPIポートを備えたホスト・コントローラが、アナログ・デバイスズのバッテリー管理UARTプロトコルを用いる1つ以上のバッテリー管理モニタとの間で、信頼性の高い通信を行うことを可能にします。ホスト・コントローラと併用することで、MAX17851は、スレーブ・デバイスとの通信を行うマスタとなります。

MAX17851には、以下に示すように、際立った特徴がいくつかあり、自動車の高い安全完全性を達成する上で最適なデバイスになっています。

- 簡素なホスト検証に対しUARTデータの完全性を確保できる、デュアル自動ロックステップUARTプロセス・コア(ロックステップ安全対策(LSSM))。詳細については、[ロックステップ安全対策の検証](#)のセクションを参照してください。
- ホスト・マイクロコントローラを回復し、失敗した場合には、再構成してデジタイゼーションをポーリングしてエラーを検出する機能を備える、ホストの通信の完全性を検証するための独立した組み込みコントローラ。詳細については、[BMS安全モニタリング・モード](#)のセクションを参照してください。
- システムのコンタクタ制御を可能にするGPIOプログラマビリティ。
- 安全性の検証を加速するための、ユーザ利用可能な組み込みテスト・モード。
- ALERTPACKETを用いる、デジタイゼーション・バッテリー管理モニタの自動ポーリング。
- デジタル動作を検証できるデュアル・オシレータ。
- 過電圧および低電圧を検出できる複数のLDOレギュレータ。

動作モード

デバイスの動作モードには次の3種類があります。

- コマンド動作モード
- 自動BMS安全モニタリング・モード
- スリープ・モード

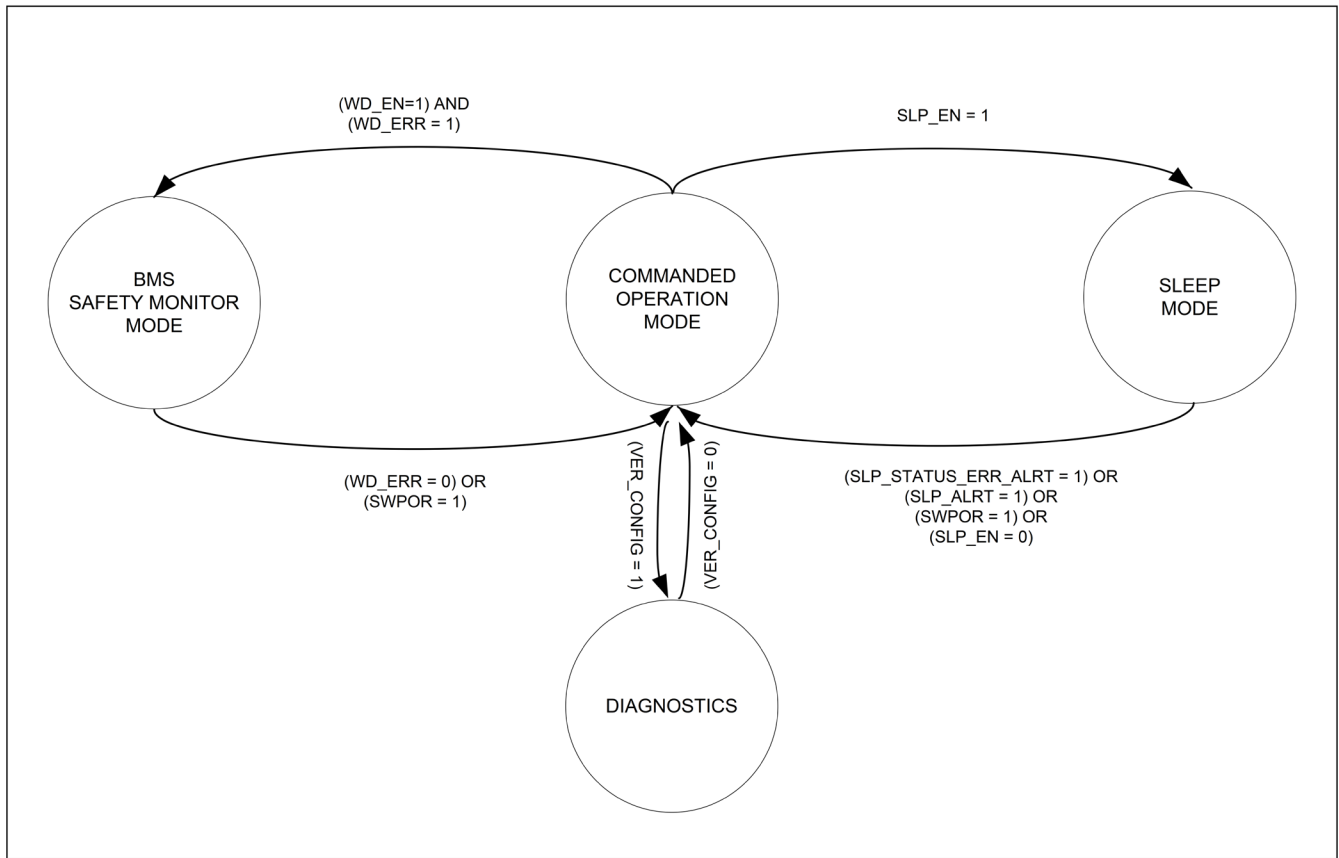


図5. 動作モード

コマンド動作モード

電源投入時のデフォルト状態は、ウォッチドッグ機能が無効化されたコマンド動作モードです。コマンド動作モードの場合、MAX17851は、ホスト・マイクロコントローラに全て制御されます。

BMS安全モニタリング・モードにできるようにするには、WD_ENを設定してウォッチドッグ・ステート・マシンの動作を有効化する必要があります。WD_DBNCを超える無効なウォッチドッグ応答が生じると、MAX17851がBMS安全モニタリング・モードになる原因となります。ウォッチドッグ・タイムアウトの時間は、WD_OPEN、WD_CLOSED、WD_DIV、WD_1UDの各レジスタ設定に依存します。

コマンド動作モードに復帰できるのは、有効な応答でウォッチドッグ・キー・レジスタを更新した場合 (WD_KEY)、または、ソフトウェアPORを発行した場合 (SWPOR = 1) のみです。

スリープ・モード

スリープ・モードは、ホスト・マイクロコントローラが低消費電力状態になることができる長期セル・バランシングに対応できるよう、設計されています。MAX17851は、コントロール・エリア・ネットワーク (CAN) のバス・アクティビティとは無関係に、ホストが動作するよう定期的にウェイク・イベントを供給します。更に、スリープ・モードでは、セル・バランシングの有無に関係なく、デジタイゼーション・デバイスからのセーフティ・クリティカルなバッテリー・パラメータ (OV、UV、OT、UT) について安全性モニタリングを行います。

スリープ・モードは、SLP_ENをアサートすることで開始できます。これによって、内部タイマーのSLP_CBNTFYによる制御が始まります。スリープ・モードになるとウォッチドッグの動作が無効化されるため、スリープ・モード中はBMS安全モニタリング・モードに遷移することができなくなります。そのため、スリープ・モード中はウォッチドッグを更新する必要がありません。

スリープ・モードは、SLP_ENに0を書き込む、SLP_CBNTFYタイマー時間が経過する、ALRTPCKT_DBNCカウンタ・スレッシュホールドを超過する、SWPORコマンドを送信する、のいずれかによって終了できます。図6を参照してください。

ALRTPCKT_TIMINGレジスタでプログラムされた自動キープアライブおよびアラート・パケット通信は、このモードでは無効化されます。そのため、デジチーチェーン・デバイスがプログラムに従ってSHDNLPINをハイにする（スタンバイ状態にする）必要があります。

スリープ・モードのALERTPACKETを設定する (SLP_ALRTPCKTEN = 1) と、ALERTPACKETコマンドの送信を通じて、セーフティ・クリティカルなバッテリー監視が可能になります。ALERTPACKETの頻度は、SLP_SCAN_DLYレジスタで設定します。ALRTPCKT_DBNCレジスタに従って永続的なALERTPACKETエラーが生じた場合は、SLP_STATUS_ERR_ALRTステータス・レジスタがアサートされ、スリープ・モードは終了します。関連するSLP_STATUS_ERR_ALRTENが有効化されている場合は、このアラートは、マイクロコントローラの割込みピンに接続されたALERTを通じてホスト・コントローラに伝達されます。

割込みピンを使用することは、CANバスとは独立なマイクロコントローラをウェイクするために必要です。

注: スリープ・モード時に設定レジスタ空間またはTXコマンド・レジスタ空間に書込みを行おうとしても無視され、SPI_ERRステータス・ビットがセットされてALERTピンがアサートされる原因になります。



BMS安全モニタリング・モードには、以下の機能があります。

- Maxim Integrated | 23

3. BMSデータ・アキュイジション・システムがユーザ定義のセーフティ・クリティカルなスレッシュホルドを超過した場合に、冗長なGPIO信号を介してバッテリー・コンタクトに車両安全状態信号を送信。

連続的なBMSデータ・アキュイジション・モニタリングは、ALERTPACKETコマンドの自動的な生成と解析を通じて行われます。これらのコマンドは、BMSデータ・アキュイジション・システムのリアルタイムのステータスを提示し、ALERTPACKETバッファのSTATUSビットに反映されます。各STATUSビットはいずれも、STATUS_DBNC_MASK[15:0]レジスタを用いてマスクできます。これにより、アプリケーションにおいて、セル過電圧、セル低電圧、セル過熱、セル低温、過電流、BMSモジュールの内部不良、セル・バランシング不良など、任意のシステム・クリティカルなイベントあるいはイベントの組み合わせを選択できる柔軟性がもたらされます。BMS安全モニタリング・モードでは、BMSデータ・アキュイジション・システムが重要なシステム・パラメータの自動ポーリングを行うよう設定する必要があります。これは、BMS安全モニタリング・モードにおいて、MAX17851が設定メモリのロード動作を自動的に実行することで行われます。詳細については、[設定メモリの伝送](#)のセクションを参照してください。

注:フォルトをクリアするためにロジック0を書き込む必要のあるSTATUSビットは全てマスクすることを推奨します。マスクしない場合、単一のエラーが永続的なエラーと誤って示されてしまいます(例えば、PECやALRTRSTなどのBMSデータ・アクイジション・システムのエラー)。通信エラー(つまり、PEC)は、全ての動作モードにおいてMAX17851のLSSMによって個別に検証されます。

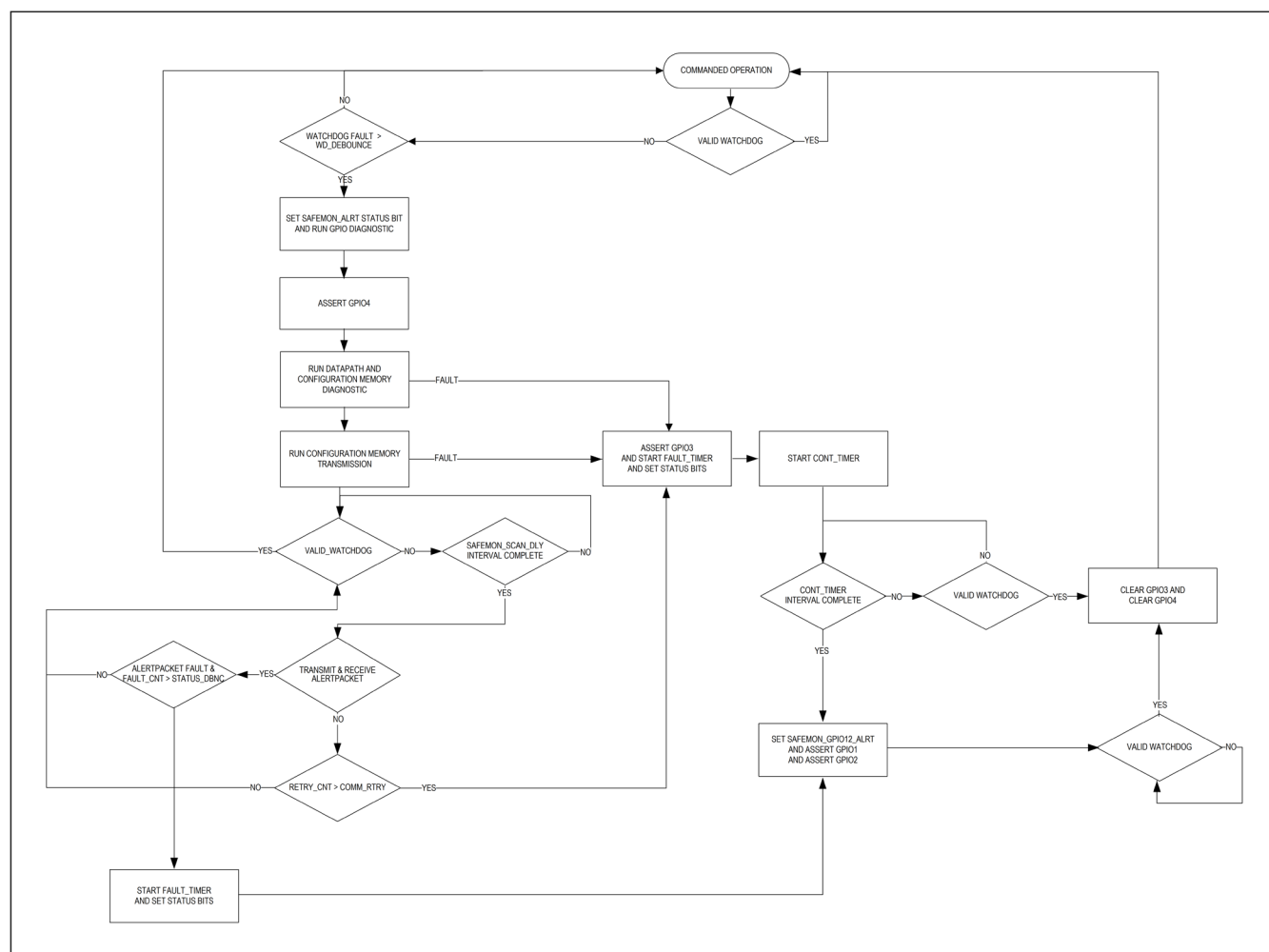


図7. BMS安全モニタリング・モードのフロー・チャート

BMS安全モニタリング・モードおよび安全対策診断状態への移行

MAX17851は、マスタに設定されている場合 (MS_EN = 1x)、連続した無効ウォッチドッグ応答の数がウォッチドッグ・バウンズ防止スレッシュホールド (WD_DBNC) を超えた後にはのみ、BMS安全モニタリング・モードになります。WD_ENが有効化されていない場合、マスタに設定されたMAX17851がBMS安全モニタリング・モードになることはありません。

スレープに設定されている場合 (MS_EN = 0x)、MAX17851は、GPIO3でロジック・ハイがアサートされた場合にのみ、BMS安全モニタリング・モードになります。

BMS安全モニタリング・モードになると、SAFEMON_ALERTステータス・ビットが直ちにアサートされます。ALERTピンはデフォルトでアサートされます。これとは異なる動作が必要な場合は、SAFEMON_ALERTENレジスタ・ビットを用いてALERTピンのアサートを無効化できます。

SAFEMONステータス・ビットがアサートされると、MAX17851は、安全対策診断状態になります。安全対策診断状態では、BMS安全モニタリング制御ロジック、ウォッチドッグ・ロジック、MAX17851の自律的なBMS安全モニタリング・モード動作の範囲外のGPIOが正しく機能しているかどうかを、ホストが検証できます。MAX17851は、GPIOREC_DLYにプログラムされた値に従って診断状態にとどまります。

診断状態の間、全てのGPIO出力は、BMS安全モニタリング・コントローラにより同時にアサートされます。コントローラによりアサートされますが、駆動されるピンの出力は、SM_GPIO[x]_MASKレジスタ・ビットを介して個別にマスクできます。マスクされた場合、GPIOピンはそれぞれの非アクティブな状態でアイドルリングします。その間、内部GPIO駆動レベルは、GPIO[x]_RDレジスタで読み出せます。GPIO[x]_RDレジスタを読み出すことで、ホスト・マイクロコントローラは、正しいGPIO動作をアサートできます。

例えば、GPIO1_CFG = 110 (SAFEMON HI出力、アイドル状態はLO) でSM_GPIO1_MASK = 1の場合、GPIO1_RDはハイ (アクティブ) にアサートされますが、GPIO1ピンの出力はロー (非アクティブ) に駆動されます。

注: GPIOのマスクが無効化されている場合 (SM_GPIO[x]_MASK = 0)、GPIO信号は、ホスト・マイクロコントローラによって検出され、基板レベルでの適切な機能が検証されます。車両操作が不必要に妨害されるのを防止するために、SM_GPIO[x]_MASKレジスタ・ビットはデフォルトで有効化されています。

コンタクトに意図しない信号が送られたり、車両操作が不必要に妨害されたりすることのないよう、GPIO1およびGPIO2は常にマスクしておくことを推奨します。GPIO3およびGPIO4をシステム診断用にマスク解除するかどうかは、アプリケーションによって判断する必要があります。そのような考慮をせずにGPIO3やGPIO4のマスクを解除すると、MAX17851が予期せぬ動作をする可能性があります。

GPIO[x]_RDレジスタの読出しに成功すると、ウォッチドッグが更新され、通常のコマンド動作モードが再開します。GPIOREC_DLYの終了までにウォッチドッグが更新されない場合、BMS安全モニタリング・モードが続行します。

マイクロコントローラおよび電源の回復

GPIO3およびGPIO4は、マイクロコントローラおよび電源とインターフェースしBMS安全モニタリング・モード時に回復を試行するよう、設計されています。

シングルUARTアプリケーションの場合、これを行うには、GPIO3およびGPIO4をマイクロコントローラおよび電源のイネーブル・ピンまたはリセット・ピンに接続します。ウォッチドッグがGPIOREC_DLYの時間が経過するまでにBMS安全モニタリング・モードで更新されない場合は、MAX17851は、ピン出力でGPIO4をアサートし、マイクロコントローラの回復を試行します。BMS安全モニタリング・モードの実行中にMAX17851に通信フォルトが発生した場合は、ピン出力でGPIO3をアサートし、電源の回復を試行します。図および詳細については、[アプリケーション情報の](#)セクションの[シングルUART動作](#)を参照してください。

デュアルUARTアプリケーションの場合は、マスタのGPIO4をマイクロコントローラのイネーブル・ピンまたはリセット・ピンに接続し、スレーブのMAX17851のGPIO3を電源に接続することで、回復が試行されます。ウォッチドッグがGPIOREC_DLYの時間が経過するまでにBMS安全モニタリング・モードで更新されない場合は、MAX17851は、ピン出力でGPIO4をアサートし、マイクロコントローラの回復を試行します。BMS安全モニタリング・モードの実行中にマスタに通信フォルトが発生した場合は、ピン出力でGPIO3をアサートします。これにより、スレーブにロジック・インジケータが備わり、マスタのフォルトに対応して冗長な通信経路上でフェールセーフのモニタリングが開始されます。BMS安全モニタリング・モード中にスレーブに通信フォルトが発生した場合は、ピン出力でGPIO3をアサートし、電源の回復を試行します。図および詳細については、[アプリケーション情報の](#)セクションの[デュアルUART動作](#)を参照してください。

電源投入時に基板レベルの競合が生じる可能性をなくすため、全てのGPIOピンは、デフォルトで汎用入力に設定されています。マイクロコントローラまたは電源に接続されているGPIOピンは、通常、SAFEMON_1-Shot LO出力として設定する必要があります。それにより、GPIOピンは、BMS安全モニタリング・コントローラによってアサートされた場合に、100msの間ローに引き下げられます。2個のMAX17851デバイスを使用するデュアルUARTアーキテクチャでは、マスタのGPIO3をSAFEMONアクティブ・ハイ出力に設定し、スレーブMAX17851のGPIO4をSAFEMON入力に設定する必要があります。ホスト・マイクロコントローラは、全てのGPIOをアプリケーションに対応してプログラムする必要があります。詳細については、[GPIO制御](#)のセクションを参照してください。

注:GPIOREC_DLYの時間が経過した後は、SM_GPIO[x]_MASKレジスタを使用することはできません。

NOMON=1の場合、GPIOREC_DLYの時間が経過した後、全てのGPIOがピン出力で同時にアサートされます。これにより、上述のように電源とマイクロコントローラの回復が試行されますが、また、GPIO1およびGPIO2もアサートされます。これらは、バッテリー・コンタクタを開放にするよう設計されています。

注:回復およびコンタクタのアサートを直ちに行うには、与えられたアーキテクチャの全てのMAX17851デバイスについて、NOMONをセットする必要があります。

BMS安全モニタリング動作

NOMON=0の場合、MAX17851は、GPIOの検証とシステム回復の試行を終えた後に、BMS安全モニタリング・モードの初期化を開始します。これは次のようにして行われます。

- 1) 自己診断を行い、UARTのデータバスおよび設定メモリが内部で正しく動作していることを確認します。
- 2) 設定メモリの内容をデジタイゼーション・デバイスに伝達します。

デジタイゼーションの自己診断およびプログラミングは、ASIL-Dシステムに対するISO26262の条件を満たすように構築されています。

設定メモリの内容をデジタイゼーション・デバイスに伝達することで、システム・フォルト時の動作で定義される新たな要件に対して、安全スレッシュホールドを再プログラムできます。なお、この条件下では、バッテリー・セルの状態は、通常動作の境界内にあるものとして引き続き適格性の判定をする必要があります。

それまでのいずれかの手順においてエラーが検出された場合、STATUS_SAFEMONレジスタにエラーの説明を示すステータス・フォルトがセットされます。その後、CONT_TIMERが起動し、設定された値までインクリメントするため（デフォルトではタイマーには無限の時間が設定されており実質的にタイマーが無効化されています）、初期化フォルト時にGPIO1ピンとGPIO2ピンが駆動され、それによって標準的なアプリケーションにおいてバッテリー・コンタクタが開放されるタイミングを、柔軟に設定できます。

上記の手順の終了後、MAX17851は、ALERTPACKETを継続的に使用してBMSデータ・アクイジション・システムをポーリングし、設定メモリで定義されたスレッシュホールドを用いてフォルトの有無を調べます。この動作は、[図7](#)に示す有効な終了基準に達するまで行われます。

NOMONレジスタ・ビットが1の場合、MAX17851は、GPIOの検証およびシステム回復試行を終えた後は、BMSデータ・アクイジション・システム・デバイスをモニタしません。その代わりに、[図8](#)に示すように、FAULT_TIMERを起動し、全てのGPIOピンをその設定に従ってアサートし、有効なウォッチドッグ更新が行われるまで待機します。

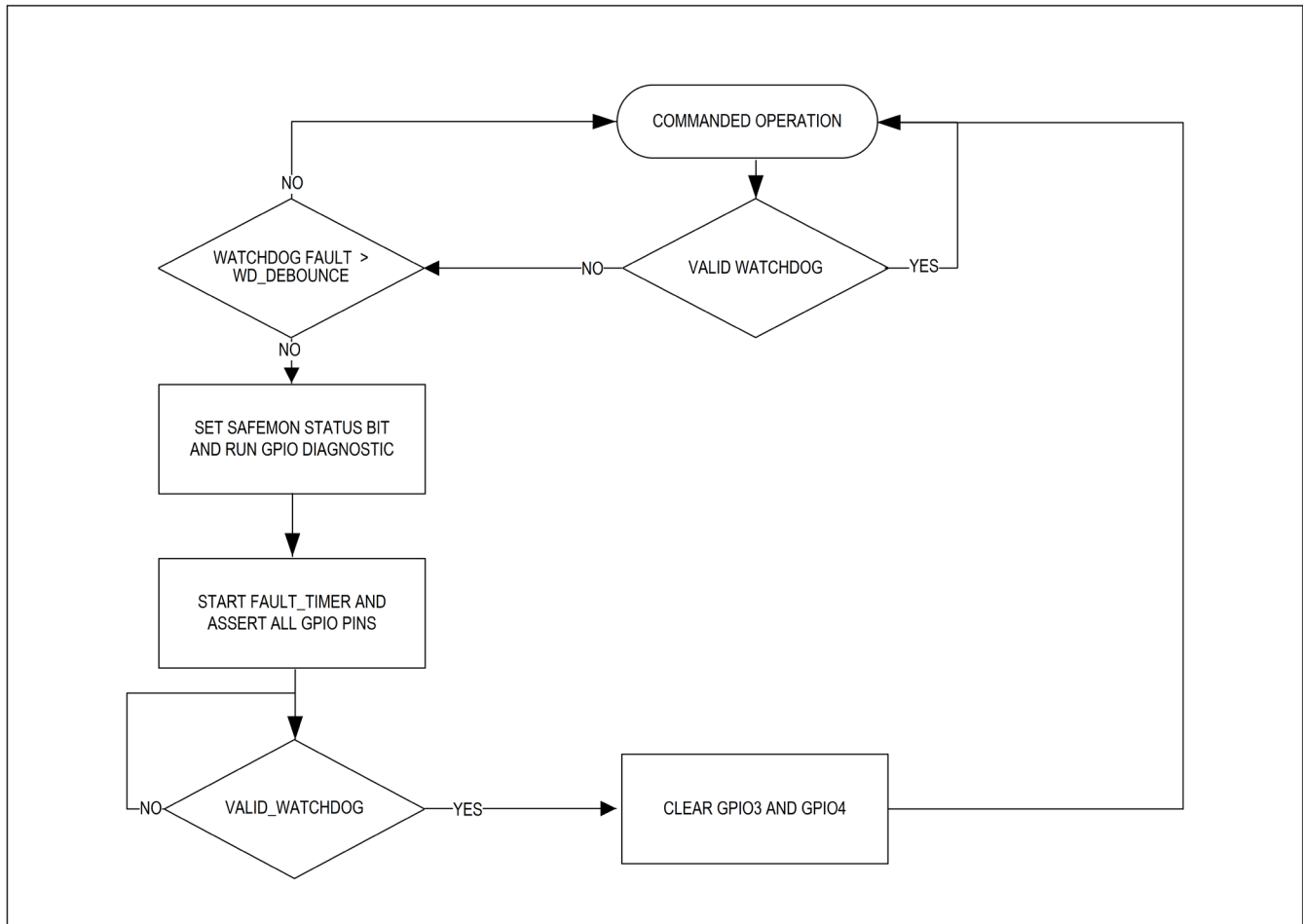


図8. BMS安全モニタリング(NOMON)モードのフロー・チャート

ALERTPACKETを用いたフォルト・ポーリング

診断およびデジチェーン設定に成功した後、SAFEMON_SCAN_DLYの時間間隔に基づいてALERTPACKETが生成されます。ALERTPACKETのデータ・ペイロードは、アラート・パケット・バッファに格納され、その後、マスクされて、ユーザ指定のフォルトを特定するために解析が行われます。マスクされていないフォルトが、SAFEMON_ALERT_DBNCレジスタにプログラムされた回数だけ継続する場合のみ、SAFEMON_ERRステータス・ビットがセットされ、GPIO1ピンおよびGPIO2ピンがアサートされます。

$$\text{SAFEMON_ERR} = |(\text{ALERTPACKET.STATUS}[n] \& \sim \text{ALRT_STATUS_MASK}[n]) \text{ (Debounced SAFEMON_ALRT_DBNC Times)}|$$

注: BMSデジチェーン・モニタリング・デバイスのポーリング間隔は、MAX17851のレート(SAFEMON_SCAN_DLYで設定)とは異なる可能性があります。そのため、それまでの同じサンプリングに基づいてフォルト条件が繰り返される可能性があります。アプリケーションでは、SAFEMON_SCAN_DLY値がBMSモニタリング・デバイスの測定遅延よりも大きくなるようにして、サンプリングが確実に一意のものとなるようにする必要があります。

フォルト・ポーリング・プロセスの間にALERTPACKETコマンドを受信しなかった場合、COMM_TO_ERRフォルトが発行され、COMM_RETRYカウント数を超えるまで連続的に再試行されます。BMS安全モニタリング・モード中にいずれかのタイミングでアラート・パケット・バッファのハードウェアが完全性チェックに失敗した場合、ALRTPCKTBUF_HW_ERRフォルトがアサートされます。BMS安全モニタリング・モード中にALRTPCKTBUF_HW_ERRまたは通信エラーのいずれかが発生した場合、GPIO3ピンがアサートされ、CONT_TIMERが起動し、このタイマーが終了するとGPIO1ピンおよびGPIO2ピンがアサートされます。

BMS安全モニタリング・モードの終了

BMS安全モニタリング・モードは、有効なウォッチドッグ応答をWD_KEYに書き込むことで終了します。終了は、安全モニタリング・サイクルが完了すると(つまり、生成されたALERT_PACKETが受信された後)行われます。

終了してもSAFEMON_ALERTビットはクリアされず、ALERTピンは、SAFEMON_ALERTがクリアされるまでアサートされたままになります。ALERTピンがクリアされないままの場合、アラート・ステータスをクエリして、MAX17851およびBMSデিজィチェーンのステータスを判定する必要があります。デバイスが依然としてBMS安全モニタリング動作を行っている間にSAFEMON_ALERTがクリアされても、SAFEMON_ALERTは直ちに再アサートされます。SAFEMON_ALERTがクリアされるのは、BMS安全モニタリング・モードが正常に終了した後のみです。

BMS安全モニタリングの終了時にGPIO1/2がアクティブに駆動されている場合、これらのピンは、SAFEMON_GPIO12_ERRビットがクリアされるまでアサートされたままになります。

BMS安全モニタリングの終了時にGPIO3/4がアクティブに駆動されている場合、これらのピンは、デアサートされます。

SPIパターンが断線しているためにマスタのデュアルUART MAX17851が応答しない場合は、スレーブのデュアルUART MAX17851のMS_ENビットをマスタのシングルUARTに設定し、そのウォッチドッグを更新して、スレーブのデュアルUART MAX17851でBMS安全モニタリング・モードを終了する必要があります。この場合、それ以前に設定したスレーブのデュアルUARTが、マスタのシングルUARTになります。この状況では、マイクロコントローラは、新たに設定されたマスタのシングルUART MAX17851を用いて、BMSデバイスをその通常動作状態に再初期化する必要があります。なお、RX_SWAP_ENは変更せずにおき、デিজィチェーン経路を反映するようにする必要があります([コマンド動作モードのスレーブ](#)のセクションを参照)。

注: MAX17851は、デバイスをリセットする(SWPOR = 1)ことでBMS安全モニタリング・モードを終了することもできます。SWPORを終了条件として用いる場合、MAX17851はデフォルト状態になることが前提であり、レジスタのプログラミングはリセットされ、アプリケーションに合わせてSPIで再初期化する必要があります。

設定メモリ

BMSデিজィチェーンのステータスは、目的とは異なる設定状態になる可能性があります(BMSソフトウェアの実行時にマイクロコントローラが無応答となるタイミングに依存)。更に、不明なマイクロコントローラ・フォルトが原因で、システムがセーフティ・クリティカルなスレッシュホールドを変更する必要がある場合もあります。設定メモリには、BMS安全モニタリング・モード内で実行して全てのデিজィチェーン・デバイスを既知の状態にするのに必要なUARTコマンド・セットが格納されています。BMSデータ・アキュイジション・システムがこれらの設定により設定された後、ホスト・マイクロコントローラが応答しない場合に、システムはこれらのパラメータを定期的に自動ポーリングして、システムの安全性を確認します。システムにフォルトが生じた場合、および、ユーザ定義の重要なスレッシュホールド/バウンス防止に応じて、MAX17851は、ホスト・マイクロコントローラやコンタクトにGPIO信号を発することでシステムを安全な状態にします。

設定メモリは、それぞれ30データ・バイトからなる3つのキューと、6データ・バイトからなる4番目のキューで構成されます。最初の3つのキューは、7つのデータ・ブロック、キューを保護する1つのPECバイト、および予約済みの1つのバイトに分割されます。4番目のキューは、同じ構造を持つ1つのデータ・ブロックで構成されています。設定メモリのデータ・ブロックはそれぞれ2つの個別のレジスタ・アドレス・バイトおよび2つのデータ・バイトで構成されます。データ・バイトは、BMSデিজィチェーン・ユニットに16ビットのデータ・ペイロードを書き込むために用いられます。データはLSBが最初に格納され、次にMSBが格納されます。

0xFFのアドレス値は、ヌル・アドレスであり、設定メモリ・ロード・ステート・マシンおよび設定メモリ検証ステート・マシンを動作させないことを示すものです。同じブロックに2つのヌル・アドレスが連続した場合、そのデータ・キューに対するデバイス設定の終了を意味します。デフォルトでは、全てのバイトは0xFFにリセットされます。

PECバイトは、2つの連続するヌル・アドレスの前にある1つのキューの全バイトを保護します。2つの連続するヌル・アドレスがそのキュー内に生じない場合は、PECバイトはキュー全体を保護します。ヌル・アドレスが1つの場合はスキップされ、PECバイトの計算には含まれません。2つのヌル・アドレスで始まるキューは、設定メモリのロード処理や設定メモリの検証処理が行われません。

CONFIGメモリには、CONFIGQ、CONFIG_QUEUE_PTR、CONFIG_BYTE_PTRの各レジスタを用いてアクセスできます。詳細については、[MAX17851ユーザ・レジスタ・マップ](#)を参照してください。

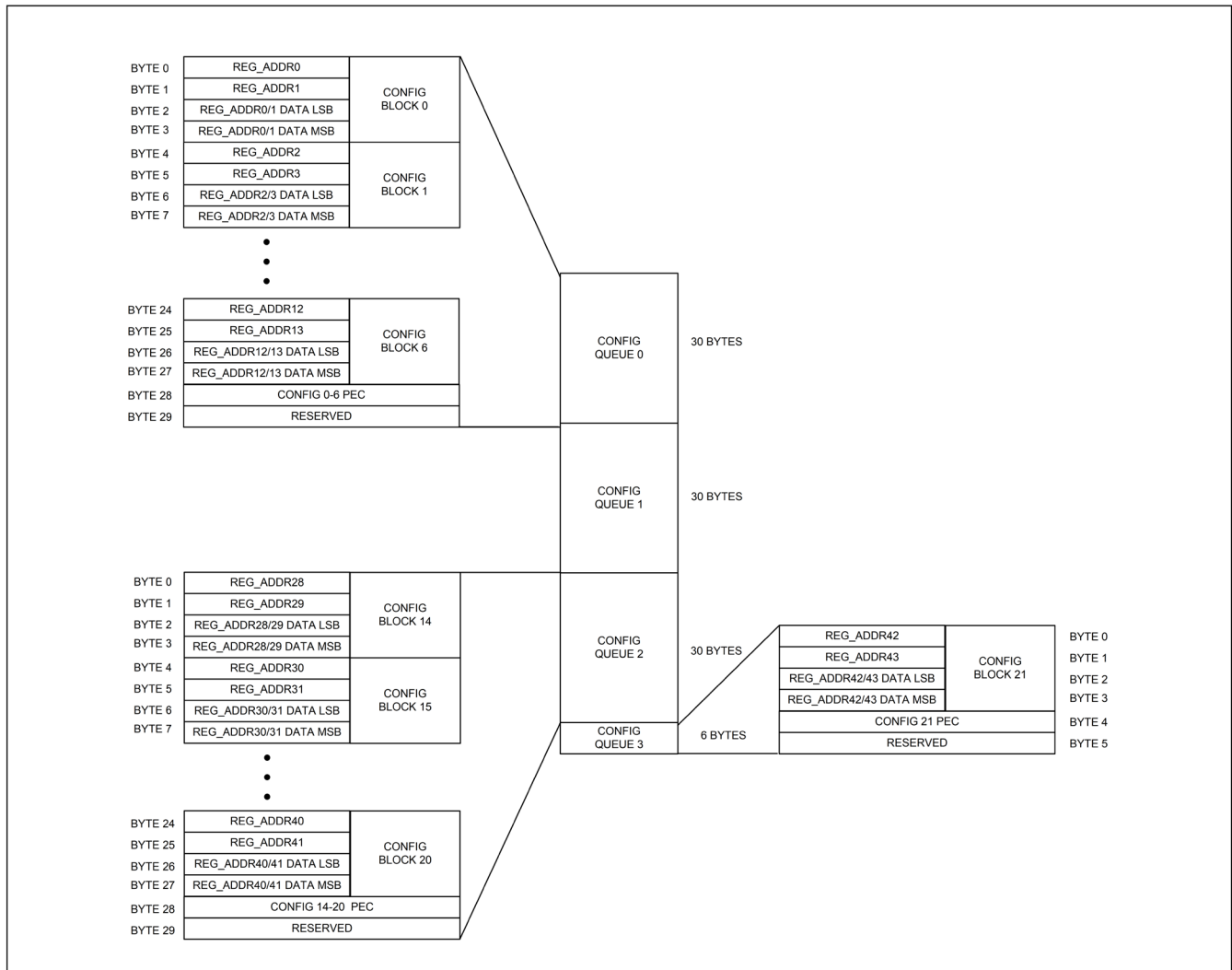


図9. 設定メモリ

データバスおよび設定メモリの検証

設定メモリをUARTのWRITEALLパケットとしてフォーマットして送信する前に、内部のループバック回路を用いて、設定メモリの内容、および、UARTデータバス、PECエンコーダ、PECデコーダの機能が検証されます。このプロセスの間、アラート・パケットの生成(有効化されている場合)は中断され、キープアライブの生成が有効化されます。キープアライブ・キャラクタは、設定メモリ・バイトごとの検証プロセスの間に生じます。これにより、BMSデジタイゼーションがこの検証プロセスの間に意図せずシャットダウンされることがなくなります。

このプロセスの間にフォルト条件が発生すると、DATAPATH_ERRビットおよびSAFEMON_CONFIG_ERR_ALERTビットで示されます。マスタのみの設定またはスレーブ設定の場合、MAX17851はGPIO3をアサートし、FAULT_TIMERおよびCONT_TIMERを起動した後、GPIO_CFGビットフィールドの定義に従いGPIO1とGPIO2をアサートします。スレーブ付きマスタの設定の場合は、MAX17851は、GPIO3をアサートし、FAULT_TIMERを起動してウォッチドッグの更新を待ちます。

設定メモリとUARTデータバスの検証は、VER_CONFIGコマンドを発行することで、BMS安全モニタリングの外部で実行できます。動作が完了するとこのビットは自動クリアされ、その結果は上述のステータス・ビットで示されます。内部ループバックを用いてMAX17851の内蔵ハードウェアをテストする方法の詳細については、[TX_AUTO機能を用いたハードウェア・イン・ザ・ループ\(HIL\)テスト](#)のセクションを参照してください。

データバスおよび設定メモリの検証診断の実行時間 ($t_{\text{VER_CONFIG}}$) は、設定されたボー・レートおよび設定メモリに格納されたデータの量によって異なります。

表2. データバスおよび設定メモリの実行時間

Configuration Memory Banks Populated	$t_{\text{VER_CONFIG}}$
1	$30\mu\text{s} + 72 \cdot t_{\text{BIT}}$
2	$60\mu\text{s} + 72 \cdot t_{\text{BIT}}$
3	$90\mu\text{s} + 72 \cdot t_{\text{BIT}}$
4	$120\mu\text{s} + 72 \cdot t_{\text{BIT}}$

設定メモリの伝送

各CONFIGデータ・ブロックは、WRITEALLメッセージに順次フォーマットされ、送信バッファにロードされて、UARTデジタイゼーション・デバイスに送信されます。データ伝送のフロー・チャートの詳細図については、[図10](#)を参照してください。

データ伝送は、CONFIGデータ・ブロック内の各レジスタ・アドレス (REG_ADDR1 および REG_ADDR2) について2度行われます。それによって、同じデータ・バイトを2つの異なるレジスタ・アドレスに送信したり、伝送の冗長性を確保するために同じデータと同じレジスタ・アドレスに送信したりできます。

表3. 設定メモリからのWRITEALLコマンド・シーケンス

MAX17851 TX	MAX17851 RX
Preamble	Preamble
02h	02h
[REG_ADDR1/2]	[REG_ADDR1/2]
[DATA LSB]	[DATA LSB]
[DATA MSB]	[DATA MSB]
[PEC]	[PEC]
[ALIVE]*	[ALIVE]*
Stop	Stop

* 自動アラライブ・カウンタが有効化されている場合

発行された設定メモリ・コマンドのPECは、MAX17851によって自動的に計算されます。アラライブ・カウンタが有効な場合は、デフォルトのアラライブ・カウンタ・シード値またはローリング・アラライブ・カウンタに応じて、アラライブ・カウンタも発行されます。

次のレジスタ・アドレス (REG_ADDR2) または、次のCONFIGデータ・ブロックを伝送できるようになる前に、通信パケットを正しく実行できるかが検証されます。データ・パケットの完全性あるいはデータ・パケットのタイミングにエラーがある場合、COMM_ERRステータス・ビットまたはCOMM_MSMTCH_ERRステータス・ビットにフォルトが格納されます。あるいは、COMM_TO_DLYで設定された時間が経過しても設定メモリ・データ伝送が受信されない場合は、SAFEMON_CONFIG_ERRビットにフォルトが格納されます。正常に送信されなかったデータ・パケットは、COMM_RTRYレジスタにプログラムされた値になるまで再送信されます。COMM_RTRYカウントを超えると、SAFEMON_CONFIG_ERRビットがアサートされます。マスタのみの設定またはスレーブ設定の場合、MAX17851はGPIO3をアサートし、FAULT_TIMERおよびCONT_TIMERを起動した後、GPIO_CFGビットフィールドの定義に従いGPIO1とGPIO2をアサートします。スレーブ付きマスタの設定の場合は、MAX17851は、GPIO3をアサートし、FAULT_TIMERを起動してウォッチドッグの更新を待ちます。

通信の安全性検証の詳細については、[ロックステップ安全対策の検証](#)のセクションを参照してください。

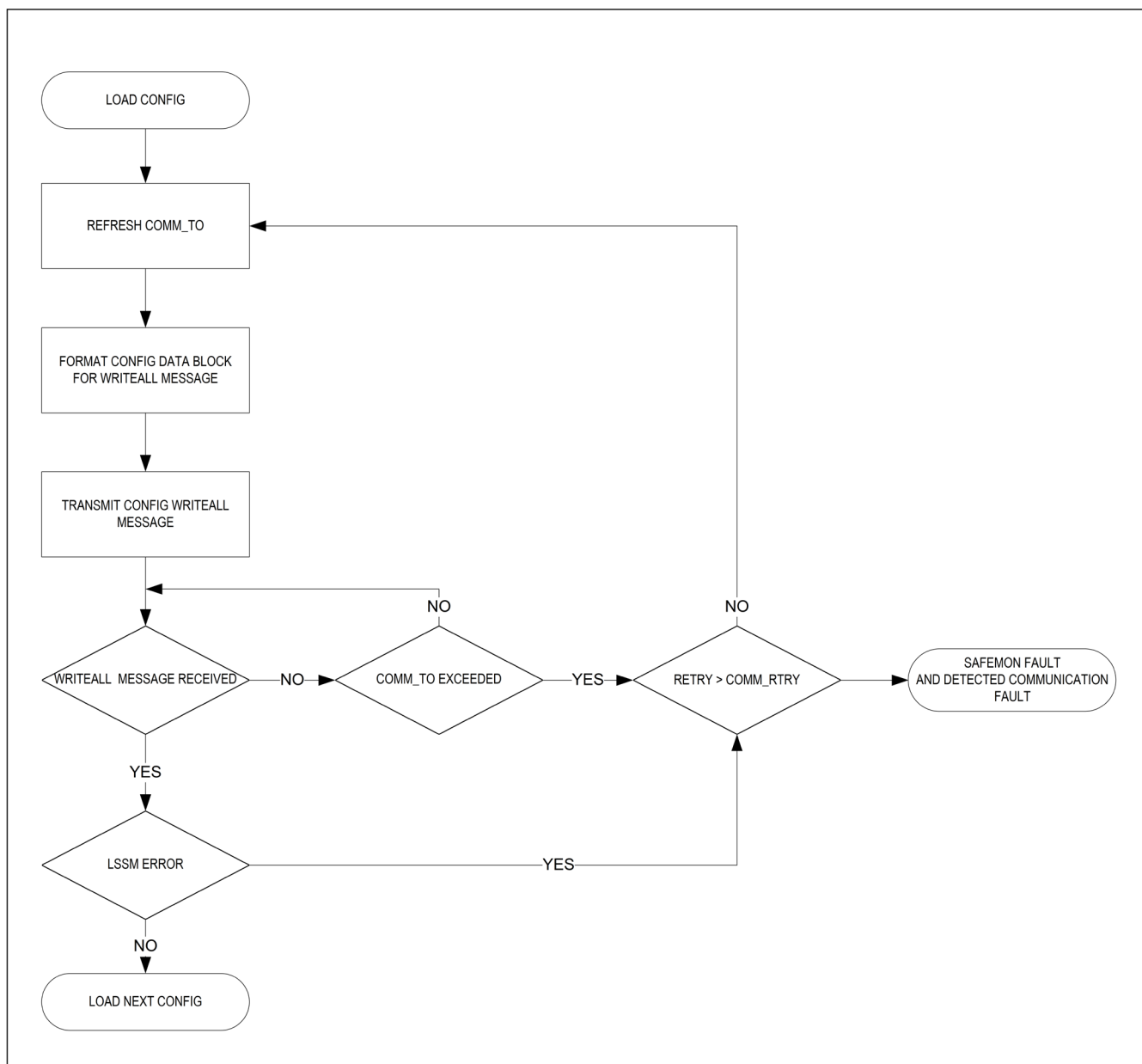


図10. 設定メモリ・データの伝送

設定メモリのセットアップ

設定メモリは、MAX17851に次の動作が可能となるようにプログラムする必要があります。

1. デイジーチェーン・デバイスの設定の再初期化
2. 全てのデイジーチェーン・デバイスに対する自動ポーリングの起動

デイジーチェーン・デバイスの設定が必要なのは、BMS動作のいずれかの動作サイクル時に、ウォッチドッグ・アラートの原因となる不具合が生じる可能性があるためです。不具合が生じた場合、デイジーチェーンは、診断サイクルや部分的な測定サイクルなどにおいて、意図せぬ設定になることがあります。デイジーチェーンが意図せぬ状態になる可能性があるままにしておくと、MAX17851がシステム・コンタクトを誤って開放してしまう場合があります。

デジタイゼーション・デバイスの再初期化には、セル測定設定（有効化されたセル、極性）、温度測定設定（有効化された補助入力、補助スキャン・タイミング、補助レシオメトリック・ソース）、診断設定および測定（有効化された電流源、電流源のレベル、組み込みスキャン診断）、スキャン制御（スキャン・タイプ、デジタル・ノイズ・フィルタの設定および制御）などの変更が含まれます。

設定の例については、[デジタイゼーション設定用の設定メモリ・シーケンス](#)のセクションを参照してください。

マスタ/スレーブのデバイス設定

MS_ENビットは、アプリケーション回路におけるMAX17851のウォッチドッグおよびBMS安全モニタリングの動作を、マスタのみ、スレーブ付きマスタ、スレーブ・デバイスのいずれかに指定します。

BMS安全モニタリング・モードにおけるマスタ

MAX17851をマスタに設定する（MS_EN = 1X）には、BMS安全モニタリング・モードにならないようにウォッチドッグが機能することが必要です。

GPIOREC_DLYタイムアウトに伴って安全対策状態が終了するとGPIO4がアサートされます。詳細については、[GPIO制御](#)のセクションを参照してください。GPIO4のアサートは、マイクロコントローラまたは電源の回復を試行するために用いることができます。

GPIO3は、データ破損、内部処理エラー、あるいはUART通信の妨げとなるフォルトが示された場合にアサートされます。GPIO3のアサートは、BMS安全モニタリング動作の制御をデュアルUARTアプリケーションのスレーブ・デバイスに移行する場合に用いることができます。

詳細およびアプリケーション回路図については、[アプリケーション情報の](#)、[シングルUART動作](#)および[デュアルUART動作](#)の各セクションを参照してください。

BMS安全モニタリング・モードにおけるスレーブ

MAX17851デバイスをスレーブに設定している場合（MS_EN = 0x）、WD_ENの設定とは無関係にウォッチドッグ機能は無効化され、全てのウォッチドッグ・アラートはマスクされます。したがって、SPI通信の喪失やウォッチドッグ・タイムアウトが発生しても、スレーブはBMS安全モニタリング・モードになることができません。

スレーブ・デバイスがBMS安全モニタリング・モードになるのは、そのGPIO4ピンにロジック・ハイをアサートした場合のみです（GPIO4ピンがSAFEMONスレーブ入力に事前設定されていることが前提）。スレーブ・デバイスがBMS安全モニタリング・モードになると、DOWNHOSTコマンドを発行し、BMS安全モニタリング・モードを開始します。これにより、マスタ・デバイスの通信障害が発生した場合に冗長性が確保されます。

GPIO3は、データ破損、内部処理エラー、あるいはUART通信の妨げとなるフォルトが示された場合にアサートされます。GPIO3のアサートは、マイクロコントローラまたは電源の回復を試行するために行うことができます。

スレーブ・デバイスは、2通りの方法でBMS安全モニタリング・モードを終了できます。1つ目は、マスタ・デバイスのウォッチドッグを更新する方法で、これによりマスタのGPIO3出力がデアサートされます。2つ目は、SWPORコマンドをスレーブに発行し（これによりマスタ・シングルUARTにデフォルト設定されます）、デジタイゼーションを再設定する方法です。この2つ目の状況が必要となるのは、マスタとのSPI通信が不能となりそのウォッチドッグを更新できない場合のみです。

スレーブ設定は、デュアルUART設定のみを目的としています。詳細については、[アプリケーション情報のデュアルUART動作](#)のセクションを参照してください。

コマンド動作モードのスレーブ

UARTの下り経路からRXバッファに入るデータは、UARTの上り経路からとは逆の順になります。RXSWAP_ENを用いると、下り経路のデータを上り経路のデータと同じ順にすることができるため、システム・マイクロコントローラによる処理が容易になります。

注: コマンド動作モードでは、下り経路に接続されたMAX17851デバイスは、読出しコマンドのみを発行するスレーブ・デバイスであることが必要です。

注: 物理的な下り経路に接続されたスレーブ・デバイスでは、RXSWAP_ENを1に設定する必要があります。ただし、上り経路と下り経路の読出し／書込み機能を変更する必要がある場合、ホスト・マイクロコントローラは、DOWNHOSTあるいはUPHOSTを発行できますが、RXSWAP_ENは、物理的な経路に対し設定されているため、変更せずにおく必要があります。例については、[表4](#)を参照してください。

表4. デュアルUART下り経路でのREADALLコマンドのシーケンス(zはデバイス数)

MAX17851 UART TX	MAX17851 UART RX	MAX17851 BUFFER STORAGE (RXSWAP = 0)	MAX17851 BUFFER STORAGE (RXSWAP = 1)
Preamble	Preamble		
03h	03h	03h	03h
[REG ADDR]	[REG ADDR]	[REG ADDR]	[REG ADDR]
[DC] = 0x00	[DATA LSB(z)] = [DATA LSB(BA)]	[DATA LSB(z)] = [DATA LSB(BA)]	[DATA LSB(1)] = [DATA LSB(TA)]
[PEC]	[DATA MSB(z)] = [DATA MSB(BA)]	[DATA MSB(z)] = [DATA MSB(BA)]	[DATA MSB(1)] = [DATA MSB(TA)]
[ALIVE]*	[DATA LSB(z-1)] = [DATA LSB(BA + 1)]	[DATA LSB(z-1)] = [DATA LSB(BA + 1)]	[DATA LSB(2)] = [DATA LSB(TA-1)]
[FD(1) C2h]	[DATA MSB(z-1)] = [DATA MSB(BA + 1)]	[DATA MSB(z-1)] = [DATA MSB(BA + 1)]	[DATA MSB(2)] = [DATA MSB(TA-1)]
[FD(1) D3h]	...	...	...
[FD(2) C2h]	...	...	...
[FD(2) D3h]	...	...	...
...	...	...	...
...	...	...	[DATA LSB(z-1)] = [DATA LSB(BA + 1)]
...	...	...	[DATA MSB(z-1)] = [DATA MSB(BA + 1)]
...	[DATA LSB(1)] = [DATA LSB(TA)]	[DATA LSB(1)] = [DATA LSB(TA)]	[DATA LSB(z)] = [DATA LSB(BA)]
...	[DATA MSB(1)] = [DATA MSB(TA)]	[DATA MSB(1)] = [DATA MSB(TA)]	[DATA MSB(z)] = [DATA MSB(BA)]
...	[DC]	[DC]	[DC]
[FD(z) C2h]	[PEC]	[PEC]	[PEC]
[FD(z) D3h]	[ALIVE]*	[ALIVE]*	[ALIVE]*
Stop	Stop	Stop	Stop
12+(4 x z) characters	12+(4 x z) characters	6+(2 x z) bytes	6+(6 x z) bytes

* アライブ・カウンタ・モードを有効化した場合

MAX17851のインターフェースによって送信されるフィル・バイトの値は、表に示すようにC2hとD3hの間で変化します。

スリープ・モードのスレーブ

システム割込みが競合するのを防止するため、スレーブのMAX17851にスリープ・モードになるようコマンドを送るのは、マスタのMAX17851がスリープ・モードになっている間である必要があります。ただし、スレーブにこのコマンドを送ることが禁じられているわけではありません。

マスタ／スレーブの設定の詳細については、[アプリケーション情報](#)のセクションの[デュアルUART動作](#)を参照してください。

診断および動作の検証

MAX17851は、以下に示す2つの診断を実行し、MAX17851の適切な動作状態を確保します。

- データバスおよび設定メモリの診断
- WD/GPIOの診断

データバスおよび設定メモリの診断では、設定メモリおよびUARTデータ経路を完全にチェックして、UARTエンコーダおよびデコーダ、受信PECデコーダ、送信PECエンコーダ、ピンの前段のUART TX/RX PHYを検証します。ハードウェア・エラー

または設定メモリ/PECエラーの場合、MAX17851は、STATUS_GENレジスタのDATAPATH_ERRビットにフラグを設定し、UARTデータバス・エラーが生じていることを示します。この診断機能は、コマンド動作モードでVER_CONFIGビットに1を書き込むことで、手動で開始できます。デバイスがスリープ・モードまたはBMS安全モニタリング・モードになっている場合は、この診断機能を手動で実行することはできません。

WD/GPIOの診断では、ウォッチドッグとGPIOの機能が適切かどうかを検証します。この診断は、ウォッチドッグ・タイムアウトによりBMS安全モニタリング・モードになった場合に行われます。このとき、ホストはGPIOステータス・レジスタの読出しを実行する必要があります。有効なウォッチドッグ・コマンドが書き込まれると、この診断は終了します。GPIOREC_DLYの時間が経過するまでにこの診断が終了しない場合は、MAX17851は、NOMONビットの状態に従って動作を進めます。詳細については、[BMS安全モニタリング・モードおよび安全対策診断状態への移行](#)のセクションを参照してください。

ウォッチドッグ

MAX17851のウォッチドッグ機能を用いることで、一定の時間間隔でSPIインターフェースによる通信を行い、ホスト・マイクロコントローラの動作を検証できます。ホスト・マイクロコントローラが応答しなくなった場合（つまり、ウォッチドッグがタイムアウトした場合）、カウンタがインクリメントします（[CONFIG_WD2\[3:0\]のWD_DBNCビット](#)）。このカウンタは、1から128まで設定可能です。WD_DBNCの値が一致すると、MAX17851はSTATUS_WDレジスタのWD_TO_ERRビットをセットし、BMS安全モニタリング・モードに入ります。

注:スリープ・モードになるとウォッチドッグの動作が無効化されるため、スリープ・モード中はBMS安全モニタリング・モードに遷移することができなくなります。そのため、スリープ・モード中はウォッチドッグを更新する必要がありません。

MAX17851のウォッチドッグ機能は、WD_CONFIG2レジスタのWD_ENビットを用いて有効化できます。WD_ENビットがデアサートされると（あるいはSWPORビットがアサートされると）、ウォッチドッグが無効になり、デバイスは、BMS安全モニタリング・モードに入ることができなくなります。MAX17851がBMS安全モニタリング・モードになっており、SWPORビットがアサートされた場合、デバイスはBMS安全モニタリング・モードを終了し、デフォルトのレジスタ状態でコマンド動作モードになります。

ウォッチドッグの時間ベースは、[CONFIG_WD1](#)レジスタのWD_DIVビットを用いて設定されます。ウォッチドッグのクロック時間は、WD_DIVの値に1を加え、それに256μsを掛けた値に等しくなります（ $t_{WDCLK} = (WD_DIV[4:0] + 1) * 256\mu s$ ）。

ウォッチドッグでは、ホスト・マイクロコントローラが有効なWD_KEYレジスタ書き込みを定期的送信する必要があります。応答は、 t_{WD1} の時間が経過した後、 t_{WD2} の時間が経過するまでに書き込まなくてはなりません。

t_{WD1} 時間および t_{WD2} 時間の計算方法は次のとおりです。

$$t_{WD1} = t_{WDCLK} * (WD_OPN[3:0] + 1) * 8$$

$$t_{WD2} = t_{WD1} + t_{WDCLK} * (WD_CLO[3:0] + 1) * 8$$

ここで、 t_{WD1} はクローズド・ウィンドウの時間、 t_{WD2} はクローズド・ウィンドウの時間にオープン・ウィンドウの時間（最大値）を加えた値、 t_{WDCLK} はウォッチドッグのクロック周期です。

最初に有効化されたとき、ウォッチドッグは延長されたオープン・ウィンドウに入ります。最初のオープン・ウィンドウは、延長されており、ホスト・マイクロコントローラが初期化するのに十分な時間が確保されます。この延長時間は、CONFIG_WD1レジスタのWD_1UDビットフィールドで設定され、 $t_{WD2} * (WD_1UD[2:0] + 1)$ と定義されます。この時間は、ウォッチドッグが最初に有効化されるときにのみ、 t_{WD2} に追加されます。

応答は、 t_{WD1} の時間が経過した後、 t_{WD2} の時間が経過するまでに書き込まなくてはなりません。ウォッチドッグの有効な更新によって、オープン・ウィンドウが終了し、クローズド・ウィンドウが開始され、WD_KEYレジスタが更新されます。

注:ウォッチドッグが有効化されている間にウォッチドッグ設定レジスタをプログラムすることは、予期せぬ動作の原因となる可能性があるため、推奨しません。

WD_FAULT_CNTが、WD_DBNCで設定された値に等しくなると、MAX17851はBMS安全モニタリング・モードを開始します。BMS安全モニタリング・モードの間に有効なWD_KEY書き込みが送信されると、MAX17851はコマンド動作モードに戻り、ウォッチドッグは延長オープン・ウィンドウに入って通常動作を再開します。

WD_FAULT_CNTレジスタに累積されたウォッチドッグ違反の数は、何らかの有効なウォッチドッグ更新が受信されるとリセットされます。

ウォッチドッグがチャレンジ応答モードになっているときに無効なウォッチドッグ・キーが書き込まれると、STATUS_WDレジスタのWD_LFSR_ERRビットがセットされます。チャレンジ応答モード時のクローズド・ウィンドウの時間にウォッチドッグ・キーが書き込まれると、WD_RJCT_ERR (STATUS_WDレジスタ) がセットされます。これらのビットは、WD_ENが0に設定されるとクリアされます。

オープン・ウィンドウの間に有効なWD_KEYの書込みに失敗すると、ウォッチドッグがタイムアウトし、WD_FAULT_CNTがインクリメントする原因になります。ウォッチドッグがタイムアウトするたびに、WD_EXP_ERRビット (STATUS_WDレジスタ) がセットされます。このビットは、WD_ENが0に設定されるとクリアされます。

WD_FAULT_CNTが設定済みのWD_DBNC値に等しくなると、WD_TO_ERRビット (STATUS_WDレジスタ) がセットされます。このビットは、WD_ENが0に設定されるとクリアされます。

WD_ERR_ALRTENビットがセットされると (デフォルト)、WD_ERR_ALRTビット (ALERT_GENレジスタ) がセットされ、ウォッチドッグ・エラーが生じるとALERTピンがアサートされます。WD_ERR_ALRTビットをクリアしてからALERTピンをデアサートする必要があります。WD_ERR_ALRTビットがアサートされる原因となるウォッチドッグ・エラーは、WD_TO_ERR、WD_LFSR_ERR、WD_RJCT_ERR、WD_EXP_ERRです。

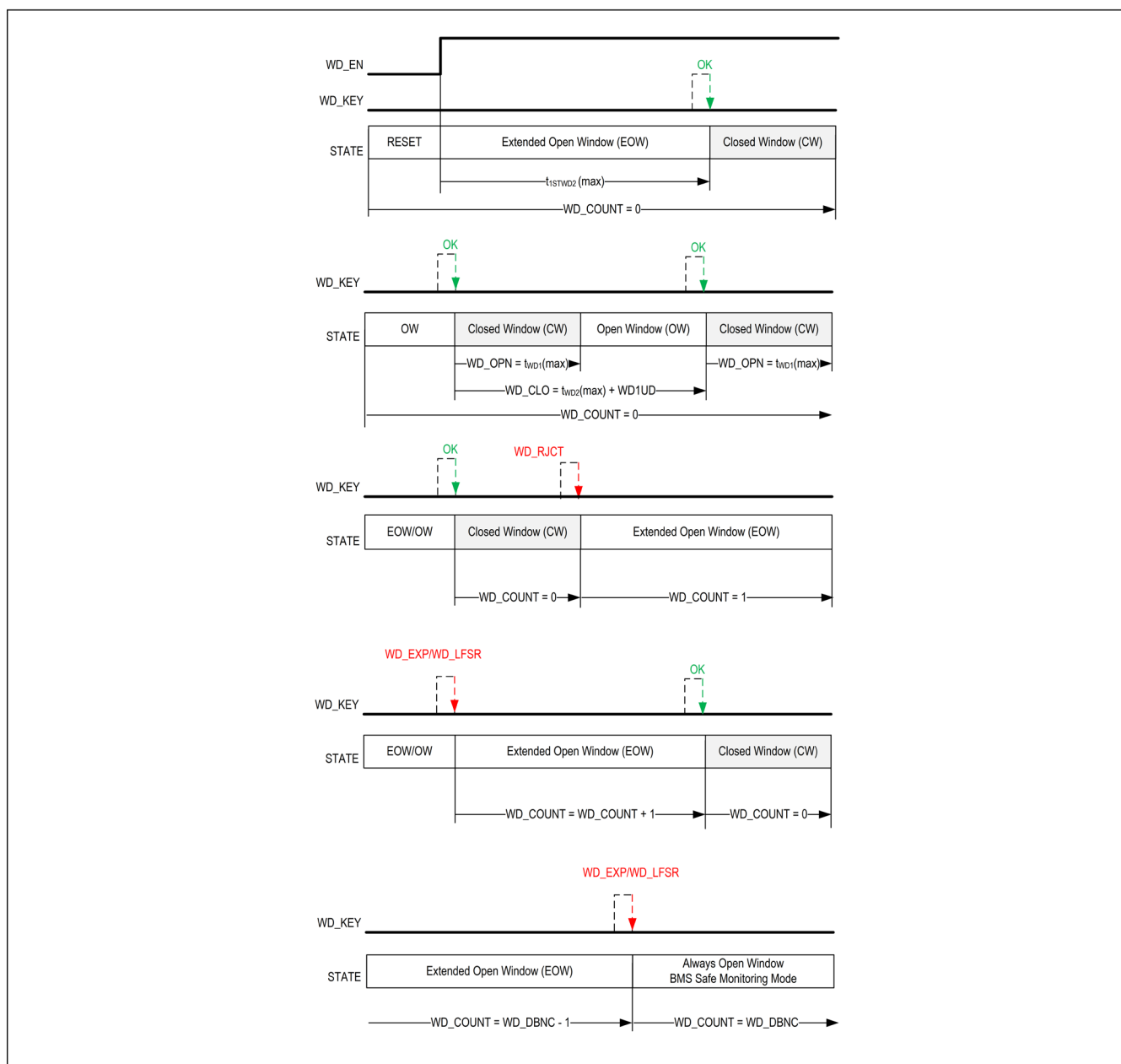


図11. ウォッチドッグ動作

チャレンジ／応答モード

ウォッチドッグのチャレンジ応答設定(WD_SWW=0)では、WD_KEYレジスタに書き込まれたデータを、線形帰還シフト・レジスタ(LFSR)アルゴリズムを用いて計算し、最新のキーに対して必要な書込み応答を計算する必要があります(詳細については、[LFSR/CRCコードの例](#)のセクションを参照)。

ホスト・コントローラは、WD KEYレジスタからいつでも最新のキーを読み出すことができます。

単純なウィンドウ・モード

単純なウィンドウ・ウォッチドッグ (WD_SWW = 1) を用いると、WD_KEY にどの値を書き込んでも有効なウォッチドッグ更新信号を供給できます。レジスタに書き込まれた値は無視されます。

LFSR/CRCコードの例

```
// feedback polynomial: x^8 + x^6 + x^3 + x^2 + 1
uint8 lfsr(uint8 iKey)
{
    lfsr = iKey;
    uint8 bit = ((lfsr >> 7) ^ (lfsr >> 5) ^ (lfsr >> 2) ^ (lfsr >> 1)) & 1;
    lfsr = (lfsr << 1) | bit;
    return lfsr;
}
```

GPIO制御

GPIOの機能は、個々のGPIO[n]_CFGレジスタの設定によって決まります。

GPIOピンを汎用I/Oに設定して明示的なレジスタ書き込みによってコマンドを送信することもできますが、BMS安全モニタリング・モード (SAFEMON) 内の特定の機能を実行するよう個別に設定することもできます。

シングルUART設定 (MS_EN=0b10) の場合、GPIO1とGPIO2は、バッテリー・コンタクタのリレー・ドライバ・インターフェースにロジック信号を送るための静的出力 (ハイまたはロー) として設定されます。GPIO3とGPIO4は、無応答状態になった場合にマイクロコントローラまたはマイクロコントローラの電源ネットワークをリセットするよう設定されます。このリセット機能は、特定のデバイス要件に応じて様々に設定できますが、一般的には、ワンショットの負パルスの形を取ります (デフォルトはワンショットのロー信号)。リセットが適用されリリースされるよう、ワンショットの発生時間は100msになっています。

デュアルUART設定 (MS_EN=0b11) の場合、[BMS安全モニタリング・モードおよび安全対策診断状態への移行](#)のセクションで説明されているように、マスタ・デバイスは、スレーブ・デバイスへの制御を停止します。この設定では、マスタはGPIO3をSAFEMONアクティブ・ハイに設定する必要があります。また、オプションでGPIO4を使用し、マイクロコントローラまたはマイクロコントローラの電源ネットワークを駆動することができます。スレーブは、GPIO4をSAFEMONスレーブ入力に設定する必要があります。また、オプションでGPIO3を使用して、マスタが制御しないマイクロコントローラまたはマイクロコントローラの電源ネットワークを駆動することができます。これによって、GPIO3マスタとGPIO4スレーブの間でハンドシェイクが可能になります。

マスタ/シングルUARTまたはマスタ/スレーブ・デュアルUARTのいずれかにおいて、汎用入力、汎用ハイ、汎用ローが選択されている場合、これらは、デバイスが安全モニタリング・モードまたはスリープ・モードになった場合も、その設定が維持されます。

GPIO3およびGPIO4がSAFEMON出力に設定されていると、コマンド動作モードまたはスリープ・モードの場合に、これらは非アクティブ状態でアイドルリングします。

GPIO1およびGPIO2がSAFEMON出力に設定されていると、SAFEMON_GPIO12_ALRTビットがセットされていない限り、コマンド動作モードまたはスリープ・モードの場合に、これらは非アクティブ状態でアイドルリングします。

予約済みのGPIOピン設定は、高インピーダンス入力としてアイドルリングします。

注: 高インピーダンス入力でアイドルリングしている場合、GPIOピンにはプルダウン抵抗RGPIOがあり、ピンをグランドにします。

表5. GPIO制御

INPUT			OUTPUT					
GPIO1	General Purpose Input (Default)		General Purpose HI	General Purpose LO			SAFEMON Active HI	SAFEMON Active LO
GPIO2	General Purpose Input (Default)		General Purpose HI	General Purpose LO			SAFEMON Active HI	SAFEMON Active LO
GPIO3	General Purpose Input (Default)		General Purpose HI	General Purpose LO	SAFEMON 1-Shot HI	SAFEMON 1-Shot LO	SAFEMON Active HI	SAFEMON Active LO

表5. GPIO制御（続き）

GPIO4	General Purpose Input (Default)	SAFEMON Slave Input	General Purpose HI	General Purpose LO	SAFEMON 1-Shot HI	SAFEMON 1-Shot LO	SAFEMON Active HI	SAFEMON Active LO
-------	---------------------------------	---------------------	--------------------	--------------------	-------------------	-------------------	-------------------	-------------------

GPIO[n]_RDビットは、ポートが入力として定義されているか出力として定義されているかに関わらず、そのピンのロジック・レベルをモニタします。GPIOで競合がなくなり、読出しレベルが設定出力に等しくなくなると、GPIO[n]_ERRステータス・ビットがアサートされます。

注：GPIO[n]_ERRは、GPIO出力設定にのみ適用されます。

シリアル・ペリフェラル・インターフェース (SPI)

SPIポートは同期型のデータ・リンクであり、ホストは、これを用いてMAX17851レジスタおよびUART通信バッファとの間で読書きを行います。

SPIトランザクション

ホストがCSピンをローに駆動すると、SPIトランザクションが開始されます。ホストは、常に最上位ビット (MSB) ファーストでMAX17851にデータを送信します。最初のバイト後、ホストは、トランザクションを終了する (シングルバイト・コマンド・トランザクション)、データのクロック同期出力を続ける (書込みトランザクション)、あるいは、データのクロック同期入力を開始する (読出しトランザクション) ことができます。ただし、データの送信と受信を同時に行うこと (半2重動作) はできません。

全てのトランザクションについて、ホストは、まず7ビットのレジスタ・アドレスを送信します。読出しおよび書込みは、8番目のビットで示されます (R/Wbビット)。読出しトランザクションの場合、2番目のバイトは、MAX17851がホストに送信する読出しデータです。書込みトランザクションの場合、2番目のバイトは、ホストがMAX17851に送信する書込みデータです。

CSがアクティブ・ローである限り、複数のデータ・バイトの読出しまたは書込み (バースト・モード動作) が可能です。MAX17851は、次の読出し専用レジスタ・アドレスまたは次の書込み専用アドレスを、自動的にインクリメントします。インコヒーレントなバースト・モード・アクセスを防止するため、定められた特定のアドレスについては自動アドレス・インクリメントが無効化されています。

自動アドレス・インクリメントが無効化されているのは、以下に示すUARTバッファ・レジスタ・アドレスです。

- RX_RD_MSG
- RX_RD_NXT_MSG
- ALRTPCKTBUF_RD_MSG
- NXT_LDQ
- LDQ
- CONFIGQ

これらのレジスタがバースト・モード・トランザクションでアクセスされた場合、内部のSPIアドレスは変化せず、代わりに、そのアドレスへの関連バッファ・ポインタがインクリメントされます。

自動アドレス・インクリメントは、以下のコマンド・レジスタ・アドレスについても無効化されています。

- CLR_TXBUF
- CLR_RXBUF
- CLR_LSSM
- CLR_ALIVECOUNT_SEED
- SWPOR
- SLP_EN
- VER_CONFIG
- LOAD_CONFIG

ホストがCSをハイに駆動すると、SPIトランザクションは終了します。

注：自動アドレス・インクリメント動作が無効化されたレジスタがバースト・モード・トランザクションに含まれている場合、自動アドレス・インクリメントは、そのレジスタの前のアドレスで停止します。

送信キューの詳細については、[送信バッファ](#)のセクションを参照してください。

受信キューの詳細については、[受信バッファ](#)のセクションを参照してください。

表6. SPIレジスタの一覧

アドレス	名称	説明
0x00 to 0x0F	Status Registers (See Register Table)	MAX17851の現在のステータス。レジスタは読出し専用です。
0x10 to 0x1F	Alert Registers (See Register Table)	MAX17851アラート・レジスタの現在のステータス。ステータスは、読出しまたはクリア(0の書き込み)のみが可能です。レジスタに1を書き込んでも何の影響もありません。
0x20 to 0x2F	Alert Enable Registers (See Register Table)	MAX17851アラート・イネーブル・レジスタの現在のステータス。ステータスは読出しあるいは書き込みが可能です。
TXバッファ・レジスタ		
0x40	CLR_TXBUF	コマンド:送信バッファをデフォルト状態にリセットし、TX_QおよびLD_Qをクリアします。
0xB0	NXT_LDQ	コマンド:LDQをインクリメントし、その後、送信バッファ・ロード・キューを書き込みます。
0xC0	LDQ	コマンド:送信バッファ・ロード・キューの読出し／書き込みを行います。
0xC2	LDQ_PTR	送信LDQバッファ内でホストが読出しまたは書き込みを行う場所。
0xD0	CONFIGQ	コマンド:CONFIG_BYTE_PTRレジスタおよびCONFIG_QUEUE_PTRレジスタの指示に従い、設定データ・キュー・バイトの読出し／書き込みを行います。
0xD2	CONFIG_PTR	CONFIG_QUEUE_PTRおよびCONFIG_BYTE_PTRの場所。
RXバッファ・レジスタ		
0x42	CLR_RXBUF	コマンド:受信バッファおよび受信バッファ・ポインタをデフォルト状態にリセットします。
0x8C	ALRTPCKTBUF_RD_PTR	アラート・パケット・バッファ用の読出しポインタの場所。
0x8E	ALRTPCKTBUF_RD_PTR	コマンド:アドレスALRTPCKTBUF_RD_PTRを始点として受信バッファを読み出します。バイトが読み出された後に読出しポインタを自動的にインクリメントしますが、次のメッセージへの読出しポインタはインクリメントしません。
0x90	RX_RD_MSG	コマンド:アドレスRX_RD_PTRを始点として受信バッファを読み出します。バイトが読み出された後に読出しポインタを自動的にインクリメントしますが、次のメッセージへの読出しポインタはインクリメントしません。
0x92	RX_RD_NXT_MSG	コマンド:アドレスRX_NXT_MSG_PTR(最も古い未読メッセージ)を始点として受信バッファを読み出します。バイトが読み出された後に読出しポインタを自動的にインクリメントしますが、次のメッセージへの読出しポインタはインクリメントしません。
0x96	RX_RD_PTR	受信バッファ内のホストが読出しを行う場所。UARTはこのポインタを自動的にインクリメントします。
0x98	RX_WR_PTR	データ受信時にUARTによって書き込まれる受信バッファ内の場所。
0x9A	RX_NXT_MSG_PTR	受信バッファ内の次の未読メッセージの先頭。RX_RD_Pointerには、RD_NXT_MSG SPIトランザクションによってこの値がロードされます。
0x9C	RX_SPACE	受信バッファでアクセス可能なバイトの数。
コマンド・レジスタ		
0x44	CLR_LSSM	コマンド:LSSMをリセットします。
0x48	CLR_ALIVECOUNT_SEED	コマンド:ALIVECOUNT_SEEDレジスタをクリアします。
0x4A	SWPOR	コマンド:ソフトウェア・パワーオン・リセット。
0x4C	SLP_EN	コマンド:スリープ・モードを有効化します。
0x4E	VER_CONFIG	コマンド:設定メモリを検証します。
0x50	LOAD_CONFIG	コマンド:設定メモリをロードします。
0x52	WD_KEY	コマンド:ウォッチドッグ・キーの値の読出し／書き込みを行います。

SPIのタイミング

AX17851は、SPIモード0(CPOL = 0/CPHA = 0)にのみ対応しています。このモードでは、データは常にSCLKの立下がりエッジで出力され、SCLKの立上がりエッジでサンプリングされます。読出しの場合、MAX17851は、R/Wbビット後の最初のSCLKの立下がりエッジで、デジタル出力(DOUT)の駆動を開始します。読出しの間にDOUTデータを駆動しても、データ入力(DIN)には影響はありません。アドレス空間を越えて読出しを行おうとしてもゼロが返されます。

書込みの場合、最後のビットがサンプリングされた後、SCLKの立下がりエッジでレジスタに書込みが行われます。ただし、SCLKの最終ビットの立下がりエッジ前に $\overline{CS}$ がハイになると、そのレジスタには書込みは行われません。

表7. SPI通信の概要

PARAMETER	VALUE
Communication Mode	Half-duplex
Maximum Clock Frequency	10 MHz
Bit Order	Most-significant bit first
Clock Polarity (CPOL)	0 (leading clock edge is rising edge)
Clock Phase (CPHA)	0 (data sampled on leading clock edge)

SPIでの読出しと書込みに対する制限

SPIの読出しおよび書込みは、コマンド動作モードでは制限がありません。しかし、指定された動作モードにおいては、MAX17851内で状態が競合するのを防止するため、一部のレジスタの書込みおよび読出しが制限されます。

これらの指定されたモードでは、無効化されたレジスタ書込みが行われても無視され、SPI_ERR_ALRTビットがセットされます。無効化されたレジスタ読出しはゼロが返され、SPI_ERR_ALRTビットがセットされます。

表8. SPIでの読出しと書込みに対する制限

REGISTER	COMMANDED OPERATION	SLEEP MODE	SAFETY MONITORING MODE	LOAD CONFIG	VERIFY CONFIG
STATUS Registers	Unrestricted	Unrestricted	Unrestricted	Unrestricted	Unrestricted
ALERT Registers	Unrestricted	As Defined	As Defined	As Defined	As Defined
ALERTEN Registers	Unrestricted	WR Disabled	WR Disabled	WR Disabled	WR Disabled
COMMAND CLR_TXBUF	Unrestricted	WR Disabled	WR Disabled	WR Disabled	WR Disabled
COMMAND CLR_RXBUF	Unrestricted	WR Disabled	WR Disabled	WR Disabled	WR Disabled
COMMAND CLR_LSSM	Unrestricted	WR Disabled	WR Disabled	WR Disabled	WR Disabled
COMMAND CLR_ALIVECOUNT_SEED	Unrestricted	WR Disabled	WR Disabled	WR Disabled	WR Disabled
COMMAND SWPOR	Unrestricted	Unrestricted	Unrestricted	Unrestricted	Unrestricted
COMMAND SLP_EN	Unrestricted	Unrestricted	WR Disabled	WR Disabled	WR Disabled
COMMAND VER_CONFIG	Unrestricted	WR Disabled	WR Disabled	WR Disabled	Unrestricted
COMMAND LOAD_CONFIG	Unrestricted	WR Disabled	WR Disabled	Unrestricted	WR Disabled
COMMAND WD_KEY	Unrestricted	Unrestricted	Unrestricted	Unrestricted	Unrestricted
CONFIG Registers	Unrestricted	WR Disabled	WR Disabled	WR Disabled	WR Disabled
RX_COMMAND Registers	Unrestricted	Unrestricted	Unrestricted	Unrestricted	Unrestricted
TX_COMMAND Registers	Unrestricted	RD/WR Disabled	RD/WR Disabled	RD/WR Disabled	RD/WR Disabled
INFO Registers	Unrestricted	Unrestricted	Unrestricted	Unrestricted	Unrestricted

UARTインターフェース

アナログ・デバイスズのバッテリー管理UARTプロトコルを用いるスレーブ・デバイスは、複数のバッテリーセル・スタックを管理するために、デジタイズ形式で接続できます。BMSでは、コントローラが全てのスレーブ・デバイスのホストとなり、全ての通信を開始します。[図12](#)に示すように、データ・フローは常にホストから始まり、デジタイズチェーンを経由してホストに戻ります。

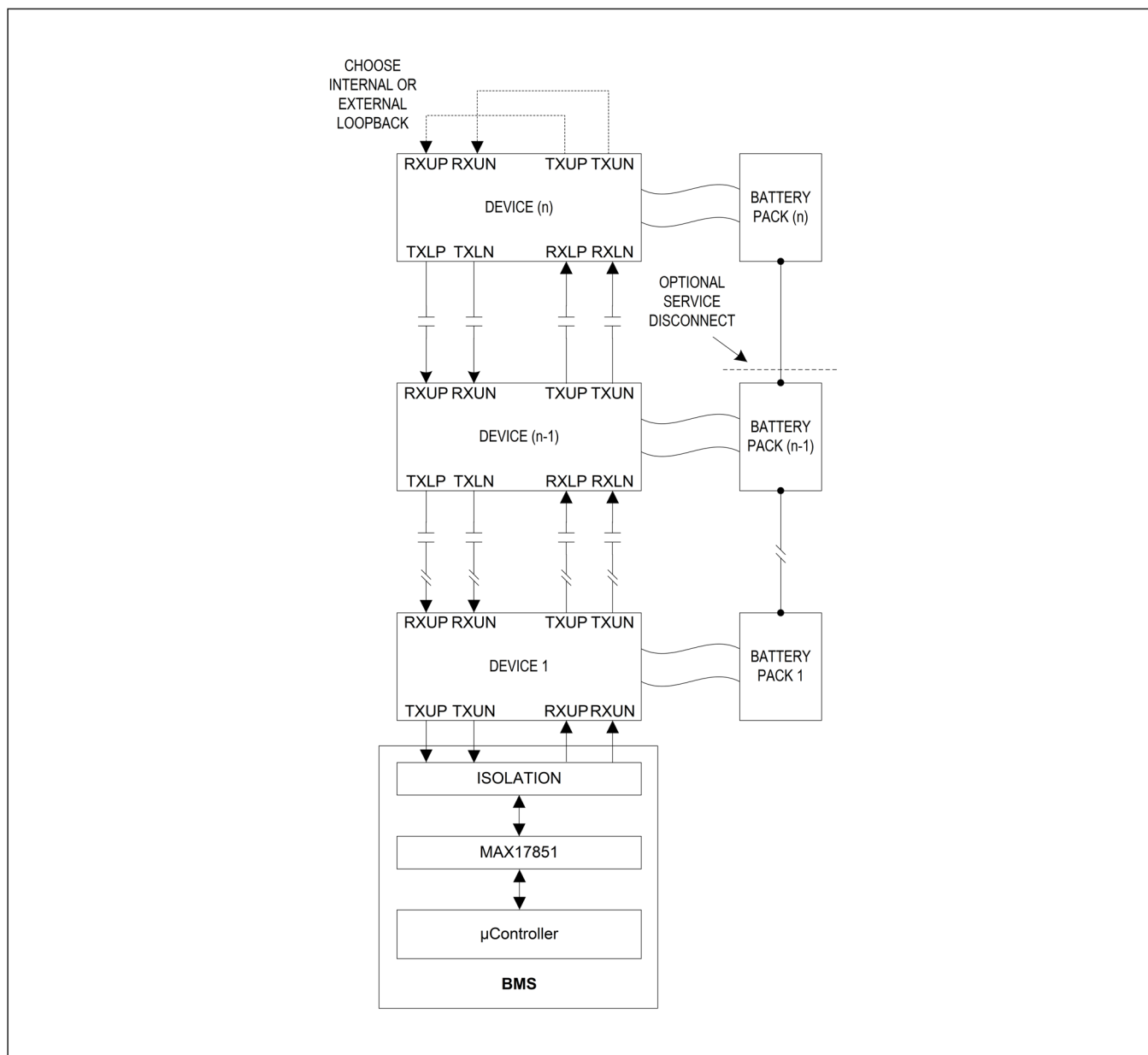


図12. システムのデータ・フロー

バッテリー管理UARTプロトコル

MAX17851は、アナログ・デバイスズのバッテリー管理デバイス専用に設計されたUARTプロトコルを用いています。このプロトコルは、以下の機能を使用して通信の完全性を最大限まで高めます。

- 送信された全てのデータはマンチェスター・エンコーディング (G.E. Thomas方式) されて各データ・ビットが2回送信され、2回目に送るビットは反転されます。
- 送信されるキャラクタは全て12ビットで構成され、これにはSTARTビット、パリティ・ビット、および2つのSTOPビットが含まれます。
- 各メッセージは、プリアンプル・キャラクタとオプションのストップ・キャラクタでフレーム化されます。
- 受信される各読み出しコマンド・メッセージ (READALL、READBLOCK、READDEVICE) には、送信の完全性を検証するためにオプションでデータチェック・バイトを含めることができます。

- READALL、READBLOCK、READDEVICE、WRITEALL、WRITEDEVICE、ALERTPACKETの各メッセージは、CRC-8のバケット・エラー・チェック(PEC)バイトで保護されています。
- READALL、READBLOCK、READDEVICE、WRITEALL、WRITEDEVICE、ALERTPACKETの各メッセージは、オプションで後ろにアライブ・カウント・バイトを伴うことができます。

また、このプロトコルは、データ・リンクのアイドル状態が指定された時間続いた場合にスレーブ・デバイスをシャットダウンできるようにすることによって、消費電力を最小限に抑えるように設計されています。スレーブ・デバイスの意図しないシャットダウンを防止するために、ホストは、MAX17851のキープアライブ・モードを有効にして、キープアライブ・キャラクタを定期的を送信する必要があります。キープアライブ・キャラクタ同士の時間間隔は、ホストによって設定可能です。

UARTメッセージ

メッセージは、一連のUARTキャラクタとして定義されます。メッセージは、プリアンブル・キャラクタで始まり、データ・キャラクタが続き、ストップ・キャラクタで終了します。各キャラクタは、以下に示す12ビットで構成されます。

- 1個のSTARTビット
- 8個のデータ・ビット (LSBファースト)
- 1個のパリティ・ビット (偶数)
- 2個のSTOPビット

各データ・バイトは、4ビットのデータ・ニブルごとに1つの12ビット・キャラクタを持つ、2つの別々のキャラクタとして送受信されます。各マンチェスター・エンコード・ニブルには、8個のデータ・ビット (4個の真のビットと4個の反転ビット) が必要です。デフォルト設定では、MAX17851がメッセージを送信する場合、以下の機能が自動的に実行されます。

- メッセージの最初に必要なプリアンブル・キャラクタを使ってメッセージをフレーム化。
- 各データ・ニブルをマンチェスター・エンコーディングし、必要なSTARTビット、パリティ・ビット、STOPビットと共にエンコードされた各ニブルを送信。
- 0.5Mbps、1Mbps、2Mbps、4Mbpsのいずれかに設定されたボー・レートでメッセージを送信。
- メッセージの最後に必要なストップ・キャラクタを使ってメッセージをフレーム化。

これらの自動機能が無効化するには、次の特別な送信モードを有効化します。

- プリアンブルなし送信モード (プリアンブル・キャラクタを削除)
- ストップなし送信モード (ストップ・キャラクタを削除)
- 生データ送信モード (マンチェスター・エンコーディングを行わずにデータを送信)
- 生データ受信モード (マンチェスター・エンコーディングされていないものとしてデータを受信)

プリアンブル・キャラクタ

プリアンブルは、メッセージの開始を知らせるためにUARTが生成するフレーミング・キャラクタです。エンコードされない15hとして送信されますが、なおもDCバランスの取れたキャラクタです。ストップ・ビット以外のいずれかのビットが固有のプリアンブル・シーケンスから外れている場合、そのキャラクタは有効なプリアンブルと見なされず、データ・キャラクタとして扱われます。

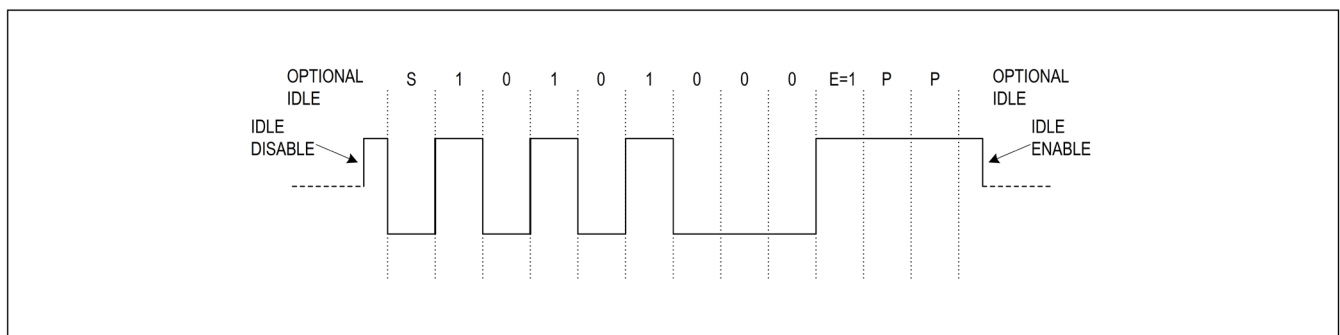


図13. UARTでのプリアンブルのタイミング

ストップ・キャラクタ

ストップ・キャラクタは、メッセージの終了を知らせるためにUARTが生成するフレーミング・キャラクタです。エンコードされない54hとして送信されますが、なおもDCバランスの取れたキャラクタです。

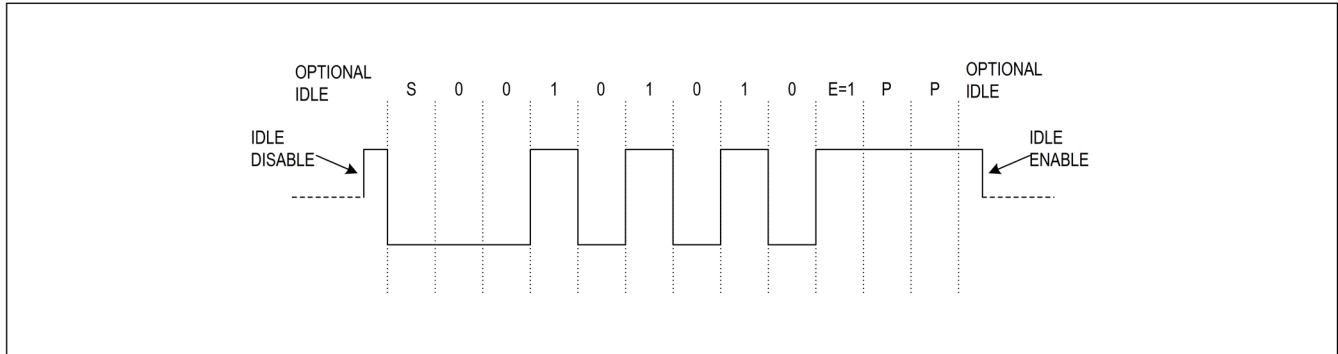


図14. UARTでのストップ・キャラクタのタイミング

マンチェスター・エンコーディング

各データ・バイトは、マンチェスター・エンコーディングされた2つの別々のニブル(4ビット)として送信されます。各データ・ビットでは、最初のビットが情報を表し、2番目のビットがその補数となります。パリティは偶数となるため、その値は常に、偶数のハイ・ビットになります。データがマンチェスター・エンコーディングされ、2つのSTOPビットがあるため、データ・キャラクタ(ただしフレーミング・キャラクタは除く)のパリティ・ビットは常にゼロとなるはずで

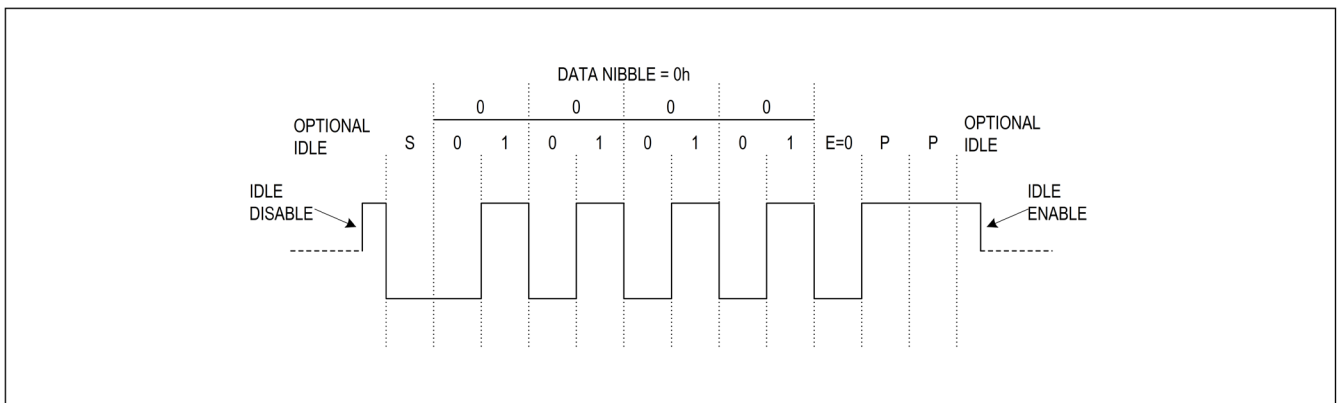


図15. UARTでのマンチェスター・エンコーディング・データ・ニブル0hのタイミング

データ・タイプ

アナログ・デバイスズのバッテリー管理UARTプロトコルは、表9に示すいくつかの異なるデータ・タイプに対応しています。送信するデータの構成と受信するデータの解析は、ホストが行います。例えば、ホストは、各送信メッセージのPEC値を正しく計算し、各受信メッセージのPEC値を検証しなくてはなりません。

スレーブ・デバイス・アドレスの割り当て

バッテリー管理UARTプロトコルでは、ホストが、必要に応じて各UARTスレーブ・デバイスに0~31の一意の連続的なアドレスを割り当てる必要があります。ホストは、HELLOALLコマンド・シーケンスのシード・アドレスを指定することで、この割り当てを実行します。コマンドがデジチェーン内を伝搬するに伴い、各スレーブ・デバイスは、その固有アドレスを指定します。HELLOALLシーケンスは、1つの値を返し、その値からホストはデジチェーン内のデバイス数とデバイス・アドレスを決定できます。

UARTのメッセージ・データ・タイプ

表9. UARTのメッセージ・データ・タイプ

データ・タイプ	説明
Command	メッセージのタイプ(書き込みコマンドまたは読み出しコマンド)を定めます。
Address	読み出しまたは書き込みの対象となるレジスタ・アドレス。
Data	読み出しまたは書き込みの対象となるレジスタ・データ。
Fill	送信される合計バイト数が受信バイト数に等しくなるように読み出しコマンドの一部として送信される、値がC2hまたはD3hのバイト。ただし、これらのバイトは元の値でレシーバに返されることはありません。その代わりに、各スレーブ・デバイスが、フィル・バイトをホストから要求されたレジスタ・データで置き換えます。
Data-Check	スレーブ・デバイスから提供されたエラー・ステータス。読み出し時にのみ返されます。
PEC	CRC-8の packets・エラー・チェック・バイト。メッセージごとに送信と返信が行われます。
Alive-Counter	送信されたメッセージに回答しているデバイスの数を検証するために用いられます。このバイトはオプションですが、エラー・チェックのために推奨します。

一般的なUARTコマンド

表10. 一般的なUARTコマンド

コマンド・バイト	値	説明
HELLOALL	57h	デジタイチェーン内の各デバイスに一意のデバイス・アドレスを書き込みます。システムの初期化が必要です。
ALERTPACKET	21h	デジタイチェーンの全てのデバイスからアラート・パケットを読み出します。
WRITEDEVICE	{DA[4:0], 0b100}	指定されたレジスタを1つのデバイスに書き込みます。
WRITEALL	02h	指定されたレジスタを全デバイスに書き込みます。
READDEVICE	{DA[4:0], 0b101}	指定されたレジスタを1つのデバイスから読み出します。
READALL	03h	指定されたレジスタを全デバイスから読み出します。
READBLOCK	{BS[4:0], 0b110}	1組のレジスタを1つのデバイスから読み出します。
DOWNHOST	09h	デジタイチェーンのデバイスを下り経路での書き込みアクセス用に設定します。
UPHOST	08h	デジタイチェーンのデバイスを上り経路での書き込みアクセス用に設定します。
Reserved	{_, 0b0000}, {_, 0b1010}, {_, 0b1011}, {_, 0b1111}	予約済み。 注: 予約済みのコマンド・バイトは、ロックステップ安全対策機能によって有効と認識され、読み出し動作と同様に処理されます。

UARTの動作

UARTは、UARTスレーブ・デバイスとの間でメッセージの送受信を行うためのサブシステムです。図16に示すように、ホストはSPIバッファ・トランザクションを用いて、コード化されていない送信メッセージを送信バッファに格納し、デコードされた受信メッセージを受信バッファから読み出します。UARTバッファは、UARTの各バッファのサイズと構成を示します。

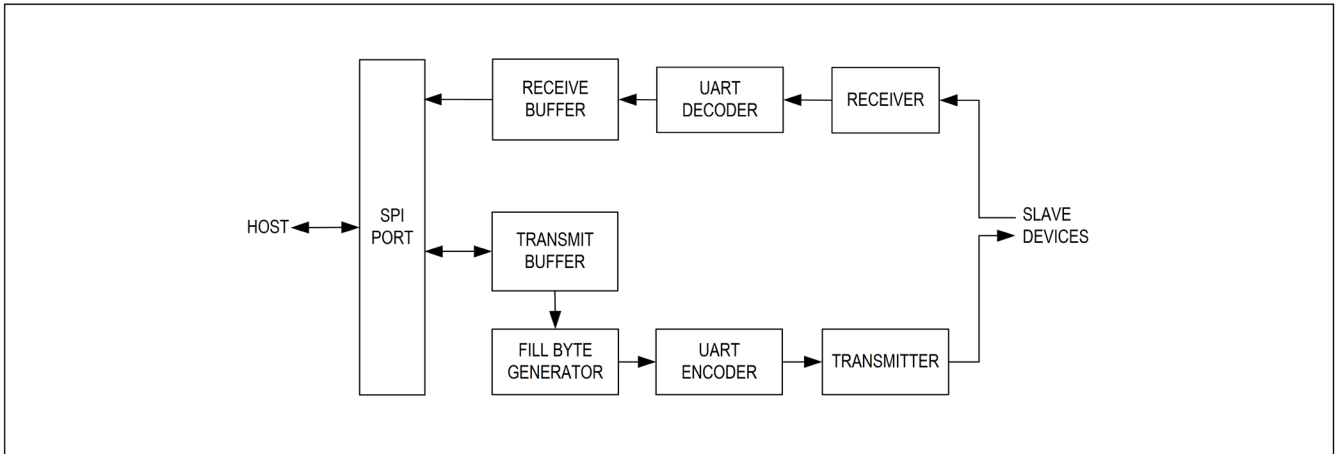


図16. UARTのデータ・フロー

UARTの動作モード

表11. UARTの動作モード

モード	CONFIG_GEN2レジスタ またはCONFIG_GEN3 レジスタのビット	説明
Transmit Preambles	TX_PREAMBLES (CONFIG_GEN2)	プリアンプルを連続的に送信します(アイドル状態なし)。UARTスレーブ・デバイスを起動し各スレーブ・デバイスのUARTボー・レートを初期化するために用います。このモードは、送信停止モードを除く全ての送信モードに優先します。
Keep-Alive	ALRTPCKT_TIMING (CONFIG_GEN3)	ストップ・キャラクタを定期的に送信し、通信のない期間(アイドルリング状態)にUARTスレーブ・デバイスがシャットダウンするのを防止します。定期的なストップ・キャラクタ同士の間のアイドルリング時間は、ALRTPCKT_TIMING[3:0]の設定を用いて0ms~10.24msの範囲で設定できます。デフォルト設定は無限です(モードを無効化)。送信停止、プリアンプル送信、キュー送信の各モードは、このモードに優先します。なお、CO_ALRTPCKTEN=0の場合、キープアライブ(ストップ・キャラクタ)が送信される点に注意してください。CO_ALRTPCKTEN=1の場合は、アラート・パケットが送信されます。詳細については、 アラート・パケット・バッファ のセクションを参照してください。
Transmit Queue (default mode)	TX_QUEUE (CONFIG_GEN2)	以下の場合に、送信キューにロードされたメッセージの送信を開始します。1)メッセージに対し十分な容量が受信バッファにある(RX_FULLステータスが偽)。または2)メッセージ長に対する制限が解除されている(TX_Unlimitedがセット)。デフォルトでこの動作モードは有効化されています。
Transmit Unlimited	TX_UNLIMITED (CONFIG_GEN3)	このモードでは、送信キューは、メッセージ長をデフォルトの制限値である86バイトではなく、255バイトに自動的に制限します。メッセージ長が受信バッファに書き込み可能な容量を超えていても、メッセージの送信は許可されます。なお、このモードでは、LSSMバッファがオーバーフローする可能性があるため、LSSMエラーが発生することがあります。メッセージが送信され、通常のメッセージが再開された後は、LSSMをクリアする必要があります。
Transmit Pause	TX_PAUSE (CONFIG_GEN2)	UARTが最新のバイトの送信を終了した後、トランスミッタをアイドルリング状態にします。ただし、TX_BUSYとTX_IDLEのステータス・ビットは不変です。このビットがクリアされると送信が再開されます。このモードはその他の全ての送信モードに優先します。
Transmit Odd Parity	TX_ODDPARITY (CONFIG_GEN2)	奇数パリティのキャラクタを送信します。バッテリー管理UARTプロトコルは偶数パリティを用いるため、このモードは、システムがパリティ・エラーを検出できるかどうかをテストするために用いることができます。偶数パリティがデフォルトです。
Transmit No Stop	TX_NOSTOP (CONFIG_GEN2)	ストップ・キャラクタのないメッセージを送信します。

表11. UARTの動作モード（続き）

モード	CONFIG_GEN2レジスタ またはCONFIG_GEN3 レジスタのビット	説明
Transmit No Preamble	TX_NOPREAMBLE (CONFIG_GEN2)	プリアンプルのないメッセージを送信します。
Transmit Raw Data	TX_RAW_DATA (CONFIG_GEN2)	送信データのマンチェスター・エンコーディングを無効化します。このモードでは、各データ・バイトは、2個のキャラクタではなく1個のキャラクタとして送信されます。
Receive Raw Data	RX_RAW_DATA (CONFIG_GEN2)	受信データのマンチェスター・デコーディングを無効化します。このモードでは、1つのデータ・バイトを、受信した2個のキャラクタごとではなく、1個のキャラクタごとに格納します。

送信バッファ

17に、送信バッファのメモリ・マップを示します。送信バッファは、固定長の4個のキューで構成され、ホストはこれを用いて、出力するメッセージを格納します。常に、これらのキューの1つがロード・キュー（ロードされるキュー）に指定され、1つが送信キュー（アンロードされるキュー）に指定されます。ロード・キューは、2ビットのレジスタLD_Qで選択され、送信キューは、2ビットのレジスタTX_Qで選択されます。各キューは、31バイトで構成されます。

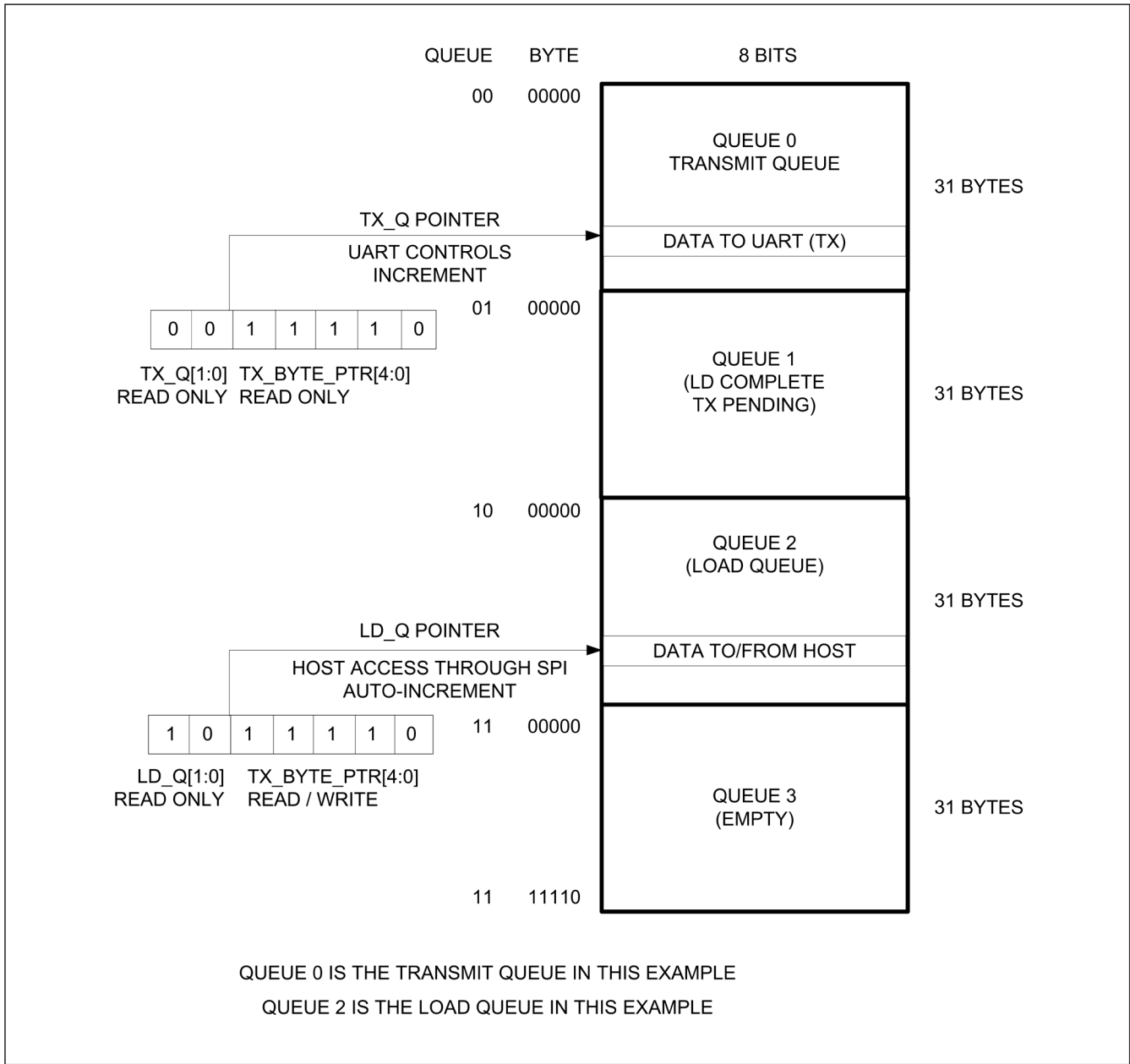


図17. 送信バッファのメモリ・マップ

送信バッファのキュー

それぞれのキューでは、場所「0」がメッセージ長のために予約済みとなっており、残りの31個の場所が特定のメッセージ・データ用となっています。各キューのデフォルト状態を表12に示します。

送信バッファのクリア

UARTの初期化の間、ホストがCLR_TXBUF SPIコマンドを発行して送信バッファをリセットすることを推奨します。これにより、送信バッファは以下のようにリセットされます。

- TX_Q [1:0] = 00b
- LD_Q [1:0] = 00b
- 送信バッファの全データが表12に従ってデフォルト状態にリセット。

メッセージ長

メッセージを作成する前に、ホストは、コマンドのタイプ(読み出しまたは書き込み)とデバイス数に応じて、メッセージの長さ(キャラクタ単位ではなくバイト単位)を計算する必要があります。メッセージ長には、必要なフィル・バイトを含める必要がありますが、プリアンブル・キャラクタおよびストップ・キャラクタは必ずしも含める必要はありません。ホストは、メッセージ長をロード・キューの場所「0」に書き込みますが、指定メッセージ長が86dを超える場合は、86d(56h)のみが書き込まれます。TX_UNLIMITED = 1の場合は、書き込める最大メッセージ長が255d(FFh)に増加しますが、ホストは、受信バッファに対しそれに応じた対応を行い、オーバーフローの可能性を避ける必要があります。

指定されたメッセージ長がユーザ・データの30バイトを超える場合は、UARTは、メッセージの後ろの部分を送信する間に、バッテリ管理UARTプロトコルの要件に従って、フィル・バイトを自動的に交互に(D3h、C2h)付加します。

メッセージ長がデジizerチェーン・デバイスの数およびトランザクション・タイプで指定される適切な長さに対応しない場合、LSSMが複数のステータス・エラーを通知します。この要件により、ホストから伝搬されたエラーは、デバイス内、またはデジizerチェーン内において、ISO-26262 ASIL-Dの基準に従って確実に検出されます。

表12. キューのメモリ・マップ

LOCATION	DESCRIPTION	DEFAULT VALUE	MAXIMUM DEFAULT PERMITTED	
			TX_UNLIMITED = 0	TX_UNLIMITED = 1
0	Message length	00h	56h	FFh
1	Data bytes and/or fill bytes	D3h	FFh	FFh
2		C2h		
3		D3h		
4		C2h		
5		D3h		
6		C2h		
7		D3h		
8		C2h		
9..29		C2h..D3h		
30		C2h		

ロード・キューの書き込み

メッセージは、フィル・バイトを除き、3〜31バイトで構成されます。例えば、HELLOALLシーケンスは、57h、00h、00h(最初のアドレスを0にセット)の3バイトです。フィル・バイトは必要ないため、全メッセージ長は3バイトです。したがって、ホストは、ロード・キューに表13に示すデータを書き込む必要があります。

ホストは、SPIトランザクションのセクションで示した該当のSPIコマンドを用いて、キュー内の任意の場所を始点としてロード・キューの書き込みを行うことができます。ただし、ホストがキューの場所「31」を越えて書き込みを行おうとしても、それ以上のデータは無視されます。

UARTは、LD_Qで選択されたキューの送信は試行しません。その理由は、ホストがそれをロードしている最中であるか、ロードを終えた場合でもロード・キューの内容を検証する(読み出す)必要がある可能性があるためです。その後、ホストは、NXT_LD_Qコマンドを書き込んで、ロード・シーケンスの次のキューを選択できます。これによりLD_Q値がインクリメントします。このインクリメントが生じた場合のみ、UARTはそれ以前にロードされたキューのデータ送信を開始します。LD_Q[1:0]とTX_Q[1:0]のどちらも、3hの値がインクリメントすると0hになります。

表13. HELLOALLメッセージでロードされるキューの例

LOCATION	VALUE	DESCRIPTION
0	03h	Message length
1	57h	Command byte
2	00h	Address byte
3	00h	Data byte
4	C2h	Not written
5	D3h	Not written
6..29	...	Not written
30	C2h	Not written
31	D3h	Not written

送信バッファのフィル

ホストは、LD_Q = TX_Q - 1となるまで、使用できる全てのキューをロードできます。LD_Q = TX_Q - 1の状態では、送信バッファは満杯になります (TX_FULLが真)。この状態では、ホストは、送信キューのロードを開始できません。UARTが依然としてデータのアンロード／送信を行っている可能性があるためです。送信バッファが満杯のときに、ホストがNXT_LD_Qコマンドを書き込もうとし、送信キューをロードしようとしている場合、インクリメントは行われず、オーバーフロー条件になっていることが通知されます (TX_OVERFLOWステータスが真)。送信バッファが満杯の場合は、ホストは送信キューを書き込むことができません。

メッセージの送信

送信されていないトランザクションがない場合は、送信バッファは空と見なされます (TX_EMPTYが真)。ホストは、キューの対処を終了すると、WR_NXT_LD_Qトランザクションを実行し、それまでにロードされたメッセージを送信して次のキューを選択します。

UARTは、以下の条件が満たされた場合に、WR_NXT_LD_Qコマンドで送信キューのアンロード／送信を行います。

- UARTがTransmit_Queueモードになっている (TX_QUEUEビットがセット)
- 送信バッファに少なくとも1つのロード・キューがある (TX_EMPTYが偽)
- 受信バッファにメッセージ用の十分な容量がある (RX_SPACEがメッセージ長より大きい)
- LSSMバッファにメッセージ用の十分な容量がある (LSSM_FULL != 1)

注: 受信バッファで使用できる容量の制限は、TX_UNLIMITEDビットを設定することで解除できます。

注: 送信バッファと自動生成ALERTPACKETコマンドの間に競合が存在する場合、送信バッファが最初に処理されます。

送信条件が満たされると、UARTは、必要なフィル・バイトを含め、メッセージ全体が送信されるまで、自動的に送信キューのアンロードを開始します。送信が完了した後、送信キューの内容がそのデフォルト値にリセットされ、キューは再びホストがロードできるようになります。

受信バッファ

受信バッファは、86バイトのサーキュラ・バッファで、ホストはこれをSPIで読み出せますが、データ受信時にはUARTでのみロードできます。このバッファは、受信バッファ・メモリ・マップ (図18) に示すように、以下の3つのポインタを用います。

- RX_RD_PTR: 読出しポインタ。つまり、ホストが読み出すバッファ位置 (デフォルトは00h、読出し専用)
- RX_WR_PTR: 書込みポインタ。つまり、UARTが書き込むバッファ位置 (デフォルトは01h、読出し専用)
- RX_NXT_MSG_PTR: 次の未読メッセージの開始点となるバッファ位置 (デフォルトは00h、読出し専用)

デフォルト状態では、読出しポインタが書込みポインタより1少なく、受信バッファは空と見なされます (RX_EMPTYが真)。この状態での受信バッファ・データ読出しはゼロです。

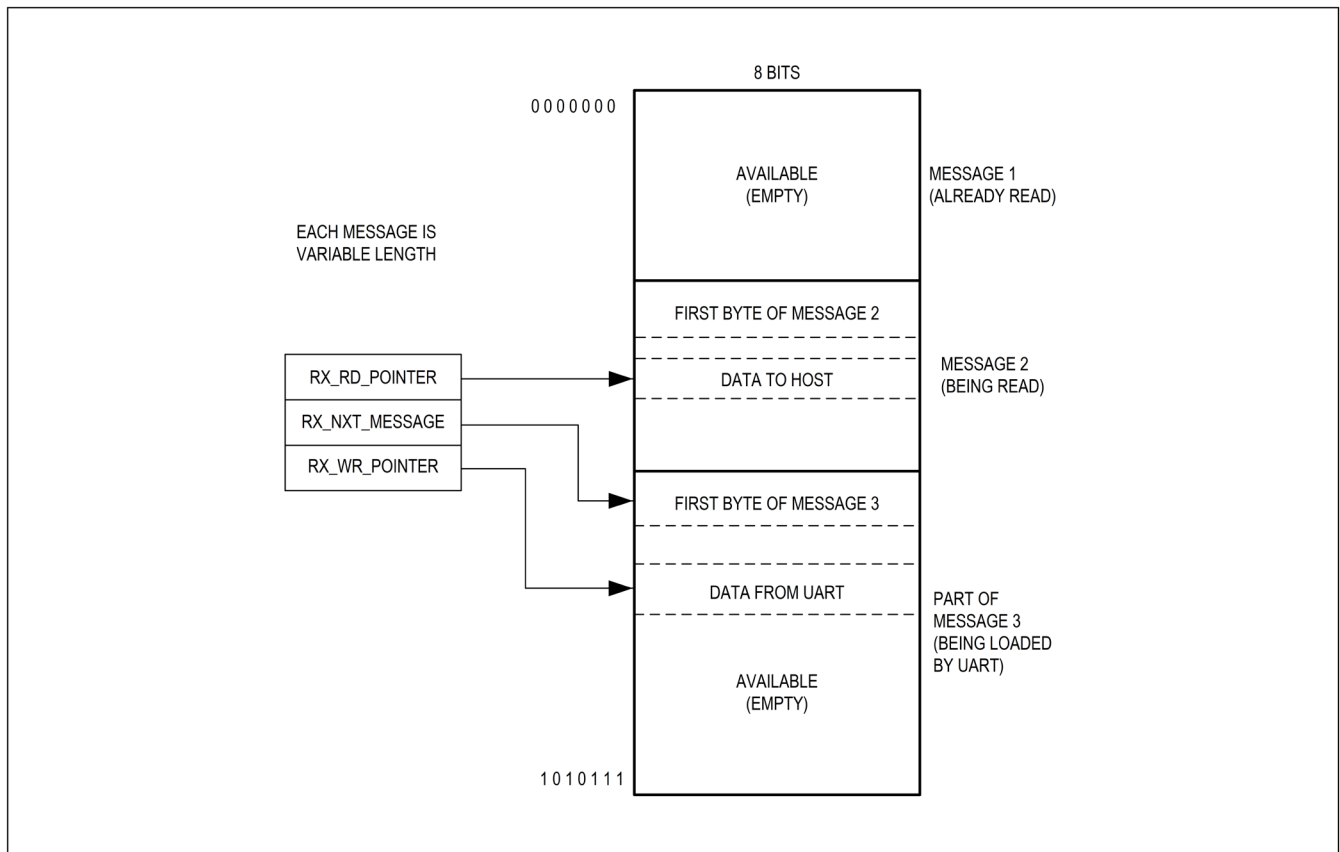


図18. 受信バッファのメモリ・マップ

受信バッファのクリア

UARTの初期化の間、ホストがCLR_RXBUF SPIコマンドを発行して受信バッファをクリアすることを推奨します。これにより、受信バッファは以下のようにリセットされます。

- RX_RD_Pointer:00h
- RX_WR_Pointer:01h
- RX_NXT_MSG_POINTER:00h
- 受信バッファ(86バイト)のデータは00hにクリア

プリアンブル送信モードの間に受信バッファがクリアされた場合、バッファの状態は確保されません。そのため、プリアンブル送信モードを無効化した後、ホストは、送信された全てのプリアンブルが受信されるまで待機してから、バッファをクリアする必要があります。

最後のプリアンブル後に受信した最初のキープアライブ・ストップ・キャラクタはヌル・メッセージとなるため、ホストは、単純にバッファが空でなくなる(RX_EMPTYが偽)まで待機してから、バッファをクリアできます。RXクリア・バッファ・コマンドは、RXバッファに対するだけでなく、UARTレシーバ・ロジックに対しても、非同期リセットの作用をします。UARTレシーバ・ロジックがリセットされると、受信するUART信号に再同期してからバイトを適切に処理しなくてはなりません。これを行うには、プリアンブル・バイトを受信するか、少なくともUARTバイト1つ分の期間アイドル状態を継続させます。

データ・ストリームを再同期するために用いるプリアンブルは、次に送信するメッセージの先頭にあるプリアンブルと同じではありません。アプリケーションでは、RXバッファのクリアに続いてこれらの条件の1つを満たしてから、MAX17851から次のメッセージを送信する必要がある必要があります。

メッセージの受信

UARTメッセージは、プリアンブル・キャラクタおよびストップ・キャラクタでフレーム化されます。UARTレシーバが有効なプリアンブルをデコードした場合、レシーバは、メッセージを受信する準備を行います。プリアンブルを受信バッファには格納しません。データが受信されると、バッファは空ではなくなり(RX_EMPTY = 0)、UARTは、ストップ・キャラクタを受信する、メッセージ・フレーム・タイムアウト(t_{FRMTO})を超過する、別のプリアンブルを受信する、のいずれかが生じるまで、デコードされたデータ・バイトを順次受信バッファに格納します。メッセージが終了したことを検出すると、UARTは受信バッファにヌル・バイト(00h)を格納し、RX_READYビットをセットします。RX_READYビットは、その後、全ての未読メッセージを読み出す(バッファが空)か、次のプリアンブルを検出すると、クリアされます。

注: RX_READYは、メッセージが受信され、RXバッファからリードバックする用意ができていることを通知します。ストップ・キャラクタが有効化されている場合(TX_STOP = 1)、RX_STOPステータス・ビットは、正しくフレーム化されたメッセージを受信したことを通知します。RX_READYがセットされRX_STOPがセットされていない場合、ストップ・キャラクタ内に破損が生じていることを示します。破損したストップ・キャラクタは、必ずしも、データ・ペイロード内の破損を示すわけではありません。この状況では、LSSM_BYTEを用いてデータ・ペイロードの完全性を検証する必要があります。

メッセージを受信するとALERTBピン(RX_READY_ALRTEN = 1)がアサートされるようにホストがMAX17851を設定することを推奨します。これにより、受信バッファをいつ処理できるかが明確になります。RX_READY_ALRTを受信する前にバッファを処理してはいけません。

複数のメッセージを受信バッファに格納して、ホスト・マイクロコントローラの処理効率を向上することができます。ALERTBピンのメッセージを介してメッセージ全体の正常な通信を確実なものにするには、次のメッセージを受信する前にRX_READY_ALRTをクリアします。RX_READY_ALRTをクリアしない場合、永続的なアラートが通信されます。

ホストが受信バッファを処理する場合、以下に示すRX_BYTEレジスタの3つのビットが、読み出されるバイト(RX_RD_PTRでアドレス指定されたバイト)に関する特定の情報を示すため、エラー・チェックの際に役に立ちます。

- RX_FIRST_BYTEビット: バイトがメッセージの最初のデータ・バイトであることを示します。対応するキャラクタの前にはプリアンブル・キャラクタが置かれます。
- RX_BYTE_ERRORビット: バイトにエラーが含まれている可能性があることを示します。対応するキャラクタには、マンチェスターやパリティのエラーが含まれます。
- RX_LAST_BYTEビット: バイトがメッセージの最後のデータ・バイトであることを示します。対応するキャラクタはストップ・キャラクタであり、ヌル・バイトとして格納されます。

メッセージの例外

有効なプリアンブルのあるメッセージを受信しなかった場合、UARTはそのデータを無視し、格納はしません。

メッセージがストップ・キャラクタで終了していない場合は、次のメッセージのプリアンブル(または受信データのタイムアウト)がこれら2つのメッセージの境界を示し、その結果、最初のメッセージが完了します。

受信データのタイムアウトが生じるのは、最後のデータ・バイトの2つのUARTフレーム長の範囲内でLSSMがデータを受信しなかった場合です。その結果、LSSMのステータス・バイトが書き込まれ、RX_READYビットがアサートされ、メッセージが終了します。

UARTが、プリアンブルと、それに続いてストップ・キャラクタを受信した場合は、1つのヌル・バイト(00h)で構成される受信バッファにヌル・メッセージを格納します。これが生じるのは、プリアンブル送信モードが無効化された後にキープアライブ・ストップ・キャラクタを受信した場合です。この場合、受信バッファは空にはなりません。ホストは、受信バッファをクリアするか、ヌル・メッセージを読み出して破棄することで、ヌル・メッセージを分配する必要があります。

通信タイムアウト・エラー(COMM_TO_ERR)は、TXポートがプリアンブルを送信してからRXポートが受信するまでの時間として定義されます。

受信バッファ・オーバーフローは、UARTがデータを受信しながらもそれを格納する容量がない場合に発生します。これが生じる可能性があるのは、TX_UNLIMITEDがセットされている場合か、デジタイズチェーンに十分なレイテンシがある場合です。UARTは、読出しポインタに先行することや、読出されるデータを上書きすることはできません。そのため、書き込むことのできる最後のアドレスは、読出しポインタに続くアドレスです。最後のアドレスを書き込んだ後に更に多くのデータを受信した場合、UARTは最後のアドレスを上書きし、RX_OVERFLOWビットをセットします。このビットは、受信バッファを読み出すとクリアされます。これにより、より多くの書き込みスペースが生じます。オーバーフローを検出するには、受信バッファを処理する前にステータスをチェックする必要があります。その後、再度ステータスをチェックしてデータ・エラー(パリティ・エラーなど)がないことを確認してから、次のメッセージの送信を開始します。

読出しを行わずに複数のメッセージを受信する場合、オーバーフローが発生する可能性があり、UARTはRX_OVERFLOWビットをセットします。これは、書き込みポインタが読出しポインタより1つ少なくなるまでインクリメントされた場合に生じます。この時

点でUARTはインクリメントを停止します。この場合、最後のデータ・バイトは上書きされます。

メッセージの読出し

ホストは、以下に示す2つの異なるSPIトランザクションを用いて受信(RX)バッファを読み出すことができます。

- RX_RD_NXT_MSG:RX_RD_PTRの場所(次の未読メッセージ)で読出しを開始します。
- RX_RD_MSG:RX_RD_PTRの場所で読出しを開始します。

どの読出しトランザクションの間も、ホストは、メッセージの最後までデータの読出しを続けることができ、その後、データ読出しが0x00になります。次のメッセージがある場合でも、ホストはそこにいたるまで読出しを続けることはできません。

RX_RD_NXT_MSGトランザクションまたはRX_RD_MSGトランザクションの間、デバイスは、関連するポインタを自動的にインクリメントします。そのため、複数のSPIトランザクションを用いて、与えられたメッセージを読み出すことができます。例えば、RX_RD_NXT_MSGを用いてメッセージをリードバックし、メッセージの末尾の前にトランザクションが終了する場合です。その場合、後続のRD_RX_MSGを用いてそれと同じメッセージのリードバックを再開できます。これにより、ホストは読出しを停止し、データを失うことなくリードバックを再開できます。バッファの1バイトが読み出された後、マイクロコントローラ・ユニット(MCU)がこれを使用できるようになり、新たな受信データの格納に備えます。

注:SWAP_ENビット・セットがREADALLのコマンド・タイプである場合、MAX17851は、DEVCOUNTレジスタを用いて自動的にメッセージ・データを並べ替えます。そのような状況では、受信した元のPECは、メッセージ・データの並び替えに対応するよう、再計算されたパケット・エラー・チェック(PEC)に置き換えられます。

注:バイトの一部分のみへのSPIアクセスはサポートされておらず、これを行うと予期せぬ動作を招く可能性があります。ただし、これがRD_RX_NXT_MSGまたはRD_RX_MSGの間に生じた場合は、MAX17851は、データ喪失を防ぐために、部分的に読み出されたバイトのポインタを復元しようとします。これによって、メッセージのリードバックを後続のRD_RX_MSG SPIトランザクションを用いて再開できます。

アラート・パケット・バッファ

ALERTPACKETコマンドは、受信UARTデータを基に解析され、2つ(メインおよび冗長性確保用)の6バイトのALERTPACKETバッファに格納されます。これらのバッファは、BMSデジタイゼーション・モジュールのフォルト位置およびBMSデジタイゼーションのSTATUS1レジスタの通知を格納します。ALERTPACKETデータは、メインのALERTPACKETバッファからのみ読み出せます。

ALERTPACKETデータ内にPECエラーがあると、データが破棄されALRTPCKTBUFには格納されません。これにより、ノイズの多い環境でホスト・マイクロコントローラが頻繁に中断されるのが防止されます。ALERTPACKETがCOMM_RTRYより多数のインスタンスに対し失敗する場合は、ALRTPCKT_PEC_ERRがアサートされます。このインスタンスでは、ホストがREADALLを実行して、デジタイゼーション内の全デバイスのステータスをクエリすることを推奨します。

データの完全性を確保するために、新しいブリアンブルまたはALERTPACKETストップ・キャラクタの受信後に、メインと冗長性確保用のALERTPACKETバッファが比較されます。2つのバッファ間でALERTPACKETデータの完全性が保たれない場合、ALRTPCKTBUF_HW_ERRステータス・ビットがアサートされます。ALRTPCKTBUF_HW_ERRビットがアサートされた場合、ALERTPACKETのデータは無視する必要があり、また、各デジタイゼーション・デバイスのSTATUS1レジスタを、ユーザ指示によるREADDEVICEおよびREADALLによってモニタする必要があります。

PECエラーのない新たなALERTPACKETを受信すると、ALRTPCKTBUF_FULLステータス・ビットがセットされます。ALRTPCKTBUFのバイトは、バイトの読出し後にゼロにクリアされます。いずれかのバイトの読出し／クリアが行われた場合、新しいALERTPACKETは受信されず、ALRTPCKTBUF_FULLレジスタはクリアされます。BMS安全モニタリング・モードおよびスリープ・モードの場合、ALRTPCKTBUFは自動的にクリアされ、ALRTPCKTBUF_FULLが自動的にトリグリングします。

コマンド動作モードの間、ALERTPACKET_STATUSビットは、STATUS_ERR_MASKレジスタによってマスクされ、全ビットに論理OR操作が行われ、ALRTPCKT_STATUS_ERRステータス・ビットにレポートされます。クリアされない可能性のあるPECが永続的にレポートされるのを防止するため、ALRTPACKET_STATUS[5]をマスクすることを推奨します。デフォルトでは、ALRTPACKET_STATUS[5](BMSデジタイゼーションからのALRTPPEC)は、STATUS_ERR_MASKおよびSTATUS_DBNC_MASKによりマスクされています。

MAX17851のその他のステータス・ビットと同様、ALRTPCKT_STATUS_ERR_ALRTビットは、ALRTPCKT_STATUS_ERR_ALRTENビットを用いて無効化できます。このビットを無効化することで、ユーザ指示以外の通信がALERTピンを通じてマイクロコントローラのタスクを中断するのを防止できます。ALERTPACKETステータスは、引き続き、ALERTPACKETバッファの断続的なポーリング、または、ALRTPCKT_STATUS_ERRステータス・ビットを通じて検証できます。

ALRTPCKT_STATUS_ERRビットは、LSSMステータス・バイトに提示されます。

BMS安全モニタリング・モードまたはスリープ・モードの間、ALERTPACKET_STATUSビットは、STATUS_DBNC_MASKレ

ジスタでマスク／マスク解除できます。マスク解除されたSTATUSビットは、BMS安全モニタリングまたはスリープのロジックによって、全ビットに対し論理OR操作が行われバウンス防止がなされます。BMS安全モニタリング・モードまたはスリープ・モードの間、マスク解除されたSTATUSビットがALERTPACKETデータでアサートされ、対応するモードのDBNCLレジスタより長い時間続く場合は、FAULT_TIMERが始動します。

新たなALERTPACKETデータを受信すると、ALRTPCKTBUFはそれ以前の値を常に上書きします。

それ以外の全てのトランザクションと同様、ALERTPACKETデータはLSSMによってモニタされ、オーバーフロー、アンダーフロー、PEC、タイムアウトなどのデータ完全性問題がないか確認されます。

自動化されたALERTPACKETデータがALERTPACKETバッファに格納された場合、そのデータは、MAX17851からホスト・マイクロコントローラへのPEC保護は行われません。ただし、TXポート送信からRXポートで受信された場合にはPEC保護が行われ、データの完全性が確保されます。ユーザが、ALERTPACKETをTXバッファへのコマンドに書き込むことでこれを発行した場合は、通常のUARTデータとして処理され、RXバッファに格納されます。MAX17851からホスト・マイクロコントローラへはPEC保護されます。ALERTPACKETバッファにエラーが検出された場合、ユーザは、ALERTPACKETを発行して、マイクロコントローラへの正しい通信を確保する必要があります。

自動およびユーザ指定アライブ・カウンタ

自動アライブ・カウンタは、UARTトランザクションの最後にフレーム・カウンタを明示的に付加することにより、通信の安全性を更に高めます。これによって、各トランザクションを1対1で比較でき、パケットの挿入、削除、並び替え、非更新がないことを確認できます。

ALIVECOUNT_EN = 0x11の場合、MAX17851の自動アライブ・カウンタは、出力される読み出しおよび書き込みトランザクションの末尾に自動的にローリング・カウンタを付加します。最初に送信されるトランザクションは、デフォルトのALIVECOUNT_SEEDである0x00を使用します。これに対し、後続の全てのトランザクションは、このシード値を1ずつインクリメントします。ALIVECOUNT_SEEDが0xFFに達すると、0x00にロール・オーバーして続行します。

送信されたALIVECOUNT_SEEDは、デジチェーンを通じて伝搬し、コマンドでアドレス指定されたユニットごとにインクリメントされます。返されたアライブ・カウント値はALIVECOUNT_RETレジスタに格納され、あらゆるトランザクションについて予測値と比較されます。ALIVECOUNT_RETが予測値と一致しない場合、ALIVECOUNT_ERRビットがアサートされます。ALIVECOUNT_RETの予測値は次のように計算されます。

$$\text{Expected ALIVECOUNT_RET} = \text{ALIVECOUNT_SEED} + \text{DEV_COUNT (WRITEALL/READALL)}$$

$$\text{Expected ALIVECOUNT_RET} = \text{ALIVECOUNT_SEED} + 1 \text{ (WRITEDEVICE/READDEVICE/READBLOCK)}$$

注: ALIVECOUNT_ERRが生じた場合、DEV_COUNTレジスタの内容の正確さを検証します。有効化エラーが検出された場合、ALIVECOUNT_Qにクエリを行い、0x0の値が読み出されるまで続けます。トランザクションを再度整えるには、RX、TX、LSSMの各バッファをクリアします（それぞれ、CLR_RXBUF、CLR_TXBUF、CLR_LSSM）。

自動アライブ・カウンタが有効化されている場合、返されるALIVECOUNTデータは、ALIVECOUNT_RETとして内部で格納されるため、RXバッファには格納されません。全てのトランザクションはLSSMで検証されます。

長いデジチェーンを使用するシステム・アーキテクチャでは、所定のトランザクションが受信されるまでに複数のトランザクションが送信される可能性があります。これらの伝送中のトランザクションの数は、ALIVECOUNT_Qレジスタに格納されます。トランザクションの検証に成功するまでに新たなコマンドが送信されないようなコマンド／応答トランザクションが行われている場合は、通常、ALIVECOUNT_Qレジスタからは0x0が読み出されます。

注: ALIVECOUNT_Qは、RXバッファがまだ受信していないLSSMバッファ内のメッセージ数を提供します。

ALIVECOUNT_Qレジスタがゼロ以外の値の場合にCLR_ALIVECOUNT_SEEDコマンドが発行された場合は、このコマンドは無視され、ALIVECOUNT_ERRステータス・ビットがセットされます。

自動アライブ・カウンタを有効化すると、全ての動作モード（コマンド動作、安全モニタリング、スリープ）でアクティブになります。

ALIVECOUNT_EN = 0x10の場合、MAX17851は、ユーザ指定のアライブ・カウンタ・モードになります。このモードでは、ホストは、SPIを介してUARTの送信データをMAX17851に書き込む場合に、サポートするトランザクションの最後にアライブ・カウントを手動で付加する必要があります。返されたアライブ・カウント・バイトは、このモードのRXバッファに格納され、エラーが生じてもALIVECOUNT_ERRはアサートされません。

2つのモジュール、1つのUARTデジチェーンでの、READALLコマンドの伝搬の例を以下に示します。

表14. ユーザ指定アライブ・カウンタを伴うコマンドの伝搬 (ALIVECOUNT_EN = 10)

TX BUFFER		RECEIVED UART PACKET		RX BUFFER	
Message Length	0x09	Preamble		Command Byte	0x03
Command Byte	0x03	Command Byte	0x03	Register Address	0x12
Register Address	0x12	Register Address	0x12	LSB Data Device Address 1	0xB1
Data Check	0x00	LSB Data Device Address 1	0xB1	MSB Data Device Address 1	0xB2
PEC	0xCB	MSB Data Device Address 1	0xB2	LSB Data Device Address 0	0xB1
Alive Counter	0x00	LSB Data Device Address 0	0xB1	MSB Data Device Address 0	0xB2
		MSB Data Device Address 0	0xB2	Data Check	0x00
		Data Check	0x00	Alive Counter	0x02
		PEC	0x67	LSSM Status Byte	0x00
		Alive Counter	0x02	PEC	0x3E
		Stop			

上記の例 (ALIVECOUNT_EN = 10) では、ユーザはアライブ・カウンタ・バイトをTXバッファに付加し、デジチェーンは0x02の値を返してデバイスの数を通知します。

表15. 自動アライブ・カウンタを伴うコマンドの伝搬 (ALIVECOUNT_EN = 11)

TX BUFFER		RECEIVED UART PACKET		RX BUFFER	
Message Length	0x08	Preamble		Command Byte	0x03
Command Byte	0x03	Command Byte	0x03	Register Address	0x12
Register Address	0x12	Register Address	0x12	LSB Data Device Address 1	0xB1
Data Check	0x00	LSB Data Device Address 1	0xB1	MSB Data Device Address 1	0xB2
PEC	0xCB	MSB Data Device Address 1	0xB2	LSB Data Device Address 0	0xB1
		LSB Data Device Address 0	0xB1	MSB Data Device Address 0	0xB2
		MSB Data Device Address 0	0xB2	Data Check	0x00
		Data Check	0x00	LSSM Status Byte	0x00
		PEC	0x67	PEC	0x7D
		Alive Counter Return (TX Count = 5, ALIVECOUNT_SEED = 4)	0x06		
		Stop			

上記の例 (ALIVECOUNT_EN = 11) では、自動アライブ・カウンタが有効になり、シーケンスの5番目のトランザクションになっています (ALIVECOUNT_SEED = 4)。

LSSMステータス・バイトが0のRX_READY値をリードバックした場合、RXバッファの読み出しデータは、正しく終了されずに読み出されます。この状況では、RXバッファ・データは無効と見なされ、メッセージは再送信する必要があります。

注: TX_NOSTOPが有効化されている場合、RX_STOPステータス・ビットを用いてトランザクションの完了を示すことはできません。RX_READYステータス・ビットは、全てのトランザクション完了ユース・ケースに対し有効です。

データチェック・パーサ

データチェック・バイトは、DC_ENレジスタを介して有効化されている場合、ホストが生成する必要があります。MAX17851は、受信するデータチェック・バイトを自動的に解析し、返されたデジチェーンDC_PECERRをCOMM_ERRステータス・ビットを介してレポートします。これ以外のデータチェック・バイトからのビットはレポートされません。その他の全てのデータチェック・バイト・エラーは、ALRTPCKTBUFデータでレポートされます。データチェック・バイトが有効化されている場合、

ユーザには、RXバッファに格納するかしないかのオプションがあります(10b=データチェックを有効化しRXバッファに格納、11b=データチェックを有効化しながらもRXバッファには格納しない)。

ロックステップ安全対策の検証

UART通信は、送信されたUARTパケットを受信されたUARTパケットと比較することで、完全性を検証します。この検証は、サポートされるコマンド・バイトと予約済みのコマンド・バイトに対する全てのUARTトランザクションについて行われます(表10を参照)。

ロックステップ安全対策検証エンジンには、データとメッセージの両方のプロパティを解析してデータ・ペイロード内に破損があるかどうかを判定する、2つの個別コアがあります。これらのフォルトは、LSSMステータス・バイトとして格納され、受信される各通信メッセージに挿入されます。LSSMステータス・バイトは、RXバッファ・ペイロードの末尾かつPECバイトの前に付加されます。PECは、LSSMステータス・バイトを含みかつ保護するよう、MAX17851によって再計算されます。これにより、RXペイロード全体のフォルトに対する保護が確実に行われます。LSSMステータス・バイトは、オプションで、STATUS_LSSM_BYTEレジスタを通じて読み出し、関連するALRTENレジスタ・ビットがセットされている場合には、ALERTピンを通じてアサートできます。

次に示す表は、LSSMステータス・バイトの各ビットに対し表されるエラーのリストです。

表16. LSSMステータス・バイトのエラー・マッピング

ビット	LSSMステータス・レジスタ・ビット	レジスタ・ビットに関連するエラー	説明
7	RX_READY	N/A	UARTは、正しくフレーム化されたメッセージ(ストップ・キャラクタ、プリアンブル、またはフレーム・タイムアウト)の受信を終了し、読み出し可能な状態になっています。
6	ALRTPCKT_STATUS_ERR	Alert Packet Buffer Status Error	自動化ALRTPACKETでの全ての非マスクSTATUSビットの論理OR。
5	COMM_ERR	Communication Timeout Error	COMM_TO_DLYの時間が経過するまでに送信コマンドが受信されません。
		Command Byte Invalid	コマンド・バイト無効。定義されたUARTコマンド以外のコマンドは全て無効です。
		PEC Received Error	PEC RXエラー:デコードしたPECが受信したPECと一致しません。このエラーには、自動生成ALRTPACKETコマンドによって生成されたPEC RXエラーは含まれません。
		Data Check Error	データチェック・パーサ(有効化している場合)によって示されるPECエラー。
		Register Address Invalid	レジスタ・アドレスのバイトが無効。
4	ALRTPCKT_ERR	Alert Packet Communications Error	自動化ALRTPACKETに対する応答がタイムアウトしたか、受信した自動化ALRTPACKETのPECが正しくありません(ALRTPCKT_COMM_ERR)。
		Alert Packet Hardware Error	冗長アラート・パケット・バッファとメイン・アラート・パケット・バッファが一致しません(ALRTPCKT_HW_ERR)。
3	COMM_MSMTCH_ERR	Command Byte Mismatch	受信したコマンド・バイトが送信されたコマンドと異なっています。
		Register Address Mismatch	受信トランザクションと送信トランザクションのレジスタ・アドレス・バイトが一致しません。
		Write Message Mismatch	受信トランザクションと送信トランザクションのWRITE_ALLメッセージおよびWRITE_DEVICEメッセージが一致しません。
		LSSM Hardware Error	冗長LSSM/バッファとメインLSSM/バッファが一致しません。
		Message Length Mismatch	受信トランザクションと送信トランザクションの間のメッセージ長ミスマッチ・エラー。
		LSSM Over/Underflow	LSSMバッファがオーバーフローまたはアンダーフローしています。

表16. LSSMステータス・バイトのエラー・マッピング (続き)

ビット	LSSMステータス・レジスタ・ビット	レジスタ・ビットに関連するエラー	説明
2	COMMAND_OP	N/A	このビットは、MAX17851が通常動作している場合にセットされます。
1	ALIVECOUNT_ERR	Incorrect Alive Count	返されたアライブ・カウントがALIVECOUNT_SEED - ALIVECOUNT_Q + DEV_COUNTに等しくありません (WRITEALL/READALL) 返されたアライブ・カウントがALIVECOUNT_SEED - ALIVECOUNT_Q + 1に等しくありません (WRITEDEVICE/READDEVICE/READBLOCK)
0	HW_ERR	Oscillator HW Error	低周波発振器または高周波発振器が範囲外になっているか、動作しません。
		VDDL1 HW Error	VDDL1が低電圧状態または過電圧状態になっています。
		VDDL2 HW Error	VDDL2が低電圧状態または過電圧状態になっています。
		Register HW Error	内部レジスタがテスト・チェックに失敗しました。
		OTP HW Error	ワンタイム・プログラマブル・メモリがテスト・チェックに失敗しました。
		VDCIN1 HW Error	VDCIN1が低電圧状態または過電圧状態になっています。
		VDCIN2 HW Error	VDCIN2が低電圧状態または過電圧状態になっています。

LSSM_UNDRFLW_ERRビットがアサートされるのは、受信トランザクション数が予想トランザクションの数を上回った場合です (オーバーフロー)。

オーバーフロー状態を避けるために、LSSMは、LSSMが満杯 (LSSM_FULLが真) の場合に送信バッファによるデータ送信が行われないようにします。

注: LSSMアンダーフロー状態が生じた場合、ユーザは、CLR_LSSMビットでLSSMを再初期化する必要があります。この時間内に実行中のトランザクションがある場合 (ALIVECOUNT_Qが0でない場合)、この状態が継続することがあるため、動作を繰り返さなくてはならない可能性があります。

フォルトやメッセージの例外の処理の詳細については、[フォルト処理のガイドライン](#)のセクションを参照してください。

デュアル・ロックステップ・データバス処理

ロックステップ安全対策検証エンジン (LSSM) は、MAX17851が送受信する全てのUARTトラフィックを検証します。

送信バッファからのメッセージは、LSSM (1) によってチェックされ格納されます。信号に混入するノイズを検出するために追加される短い遅延 (2) の後、冗長LSSMコア (3) もメッセージのチェックと格納を行います。2つのLSSMコアを相互に比較し、MAX17851内のハードウェア・エラーを検出します。送信メッセージに、有効なコマンド、有効な長さ、有効なデバイス・カウント、正しいPECがあることをチェックします。

受信メッセージを同様に処理し、2つのLSSMコアによる冗長チェックを行います。受信メッセージを、出力される送信メッセージに対してチェックし、有効なトランザクションが行われたことを確認します。不適切に終了されたメッセージは検出されフラグ通知されます。

デジチェーン・デバイスが予期せぬメッセージや予期せぬデータを混入した場合、LSSM内に一致する出力メッセージがないため、LSSMはエラーを検出しフラグ通知します。同様に、デジチェーンで削除されたり失われたりしたメッセージもLSSMによって検出されフラグ通知されます。

アラート・パケット・バッファ(5)および冗長アラート・パケット・バッファ(4)は、アラート・パケットの処理と格納を行い、これを用いてホスト・マイクロコントローラが直接読み出すことができます。

図19にMAX17851の送受信データ処理を図示します。

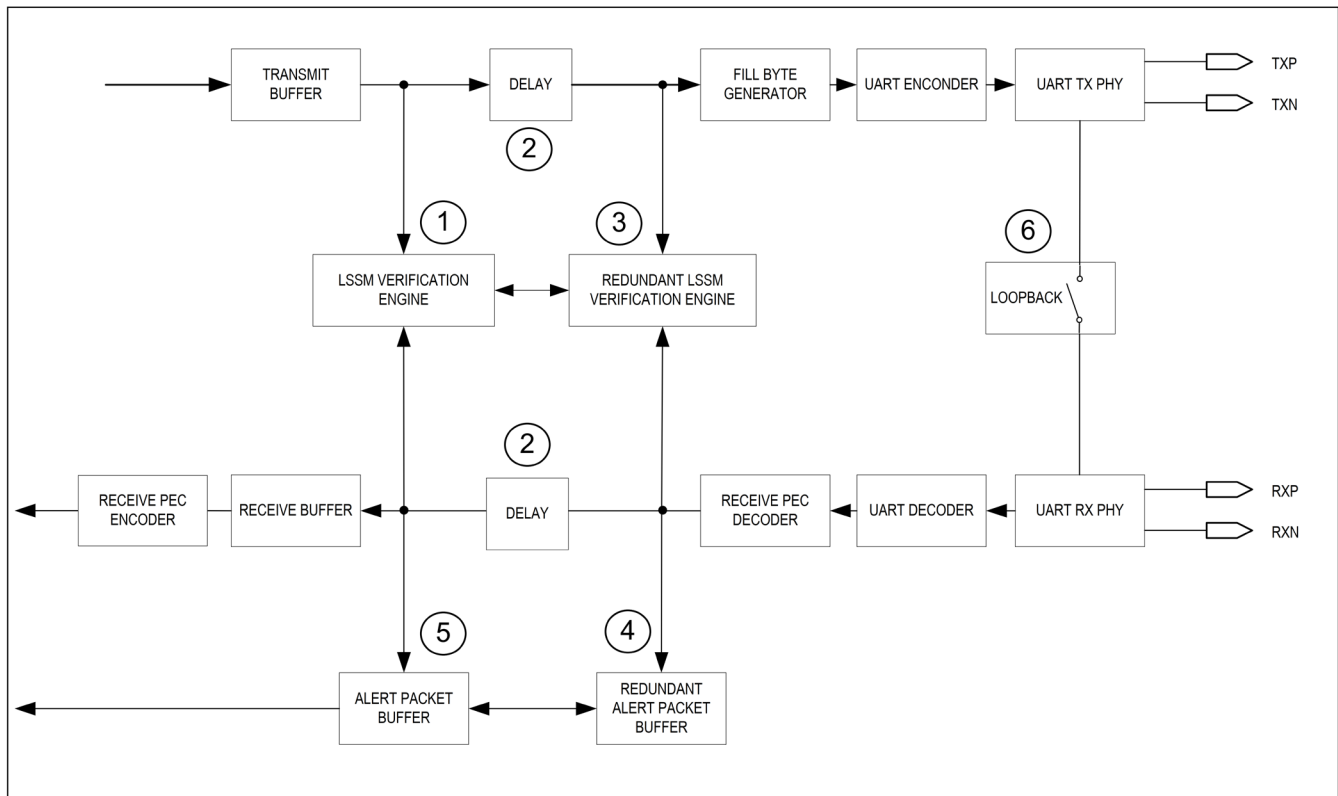


図19. LSSMのデータパス

アラートおよびフォルト

ALERTピン、ステータス、アラート・フラグの動作

ALERT $\bar{P}$ ピン(アクティブ・ロー)は、ALERTレジスタ空間にあるアラート・フラグ・レジスタのいずれかがアサートされた場合にアサートされ、イベントが発生したことをホストに通知します。イベントとは、MAX17851のフォルト、デジチェーンのフォルト、通常のUARTトランザクション・イベントを指します。詳細については、[レジスタの表](#)を参照してください。

デジチェーンに1つのトランザクションのみが送られ直ちに読み出されるようなコマンド/応答プロトコルを使用するアプリケーションには、イベントとトランザクションの直接的な相関が与えられます。

MAX17851を通じて複数のトランザクションをパイプライン伝送する(複数のトランザクションがデジチェーン内で同時に処理される)アプリケーションでは、ALERTピンを使用してもアラート・フラグの原因となったトランザクションを特定することはできません。このような状況では、ホストはバッファをフラッシュして処理中の全てのデータを再送信する必要があります。

ステータス・レジスタ・ビットがセットされる原因となるイベントは複数あります(詳細については、[レジスタの表](#)を参照)。イベントごとに、ステータス・ビット、アラート・イネーブル・ビット、アラート・フラグ・ビットがあります。ステータス・ビットは、イベントのリアルタイムのステータスであり、自律的にセットやクリアが行われます。ステータス・ビットに対応するアラート・イネーブル・ビットは、イベントによってアラート・フラグがセットされそれに伴ってALERTピンがアサートされるかどうかを決定します。

アラート・イネーブル・ビットのプログラミングに関わらず、全てのアラート・フラグ・ビットのアサーションは、ALRTRSTによってゲート処理されます。そのため、ホスト・マイクロコントローラは、いずれかのPORイベントまたはSWPORイベント後にALRTRSTビットをクリアする必要があります。これによってALERTピンがデアサートされ、アラート・フラグのアサーションが可能になります。ALRTRSTビットには、対応するアラート・イネーブル・ビットがなく、これを無効化することはできません。

アラート・フラグ・ビットのアサーションは、エッジトリガです（つまり、アラート・フラグがセットされるのは、アラート・イネーブル・ビットが真であり、対応するステータス・ビットがロジック0状態からロジック1状態に遷移した場合です）。1つの例外として、アラート・フラグ・ビット(SAFEMON_GPIO12_ALERT)のみは、ホストによってクリアできます。アラート・フラグが有効化されている場合でも、対応するステータス・ビットがロジック0状態からロジック1状態に遷移するまでは、アラート・フラグはセットされません。対応するステータス・ビットが真の場合にアラート・フラグ・ビットがクリアされると、ステータス・ビットがロジック0状態からロジック1状態に遷移するまで、アラート・フラグ・ビットは再度セットされることはありません。いずれかのアラート・フラグ・ビットがセットされると、MAX17851はALERTピンをアサートします。ALERTピンをデアサートするには、全てのアラート・フラグ・ビットがクリアされていなくてはなりません。

アラート・フラグ・ビットはイベントによって駆動されるか、永続的であるかのどちらかです。永続的アラート・フラグは、対応するステータス・ビットが真のままである場合、クリアできません。イベント駆動アラート・フラグはいつでもクリアできます。詳細については、[レジスタの表](#)を参照してください。

注: LSSMバイトには、トランザクションに関連した特定のデータが含まれています。したがって、アプリケーションが複数のトランザクション、あるいはパイプライン処理されたトランザクションをデジイチチェーンに対して使用する場合、ALERTピンを用いてLSSM_BYTE通知を行うことは推奨しません。

フォルト・タイマー

フォルト・タイマーは、通信フォルトあるいはデジイチチェーン・フォルト(ALERTPACKETにより示される)がスリープ・モードまたはBMS安全モニタリング・モードで発生してから経過時間を、秒単位で示します。

フォルト・タイマーは、フォルトがバウンス防止されると開始し、SLP_ALERTビットまたはSAFEMON_ALERTビットがクリアされると停止します。

フォルト・タイマーは、スリープ・モードまたはBMS安全モニタリング・モードの開始時にリセットされます。

フォルト・タイマーは読み出し専用で長さは2バイトです。FAULT_TIMER0レジスタは下位バイトを保持し、FAULT_TIMER1は上位バイトを保持します。

ハードウェア・フォルトの検出

MAX17851は、デバイス動作に影響する可能性のあるハードウェア・フォルトがないか、継続的に自己モニタしています。フォルトには、仕様規定された動作範囲を超える電源電圧、通信のタイミングに影響する可能性のある発振器のドリフト誤差、破損したデバイス・トリム、破損したメモリ(スタックまたはトランジェント)、内部レジスタ・エラーなどがあります。検出されたフォルトは、HW_ERRステータス・ビットでレポートされます。

レジスタ・フォルトの検出

MAX17851は、CONFIGレジスタおよびALRTENレジスタに対し連続的にフォルト検出を行うことを特長としています。これによって、意図しないデバイス動作の原因となる可能性があるトランジェント・フォルトやハードウェア・エラーから保護します。

SPI入力エラー検出は、SPI入力データを書込み後のレジスタ・データと比較することで実行できます。この比較は、CONFIGレジスタ・ブロックまたはALRTENレジスタ・ブロックへの書込みごとに実行されます。入力SPIデータとレジスタ・データに不一致があると、HW_ERRステータス・ビットがセットされます。

CONFIGレジスタおよびALRTENレジスタにおける補助的なエラー検出では、レジスタの現在の状態と初めにプログラムされた状態との差を検出します。したがって、このビットはSPI通信が行われていない場合でも、常にアサートできます。フォルトが発生すると、HW_ERRステータス・ビットがセットされます。

MAX17851ユーザ・レジスタ・マップ

READ ADDRESS	WRITE ADDRESS	NAME	MSB							LSB
STATUS Registers										
0x01	0x00	STATUS_RX[7:0]	RX_ER R	-	RX_BU SY	RX_ID LE	RX_OV RFLW_ ERR	RX_FU LL	RX_ST OP	RX_E MPTY
0x03	0x02	STATUS_TX[7:0]	TX_HE LD	-	TX_BU SY	TX_IDL E	TX_OV RFLW_ ERR	TX_FU LL	TX_AV AIL	TX_EM PTY

READ ADDRESS	WRITE ADDRESS	NAME	MSB							LSB
0x05	0x04	STATUS_LSSM_BYTE[7:0]	RX_RE ADY	ALRTP CKT_S TATUS _ERR	COMM _ERR	ALRTP CKT_E RR	COMM MSM TCH_E RR	COMM AND_ OP	ALIVE COUN T_ERR	HW_E RR
0x07	0x06	STATUS_GEN[7:0]	HFOS C_HW _ERRB	DEV_C OUNT_ ERR	WD_E RR	GPIO_ ERR	DATAP ATH_E RR	–	ALRTP CKT_C OMM_ ERR	ALRTP CKTBU F_HW_ ERR
0x09	0x08	STATUS_OPSTATE[7:0]	–	–	–	–	SAFE MON	–	–	–
0x0B	0x0A	STATUS_BUF[7:0]	ALRTP CKTBU F_FUL L	–	–	–	LSSM_ FULL	–	–	–
0x0D	0x0C	STATUS_WD[7:0]	–	–	–	WD_T O_ER R	WD_O PEN	WD_LF SR_ER R	WD_R JCT_E RR	WD_E XP_ER R
0x0F	0x0E	STATUS_GPIO[7:0]	GPIO4 _RD	GPIO3 _RD	GPIO2 _RD	GPIO1 _RD	GPIO4 _ERR	GPIO3 _ERR	GPIO2 _ERR	GPIO1 _ERR
ALERT Registers										
0x11	0x10	ALERT_RX[7:0]	RX_ER R_ALR T	–	RX_BU SY_AL RT	RX_ID LE_AL RT	RX_OV RFLW_ ERR_A LRT	RX_FU LL_AL RT	RX_ST OP_AL RT	RX_E MPTY_ ALRT
0x13	0x12	ALERT_TX[7:0]	TX_HE LD_AL RT	–	TX_BU SY_AL RT	TX_IDL E_ALR T	TX_OV RFLW_ ERR_A LRT	TX_FU LL_AL RT	TX_AV AIL_AL RT	TX_EM PTY_A LRT
0x15	0x14	ALERT_LSSM_BYTE[7:0]	RX_RE ADY_A LRT	ALRTP CKT_S TATUS _ERR_ ALRT	COMM _ERR_ ALRT	ALRTP CKT_E RR_AL RT	COMM MSM TCH_E RR_AL RT	COMM AND_ OP_AL RT	ALIVE COUN T_ERR_ ALRT	HW_E RR_AL RT
0x17	0x16	ALERT_GEN[7:0]	–	DEV_C OUNT_ ERR_A LRT	WD_E RR_AL RT	GPIO_ ERR_A LRT	DATAP ATH_E RR_AL RT	SPI_E RR_AL RT	ALRTP CKT_C OMM_ ERR_A LRT	ALRTP CKTBU F_HW_ ERR_A LRT
0x19	0x18	ALERT_OPSTATE[7:0]	ALRTR ST	–	SLP_A LRT	SLP_S TATUS _ERR_ ALRT	SAFE MON_ ALRT	SAFE MON_ GPIO1 2_ALR T	SAFE MON_ STATU S_ERR_ ALRT	SAFE MON_ CONFI G_ER R_ALR T
0x1B	0x1A	ALERT_BUF[7:0]	ALRTP CKTBU F_FUL L_ALR T	–	–	–	LSSM_ FULL_ ALRT	–	–	–
0x1D	0x1C	ALERT_WD[7:0]	–	–	–	WD_T O_ER R_ALR T	WD_O PEN_A LRT	WD_LF SR_ER R_ALR T	WD_R JCT_E RR_AL RT	WD_E XP_ER R_ALR T

READ ADDRESS	WRITE ADDRESS	NAME	MSB							LSB
0x1F	0x1E	ALERT_GPIO[7:0]	–	–	–	–	GPIO4_ERR_ALRT	GPIO3_ERR_ALRT	GPIO2_ERR_ALRT	GPIO1_ERR_ALRT
ALRTEN Registers										
0x21	0x20	ALRTEN_RX[7:0]	RX_ERR_ALRTEN	–	RX_BUFFER_ALRTEN	RX_IDLE_ALRTEN	RX_OVERFLOW_ERR_ALRTEN	RX_FULL_ALRTEN	RX_STOP_ALRTEN	RX_EMPTY_ALRTEN
0x23	0x22	ALRTEN_TX[7:0]	TX_HELD_ALRTEN	–	TX_BUFFER_ALRTEN	TX_IDLE_ALRTEN	TX_OVERFLOW_ERR_ALRTEN	TX_FULL_ALRTEN	TX_AVAILABLE_ALRTEN	TX_EMPTY_ALRTEN
0x25	0x24	ALRTEN_LSSM_BYTE[7:0]	RX_READY_ALRTEN	ALRTP_CKT_STATUS_ERR_ALRTEN	COMM_ERR_ALRTEN	ALRTP_CKT_ERR_ALRTEN	COMM_MSM_TCH_ERR_ALRTEN	COMM_AND_OP_ALRTEN	ALIVE_COUNT_ERR_ALRTEN	HW_ERR_ALRTEN
0x27	0x26	ALRTEN_GEN[7:0]	–	DEV_COUNT_ERR_ALRTEN	WD_ERR_ALRTEN	GPIO_ERR_ALRTEN	DATAPATH_ERR_ALRTEN	SPI_ERR_ALRTEN	ALRTP_CKT_COMM_ERR_ALRTEN	ALRTP_CKTBUF_HW_ERR_ALRTEN
0x29	0x28	ALRTEN_OPSTATE[7:0]	–	–	SLP_ALRTEN	SLP_STATUS_ERR_ALRTEN	SAFE_MON_ALRTEN	SAFE_MON_GPIO12_ALRTEN	SAFE_MON_STATUS_ERR_ALRTEN	SAFE_MON_CONFIG_ERR_ALRTEN
0x2B	0x2A	ALRTEN_BUF[7:0]	ALRTP_CKTBUF_FULL_ALRTEN	–	–	–	LSSM_FULL_ALRTEN	–	–	–
0x2D	0x2C	ALRTEN_WD[7:0]	–	–	–	WD_TIMER_ERR_ALRTEN	WD_OPEN_ALRTEN	WD_LFSR_ERR_ALRTEN	WD_RESET_ERR_ALRTEN	WD_EXPIR_ALRTEN
0x2F	0x2E	ALRTEN_GPIO[7:0]	–	–	–	–	GPIO4_ERR_ALRTEN	GPIO3_ERR_ALRTEN	GPIO2_ERR_ALRTEN	GPIO1_ERR_ALRTEN
COMMAND Registers										
0x41	0x40	CLR_TXBUF[7:0]	–	–	–	–	–	–	–	–
0x43	0x42	CLR_RXBUF[7:0]	–	–	–	–	–	–	–	–
0x45	0x44	CLR_LSSM[7:0]	–	–	–	–	–	–	–	–
0x49	0x48	CLR_ALIVECOUNT_SEED[7:0]	–	–	–	–	–	–	–	–
0x4B	0x4A	SWPOR[7:0]	–	–	–	–	–	–	–	SWPOR
0x4D	0x4C	SLP_EN[7:0]	–	–	–	–	–	–	–	SLP_EN

READ ADDRESS	WRITE ADDRESS	NAME	MSB							LSB
0x4F	0x4E	VER_CONFIG[7:0]	–	–	–	–	–	–	–	VER_CONFIG
0x51	0x50	LOAD_CONFIG[7:0]	–	–	–	–	–	–	–	LOAD_CONFIG
0x53	0x52	WD_KEY[7:0]	WD_KEY[7:0]							
CONFIG Registers										
0x61	0x60	CONFIG_GEN0[7:0]	–	–	DEV_COUNT[5:0]					
0x63	0x62	CONFIG_GEN1[7:0]	SINGLE_ENDED	BAUD_RATE[2:0]			–	–	–	–
0x65	0x64	CONFIG_GEN2[7:0]	RX_RAW_DATA	TX_RAW_DATA	TX_PREAMBLES	TX_QUEUE	TX_ODDPARITY	TX_PAUSE	TX_NOSTOP	TX_NOPREAMBLE
0x67	0x66	CONFIG_GEN3[7:0]	–	TX_AUTO	TX_UNLIMITED	SPI_DIRECTION	ALRTPCKT_TIMING[3:0]			
0x69	0x68	CONFIG_GEN4[7:0]	COAL RTPCKTEN	RXSWAP_EN	MS_EN[1:0]		DC_EN[1:0]		ALIVECOUNT_EN[1:0]	
0x6B	0x6A	CONFIG_GEN5[7:0]	TX_STARTSETUP	TX_HIZ	ALRTPCKT_DBNC[2:0]			SPI_SFTYSLK	SPI_SFTYSDI	SPI_SFTYCSB
0x6D	0x6C	CONFIG_SAFEMON0[7:0]	GPIOREC_DLY[7:0]							
0x6F	0x6E	CONFIG_SAFEMON1[7:0]	CONT_TIMER_DLY[7:0]							
0x71	0x70	CONFIG_SAFEMON2[7:0]	–	–	–	–	–	–	SAFEMON_SCAN_DLY[1:0]	
0x73	0x72	CONFIG_SAFEMON3[7:0]	NOMON	–	–	–	SM_GPIO4_MASK	SM_GPIO3_MASK	SM_GPIO2_MASK	SM_GPIO1_MASK
0x75	0x74	CONFIG_SLP[7:0]	–	–	SLP_SCAN_DLY[1:0]		SLP_ALRTPCKTEN	SLP_CBNTFY[2:0]		
0x77	0x76	CONFIG_COMM[7:0]	–	–	COMM_RETRY[1:0]		–	COMM_TO_DLY[2:0]		
0x79	0x78	STATUS_DBNC_MASK0[7:0]	STATUS_DBNC_MASK[7:0]							
0x7B	0x7A	STATUS_DBNC_MASK1[7:0]	STATUS_DBNC_MASK[15:8]							
0x7D	0x7C	STATUS_ERR_MASK0[7:0]	STATUS_ERR_MASK[7:0]							
0x7F	0x7E	STATUS_ERR_MASK1[7:0]	STATUS_ERR_MASK[15:8]							
0x81	0x80	CONFIG_GPIO12[7:0]	–	GPIO2_CFG[2:0]			–	GPIO1_CFG[2:0]		
0x83	0x82	CONFIG_GPIO34[7:0]	–	GPIO4_CFG[2:0]			–	GPIO3_CFG[2:0]		
0x85	0x84	CONFIG_WD0[7:0]	WD_OPN[3:0]				WD_CLO[3:0]			
0x87	0x86	CONFIG_WD1[7:0]	WD_1UD[2:0]			WD_DIV[4:0]				

READ ADDRESS	WRITE ADDRESS	NAME	MSB							LSB
0x89	0x88	CONFIG_WD2[7:0]	WD_EN	–	–	–	WD_SWW	WD_DBNC[2:0]		
RX COMMAND Registers										
0x8D	0x8C	ALRTPCKTBUF_RD_PTR[7:0]	–	–	–	–	–	ALRTPCKTBUF_RD_PTR[2:0]		
0x8F	0x8E	ALRTPCKTBUF_RD_MSG[7:0]	ALRTPCKTBUF_RD_MSG[7:0]							
0x91	0x90	RX_RD_MSG[7:0]	RX_RD_MSG[7:0]							
0x93	0x92	RX_RD_NXT_MSG[7:0]	RX_RD_NXT_MSG[7:0]							
0x95	0x94	TX_QUEUE_SEL[7:0]	–	–	–	–	TX_Q[1:0]		LD_Q[1:0]	
0x97	0x96	RX_RD_PTR[7:0]	–	RX_RD_PTR[6:0]						
0x99	0x98	RX_WR_PTR[7:0]	–	RX_WR_PTR[6:0]						
0x9B	0x9A	RX_NXT_MSG_PTR[7:0]	–	RX_NXT_MSG_PTR[6:0]						
0x9D	0x9C	RX_SPACE[7:0]	–	RX_SPACE[6:0]						
0x9F	0x9E	RX_BYTE[7:0]	–	–	–	–	–	RX_FIRST_BYTE	RX_BYTE_ER	RX_LAST_BYTE
TX COMMAND Registers										
0xB1	0xB0	NXT_LDQ[7:0]	NXT_LDQ[7:0]							
0xC1	0xC0	LDQ[7:0]	LDQ[7:0]							
0xC3	0xC2	LDQ_PTR[7:0]	–	–	–	LDQ_PTR[4:0]				
0xD1	0xD0	CONFIGQ[7:0]	CONFIGQ[7:0]							
0xD3	0xD2	CONFIG_PTR[7:0]	–	CONFIG_QUEUE_PTR[1:0]		CONFIG_BYTE_PTR[4:0]				
INFO Registers										
0xDD	0xDC	STATE[7:0]	–	–	–	–	–	STATE[2:0]		
0xDF	0xDE	COMM_RTRY_CNT[7:0]	–	–	–	COMM_RTRY_CNT[4:0]				
0xE1	0xE0	ALRTPCKT_ERR_CNT[7:0]	ALRTPCKT_ERR_CNT[7:0]							
0xE3	0xE2	WD_FAULT_CNT[7:0]	WD_FAULT_CNT[7:0]							
0xE5	0xE4	ALIVECOUNT_SEED[7:0]	ALIVECOUNT_SEED[7:0]							
0xE7	0xE6	ALIVECOUNT_RET[7:0]	ALIVECOUNT_RET[7:0]							
0xE9	0xE8	ALIVECOUNT_Q[7:0]	–	–	–	–	–	ALIVECOUNT_Q[2:0]		
0xEB	0xEA	FAULT_TIMER0[7:0]	FAULT_TIMER[7:0]							
0xED	0xEC	FAULT_TIMER1[7:0]	FAULT_TIMER[15:8]							
0xEF	0xEE	SLP_CBTIMER0[7:0]	SLP_CBTIMER[7:0]							
0xF1	0xF0	SLP_CBTIMER1[7:0]	SLP_CBTIMER[15:8]							
0xF3	0xF2	VERSION[7:0]	MODEL[3:0]				VERSION[3:0]			
0xF5	0xF4	MODEL[7:0]	MODEL[11:4]							

レジスタの詳細

STATUS_RX (R/W = 0x01/0x00)

BIT	7	6	5	4	3	2	1	0
Field	RX_ERR	–	RX_BUSY	RX_IDLE	RX_OVRFLW_ERR	RX_FULL	RX_STOP	RX_EMPTY
Reset	0x0	–	0x0	0x1	0x0	0x0	0x0	0x1
Access Type	Read Only	–	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only

ビットフィールド	ビット	説明
RX_ERR	7	受信エラー RX_RD_PTRにあるデータ・バイトにエラー (マンチェスターやパリティのエラーが含まれる、対応するキャラクタ)が含まれている可能性があります。 このビットは、バイトが読み出される場合にセットされ、バイトの受信時や書き込み時にはセットされません。
RX_BUSY	5	受信ビジー UARTはデータ受信のためビジー状態です。
RX_IDLE	4	UARTはデータ受信中ではありません。
RX_OVRFLW_ERR	3	受信バッファが満杯のため、受信バッファのRX_WR_POINTERの位置にあるデータ・バイトは上書きされます。受信バッファが満杯でなくなると (バッファが読み出されると) クリアされます。
RX_FULL	2	受信バッファの空バイトの数がゼロです。
RX_STOP	1	RX停止を受信 UARTは正しくフレーム化されたメッセージ (ストップ・キャラクタ) の受信を終了し、読出し可能な状態になっています。全ての未読メッセージが読み出された後、あるいは、UARTが新たなブリアンブル・キャラクタを検出した場合、UARTはこのビットを自動的にクリアします。 バッファが空の状態ですトップ・キャラクタを受信した場合、UARTはこのビットをセットしません。 このビットは、自動化ALERTPACKETトランザクションに対してはセットされません。
RX_EMPTY	0	受信バッファがクリアされ、未読データがありません (RX_RD_PTR = RX_WR_PTR – 1)。

STATUS_TX (R/W = 0x03/0x02)

BIT	7	6	5	4	3	2	1	0
Field	TX_HELD	–	TX_BUSY	TX_IDLE	TX_OVRFLW_ERR	TX_FULL	TX_AVAIL	TX_EMPTY
Reset	0x0	–	0x0	0x1	0x0	0x0	0x1	0x1
Access Type	Read Only	–	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only

ビットフィールド	ビット	説明
TX_HELD	7	以下の条件が真の場合にこのビットがセットされます。 受信バッファの空バイトの数 < (送信キューのメッセージ長 + 1) このステータス・ビットが真、かつ、TX_UNLIMITED = 1' bの場合、送信バッファは、RXバッファに十分な空き容量ができるまで現在のLD_Qにデータを送信しません。 TX_UNLIMITED = 1' bの場合、TX_HELDビットは決してセットされず、データはRXバッファの空き容量には無関係に送信されます。この状態でデータの完全性を確保するには、RX_OVRFLW_ERRをモニタする必要があります。
TX_BUSY	5	UARTはデータ送信のためビジー状態です。
TX_IDLE	4	UARTはデータ送信中ではありません。
TX_OVRFLW_ERR	3	次のキューに未送信データがあるため、LD_Qをインクリメントできませんでした。この状態で何らかの書き込みを行うとロード・キューを上書きします。
TX_FULL	2	ロード・キューを除いて送信バッファの全てのキューは満杯です (LD_Q = TX_Q - 1)。
TX_AVAIL	1	送信バッファの少なくとも1つのキューがロード可能です (TX_FULLが偽)。
TX_EMPTY	0	送信バッファの全てのキューがクリアされ、ロード可能です (LD_Q = TX_Q)。

STATUS_LSSM_BYTE (R/W = 0x05/0x04)

BIT	7	6	5	4	3	2	1	0
Field	RX_READY	ALRTPCKT_STATUS_ERR	COMM_ERR	ALRTPCKT_ERR	COMM_MSMTCH_ERR	COMMAND_OP	ALIVECOUNT_ERR	HW_ERR
Reset	0x0	0x0	0x0	0x0	0x0	0x1	0x0	0x0
Access Type	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only

ビットフィールド	ビット	説明
RX_READY	7	UARTは、正しくフレーム化されたメッセージ(ストップ・キャラクタ、プリアンプル、またはフレーム・タイムアウト)の受信を終了し、読出し可能な状態になっています。 全ての未読メッセージの読出し後、UARTはこのビットを自動的にクリアします。 バッファが空の状態 でストップ・キャラクタを受信した場合、UARTはこのビットをセットしません。 このビットは、自動化ALERTPACKETトランザクションに対してはセットされません。
ALRTPCKT_STATUS_ERR	6	ALERTPACKETステータス・エラー 非マスク・アラート・パケット・ステータス・ビットがアラート・パケット・バッファで受信されています。 コマンド動作モードでは、受信ALERTPACKETステータス・ビットのマスキングは、STATUS_ERR_MASKレジスタの機能の1つです。 スリープ・モードまたはBMS安全モニタリング・モードでは、受信ALERTPACKETステータス・ビットのマスキングは、STATUS_DBNC_MASKレジスタの機能の1つであり、ステータスがバウンス防止されるときに自動クリアされます。
COMM_ERR	5	通信エラー UARTメッセージ内でデータが破損、挿入、あるいは削除されたことによる通信エラーが検出されています。 注:このビットは、プリアンプルの受信時にクリアされ、エラーを含むトランザクションの完了時にセットされます。
ALRTPCKT_ERR	4	ALERTPACKETエラー このステータス・ビットは、以下のALERTPACKETステータス・エラーのいずれかでアサートされます。 ALRTPCKT_COMM_ERR ALRTPCKT_HW_ERR
COMM_MSMTCH_ERR	3	通信ミスマッチ・エラー LSSMは、受信データが送信データに一致しない通信問題を検出しています。
COMMAND_OP	2	コマンド動作モード・ステータス このビットは、コマンド動作モード(デバイスはスリープ・モードまたはBMS安全モニタリング・モードではない)を表します。このビットがロジック1でない場合、デバイスはその他の動作モードに入っています。

ビットフィールド	ビット	説明
ALIVECOUNT_ERR	1	自動アライブ・カウンタ・エラー 自動アライブ・カウンタ・モードが有効な場合、このビットは以下の2つの状況でアサートされます。 1) 受信ALIVECOUNTバイト (ALIVECOUNT_RET)が以下の予想値とは異なる $\text{ALIVECOUNT_RET} = \text{ALIVECOUNT_SEED} - \text{ALIVECOUNT_Q} + \text{DEV_COUNT (WRITEALL/READALL)}$ $\text{ALIVECOUNT_RET} = \text{ALIVECOUNT_SEED} - \text{ALIVECOUNT_Q} + 1 \text{ (WRITEDevice/READDevice/READBLOCK)}$ 2) ALIVECOUNT_Qレジスタがゼロ以外の場合にCLR_ALIVECOUNT_SEEDコマンドが実行される。 アライブ・カウンタが無効化されている場合、または手動モードに設定されている場合、このビットはセットされません。
HW_ERR	0	ハードウェア・エラー デバイス動作に影響する可能性のあるハードウェア・エラーが報告されています。 このステータス・ビットのアサートの原因となる可能性のある条件には、以下のようなものがあります。 - 仕様範囲外の電源電圧または発振器が検出される。 - トランジェント・ロジック・フォルト。

STATUS GEN (R/W = 0x07/0x06)

BIT	7	6	5	4	3	2	1	0
Field	HFOSC_HW_ERRB	DEV_COUNT_ERR	WD_ERR	GPIO_ERR	DATAPATH_ERR	–	ALRTPCKT_COMM_ERR	ALRTPCKT_BUF_HW_ERR
Reset	0x1	0x0	0x0	0x0	0x0	–	0x0	0x0
Access Type	Read Only	Read Only	Read Only	Read Only	Read Only	–	Read Only	Read Only

ビットフィールド	ビット	説明
HFOSC_HW_ERRB	7	発振器ハードウェア・エラー 高周波発振器の安定性が、低周波発振器に対して測定した場合の予定値の+/-5%以内に入っていないことを示します。このステータスは、低周波発振器の4サイクル(64kHz)ごとに更新されます。 ドリフト・アラート状態でもSPIインターフェースを引き続き機能させることはできますが、高周波発振器が停止したり速度が極端に増加／減少したりする場合は機能しません。 注:このビットはPORイベントによってのみクリアできます。 注:このステータス・ビットはアクティブ・ローであり、エラーが検出されない場合はロジック・ハイとしてリードバックされます。
DEV_COUNT_ERR	6	デバイス・カウント・エラー このエラーがアサートされるのは、HELLOALL、UPHOST、またはDOWNHOSTのコマンドから返されたデバイス・カウントが、ユーザ設定されたDEV_COUNTレジスタに一致しない場合です。 HELLOALL、UPHOST、またはDOWNHOSTが、設定されたDEV_COUNTレジスタと一致する値を返した場合、このエラーはクリアされます。
WD_ERR	5	ウォッチドッグ・エラーのまとめ このステータス・ビットは、以下のウォッチドッグ・ステータス・エラーのいずれかと同時にアサートされます。 WD_TO_ERR WD_LFSR_ERR WD_RJCT_ERR WD_EXP_ERR
GPIO_ERR	4	GPIO HWエラーのまとめ このステータス・ビットは、以下のGPIOステータス・エラーのいずれかと同時にアサートされます。 GPIO1_ERR GPIO2_ERR GPIO3_ERR GPIO4_ERR

ビットフィールド	ビット	説明
DATAPATH_ERR	3	データバス・エラー このビットがセットされるのは、設定検証動作において内部UARTハードウェア・エラーまたは設定メモリ・ハードウェア／PECエラーを検出した場合です。 設定検証動作を再初期化すると、このビットはクリアされます。
ALRTPCKT_COMM_ERR	1	アラート・パケット通信エラー このビットは、ALRTPACKET通信エラーを通知する2つの状態をまとめたものです。 状態1 = 自動生成ALRTPACKETがデコードしたPECが、COMM_RTRYレジスタで設定された回数だけ、受信PECに一致しません。 この状態の場合、新たな自動生成ALRTPACKETがPECエラーなしに受信されると、このビットはクリアされます。 注:デコードされたPECが受信PECに一致しない場合、ALRTPCKTBUFは、このビットがセットされない場合でも受信ALRTPACKETデータを格納しません。 状態2 = BMS安全モニタリング・モードまたはスリープ・モードにおいて、自動生成ALRTPACKETが、COMM_TO_DLYレジスタで設定された通信タイムアウトを検出しています。 この状態の場合、BMS安全モニタリング・モードまたはスリープ・モードが終了すると、このビットはクリアされます。
ALRTPCKTBUF_HW_ERR	0	冗長アラート・パケット・バッファとメイン・アラート・パケット・バッファが一致しません。

STATUS_OPSTATE (R/W = 0x09/0x08)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	SAFEMON	–	–	–
Reset	–	–	–	–	0x0	–	–	–
Access Type	–	–	–	–	Read Only	–	–	–

ビットフィールド	ビット	説明
SAFEMON	3	安全モニタリング・モード デバイスが、ウォッチドッグ応答失敗の数を超え、BMS安全モニタリング・モードに入っています。

STATUS_BUF (R/W = 0x0B/0x0A)

BIT	7	6	5	4	3	2	1	0
Field	ALRTPCKT BUF_FULL	–	–	–	LSSM_FUL L	–	–	–
Reset	0x0	–	–	–	0x0	–	–	–
Access Type	Read Only	–	–	–	Read Only	–	–	–

ビットフィールド	ビット	説明	デコード
ALRTPCKTBUF_FULL	7	アラート・パケット・バッファが満杯。 PECエラーのない新たなALRTPACKETを受信すると、ALRTPCKTBUF_FULLステータス・ビットがセットされます。ALRTPCKTBUFのデータは、バイトの読出し後にゼロにクリアされます。いずれかの与えられたALRTPACKETバイトの読出し／クリアが行われた場合、新しいALRTPACKETは受信されず、ALRTPCKTBUF_FULLレジスタはクリアされます。 BMS安全モニタリング・モードおよびスリープ・モードの場合、ALRTPCKTBUFは自動的にクリアされ、ALRTPCKTBUF_FULLが自動的にトグルします。	0 = ALRTPCKTBUF is empty 1 = ALRTPCKTBUF is full
LSSM_FULL	3	LSSMが満杯。 LSSMキューに4つの未処理メッセージがある場合、LSSM_FULLステータス・ビットがセットされます。 LSSMが満杯の場合、送信バッファは、それ以上のトランザクションが送信されないようにして、オーバーフロー状態になるのを防止します。	0 = LSSM is not full 1 = LSSM is full

STATUS_WD (R/W = 0x0D/0x0C)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	WD_TO_ER R	WD_OPEN	WD_LFSR_ ERR	WD_RJCT_ ERR	WD_EXP_E RR
Reset	–	–	–	0x0	0x0	0x0	0x0	0x0
Access Type	–	–	–	Read Only	Read Only	Read Only	Read Only	Read Only

ビットフィールド	ビット	説明	デコード
WD_TO_ERR	4	ウォッチドッグ・タイムアウト 無効な応答の数がWD_DBNCレジスタに設定された回数に等しくなっています。 注: このビットがセットされた場合、WD_EN = 0になるとクリアされます。	
WD_OPEN	3	ウォッチドッグ・ウィンドウが開いています	0 = Watchdog window not open. Do not update. 1 = Watchdog window open. Update.
WD_LFSR_ERR	2	ウォッチドッグ無効キー オープン・ウィンドウ(延長オープン、オープン、または常時オープン)の間に、無効なキーがWD_KEYレジスタに書き込まれました 有効な更新時にクリアされます。 注: このビットがセットされた場合、WD_EN = 0になるとクリアされます。	0 = LFSR key match 1 = LFSR key mismatch since last read
WD_RJCT_ERR	1	ウィンドウが閉じている間のウォッチドッグ書き込み 有効な更新時にクリアされます。 注: このビットがセットされた場合、WD_EN = 0になるとクリアされます。	
WD_EXP_ERR	0	ウォッチドッグ期間終了 有効な更新時にクリアされます。 注: このビットがセットされた場合、WD_EN = 0になるとクリアされます。	0 = Watchdog timer not expired 1 = Watchdog timer expired

STATUS_GPIO (R/W = 0x0F/0x0E)

BIT	7	6	5	4	3	2	1	0
Field	GPIO4_RD	GPIO3_RD	GPIO2_RD	GPIO1_RD	GPIO4_ER R	GPIO3_ER R	GPIO2_ER R	GPIO1_ER R
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only

ビットフィールド	ビット	説明
GPIO4_RD	7	GPIO4ピンのロジック状態を示します。 注: BMS安全モニタリング・モードの安全対策状態の間、GPIO[n]_MASKレジスタ・ビットに従ってGPIOピンの出力はマスクされますが、このレジスタは、目的の出力駆動値をレポートします。

ビットフィールド	ビット	説明
GPIO3_RD	6	GPIO3ピンのロジック状態を示します。 注：BMS安全モニタリング・モードの安全対策状態の間、GPIO[n]_MASKレジスタ・ビットに従ってGPIOピンの出力はマスクされますが、このレジスタは、目的の出力駆動値をレポートします。
GPIO2_RD	5	GPIO2ピンのロジック状態を示します。 注：BMS安全モニタリング・モードの安全対策状態の間、GPIO[n]_MASKレジスタ・ビットに従ってGPIOピンの出力はマスクされますが、このレジスタは、目的の出力駆動値をレポートします。
GPIO1_RD	4	GPIO1ピンのロジック状態を示します。 注：BMS安全モニタリング・モードの安全対策状態の間、GPIO[n]_MASKレジスタ・ビットに従ってGPIOピンの出力はマスクされますが、このレジスタは、目的の出力駆動値をレポートします。
GPIO4_ERR	3	GPIO4ピン連続性エラー 駆動された出力値が入力値に一致しません。
GPIO3_ERR	2	GPIO3ピン連続性エラー 駆動された出力値が入力値に一致しません。
GPIO2_ERR	1	GPIO2ピン連続性エラー 駆動された出力値が入力値に一致しません。
GPIO1_ERR	0	GPIO1ピン連続性エラー 駆動された出力値が入力値に一致しません。

ALERT_RX (R/W = 0x11/0x10)

BIT	7	6	5	4	3	2	1	0
Field	RX_ERR_A LRT	–	RX_BUSY_ ALRT	RX_IDLE_A LRT	RX_OVRFL W_ERR_AL RT	RX_FULL_ ALRT	RX_STOP_ ALRT	RX_EMPTY_ ALRT
Reset	0x0	–	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write 0 to Clear, Read	–	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明
RX_ERR_ALRT	7	RX_RD_PTRにあるデータ・バイトにエラー(マンチェスターやパリティのエラーが含まれる、対応するキャラクタ)が含まれている可能性があります。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
RX_BUSY_ALRT	5	UARTはデータ受信のためビジー状態です。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
RX_IDLE_ALRT	4	UARTはデータ受信中ではありません。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
RX_OVRFLW_ERR_ALRT	3	受信バッファが満杯のため、受信バッファのRX_WR_POINTERの位置にあるデータ・バイトは上書きされます。 受信バッファが満杯でなくなると(バッファが読み出されると)クリアできます。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
RX_FULL_ALRT	2	受信バッファの空バイトの数がゼロです。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ビットフィールド	ビット	説明
RX_STOP_ALERT	1	RX停止を受信 UARTは正しくフレーム化されたメッセージ(ストップ・キャラクタ)の受信を終了し、読出し可能な状態になっています。全ての未読メッセージが読み出された後、あるいは、UARTが新たなプリアンブル・キャラクタを検出した場合、UARTはこのビットを自動的にクリアします。 バッファが空の状態ですトップ・キャラクタを受信した場合、UARTはこのビットをセットしません。 このビットは、自動化ALERTPACKETトランザクションに対してはセットされません。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
RX_EMPTY_ALERT	0	受信バッファがクリアされ、未読データがありません(RX_RD_PTR = RX_WR_PTR - 1)。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ALERT_TX (R/W = 0x13/0x12)

BIT	7	6	5	4	3	2	1	0
Field	TX_HELD_ALERT	—	TX_BUSY_ALERT	TX_IDLE_ALERT	TX_OVRFLW_ERR_ALERT	TX_FULL_ALERT	TX_AVAIL_ALERT	TX_EMPTY_ALERT
Reset	0x0	—	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write 0 to Clear, Read	—	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明
TX_HELD_ALRT	7	受信バッファの空のバイト数が、送信キューのメッセージ長より少なくなっています。 このステータス・ビットが真、かつ、TX_UNLIMITED = 1'b0の場合、送信バッファは、RXバッファに十分な空き容量ができるまで現在のLD_Qにデータを送信しません。 TX_UNLIMITED = 1'b1の場合、TX_HELDビットは決してセットされず、データはRXバッファの空き容量には無関係に送信されます。この状態でデータの完全性を確保するには、RX_OVRFLW_ERRをモニタする必要があります。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
TX_BUSY_ALRT	5	UARTはデータ送信のためビジー状態です。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
TX_IDLE_ALRT	4	UARTはデータ送信中ではありません。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
TX_OVRFLW_ERR_ALRT	3	次のキューに未送信データがあるため、LD_Qをインクリメントできません。この状態で何らかの書込みを行うとロード・キューを上書きします。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ビットフィールド	ビット	説明
TX_FULL_ALRT	2	ロード・キューを除いて送信バッファの全てのキューは満杯です (LD_Q = TX_Q - 1)。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
TX_AVAIL_ALRT	1	送信バッファの少なくとも1つのキューがロード可能です (TX_FULLが偽)。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
TX_EMPTY_ALRT	0	送信バッファの全てのキューがクリアされ、ロード可能です (LD_Q = TX_Q)。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ALERT_LSSM_BYTE (R/W = 0x15/0x14)

BIT	7	6	5	4	3	2	1	0
Field	RX_READY_ALRT	ALRTPCKT_STATUS_ERR_ALRT	COMM_ERR_ALRT	ALRTPCKT_ERR_ALRT	COMM_MSMTCH_ERR_ALRT	COMMAND_OP_ALRT	ALIVECOUNT_ERR_ALRT	HW_ERR_ALRT
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明
RX_READY_ALRT	7	UARTは、正しくフレーム化されたメッセージ(ストップ・キャラクタ、プリアンプル、またはフレーム・タイムアウト)の受信を終了し、読出し可能な状態になっています。 全ての未読メッセージの読出し後、UARTはこのビットを自動的にクリアします。 バッファが空の状態ですトップ・キャラクタを受信した場合、UARTはこのビットをセットしません。 このビットは、自動化ALERTPACKETトランザクションに対してはセットされません。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
ALRTPCKT_STATUS_ERR_ALRT	6	ALERTPACKETステータス・エラー 非マスク・アラート・パケット・ステータス・ビットがアラート・パケット・バッファで受信されています。 コマンド動作モードでは、受信ALERTPACKETステータス・ビットのマスキングは、STATUS_ERR_MASKレジスタの機能の1つです。 スリープ・モードまたはBMS安全モニタリング・モードでは、受信ALERTPACKETステータス・ビットのマスキングは、STATUS_DBNC_MASKレジスタの機能の1つであり、ステータスがバウンス防止されるとときに自動クリアされます。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
COMM_ERR_ALRT	5	通信エラー UARTメッセージ内でデータが破損、挿入、あるいは削除されたことによる通信エラーが検出されています。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ビットフィールド	ビット	説明
ALRTPCKT_ERR_ALRT	4	ALRTPACKETエラー このステータス・ビットは、以下のALRTPACKETステータス・エラーのいずれかでアサートされます。 ALRTPCKT_COMM_ERR ALRTPCKT_HW_ERR 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
COMM_MSMTCH_ERR_ALRT	3	通信ミスマッチ・エラー LSSMで、受信データが送信データに一致しない通信問題が検出されています。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
COMMAND_OP_ALRT	2	コマンド動作モード・アラート このビットは、コマンド動作モード(デバイスがスリープ・モードまたはBMS安全モニタリング・モードになってはいない)を表します。このビットがロジック1でない場合、デバイスはその他の動作モードに入っています。 このステータス・ビットは、SLP_ENとSAFEMON_ALRTのいずれかが真の場合に偽としてアサートされます。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ビットフィールド	ビット	説明
ALIVECOUNT_ERR_ALRT	1	自動アライブ・カウンタ・エラー 自動アライブ・カウンタ・モードが有効な場合、このビットは以下の2つの状況でアサートされます。 1) 受信ALIVECOUNTバイト (ALIVECOUNT_RET) が以下の予想値とは異なる $\text{ALIVECOUNT_RET} = \text{ALIVECOUNT_SEED} - \text{ALIVECOUNT_Q} + \text{DEV_COUNT}(\text{WRITEALL}/\text{READALL})$ $\text{ALIVECOUNT_RET} = \text{ALIVECOUNT_SEED} - \text{ALIVECOUNT_Q} + 1 (\text{WRITEDevice}/\text{READDevice}/ \text{READBLOCK})$ 2) ALIVECOUNT_Qレジスタがゼロ以外の場合にCLR_ALIVECOUNT_SEEDがセットされる。 アライブ・カウンタが無効化されている場合、または手動モードに設定されている場合、このビットはセットされません。 0の書込み = クリア 1の書込み = 効果なし 注: このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
HW_ERR_ALRT	0	ハードウェア・エラー デバイス動作に影響する可能性のあるハードウェア・エラーが報告されています。 このステータス・ビットのアサートの原因となる可能性のある条件には、以下のようなものがあります。 - 仕様範囲外の電源電圧または発振器が検出される。 - トランジェント・ロジック・フォルト。 0の書込み = クリア 1の書込み = 効果なし 注: このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。

ALERT_GEN(0xB)

(R/W = 0x17/0x16)

BIT	7	6	5	4	3	2	1	0
Field	–	DEV_COUNT_ERR_ALRT	WD_ERR_ALRT	GPIO_ERR_ALRT	DATAPATH_ERR_ALRT	SPI_ERR_ALRT	ALRTPCKT_COMM_ERR_ALRT	ALRTPCKT_BUF_HW_ERR_ALRT
Reset	–	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	–	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明
DEV_COUNT_ERR_ALRT	6	デバイス・カウント・エラー このエラーがアサートされるのは、HELLOALL、UPHOST、またはDOWNHOSTのコマンドから返されたデバイス・カウントが、ユーザ設定されたDEV_COUNTレジスタに一致しない場合です。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
WD_ERR_ALRT	5	ウォッチドッグ・エラーのまとめ このステータス・ビットは、以下のウォッチドッグ・ステータス・エラーのいずれかでアサートされます。 WD_TO_ERR WD_LFSR_ERR WD_RJCT_ERR WD_EXP_ERR 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
GPIO_ERR_ALRT	4	GPIO HWエラーのまとめ このステータス・ビットは、以下のGPIOステータス・エラーのいずれかと同時にアサートされます。 GPIO1_ERR GPIO2_ERR GPIO3_ERR GPIO4_ERR 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。

ビットフィールド	ビット	説明
DATAPATH_ERR_ALRT	3	データパス・エラー 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
SPI_ERR_ALRT	2	スリープ、設定メモリ検証、設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間に、ユーザが制限されているレジスタの読出しあるいは書込みを試行しています。そのような動作を試行すると、SPI_ERRがセットされます。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
ALRTPCKT_COMM_ERR_ALRT	1	アラート・パケット通信エラー 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
ALRTPCKTBUF_HW_ERR_ALRT	0	冗長アラート・パケット・バッファとメイン・アラート・パケット・バッファが一致しません。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。

ALERT_OPSTATE (R/W = 0x19/0x18)

BIT	7	6	5	4	3	2	1	0
Field	ALRTRST	–	SLP_ALRT	SLP_STAT US_ERR_A LRT	SAFEMON_ ALRT	SAFEMON_ GPIO12_AL RT	SAFEMON_ STATUS_E RR_ALRT	SAFEMON_ CONFIG_E RR_ALRT
Reset	0x1	–	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write 0 to Clear, Read	–	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明
ALRTRST	7	POR／リセット・アラート用の割込みフラグ パワーオン・リセットが行われたことを示します。 パワーオン直後に、その後のリセットを検出するためにこのアラートをクリアする必要があります。 このビットはマスク不可能です。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはマスク不可能であり、対応するALRTENビットはありません。
SLP_ALRT	5	デバイスがスリープ状態からウェイクアップしています。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。
SLP_STATUS_ERR_ALRT	4	スリープ・モード・ステータス・エラー スリープ・モードの間、STATUS_DBNCレジスタに設定された回数だけ、アラート・パケット・バッファに非マスクSTATUSビットが示され、その結果、デバイスがスリープ・モードを終了しました。 受信したALERTPACKETステータス・ビットのマスキングは、STATUS_DBNC_MASKレジスタの機能の1つです。 このレジスタをクリアすると、FAULTタイマーがリセットされます。 このビットは、ALRTRSTレジスタの影響は受けません。 0の書込み = クリア 1の書込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。

ビットフィールド	ビット	説明
SAFEMON_ALERT	3	安全モニタリング・モード・アラート デバイスが、ウォッチドッグ応答失敗の数を超え、BMS安全モニタリング・モードに入っています。 0の書込み = クリア 1の書込み = 効果なし このレジスタをクリアすると、FAULTタイマーがリセットされます。 このビットは、ALRTRSTレジスタの影響は受けません。 注:このビットは、デバイスがBMS安全モニタリング・モードになっている場合はクリアできません。
SAFEMON_GPIO12_ALERT	2	SAFEMON_GPIO12_ALERT GPIO1/2ピンは、安全モニタリング・モードでアサートされ、コマンド動作モードになりこのビットがクリアされるまでアサートされたままになります。 0の書込み = クリア 1の書込み = 効果なし 注:このビットは、ALRTRSTレジスタの影響は受けません。 注:このビットは、BMS安全モニタリング・モードの安全対策状態の間、自律的にトグルします。 注:このビットは、デバイスがBMS安全モニタリング・モードになっている場合、ユーザによるクリアはできません。
SAFEMON_STATUS_ERR_ALERT	1	安全モニタリング・モード・ステータス・エラー 安全モニタリング・モードの間、STATUS_DBNCレジスタに設定された回数だけ、アラート・パケット・バッファに非マスクSTATUSビットが示されます。 受信したALERTPACKETステータス・ビットのマスキングは、STATUS_DBNC_MASKレジスタの機能の1つです。 0の書込み = クリア 1の書込み = 効果なし 注:このビットは、デバイスがBMS安全モニタリング・モードになっている場合はクリアできません。

ビットフィールド	ビット	説明
SAFEMON_CONFIG_ERR_ALRT	0	安全モニタリング・モード設定エラー このステータス・ビットは、設定メモリの検証処理または設定メモリのロード処理の実行中にエラーが検出された場合にセットされます。 0の書込み = クリア 1の書込み = 効果なし 注: このビットは、デバイスがBMS安全モニタリング・モードになっている場合はクリアできません。

ALERT_BUF (R/W = 0x1B/0x1A)

BIT	7	6	5	4	3	2	1	0
Field	ALRTPCKT BUF_FULL_ ALRT	—	—	—	LSSM_FUL L_ALRT	—	—	—
Reset	0x0	—	—	—	0x0	—	—	—
Access Type	Write 0 to Clear, Read	—	—	—	Write 0 to Clear, Read	—	—	—

ビットフィールド	ビット	説明	デコード
ALRTPCKTBUF_FULL_ALRT	7	アラート・パケット・バッファが満杯。 PECエラーのない新たなALRTPACKETを受信すると、ALRTPCKTBUF_FULLステータス・ビットがセットされます。ALRTPCKTBUFのデータは、バイトの読み出し後にゼロにクリアされます。 0の書込み = クリア 1の書込み = 効果なし 注: このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。	0 = ALRTPCKTBUF is empty 1 = ALRTPCKTBUF is full

ビットフィールド	ビット	説明	デコード
LSSM_FULL_ALRT	3	LSSMが満杯。 LSSMキューに4つの未処理メッセージがある場合、LSSM_FULLステータス・ビットがセットされます。 LSSMが満杯の場合、送信バッファは、それ以上のトランザクションが送信されないようにして、オーバーフロー状態になるのを防止します。 0の書込み = クリア 1の書込み = 効果なし 注: このレジスタは永続的です。ユーザが書込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。	0 = LSSM is not full 1 = LSSM is full

ALERT_WD (R/W = 0x1D/0x1C)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	WD_TO_ERR_ALRT	WD_OPEN_ALRT	WD_LFSR_ERR_ALRT	WD_RJCT_ERR_ALRT	WD_EXP_ERR_ALRT
Reset	–	–	–	0x0	0x0	0x0	0x0	0x0
Access Type	–	–	–	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明	デコード
WD_TO_ERR_ALRT	4	ウォッチドッグ・タイムアウト 無効な応答の数がWD_DBNCレジスタに設定された回数に等しくなっています。 0の書込み = クリア 1の書込み = 効果なし 注: このレジスタは永続的です。ユーザが書込みクリアを試みてもこの状態が続く場合、このビットはセットされたままになります。	
WD_OPEN_ALRT	3	ウォッチドッグ・ウィンドウが開いています 0の書込み = クリア 1の書込み = 効果なし 注: このレジスタはイベント駆動です。ユーザが書込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。	0 = Watchdog window not open. Do not update. 1 = Watchdog window open. Update.

ビットフィールド	ビット	説明	デコード
WD_LFSR_ERR_ALRT	2	ウォッチドッグ無効キー オープン・ウィンドウ(延長オープン、オープン、または常時オープン)の間に、無効なキーがWD_KEYレジスタに書き込まれています 0の書き込み = クリア 1の書き込み = 効果なし 注:このレジスタは永続的です。ユーザが書き込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。	0 = LFSR key match. 1 = LFSR key mismatch since last read.
WD_RJCT_ERR_ALRT	1	ウィンドウが閉じている間のウォッチドッグ書き込み オープン・ウィンドウの間に有効なキーが書き込まれるとクリアされます 注:このレジスタは永続的です。ユーザが書き込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。	
WD_EXP_ERR_ALRT	0	ウォッチドッグ期間終了 0の書き込み = クリア 1の書き込み = 効果なし 注:このレジスタはイベント駆動です。ユーザが書き込みクリアを試みた場合、この状態が続いている場合でもレジスタはクリアされます。	0 = Watchdog timer not expired. 1 = Watchdog timer expired

ALERT_GPIO (R/W = 0x1F/0x1E)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIO4_ER R_ALRT	GPIO3_ER R_ALRT	GPIO2_ER R_ALRT	GPIO1_ER R_ALRT
Reset	–	–	–	–	0x0	0x0	0x0	0x0
Access Type	–	–	–	–	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read	Write 0 to Clear, Read

ビットフィールド	ビット	説明
GPIO4_ERR_ALRT	3	GPIO4ピン連続性エラー 駆動された出力値が入力値に一致しません。 0の書き込み = クリア 1の書き込み = 効果なし 注:このレジスタは永続的です。ユーザが書き込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
GPIO3_ERR_ALRT	2	GPIO3ピン連続性エラー 駆動された出力値が入力値に一致しません。 0の書き込み = クリア 1の書き込み = 効果なし 注:このレジスタは永続的です。ユーザが書き込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
GPIO2_ERR_ALRT	1	GPIO2ピン連続性エラー 駆動された出力値が入力値に一致しません。 0の書き込み = クリア 1の書き込み = 効果なし 注:このレジスタは永続的です。ユーザが書き込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。
GPIO1_ERR_ALRT	0	GPIO1ピン連続性エラー 駆動された出力値が入力値に一致しません。 0の書き込み = クリア 1の書き込み = 効果なし 注:このレジスタは永続的です。ユーザが書き込みクリアを試み、対応するSTATUSレジスタがセットされたままの場合、このビットはセットされたままになります。

ALRTEN_RX (R/W = 0x21/0x20)

BIT	7	6	5	4	3	2	1	0
Field	RX_ERR_ALRTEN	–	RX_BUSY_ALRTEN	RX_IDLE_ALRTEN	RX_OVRFLW_ERR_ALRTEN	RX_FULL_ALRTEN	RX_STOP_ALRTEN	RX_EMPTY_ALRTEN
Reset	0x0	–	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
RX_ERR_ALRTEN	7	ALERTピン・イネーブル
RX_BUSY_ALRTEN	5	ALERTピン・イネーブル
RX_IDLE_ALRTEN	4	ALERTピン・イネーブル
RX_OVRFLW_ERR_ALRTEN	3	ALERTピン・イネーブル
RX_FULL_ALRTEN	2	ALERTピン・イネーブル
RX_STOP_ALRTEN	1	ALERTピン・イネーブル
RX_EMPTY_ALRTEN	0	ALERTピン・イネーブル

ALRTEN_TX (R/W = 0x1/0x0)

BIT	7	6	5	4	3	2	1	0
Field	TX_HELD_ALRTEN	–	TX_BUSY_ALRTEN	TX_IDLE_ALRTEN	TX_OVRFLW_ERR_ALRTEN	TX_FULL_ALRTEN	TX_AVAIL_ALRTEN	TX_EMPTY_ALRTEN
Reset	0x0	–	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
TX_HELD_ALRTEN	7	ALERTピン・イネーブル
TX_BUSY_ALRTEN	5	ALERTピン・イネーブル
TX_IDLE_ALRTEN	4	ALERTピン・イネーブル
TX_OVRFLW_ERR_ALRTEN	3	ALERTピン・イネーブル
TX_FULL_ALRTEN	2	ALERTピン・イネーブル
TX_AVAIL_ALRTEN	1	ALERTピン・イネーブル
TX_EMPTY_ALRTEN	0	ALERTピン・イネーブル

ALRTEN_LSSM_BYTE (R/W = 0x25/0x24)

BIT	7	6	5	4	3	2	1	0
Field	RX_READY_ALRTEN	ALRTPCKT_STATUS_ERR_ALRTEN	COMM_ERR_ALRTEN	ALRTPCKT_ERR_ALRTEN	COMM_MISMATCH_ERR_ALRTEN	COMMAND_OP_ALRTEN	ALIVECOUNT_ERR_ALRTEN	HW_ERR_ALRTEN
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x1
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
RX_READY_ALRTEN	7	ALERTピン・イネーブル
ALRTPCKT_STATUS_ERR_ALRTEN	6	ALERTピン・イネーブル

ビットフィールド	ビット	説明
COMM_ERR_ALRTEN	5	ALERTピン・イネーブル
ALRTPCKT_ERR_ALRTEN	4	ALERTピン・イネーブル
COMM_MSMTCH_ERR_ALRTEN	3	ALERTピン・イネーブル
COMMAND_OP_ALRTEN	2	ALERTピン・イネーブル
ALIVECOUNT_ERR_ALRTEN	1	ALERTピン・イネーブル
HW_ERR_ALRTEN	0	ALERTピン・イネーブル

ALRTEN_GEN (R/W = 0x27/0x26)

BIT	7	6	5	4	3	2	1	0
Field	–	DEV_COUNT_ERR_ALRTEN	WD_ERR_ALRTEN	GPIO_ERR_ALRTEN	DATAPATH_ERR_ALRTEN	SPI_ERR_ALRTEN	ALRTPCKT_COMM_ERR_ALRTEN	ALRTPCKTBUF_HW_ERR_ALRTEN
Reset	–	0x0	0x1	0x1	0x0	0x1	0x0	0x0
Access Type	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
DEV_COUNT_ERR_ALRTEN	6	ALERTピン・イネーブル
WD_ERR_ALRTEN	5	ALERTピン・イネーブル
GPIO_ERR_ALRTEN	4	ALERTピン・イネーブル
DATAPATH_ERR_ALRTEN	3	ALERTピン・イネーブル
SPI_ERR_ALRTEN	2	
ALRTPCKT_COMM_ERR_ALRTEN	1	
ALRTPCKTBUF_HW_ERR_ALRTEN	0	

ALRTEN_OPSTATE (R/W = 0x29/0x28)

BIT	7	6	5	4	3	2	1	0
Field	–	–	SLP_ALRTEN	SLP_STATUS_ERR_ALRTEN	SAFEMON_ALRTEN	SAFEMON_GPIO12_ALRTEN	SAFEMON_STATUS_ERR_ALRTEN	SAFEMON_CONFIG_ERR_ALRTEN
Reset	–	–	0x1	0x1	0x1	0x1	0x1	0x1
Access Type	–	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
SLP_ALRTEN	5	ALERTピン・イネーブル
SLP_STATUS_ERR_ALRTEN	4	ALERTピン・イネーブル
SAFEMON_ALRTEN	3	ALERTピン・イネーブル
SAFEMON_GPIO12_ALRTEN	2	ALERTピン・イネーブル
SAFEMON_STATUS_ERR_ALRTEN	1	ALERTピン・イネーブル
SAFEMON_CONFIG_ERR_ALRTEN	0	ALERTピン・イネーブル

ALRTEN_BUF (R/W = 0x2B/0x2A)

BIT	7	6	5	4	3	2	1	0
Field	ALRTPCKT BUF_FULL_ ALRTEN	–	–	–	LSSM_FUL L_ALRTEN	–	–	–
Reset	0x0	–	–	–	0x0	–	–	–
Access Type	Write, Read	–	–	–	Write, Read	–	–	–

ビットフィールド	ビット	説明
ALRTPCKTBUF_FULL_ALRTEN	7	$\overline{\text{ALERT}}$ ピン・イネーブル
LSSM_FULL_ALRTEN	3	$\overline{\text{ALERT}}$ ピン・イネーブル

ALRTEN_WD (R/W = 0x2D/0x2C)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	WD_TO_ER R_ALRTEN	WD_OPEN _ALRTEN	WD_LFSR ERR_ALRT EN	WD_RJCT ERR_ALRT EN	WD_EXP_E RR_ALRTE N
Reset	–	–	–	0x0	0x0	0x0	0x0	0x0
Access Type	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
WD_TO_ERR_ALRTEN	4	$\overline{\text{ALERT}}$ ピン・イネーブル
WD_OPEN_ALRTEN	3	$\overline{\text{ALERT}}$ ピン・イネーブル
WD_LFSR_ERR_ALRTEN	2	$\overline{\text{ALERT}}$ ピン・イネーブル
WD_RJCT_ERR_ALRTEN	1	$\overline{\text{ALERT}}$ ピン・イネーブル
WD_EXP_ERR_ALRTEN	0	$\overline{\text{ALERT}}$ ピン・イネーブル

ALRTEN_GPIO (R/W = 0x2F/0x2E)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIO4_ER R_ALRTEN	GPIO3_ER R_ALRTEN	GPIO2_ER R_ALRTEN	GPIO1_ER R_ALRTEN
Reset	–	–	–	–	0x0	0x0	0x0	0x0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明
GPIO4_ERR_ALRTEN	3	$\overline{\text{ALERT}}$ ピン・イネーブル
GPIO3_ERR_ALRTEN	2	$\overline{\text{ALERT}}$ ピン・イネーブル
GPIO2_ERR_ALRTEN	1	$\overline{\text{ALERT}}$ ピン・イネーブル
GPIO1_ERR_ALRTEN	0	$\overline{\text{ALERT}}$ ピン・イネーブル

CLR_TXBUF (R/W = 0x41/0x40)

CLR_TXBUFコマンド

送信バッファをデフォルト状態にリセットし、TX_QおよびLD_Qをクリアします。

注:スリープ、設定メモリ検証、設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。

CLR_RXBUF (R/W = 0x43/0x42)

CLR_RXBUFコマンド

受信バッファをデフォルト状態にリセットします。

注:スリープ、設定メモリ検証、設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。

CLR_LSSM (R/W = 0x45/0x44)

CLR_LSSMコマンド

LSSMをリセットします。

注:スリープ、設定メモリ検証、設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。

CLR_ALIVECOUNT_SEED (R/W = 0x49/0x48)

ALIVECOUNT_SEEDレジスタをクリアします。

ALIVECOUNT_Qレジスタがゼロ以外の場合無視され、ALIVECOUNT_ERRステータス・ビットをフラグ通知しします。

注:スリープ、設定メモリ検証、設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。

SWPOR (R/W = 0x4B/0x4A)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	–	–	SWPOR
Reset	–	–	–	–	–	–	–	0b0
Access Type	–	–	–	–	–	–	–	Write, Read, Pulse

ビットフィールド	ビット	説明	デコード
SWPOR	0	ソフトウェアPORコマンド このビットはデバイスを完全にリセットし、POR状態をエミュレートします。	0 = Normal operation (default, no effect) 1 = Initiates software POR event

SLP_EN (R/W = 0x4D/0x4C)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	–	–	SLP_EN
Reset	–	–	–	–	–	–	–	0b0
Access Type	–	–	–	–	–	–	–	Write, Read, Ext

ビットフィールド	ビット	説明	デコード
SLP_EN	0	スリープ・イネーブル・コマンド このビットはスリープ・モードを終了すると自動クリアされます。ホストは、このビットを手動でクリアすることによってスリープ・モードを終了することもできます。 注: 設定メモリ検証または設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。	0 = Normal operation 1 = Sleep mode

VER_CONFIG (R/W = 0x4F/0x4E)

BIT	7	6	5	4	3	2	1	0
Field	—	—	—	—	—	—	—	VER_CONFIG
Reset	—	—	—	—	—	—	—	0b0
Access Type	—	—	—	—	—	—	—	Write, Read, Ext

ビットフィールド	ビット	説明	デコード
VER_CONFIG	0	設定メモリの検証コマンド この動作に失敗した場合は、DATAPATH_ERRビットまたはSAFEMON_CONFIG_ERR_ALRTビットによって通知されます。 注: スリープまたは設定メモリ・ロードの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。	0 = Normal operation (default, no effect) 1 = Executes the Verify Configuration Memory command

LOAD_CONFIG (R/W = 0x51/0x50)

BIT	7	6	5	4	3	2	1	0
Field	—	—	—	—	—	—	—	LOAD_CONFIG
Reset	—	—	—	—	—	—	—	0b0
Access Type	—	—	—	—	—	—	—	Write, Read, Ext

ビットフィールド	ビット	説明	デコード
LOAD_CONFIG	0	設定メモリのロード・コマンド この動作に失敗すると、SAFEMON_CONFIG_ERR_ALRTビットによって通知されます。 注: 設定メモリ検証またはスリープの処理がアクティブになっている間、あるいは、BMS安全モニタリング・モードの間、このコマンドを書き込むことはできません。そのような動作を試行すると、SPI_ERR_ALRTビットがセットされます。	0 = Normal operation (default, no effect) 1 = Executes the Load Configuration Memory command using Configuration Memory Queue 0-2

WD_KEY (R/W = 0x53/0x52)

ウォッチドッグ・キー・レジスタ (読出し／書込み)

BIT	7	6	5	4	3	2	1	0
Field	WD_KEY[7:0]							
Reset	0xAA							
Access Type	Write, Read, Ext							

ビットフィールド	ビット	説明
WD_KEY	7:0	ウォッチドッグ・キー。現在のキーはこのレジスタから読み出せます。ウォッチドッグを更新するには、シーケンス内で次の値をこのレジスタに書き込む必要があります。 単純なウィンドウ化されたウォッチドッグとして設定されている場合、WD_KEYレジスタに何らかの値を書き込むと、ウォッチドッグが更新され、書き込まれた値は無視されます。 チャレンジ／応答ウォッチドッグとして設定されている場合は、WD_KEYレジスタに対して書き込む応答が不適切な場合、書き込まれる値は無視され、WD_LFSR違反となります。オープン・ウィンドウの間にチャレンジ／応答モードで不適切な応答を書き込むとリフレッシュされ、WD_KEYレジスタがアップデートされます。クローズド・ウィンドウの間にチャレンジ／応答モードで正しい応答を書き込んでも、書き込みは無視され、WD_UV違反となります。 LFSR多項式 = $x^8 + x^6 + x^3 + x^2 + 1$

CONFIG_GEN0 (R/W = 0x61/0x60)

一般設定レジスタ (読出し／書込み)

BIT	7	6	5	4	3	2	1	0
Field	–	–	DEV_COUNT[5:0]					
Reset	–	–	0x0					
Access Type	–	–	Write, Read					

ビットフィールド	ビット	説明
DEV_COUNT	5:0	デバイス・カウント・レジスタ このレジスタには、ユーザがダイジェチェーン内のデバイス数をプログラムする必要があります。 このレジスタは自動アライブ・カウントおよびデータチェック・パーサ機能と共に使用し、ダイジェチェーンの完全性を確保します。 注:このレジスタを、ダイジェチェーン・デバイスのデバイス・アドレスと混同してはなりません。

CONFIG_GEN1 (R/W = 0x63/0x62)

一般設定レジスタ(読出し／書込み)

BIT	7	6	5	4	3	2	1	0
Field	SINGLE_ENDED	BAUD_RATE[2:0]			–	–	–	–
Reset	0x0	0b011			–	–	–	–
Access Type	Write, Read	Write, Read			–	–	–	–

ビットフィールド	ビット	説明	デコード
SINGLE_ENDED	7	入力スレッシュホールドを負側にシフトすることによって、UARTがシングルエンド信号を受信できるようにします(ゼロ差動電圧がロジック1)。このモードでは、RXP入力グランドに接続され、反転信号を受信する必要があります。これは差動モードの場合と同じです。このモードでは、TXポートも差動モードの場合と同じ動作をします。デフォルトは差動モードです。	0 = Normal operation 1 = Enables single-ended signal operation
BAUD_RATE	6:4	UARTのボー・レート 注:UARTがデータを送信している間、または受信している間にこのレジスタをプログラミングすると、予期せぬ動作が生じる可能性があります。	000 = 500kbps 001 = 500kbps 010 = 1Mbps 011 = 2Mbps (default) 100 = 4Mbps 101-111 = Reserved Writes to reserved values default to 500kbps.

CONFIG_GEN2 (R/W = 0x65/0x64)

一般設定レジスタ(読出し／書込み)

BIT	7	6	5	4	3	2	1	0
Field	RX_RAW_DATA	TX_RAW_DATA	TX_PREAMBLES	TX_QUEUE	TX_ODDPARITY	TX_PAUSE	TX_NOSTOP	TX_NOPREAMBLE
Reset	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明	デコード
RX_RAW_DATA	7	生データ受信モード	0 = Normal operation 1 = Disables Manchester decoding of the received data. In this mode, there is one data byte stored for every one character received (instead of every two received).
TX_RAW_DATA	6	生データ送信モード	0 = Normal operation 1 = Disables Manchester encoding of transmitted data. In this mode, each data byte is transmitted as one character (instead of two characters).
TX_PREAMBLES	5	プリアンブル送信モード	0 = Normal operation 1 = Transmits preambles continuously. This mode takes precedence over all transmit modes except Transmit Pause mode.
TX_QUEUE	4	キュー送信モード 注:TX_UNLIMITEDはTX_QUEUEモードにのみ適用されます	0 = TX Queue message not transmitted but UART could transmit based on TX_PREAMBLES and TX_PAUSE 1 = Enables transmission of the message loaded in the Transmit queue if 1) there is sufficient space in the Receive buffer for the message (RX_FULL is false) OR 2) the limitations on message length are removed (TX_UNLIMITED is set).
TX_ODDPARITY	3	奇数パリティ送信モード 奇数パリティのキャラクタを送信します。UARTプロトコルは偶数パリティを用いるため、このモードは、システムがパリティ・エラーを検出できるかどうかをテストするために用いることができます。偶数パリティがデフォルトです。	0 = Normal operation 1 = Enables Odd Parity mode

ビットフィールド	ビット	説明	デコード
TX_PAUSE	2	送信停止モード 注: このモードは、その他の全ての送信モード（プリアンプル送信モード、キュー送信モード、キープアライブ・モード）に優先します。	0 = Normal operation 1 = Places the transmitter into idle state once the UART is finished transmitting the current byte; however, the TX_BUSY and TX_IDLE bits remain unchanged. Transmission resumes when this bit is cleared.
TX_NOSTOP	1	送信無停止モード ストップ・キャラクタのないメッセージを送信します。TX_NOPREAMBLEビット・セットと共に後続メッセージを送信することによって、不定長のフレーム化メッセージを構築できます。TX_UNLIMITEDビットは、86バイトより長いメッセージに対して設定する必要があります。	0 = Stop character enabled 1 = Stop character disabled
TX_NOPREAMBLE	0	無プリアンプル送信モード プリアンプルのないメッセージを送信します。TX_NOSTOPビットがセットされているメッセージを最初に送信し、次にこのビットをセットしてメッセージを送信することによって、不定長のフレーム化メッセージを構築できます。ただし、先行メッセージがストップ・キャラクタ（フレームの終了）で終了している場合、このモードで送信されたデータはフレーム化されず（プリアンプルなし）受信バッファには格納されません。	0 = Normal operation 1 = Enables Preamble mode

CONFIG_GEN3 (R/W = 0x67/0x66)

一般設定レジスタ（読み出し／書き込み）

BIT	7	6	5	4	3	2	1	0
Field	–	TX_AUTO	TX_UNLIMITED	SPI_DOUT_EN	ALRTPCKT_TIMING[3:0]			
Reset	–	0x0	0x0	0x0	0b1111			
Access Type	–	Write, Read	Write, Read	Write, Read	Write, Read			

ビットフィールド	ビット	説明	デコード
TX_AUTO	6	送信自動モード このモードでは、プリアンプルを受信するとUARTが自動的にNXT_LDQコマンドを実行します。 このモードは、ループのテストを目的としたもので、スリープ、安全モニタ、設定ロード、設定検証のプロセスの間は機能しません。	0 = Transmit Automatic mode disabled 1 = Transmit Automatic mode enabled
TX_UNLIMITED	5	送信無制限モード このモードでは、送信キューは、メッセージ長をデフォルトの制限値である86バイトではなく、255バイトに自動的に制限します。メッセージ長が受信バッファに書込み可能な容量を超えていても、メッセージの送信は許可されます。	0 = Normal operation 1 = Enables Transmit Unlimited mode
SPI_DOUT_EN	4	SPI出力イネーブル	0 = DOUT pin is tri-stated when CSB is deasserted (default) 1 = DOUT pin is always driven

ビットフィールド	ビット	説明	デコード
ALRTPCKT_TIMING	3:0	アラート・パケット／キープアライブのタイミング このレジスタは、コマンド動作モード時に、キープアライブ(ストップ)キャラクタの生成のタイミングまたは、アラート・パケットの生成のタイミングを制御します。 CO_ALRTPCKTEN = 1の場合、アラート・パケットが生成されます。 CO_ALRTPCKTEN = 0の場合、キープアライブ・キャラクタが生成されます。 キープアライブは、通信の行われていない期間(アイドル状態)にスリープ・デバイスがシャットダウンするのを防止します。周期的なストップ・キャラクタ間のアイドル時間は、右側に示す4ビット値に基づきます。デフォルト設定は無限(モード無効化)です。 注:送信停止、プリアンブル送信、キュー送信の各モードは、このモードに優先します。 注:スリープ・モードまたはBMS安全モニタリング・モード時は、アラート・パケットは、それぞれ、SLP_SCAN_DLYまたはSAFEMON_SCAN_DLYに従って生成されます。 注:推奨するアラート・パケットのタイミング・レートは1.28ms(デフォルト)です。 注:5.12msより大きなキープアライブ設定にすると、デジチェーンのデバイスのシャットダウンを防止できません。 注:アラート・パケット生成用に設定した場合、320μs未満のタイミング設定は320μsにデフォルト設定されます。	0000 = 0μs (Continuous) 0001 = 10μs 0010 = 20μs 0011 = 40μs 0100 = 80μs 0101 = 160μs 0110 = 320μs (Alert Packet minimum Timing) 0111 = 640μs 1000 = 1.28ms 1001 = 2.56ms 1010 = 5.12ms 1011 = 10.24ms 1111 = Infinite delay/disabled (default)

CONFIG_GEN4 (R/W = 0x69/0x68)

一般設定レジスタ(読み出し／書き込み)

BIT	7	6	5	4	3	2	1	0
Field	CO_ALRTPCKTEN	RXSWAP_EN	MS_EN[1:0]		DC_EN[1:0]		ALIVECOUNT_EN[1:0]	
Reset	0x0	0b0	0b10		0b10		0x0	
Access Type	Write, Read	Write, Read	Write, Read		Write, Read		Write, Read	

ビットフィールド	ビット	説明	デコード
CO_ALRTPCKTEN	7	コマンド動作アラート・パケット・イネーブル	0 = Stop characters are generated according to the ALRTPCKT_TIMING register setting. 1 = Alert Packets are generated according to the ALRTPCKT_TIMING register setting.
RXSWAP_EN	6	受信バッファ・データ・スワップ・イネーブル このビットは、MAX17851がリードバックするRXバッファ・データの順序を反転します。	0 = Receive data is reported normally. 1 = Receive data is reversed.
MS_EN	5:4	マスタ・イネーブル	0x = Slave, dual UART 10 = Master, single UART (default) 11 = Master, dual UART
DC_EN	3:2	データチェック・バイト・イネーブル 有効化されている場合、データチェック・バイトはホストが生成する必要があります。MAX17851は、受信するデータチェック・バイトを自動的に解析し、返されたデイジーチェーンDC_PECERRをALERTピンおよびLSSMバイトを介してレポートします。これ以外のデータチェック・バイトからのビットはレポートされません。その他の全てのデータチェック・バイト・エラーは、ALRTPCKTBUFデータでレポートされます。 データチェック・バイトが有効化されている場合、ユーザは、これをRXバッファに格納するかどうかを選択できます。	0x = Data Check byte is disabled. 10 = Data Check byte enabled and stored in the RX buffer (default). 11 = Data Check byte is enabled and not stored in the RX buffer.

ビットフィールド	ビット	説明	デコード
ALIVECOUNT_EN	1:0	アライブ・カウンタ・イネーブル このレジスタは、アライブ・カウンタ・モードを選択します。 ユーザ指定アライブ・カウンタ・モードの場合、ホストは、UARTトランザクションの末尾にアライブ・カウンタ・バイトを付加します。バイトは、RXバッファに格納されます。 自動アライブ・カウンタ・モードの場合、MAX17851は、UARTトランザクションにローリング・アライブ・カウンタ・バイトを自動的に付加します。受信されたアライブ・カウンタ・バイトはALIVECOUNT_RETレジスタに格納され、RXバッファには格納されません。	0x = Alive Counter disabled (default). 10 = User specified Alive Counter enabled. 11 = Automated Alive Counter enabled. Note: This setting should match downstream daisy-chain devices; otherwise, the MAX17851 LSSM reports PEC errors.

CONFIG_GEN5 (R/W = 0x6B/0x6A)

一般設定レジスタ (読出し／書込み)

BIT	7	6	5	4	3	2	1	0
Field	TX_START_SETUP	TX_HI_Z	ALRTPCKT_DBNC[2:0]			SPI_SFTYS_CLK	SPI_SFTYS_DI	SPI_SFTYC_SB
Reset	0x1	0x0	0x0			0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read			Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明	デコード
TX_START_SETUP	7	UART TX STARTセットアップ・イネーブル このビットは、送信UARTスタート・ビットのオプションのハイ・セットアップ時間を有効にします。	0 = Not enabled. 1 = Enabled (default).
TX_HI_Z	6	UART TXアイドル状態	0 = TX pins idle at ground. 1 = TX pins idle at Hi-Z.
ALRTPCKT_DBNC	5:3	ステータスのバウンス防止 このレジスタは、安全モニタリング・モードまたはスリープ・モードで動作を開始する前に発生しなくてはならない、非マスクの連続するALERTPACKETステータス・アラートの数を制御します。	000 = 1 (default) 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 110 = 64 111 = 128
SPI_SFTYSCLK	2		0 = Disables internal 100k pulldown on SCLK. 1 = Enables internal 100k pulldown on SCLK.

ビットフィールド	ビット	説明	デコード
SPI_SFTYSDI	1		0 = Disables internal 100k pulldown on SDI. 1 = Enables internal 100k pulldown on SDI.
SPI_SFTYCSB	0		0 = Disables internal 100k pullup on CSB. 1 = Enables internal 100k pullup on CSB.

CONFIG_SAFEMON0 (R/W = 0x6D/0x6C)

安全モニタリング・モード設定レジスタ0

BIT	7	6	5	4	3	2	1	0
Field	GPIOREC_DLY[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明	デコード
GPIOREC_DLY	7:0	GPIO回復遅延 このレジスタは、GPIO回復遅延タイマーを制御します。このタイマーによって、マイクロコントローラは、GPIOピンの状態を読み出し、回復させるための機会を得ることができます。 0xFF (無効化) に設定されていない限り、タイマー遅延は次式で計算されます。 遅延 = (GPIOREC_DLY * 10) + 100 (mSec)	0x00 = 100 mSec 0x01 = 110 mSec 0xFE = 2640 mSec 0xFF = disabled

CONFIG_SAFEMON1 (R/W = 0x6F/0x6E)

安全モニタリング・モード設定レジスタ1

BIT	7	6	5	4	3	2	1	0
Field	CONT_TIMER_DLY[7:0]							
Reset	0xFF							
Access Type	Write, Read							

ビットフィールド	ビット	説明	デコード
CONT_TIMER_DLY	7:0	SAFEMONコンタクト・タイマー遅延 このレジスタは、安全モニタリング・モードのコンタクト・タイマー遅延を制御します。このタイマーによる設定時間が経過すると、GPIO1および2がアサートされます(安全モニタリング出力動作に正しく設定されている場合)。 0xFF(無限)に設定されていない限り、タイマー遅延は次式で計算されます。 遅延 = (CONT_TIMER_DLY * 4) + 16(min)	0x00 = 16 min 0x01 = 20 min 0xFD = 1028 min 0xFE = 1032 min 0xFF = Infinite (default)

CONFIG_SAFEMON2 (R/W = 0x71/0x70)

安全モニタリング・モード設定レジスタ2

BIT	7	6	5	4	3	2	1	0
Field	—	—	—	—	—	—	SAFEMON_SCAN_DLY[1:0]	
Reset	—	—	—	—	—	—	0x0	
Access Type	—	—	—	—	—	—	Write, Read	

ビットフィールド	ビット	説明	デコード
SAFEMON_SCAN_DLY	1:0	安全モニタリング・モード・スキャン遅延 このレジスタは、安全モニタリング・モードでのアラート・パケット生成のタイミングを制御します。	00 = 500mS (default) 01 = 1S 10 = 1.5S 11 = Reserved

CONFIG_SAFEMON3 (R/W = 0x73/0x72)

安全モニタリング・モード設定レジスタ3

BIT	7	6	5	4	3	2	1	0
Field	NOMON	—	—	—	SM_GPIO4_MASK	SM_GPIO3_MASK	SM_GPIO2_MASK	SM_GPIO1_MASK
Reset	0x0	—	—	—	0x1	0x1	0x1	0x1
Access Type	Write, Read	—	—	—	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明	デコード
NOMON	7	BMS安全モニタリング・モードでのモニタリングなし／ALERTPACKETなしの通信。	0 = Device begins monitoring the daisy-chain BMS Safe Monitoring states after the expiration of GPIOREC_DLY. 1 = Device immediately asserts all GPIO pins if the watchdog is not refreshed prior to the expiration of GPIOREC_DLY.
SM_GPIO4_MASK	3	GPIOピンがSAFEMON出力として設定されている場合、このビットは、SAFEMON FSMの安全対策状態の間、出力ピンの駆動をマスクします。	
SM_GPIO3_MASK	2	GPIOピンがSAFEMON出力として設定されている場合、このビットは、SAFEMON FSMの安全対策状態の間、出力ピンの駆動をマスクします。	
SM_GPIO2_MASK	1	GPIOピンがSAFEMON出力として設定されている場合、このビットは、SAFEMON FSMの安全対策状態の間、出力ピンの駆動をマスクします。	
SM_GPIO1_MASK	0	GPIOピンがSAFEMON出力として設定されている場合、このビットは、SAFEMON FSMの安全対策状態の間、出力ピンの駆動をマスクします。	

CONFIG_SLP (R/W = 0x75/0x74)

BIT	7	6	5	4	3	2	1	0
Field	–	–	SLP_SCAN_DLY[1:0]		SLP_ALRT PCKTEN	SLP_CBNTFY[2:0]		
Reset	–	–	0x0		0x0	0x0		
Access Type	–	–	Write, Read		Write, Read	Write, Read		

ビットフィールド	ビット	説明	デコード
SLP_SCAN_DLY	5:4	スリープ・モード・スキャン遅延 このレジスタは、スリープ・モードでのアラート・パケット生成のタイミングを制御します。	00 = 500mS (default) 01 = 1S 10 = 1.5S 11 = Reserved
SLP_ALRTPCKTEN	3	スリープ・モード・アラート・パケット・イネーブル	0 = Alert Packets are not generated in Sleep mode. 1 = Alert Packets are generated according to the SLP_SCAN_DLY register setting in Sleep mode.

ビットフィールド	ビット	説明	デコード
SLP_CBNTFY	2:0	スリープ・モード・セル・バランシング通知 セル・バランシング・タイマーがCBNTFYレジスタにプログラムされた値を超過した場合、デバイスはスリープ・モードを終了します。	000 = 30 min (default) 001 = 60 min 010 = 120 min 011 = 240 min 1XX = OFF (0 min)

CONFIG_COMM (R/W = 0x77/0x76)

BIT	7	6	5	4	3	2	1	0
Field	–	–	COMM_RTRY[1:0]		–	COMM_TO_DLY[2:0]		
Reset	–	–	0x0		–	0x001		
Access Type	–	–	Write, Read		–	Write, Read		

ビットフィールド	ビット	説明	デコード
COMM_RTRY	5:4	通信リトライ・カウント・スレッシュホールド このレジスタは、BMS安全モニタリング・モードおよびスリープ・モード時の通信リトライ・カウント・スレッシュホールドを制御します。	00 = 2 01 = 4 10 = 8 11 = 16

ビットフィールド	ビット	説明	デコード
COMM_TO_DLY	2:0	通信タイムアウト遅延 このレジスタは、全ての動作モードの通信タイムアウト遅延を制御します。通信タイムアウト・タイマーは、トランザクションの開始時にスタートします。通信タイムアウト時間が経過すると、COMM_ERRステータス・ビットがセットされます。 スリープ・モードまたはBMS安全モニタリング・モード時は、関連するSCAN_DLYレジスタに従ってALERTPACKTコマンドが自動的に生成されます。スリープ・モードまたはBMS安全モニタリング・モードでの通信タイムアウトは、ALRTPCKT_TO_ERRの原因にもなります。 通信タイムアウト遅延は、このレジスタ設定値をBAUD_RATEレジスタ設定値で割った値の関数です。 例えば、COMM_TO_DLY=516ビット(43UARTフレーム)で、BAUD_RATE = 2Mbpsの場合、通信タイムアウト遅延は、516ビット/2Mbps= 258μsになります。	000 = 276 bits (23 frames) 001 = 516 bits (43 frames, default) 010 = 996 bits (83 frames) 011 = 1956 bits (163 frames) 1XX = Disabled

STATUS_DBNC_MASK0 (R/W = 0x79/0x78)

BIT	7	6	5	4	3	2	1	0
Field	STATUS_DBNC_MASK[7:0]							
Reset	0x20							
Access Type	Write, Read							

ビットフィールド	ビット	説明
STATUS_DBNC_MASK	7:0	アラート・パケット・バッファのSTATUSビットをマスクして、バウンス防止SAFEMON_STATUS_ERR_ALRTビットおよびバウンス防止SLEEP_STATUS_ERR_ALRTビットを生成します。 注: 0x20というリセット値は、デジタイゼーションのデバイスのPECステータス・ビットをデフォルトでマスクするためのものです。

STATUS_DBNC_MASK1 (R/W = 0x7B/0x7A)

BIT	7	6	5	4	3	2	1	0
Field	STATUS_DBNC_MASK[15:8]							
Reset	0x40							
Access Type	Write, Read							

ビットフィールド	ビット	説明
STATUS_DBNC_MASK	7:0	アラート・パケット・バッファのSTATUSビットをマスクして、バウンス防止SAFEMON_STATUS_ERR_ALRTビットおよびバウンス防止SLEEP_STATUS_ERR_ALRTビットを生成します。 注: 0x40というリセット値は、デジタイゼーションのデバイスのALRTRSTステータス・ビットをデフォルトでマスクするためのものです。

STATUS_ERR_MASK0 (R/W = 0x7D/0x7C)

BIT	7	6	5	4	3	2	1	0
Field	STATUS_ERR_MASK[7:0]							
Reset	0x20							
Access Type	Write, Read							

ビットフィールド	ビット	説明
STATUS_ERR_MASK	7:0	アラート・パケット・バッファのSTATUSビットをマスクして、非バウンス防止ALRTPCKT_STATUS_ERRビットを生成します。 注: 0x20というリセット値は、デジタイゼーションのデバイスのPECステータス・ビットをデフォルトでマスクするためのものです。

STATUS_ERR_MASK1 (R/W = 0x7F/0x7E)

BIT	7	6	5	4	3	2	1	0
Field	STATUS_ERR_MASK[15:8]							
Reset	0x40							
Access Type	Write, Read							

ビットフィールド	ビット	説明
STATUS_ERR_MASK	7:0	アラート・パケット・バッファのSTATUSビットをマスクして、非バウンス防止ALRTPCKT_STATUS_ERRビットを生成します。 注: 0x40というリセット値は、デジタイゼーションのデバイスのALRTRSTステータス・ビットをデフォルトでマスクするためのものです。

CONFIG_GPIO12 (R/W = 0x81/0x80)

BIT	7	6	5	4	3	2	1	0
Field	–	GPIO2_CFG[2:0]			–	GPIO1_CFG[2:0]		
Reset	–	0b110			–	0b110		
Access Type	–	Write, Read			–	Write, Read		

ビットフィールド	ビット	説明	デコード
GPIO2_CFG	6:4	GPIO2ピンの動作をプログラムします。	000 = General Purpose input 001 = Reserved (idles as a high impedance input with 2MΩ impedance to ground) 010 = General Purpose HI output 011 = General Purpose LO output 100 = Reserved 101 = Reserved 110 = SAFEMON Active HI output (idles LO, default) 111 = SAFEMON Active LO output (idles HI)
GPIO1_CFG	2:0	GPIO1ピンの動作をプログラムします。	000 = General Purpose input 001 = Reserved (idles as a high impedance input with 2MΩ impedance to ground) 010 = General Purpose HI output 011 = General Purpose LO output 100 = Reserved 101 = Reserved 110 = SAFEMON active HI output (idles LO, default) 111 = SAFEMON active LO output (idles HI)

CONFIG_GPIO34 (R/W = 0x83/0x82)

BIT	7	6	5	4	3	2	1	0
Field	–	GPIO4_CFG[2:0]			–	GPIO3_CFG[2:0]		
Reset	–	0b101			–	0b101		
Access Type	–	Write, Read			–	Write, Read		

ビットフィールド	ビット	説明	デコード
GPIO4_CFG	6:4	GPIO4ピンの動作をプログラムします。	000 = General Purpose input 001 = SAFEMON Slave input 010 = General Purpose HI output 011 = General Purpose LO output 100 = SAFEMON one-shot HI output (idles LO) 101 = SAFEMON one-shot LO output (idles HI, default) 110 = SAFEMON active HI output (idles LO) 111 = SAFEMON active LO output (idles HI)
GPIO3_CFG	2:0	GPIO3ピンの動作をプログラムします。	000 = General Purpose input 001 = Reserved (idles as a high impedance input with 2MΩ impedance to ground) 010 = General Purpose HI output 011 = General Purpose LO output 100 = SAFEMON one-shot HI output (Idles LO) 101 = SAFEMON one-shot LO output (idles HI, default) 110 = SAFEMON active HI output (idles LO) 111 = SAFEMON active LO output (idles HI)

CONFIG_WD0 (R/W = 0x85/0x84)

BIT	7	6	5	4	3	2	1	0
Field	WD_OPN[3:0]				WD_CLO[3:0]			
Reset	0x0				0x0			
Access Type	Write, Read				Write, Read			

ビットフィールド	ビット	説明	デコード
WD_OPN	7:4	ウォッチドッグ・オープン・ウィンドウ このレジスタは、有効な更新後オープン・ウィンドウが始まるまでのウォッチドッグ・クロック・サイクル数である t_{WD1} を設定します。 注: ウォッチドッグ動作が有効化されている ($WD_EN = 1$) 間にこのレジスタを設定すると、予期せぬ動作が生じる可能性があります。	$t_{WD1} = t_{WDCLK} * (WD_OPN[3:0] + 1) * 8$

ビットフィールド	ビット	説明	デコード
WD_CLO	3:0	ウォッチドッグ・クローズ・ウィンドウ このレジスタは、有効な更新後クローズド・ウィンドウが始まるまでの時間(つまり、クローズド・ウィンドウとオープン・ウィンドウの和)である、t_{WD2}を設定します。 注:ウォッチドッグ動作が有効化されている(WD_EN = 1)間にこのレジスタを設定すると、予期せぬ動作が生じる可能性があります。	$t_{WD2} = t_{WD1} + t_{WDCLK} * (WD_CLO[3:0] + 1) * 8$

CONFIG_WD1 (R/W = 0x87/0x86)

BIT	7	6	5	4	3	2	1	0
Field	WD_1UD[2:0]			WD_DIV[4:0]				
Reset	0x0			0x0				
Access Type	Write, Read			Write, Read				

ビットフィールド	ビット	説明	デコード
WD_1UD	7:5	最初の更新延長 このレジスタは、WD_EN後に通常のt_{WD2}になるまでの追加t_{WD2}サイクル数を設定します。 注:ウォッチドッグ動作が有効化されている(WD_EN = 1)間にこのレジスタを設定すると、予期せぬ動作が生じる可能性があります。	$t_{1STWD2} = t_{WD2} * (WD_1UD[2:0] + 1)$
WD_DIV	4:0	ウォッチドッグ・クロック分周器 注:ウォッチドッグ動作が有効化されている(WD_EN = 1)間にこのレジスタを設定すると、予期せぬ動作が生じる可能性があります。	$t_{WDCLK} = (WD_DIV[4:0] + 1) * 256\mu s$

CONFIG_WD2 (R/W = 0x89/0x88)

BIT	7	6	5	4	3	2	1	0
Field	WD_EN	—	—	—	WD_SWW	WD_DBNC[2:0]		
Reset	0x0	—	—	—	0x0	0x0		
Access Type	Write, Read	—	—	—	Write, Read	Write, Read		

ビットフィールド	ビット	説明	デコード
WD_EN	7	ウォッチドッグ・イネーブル	0 = Disabled 1 = Enabled
WD_SWW	3	単純なウィンドウ化ウォッチドッグ・イネーブル	0 = Challenge/Response Watchdog Enabled 1 = Standard Windowed Watchdog Enabled

ビットフィールド	ビット	説明	デコード
WD_DBNC	2:0	ウォッチドッグ・バウンス防止 このレジスタは、安全モニタリング・モードに入るまでに発生する必要のある連続ウォッチドッグ違反の数を制御します。 注:ウォッチドッグ動作が有効化されている(WD_EN = 1)間にこのレジスタを設定すると、予期せぬ動作が生じる可能性があります。	000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 110 = 64 111 = 128

ALRTPCKTBUF_RD_PTR (R/W = 0x8D/0x8C)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	ALRTPCKTBUF_RD_PTR[2:0]		
Reset	–	–	–	–	–	0x0		
Access Type	–	–	–	–	–	Write, Read, Ext		

ビットフィールド	ビット	説明
ALRTPCKTBUF_RD_PTR	2:0	アラート・パケット・バッファ読出しポインタ アラート・パケット・バッファ内のホストが読出しを行う場所。このレジスタは読出し／書込み可能であり、ALRTPCKTBUF_RD_MSGがアサートされると自動的にインクリメントします。

ALRTPCKTBUF_RD_MSG (R/W = 0x8F/0x8E)

BIT	7	6	5	4	3	2	1	0
Field	ALRTPCKTBUF_RD_MSG[7:0]							
Reset	0x00							
Access Type	Read Only							

ビットフィールド	ビット	説明
ALRTPCKTBUF_RD_MSG	7:0	ALRTPCKTBUF_RD_MSGコマンド アドレスALRTPCKTBUF_RD_PTRを始点として受信バッファを読み出します。バイトが読み出された後に読出しポインタを自動的にインクリメントしますが、次のメッセージへの読出しポインタはインクリメントしません。

RX_RD_MSG (R/W = 0x91/0x90)

BIT	7	6	5	4	3	2	1	0
Field	RX_RD_MSG[7:0]							
Reset	0x00							
Access Type	Read Only							

ビットフィールド	ビット	説明
RX_RD_MSG	7:0	RX_RD_MSGコマンド アドレスRX_RD_PTRを始点として受信バッファを読み出します。バイトが読み出された後に読出しポインタを自動的にインクリメントしますが、次のメッセージへの読出しポインタはインクリメントしません。 注: SPI読出しまたは書込みバーストをこのアドレスに対し実行すると、RX_RD_PTRレジスタがインクリメントしますが、内部のSPIアドレス・ポインタはインクリメントしません。

RX_RD_NXT_MSG (R/W = 0x93/0x92)

BIT	7	6	5	4	3	2	1	0
Field	RX_RD_NXT_MSG[7:0]							
Reset	0x00							
Access Type	Read Only							

ビットフィールド	ビット	説明
RX_RD_NXT_MSG	7:0	RX_RD_NXT_MSGコマンド アドレスRX_NXT_MSG_PTR(最も古い未読メッセージ)を始点として受信バッファを読み出します。バイトが読み出された後に読出しポインタを自動的にインクリメントしますが、次のメッセージへの読出しポインタはインクリメントしません。 注: SPI読出しまたは書込みバーストをこのアドレスに対し実行すると、RX_NXT_MSG_PTRレジスタがインクリメントしますが、内部のSPIアドレス・ポインタはインクリメントしません。

TX_QUEUE_SEL (R/W = 0x95/0x94)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	TX_Q[1:0]		LD_Q[1:0]	
Reset	–	–	–	–	0x00		0x00	
Access Type	–	–	–	–	Read Only		Read Only	

ビットフィールド	ビット	説明
TX_Q	3:2	送信キュー選択 UARTがメッセージ伝送(送信)用に選択した送信バッファの4つのキューの1つをアドレス指定します。
LD_Q	1:0	ロード・キュー選択 ホストがメッセージ・ロード(書込み)用に選択した送信バッファの4つのキューの1つをアドレス指定します。

RX_RD_PTR (R/W = 0x97/0x96)

BIT	7	6	5	4	3	2	1	0
Field	–	RX_RD_PTR[6:0]						
Reset	–	0x00						
Access Type	–	Read Only						

ビットフィールド	ビット	説明
RX_RD_PTR	6:0	受信バッファ読出しポインタ 受信バッファ内のホストが読出しを行う場所。UARTはこのポインタを自動的にインクリメントします。

RX_WR_PTR (R/W = 0x99/0x98)

BIT	7	6	5	4	3	2	1	0
Field	–	RX_WR_PTR[6:0]						
Reset	–	0x01						
Access Type	–	Read Only						

ビットフィールド	ビット	説明
RX_WR_PTR	6:0	受信バッファ書き込みポインタ データ受信時にUARTによって書き込まれる受信バッファ内の場所。

RX_NXT_MSG_PTR (R/W = 0x9B/0x9A)

BIT	7	6	5	4	3	2	1	0
Field	–	RX_NXT_MSG_PTR[6:0]						
Reset	–	0x00						
Access Type	–	Read Only						

ビットフィールド	ビット	説明
RX_NXT_MSG_PTR	6:0	受信バッファ次メッセージ・ポインタ 受信バッファ内の次の未読メッセージの先頭。RX_RD_Pointerには、RD_NXT_MSG SPIランザクションによってこの値がロードされます。

RX_SPACE (R/W = 0x9D/0x9C)

BIT	7	6	5	4	3	2	1	0
Field	–	RX_SPACE[6:0]						
Reset	–	0x56						
Access Type	–	Read Only						

ビットフィールド	ビット	説明
RX_SPACE	6:0	RXスペース・レジスタ 受信バッファでアクセス可能なバイトの数。

RX_BYTE (R/W = 0x9F/0x9E)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	RX_FIRST_BYTE	RX_BYTE_ERR	RX_LAST_BYTE
Reset	–	–	–	–	–	0x0	0x0	0x0
Access Type	–	–	–	–	–	Read Only	Read Only	Read Only

ビットフィールド	ビット	説明
RX_FIRST_BYTE	2	RXバッファ先頭バイト・インジケータ RX_RD_PTRの位置にあるバイトがメッセージの最初のデータ・バイトです(対応するキャラクタの前にはプリアンブル・キャラクタが置かれます)。
RX_BYTE_ERR	1	RXバイト・エラー・インジケータ RX_RD_PTRにあるデータ・バイトにエラー(マンチェスターやパリティのエラーが含まれる、対応するキャラクタ)が含まれている可能性があります。
RX_LAST_BYTE	0	RXパケット最終バイト・インジケータ RX_RD_PTRの位置にあるバイトがメッセージの最後のバイトです(対応するキャラクタはストップ・キャラクタであり、ヌル・バイトとして格納されます)。

NXT_LDQ (R/W = 0xB1/0xB0)

BIT	7	6	5	4	3	2	1	0
Field	NXT_LDQ[7:0]							
Reset	0x00							
Access Type	Write, Read, Ext							

ビットフィールド	ビット	説明
NXT_LDQ	7:0	NXT_LDQコマンド このコマンドは、LDQをインクリメントし、その後に送信バッファ・ロード・キューを書き込みます。 ホストがデータをロードしてもしなくてもインクリメントは発生します。コマンド・バイトは、最初の書き込み場所を指定します(位置0～30)。位置30を越えるバースト書き込みは無効です。 注: SPI読み出しまたは書き込みバーストをこのアドレスに対し実行すると、LDQ_PTRレジスタがインクリメントしますが、内部のSPIアドレス・ポインタはインクリメントしません。

LDQ (R/W = 0x1/0x0)

BIT	7	6	5	4	3	2	1	0
Field	LDQ[7:0]							
Reset	0x00							
Access Type	Write, Read, Ext							

ビットフィールド	ビット	説明
LDQ	7:0	LDQコマンド 送信バッファ・ロード・キューの読み出し／書き込みを行います。コマンド・バイトは、書き込む最初のバイトを指定します(位置0～30)。位置30を越えるバースト書き込みは無効です。 注: SPI読み出しまたは書き込みバーストをこのアドレスに対し実行すると、LDQ_PTRレジスタがインクリメントしますが、内部のSPIアドレス・ポインタはインクリメントしません。

LDQ_PTR (R/W = 0xC1/0xC0)

BIT	7	6	5	4	3	2	1	0
Field	—	—	—	LDQ_PTR[4:0]				
Reset	—	—	—	0x0				
Access Type	—	—	—	Write, Read, Ext				

ビットフィールド	ビット	説明
LDQ_PTR	4:0	送信バッファ読み出し／書き込みポインタ 送信LDQバッファ内でホストが読み出しまたは書き込みを行う場所。このポインタは、LDQレジスタを読み出した場合やLDQレジスタに書き込んだ場合に自動的にインクリメントし、NXT_LDQ書き込みコマンドでクリアされます。

CONFIGQ (R/W = 0xD1/0xD0)

BIT	7	6	5	4	3	2	1	0
Field	CONFIGQ[7:0]							
Reset	0xFF							
Access Type	Write, Read, Ext							

ビットフィールド	ビット	説明
CONFIGQ	7:0	設定データ・キュー・バイト CONFIG_BYTE_PTRレジスタおよびCONFIG_QUEUE_PTRレジスタの指示に従い、設定データ・キュー・バイトの読出しおよび書込みを行います。 注: SPI読出しまたは書込みバーストをこのアドレスに対し実行すると、CONFIG_PTRレジスタがインクリメントしますが、内部のSPIアドレス・ポインタはインクリメントしません。

CONFIG_PTR (R/W = 0xD3/0xD2)

BIT	7	6	5	4	3	2	1	0
Field	–	CONFIG_QUEUE_PTR[1:0]		CONFIG_BYTE_PTR[4:0]				
Reset	–	0x00		0x00				
Access Type	–	Write, Read, Ext		Write, Read, Ext				

ビットフィールド	ビット	説明
CONFIG_QUEUE_PTR	6:5	設定キュー・ポインタ この読出し／書込みレジスタは、CONFIGQレジスタの読出し／書込みに用いるキュー・ポインタを設定します。 CONFIGQレジスタにアクセスするとこのレジスタは自動インクリメントします。また、このレジスタは`3`から`0`にロール・オーバーします。

ビットフィールド	ビット	説明
CONFIG_BYTE_PTR	4:0	設定バイト・ポインタ この読み出し／書き込みレジスタは、CONFIGQレジスタの読み出し／書き込みに用いるバイト・ポインタを設定します。 CONFIG_QUEUE_PTRに'd0～'の値が含まれている場合、'd30以上の無効な値を書き込むと、その書き込みは無視されます。 CONFIG_QUEUE_PTRに'd3の値が含まれている場合、'd6以上の無効な値を書き込むと、その書き込みは無視されます。 CONFIGQレジスタにアクセスするとこのレジスタは自動インクリメントします。また、このレジスタはロール・オーバーし、アクセスがあった場合、CONFIG_QUEUE_PTRレジスタを自動インクリメントします。

STATE (R/W = 0xDD/0xDC)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	STATE[2:0]		
Reset	–	–	–	–	–	0x0		
Access Type	–	–	–	–	–	Read Only		

ビットフィールド	ビット	説明
STATE	2:0	このレジスタは、MAX17851の現在の状態を示します。 このレジスタは、システム・レベルのデバッグを目的とするものです。

COMM_RTRY_CNT (R/W = 0xDF/0xDE)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	COMM_RTRY_CNT[4:0]				
Reset	–	–	–	0x0				
Access Type	–	–	–	Read Only				

ビットフィールド	ビット	説明
COMM_RTRY_CNT	4:0	通信リトライ・カウンタ このレジスタは、BMS安全モニタリング・モードまたはスリープ・モード時に、COMM_RTRYに関連する最新のRETRYカウンタ・データを提供します。 このレジスタはシステム・レベルのデバッグに有用です。

ALRTPCKT_ERR_CNT (R/W = 0xE1/0xE0)

BIT	7	6	5	4	3	2	1	0
Field	ALRTPCKT_ERR_CNT[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
ALRTPCKT_ERR_CNT	7:0	このレジスタは、BMS安全モニタリング・モードまたはスリープ・モード時に、ALRTPCKT_DBNCに関連する最新のカウンタ・データを提供します。 このレジスタはシステム・レベルのデバッグに有用です。

WD_FAULT_CNT (R/W = 0xE3/0xE2)

BIT	7	6	5	4	3	2	1	0
Field	WD_FAULT_CNT[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
WD_FAULT_CNT	7:0	ウォッチドッグ・フォルト・カウンタ – 発生したウォッチドッグ・フォルトの数を示します。 このレジスタは、WD_ENが無効化された場合、または有効なウォッチドッグ更新が生じた場合に、リセットされます。 このレジスタはシステム・レベルのデバッグに有用です。

ALIVECOUNT_SEED (R/W = 0xE5/0xE4)

BIT	7	6	5	4	3	2	1	0
Field	ALIVECOUNT_SEED[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
ALIVECOUNT_SEED	7:0	これは、自動アライブ・カウンタ用のシード値です。アライブ・カウンタが有効化されている場合、このシードがUARTトランザクションに付加され、送信が完了すると値が1だけインクリメントします。

ALIVECOUNT_RET (R/W = 0xE7/0xE6)

BIT	7	6	5	4	3	2	1	0
Field	ALIVECOUNT_RET[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
ALIVECOUNT_RET	7:0	自動アライブ・カウンタが有効化されている場合、この読み専用レジスタは、トランザクションごとにMAX17851が返した受信アライブ・カウンタ・バイトを格納します。

ALIVECOUNT_Q (R/W = 0xE7/0xE6)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	ALIVECOUNT_Q[2:0]		
Reset	–	–	–	–	–	0x0		
Access Type	–	–	–	–	–	Read Only		

ビットフィールド	ビット	説明
ALIVECOUNT_Q	2:0	この読み専用レジスタは、TXバッファから送信されながらもRXバッファでは未受信のトランザクションの数を格納します。 このレジスタはCLR_LSSMコマンドによってリセットされます。

FAULT_TIMER0 (R/W = 0xEB/0xEA)

フォルト・タイマーのLSB(秒)

BIT	7	6	5	4	3	2	1	0
Field	FAULT_TIMER[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
FAULT_TIMER	7:0	フォルト・タイマー (LSB) は、通信フォルトあるいはデジタイゼーション・フォルト (ALERTPACKETにより示される) がスリープ・モードまたはBMS安全モニタリング・モードで発生してから経過時間 (秒) を示します。 フォルト・タイマーは、フォルトがバウンス防止されると開始し、SLP_ALERTビットまたはSAFEMON_ALERTビットがクリアされると停止します。 フォルト・タイマーは、スリープ・モードまたはBMS安全モニタリング・モードの開始時にリセットされます。

FAULT_TIMER1 (R/W = 0xED/0xEC)

フォルト・タイマーのMSB(秒)

BIT	7	6	5	4	3	2	1	0
Field	FAULT_TIMER[15:8]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
FAULT_TIMER	7:0	フォルト・タイマー (MSB) は、通信フォルトあるいはデジタイゼーション・フォルト (ALERTPACKETにより示される) がスリープ・モードまたはBMS安全モニタリング・モードで発生してからの経過時間 (秒) を示します。 フォルト・タイマーは、フォルトがバウンス防止されると開始し、SLP_ALRTビットまたはSAFEMON_ALRTビットがクリアされると停止します。 フォルト・タイマーは、スリープ・モードまたはBMS安全モニタリング・モードの開始時にリセットされます。

SLP_CBTIMER0 (R/W = 0xEF/0xEE)

BIT	7	6	5	4	3	2	1	0
Field	SLP_CBTIMER[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
SLP_CBTIMER	7:0	スリープ・モードのセル・バランシング・タイマー・リードバックのLSB(秒)

SLP_CBTIMER1 (R/W = 0xF1/0xF0)

BIT	7	6	5	4	3	2	1	0
Field	SLP_CBTIMER[15:8]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
SLP_CBTIMER	7:0	スリープ・モードのセル・バランシング・タイマー・リードバックのMSB(秒)

VERSION (R/W = 0xF3/0xF2)

BIT	7	6	5	4	3	2	1	0
Field	MODEL[3:0]				VERSION[3:0]			
Reset	0x1							
Access Type	Read Only				Read Only			

ビットフィールド	ビット	説明
MODEL	7:4	モデル番号の末尾の桁
VERSION	3:0	マスクのリビジョン

MODEL (R/W = 0xF5/0xF4)

BIT	7	6	5	4	3	2	1	0
Field	MODEL[11:4]							
Reset	0x85							
Access Type	Read Only							

ビットフィールド	ビット	説明
MODEL	7:0	モデル番号の最初の2桁

アプリケーション情報

システム構成

MAX17851は主として、次の2種類のシステム構成をサポートします。すなわち、シングルUARTとデュアルUARTです。シングルUARTの場合、MAX17851はマスタとしてのみ設定されます。デュアルUARTの場合は、複数のMAX17851デバイスがマスタ/スレーブ設定で設定されます。ハードウェアがどちらのシステム用にセットアップされているかによらず、ホスト・マイクロコントローラは、デバイス設定、BMSデータ・アクイジション・デバイスの初期化、設定メモリ、機能動作パラメータに関し、同じシーケンスに従うことが推奨されます。

次のセクションでは、BMSデータ・アクイジション・システムの全てのデバイスを設定するために従うことができる手順を、順番に詳しく説明します。

デュアルUART動作

デュアルUARTアプリケーションでは、MAX17851をマスタ/スレーブ設定でセットアップできるため、柔軟性とスループットが向上できます。このアプリケーションでは、2つのUART通信経路があるため、1つの通信経路に問題が生じていてもUART通信を続けることができます。CONFIG_GEN4レジスタのMS_ENフィールドを用いて、2つのMAX17851デバイスのうちの1つがマスタ・デバイス、1つがスレーブ・デバイスに設定されます。

スレーブ・デバイスの受信データは、マスタ・デバイスの受信データとは逆順になっています。スレーブ・デバイスは、CONFIG_GEN4レジスタのRXSWAP_ENビットを設定する必要があります。これによって、BMS UARTデバイスが受信するデータが、マスタのMAX17851の受信データと一致するよう変更されます。

GPIO1およびGPIO2は、マスタおよびスレーブのどちらのMAX17851デバイスに対しても出力として設定されます。これらのGPIOピンはどちらも、SAFEMON出力として設定する必要があります。出力は、アクティブ・ハイにもアクティブ・ローにも設定できます。GPIO1出力とGPIO2出力には、何らかの絶縁が必要です。[図20](#)に、GPIO1のダイオード、GPIO2の抵抗を示します。これにより、どちらかのGPIO出力が不能となっても、他方がバッテリー・コンタクト信号を安全にアサートできます。

マスタMAX17851のGPIO3はSAFEMONアクティブ・ハイ出力として設定し、スレーブ・デバイスのGPIO4はSAFEMONスレーブ入力に設定する必要があります。これにより、BMS安全モニタリング・モードになったときにマスタ・デバイスがスレーブにアラートを発することができます。

[図20](#)に、マイクロコントローラのリセット入力および電源イネーブル入力に接続されたその他のGPIOピン(マスタ・デバイスのGPIO4、スレーブ・デバイスのGPIO3)を示します。これらのGPIOピンは、必要な信号の極性に応じて、SAFEMONワンショットHI出力またはSAFEMONワンショットLO出力として設定できます。ワンショット出力は100ms間のアサート後デアサートするため、接続したデバイスはパワーオン・リセット・シーケンスを経ることができます。GPIO設定の詳細については、[GPIO制御](#)のセクションを参照してください。

デュアルUART設定では、BMS UARTデジタイゼーション・デバイスからデータを読み出す場合のスループットを上げることも可能です。例えば、マスタのMAX17851を奇数番号のデバイスを読み出すよう設定し、スレーブのMAX17851を偶数番号のデバイスを読み出すように設定できます。2つのSPIインターフェース(各MAX17851につき1つ)からデータを取得するのに必要なデータ・スループットをホスト・マイクロコントローラが確実に処理できるよう、注意することが必要です。

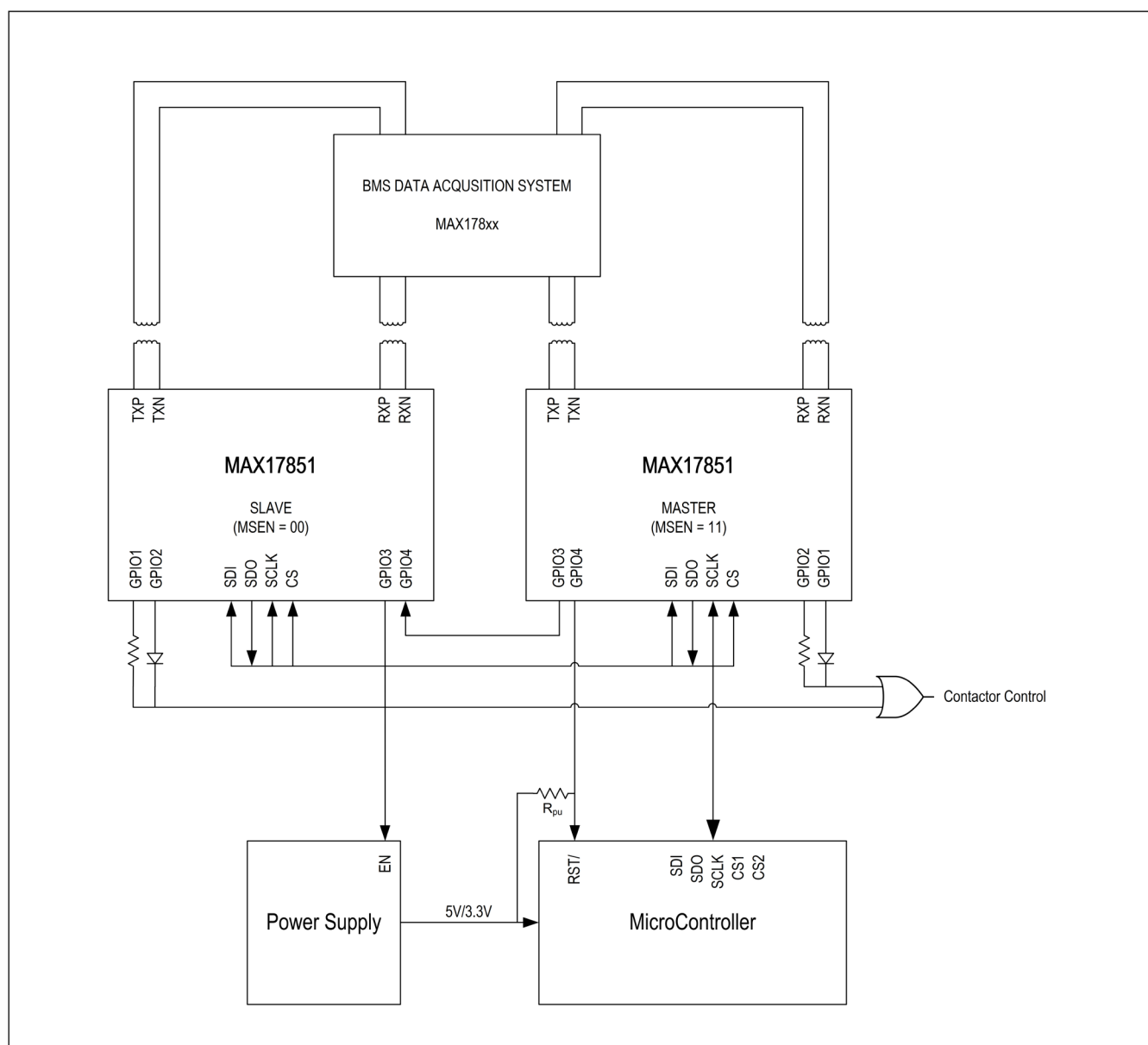


図20. デュアルUART

デュアルUART MAX17851の初期化

表17に示す例は、MAX17851デバイスをマスタ設定とスレーブ設定に設定するためのシーケンスの詳細を表しています。このシーケンスに加え、ホストはレジスタ・データをリードバックすることによって、設定レジスタへの全ての書き込みを常に検証できます。これにより検証のレイヤがもう1つ加わることになります。

表17. マスタ/スレーブ・デバイスの設定シーケンス

SPI DIN	SPI DOUT	ターゲット・ デバイス	説明
TRANSACTION 1		Master and Slave	CONFIG_GEN0を設定します。
60h	xxh		デバイス・カウントを書き込みます。
07h	xxh		デバイス・カウントを7に設定します(実際のデジタイゼーション・カウントに応じて変更)。
TRANSACTION 2		Master and Slave	CONFIG_GEN1を設定します。
62h	xxh		ポー・レートを設定します。
30h	xxh		ポー・レート = 2Mbpsに設定します。
TRANSACTION 3		Master	CONFIG_GEN4を設定します。
68h	xxh		RXSWAPENおよびMS_ENを設定します。
38h	xxh		マスタのデュアルUARTを通常のデータ・シーケンスで有効化します。
TRANSACTION 4		Slave	CONFIG_GEN4を設定します。
68h	xxh		RXSWAPENおよびMS_ENを設定します。
40h	xxh		スレーブのデュアルUARTをマスタと同じデータ・シーケンスで有効化します。
TRANSACTION 5		Master and Slave	CONFIG_GEN5を設定します。
6Ah	xxh		ALRTPCKT_DBNCを設定します。
08h	xxh		エラーがフラグ通知されるまでにBMS安全モニタリング・モードまたはスリープ・モードの2連続エラーを有効化します。
TRANSACTION 6		Master and Slave	CONFIG_SAFEMON0を設定します。
6Ch	xxh		GPIOREC_DLYを設定します。
5Ah	xxh		GPIO1、2のアサートとGPIO3、4のアサートの間に1秒の遅延を設定します。BMS安全モニタリング・モードでGPIO安全対策を実行します。
TRANSACTION 7		Master and Slave	CONFIG_SAFEMON1を設定します。
6Eh	xxh		CONT_TIMER_DLYを設定します。
FEh	xxh		通信エラー(断線または永続的な通信フォルト)の発生時、信号をオープン・コンタクトに伝達するまでに1032分の遅延を設けます。
TRANSACTION 8		Master and Slave	CONFIG_SAFEMON2を設定します。
70h	xxh		SAFEMON_SCAN_DLYを設定します。
01h	xxh		BMSデータ・アクイジション・システムの1秒間のフォルト・ポーリングを有効化します。
TRANSACTION 9		Master and Slave	CONFIG_SAFEMON3を設定します。
72h	xxh		設定します。
0Eh	xxh		BMS安全モニタリング・モードに初めて入るときにGPIO1、GPIO2、GPIO3、GPIO4をマスクします。GPIO安全対策を実行する時間を用意します。
TRANSACTION 10		Master and Slave	CONFIG_SLPを設定します。
74h	xxh		SLP_SCAN_DLY、SLP_ALRTPCKTEN、SLP_CBNTFYを設定します。
1Fh	xxh		スリープ・モードになりマイクロコントローラへの割込みによる応答が行われるときの、BMSデータ・アクイジション・システムの1秒間の自動ポーリングを有効化します。

表17. マスタ/スレーブ・デバイスの設定シーケンス (続き)

SPI DIN	SPI DOUT	ターゲット・ デバイス	説明
TRANSACTION 11			
76h	xxh	Master and Slave	CONFIG_COMMを設定します。
			COMM_RTRY、COMM_TO_DLYを設定します。
02h	xxh		デジタイゼーション = 7の場合、READALLの時間設定は251μsです。 COMM_TO_DLYはメッセージの時間設定より30%長くすることを推奨します。
TRANSACTION 12			
		Master and Slave	GPIO12を設定します。
80h	xxh		GPIO1_CFGおよびGPIO2_CFGを設定します。
66h	xxh		NMOS入力用のコンタクト・ドライバを有効化します (BMS安全モニタリング・モードのみ)。
TRANSACTION 13			
		Master	GPIO34を設定します。
82h	xxh		GPIO3_CFGおよびGPIO4_CFGを設定します。
46h	xxh		BMS安全モニタリング・モードでのマスタ/スレーブの通信トランザクションを有効化します。 マイクロコントローラのアクティブ・ハイ・リセットを有効化します。
TRANSACTION 14			
		Slave	GPIO34を設定します。
82h	xxh		GPIO3_CFGおよびGPIO4_CFGを設定します。
16h	xxh		BMS安全モニタリング・モードでのマスタ/スレーブの通信トランザクションを有効化します。 マイクロコントローラの電源アクティブ・ハイ・リセットを有効化します。
TRANSACTION 15			
		Master and Slave	ALRTEN_OPSTATEを設定します。
28h	xxh		SLP_ALRTEN、SLP_STATUS_ERR_ALRTEN、SAFEMON_ALRTEN、 SAFEMON_GPIO12_ALRTEN、SAFEMON_STATUS_ERR_ALRTEN、 SAFEMON_CONFIG_ERR_ALRTENを設定します。
3Fh	xxh		設定されたスリープ・モードおよびBMS安全モニタリング・モードのアラートに対するアラート・ピンのアサートを有効化します。
TRANSACTION 16			
		Master and Slave	ALRTEN_LSSM_BYTEを設定します。
24h	xxh		RX_READY_ALRTENを設定します。
80h	xxh		LSSMステータス・バイトのRX_READYアラートに対するアラート・ピンのアサートを有効化します。
TRANSACTION 17			
		Master and Slave	ALRTEN_RXを設定します。
20h	xxh		RX_STOP_ALRTENを設定します。
02h	xxh		RX_STOPに対するアラート・ピンのアサートを有効化します。
TRANSACTION 18			
		Master and Slave	ALRTEN_WDを設定します。
2Ch	xxh		WD_TO_ERR_ALRTENを設定します。
10h	xxh		ウォッチドッグ・タイムアウトに対するアラート・ピンのアサートを有効化します。 これにより、ホスト・マイクロコントローラとMAX17851の間の同期が可能になります。
TRANSACTION 19			
		Master and Slave	受信バッファ・エラーがないかチェックします。
11h	xxh		ALERT_RXレジスタのフラグを読み出します。
xxh	00h		エラーがなければ続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。

マスタ/スレーブUARTデバイスのパラメータが設定された後、ホスト・マイクロコントローラはBMSデータ・アキュイジション・システムを初期化すると共に設定メモリ・シーケンスを実行し、BMSデジタイゼーション・デバイスを設定できます。詳細については、[デュアルUARTのBMSデータ・アキュイジション・システムの初期化](#)のセクションおよび[デジタイゼーション設定用の設定メモリ・シーケンス](#)のセクションを参照してください。

デュアルUARTのBMSデータ・アキュイジション・システムの初期化

最初のパワーアップ時、またはデバイスのリセット後、MAX17851は正しい動作を実現するために、初期化シーケンスを実行してBMSデジタイゼーションを設定する必要があります。初期化シーケンスに先立ち、ホストは、ハードウェアが正しく設定されていることと、デバイス・カウント、ポー・レート、キープアライブ設定などの特定のパラメータが正しくプログラムされていることを検証する必要があります。

また、ホストがCONFIG_GENレジスタの必要な設定の書き込みやリードバックを行って、デバイス設定レジスタの内容を検証することも推奨します。

初期化シーケンスを実行する前に、ホストは、一般設定レジスタCONFIG_GEN4のMS_EN[5:4]に書き込みを行って、システム・セフトアップを設定する必要があります。デフォルト設定は、マスタでシングルUARTです。ハードウェアがマスタ/スレーブUART用に設定されている場合、スレーブ・デバイスは、受信設定に対し正しいデータとなるよう設定し、デバイスがデータを正しく解釈できるようにする必要があります (スレーブ・デバイスで受信するデータは逆順になっているため)。これを行うには、CONFIG_GEN4レジスタの[RXSWAP_EN[6]に書き込みを行います。

必要なUARTコマンドをBMS安全モニタリング・モードで実行するよう設定メモリをプログラムする方法の詳細については、[デジタイゼーション設定用の設定メモリ・シーケンス](#)のセクションを参照してください。

デバイス・パラメータが設定されると、ホストはBMSデータ・アキュイジション・システムを初期化する必要があります。

ホストは、デジタイゼーション内の正しいデバイス数を決定するHELLOALLシーケンスを正常に実行した後、動作を続け、コマンド動作モードとそれに続く通常動作モードでのアラート・パケット生成を有効化できます。アラート・パケットは、初期化シーケンスの前にCONFIG_GENレジスタで有効化できます。ただし、初期化シーケンスの間は、アラート・パケット生成はマスクされており、キープアライブ・パターンを用いて堅牢なシステム性能が実現されます。アラート・パケット生成の頻度は、CONFIG_GEN3レジスタのALRTPCKT_TIMING[3:0]で設定できます。アラート・パケット生成には、デジタイゼーション通信およびエラーをホストが介入する必要なく自律的にチェックできるというシステム上の利点があります。

ホストがHELLOALLシーケンス後にデバイス・カウント・エラーを受信した場合、シーケンスを再度実行するか、ハードウェア設定をチェックする必要があります。

表18で説明する例は、2つのUARTスレーブ・デバイスの初期化シーケンスの概要を示すものです。

表18. マスタ/スレーブのBMSデジタイゼーションの初期化シーケンス

SPI DIN	SPI DOUT	ターゲット・デバイス	説明
TRANSACTION 1		Master and Slave	CONFIG_GEN3を設定します。
66h	xxh		ALRTPCKT_TIMINGを設定します。
05h	xxh		キープアライブ時間を160μsに設定します。
TRANSACTION 2		Master and Slave	ALRTEN_RXを設定します。
20h	xxh		RX_ERR_ALRTENおよびRX_OVRFLW_ERR_ALRTENを設定します。
88h	xxh		RX_ERR_ALRTENビットおよびRX_OVRFLW_ERR_ALRTENビット (ALERTピン・イネーブル・ビット)を有効化します。
TRANSACTION 3		Master and Slave	CONFIG_GEN2を設定します。
64h	xxh		TX_PREAMBLESを設定します。
30h	xxh		プリアンブルの送信を有効化します (これによりデジタイゼーション・デバイスがウェイクアップします)。
TRANSACTION 4		Master and Slave	STATUS_RXを読み出します。
01h	xxh		RX_BUSYおよびRX_EMPTYをポーリングします。
xxh	21h		RX_STATUS = 21hの場合、続行します。それ以外の場合、真またはタイムアウトになるまでトランザクションを繰り返します。
TRANSACTION 5		Master and Slave	CONFIG_GEN2を設定します。
64h	xxh		TX_PREAMBLESおよびTX_QUEUE送信を設定します。
10h	xxh		プリアンブルの送信を無効化します。
TRANSACTION 6		Master and Slave	CLR_RXBUF

表18. マスタ/スレーブのBMSデ이지チェーンの初期化シーケンス (続き)

SPI DIN	SPI DOUT	ターゲット・ デバイス	説明
42h	xxh		受信バッファをクリアしリセットします。
00h	xxh		受信バッファをクリアしリセットします。
TRANSACTION 7		Master and Slave	CLR_TXBUF
40h	xxh		送信バッファをクリアしリセットします。
00h	xxh		送信バッファをクリアしリセットします。
TRANSACTION 8		Master	HELLOALLコマンド・シーケンスをロード・キューにロードします。
C0h	xxh		LD_Q SPIコマンド・バイトを書き込みます (ロード・キューを書き込みます)。
03h	xxh		メッセージ長。
57h	xxh		HELLOALLコマンド・バイト。
00h	xxh		レジスタ・アドレス (0x00)。
00h	xxh		HELLOALLの初期化アドレス。
TRANSACTION 9		Master	RX_RD_MSGを読み出します (オプション)。
C2h	xxh		ロード・キューの内容を検証します。
00h	xxh		ポインタを読み出し対象のロード・キューに変更します。
TRANSACTION 10		Master	バッファの内容を読み出します (オプション)。
C1h	xxh		ロード・キューのアドレスを読み出します。
xxh	03h		メッセージ長。
xxh	57h		HELLOALLコマンド・バイト。
xxh	00h		レジスタ・アドレス。
xxh	00h		HELLOALLの初期化アドレス。
TRANSACTION 11		Master	HELLOALLシーケンスを送信します。
B0h	xxh		NXT_LDQ SPIコマンド・バイト (次のロード・キューを書き込みます)。
TRANSACTION 12		Master	STATUS_RXを読み出します。
01h	xxh		RX_STOPステータス・ビットをポーリングします。
xxh	x2h		RX_Status[1]が真の場合、続行します。偽の場合、真になるまでトランザクションを繰り返します。
TRANSACTION 13		Master	RX_RD_NXT_MSGを読み出します。
93h	xxh		受信バッファを処理します。デ이지チェーンを通じて伝搬されたHELLOALLメッセージを読み出します。 ホストはデバイス・カウントを検証する必要があります。
xxh	57h		コマンド・バイト (HELLOALL)を送信します。
xxh	00h		アドレス = 00hを送信します。
xxh	02h		受信アドレス = 02h (2デバイス)
xxh	84h		LSSM_BYTE
TRANSACTION 14		Master	ALERT_RXを読み出します。
11h	xxh		受信バッファ・エラーがないかチェックします。
xxh	00h		エラーがなければ続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。
TRANSACTION 15		Slave	BMSデータ・アキュイジション・システムを下りホストとして設定します (トランザクション14、15、16はオプションですが、DOWNHOSTの場合には使用する必要があります)。
C0h	xxh		LD_Q SPIコマンド・バイトを書き込みます (ロード・キューを書き込みます)。
03h	xxh		メッセージ長。
09h	xxh		下りホスト

表18. マスタ/スレーブのBMSデ이지チェーンの初期化シーケンス (続き)

SPI DIN	SPI DOUT	ターゲット・デバイス	説明
00h	xxh		データLSB
00h	xxh		上位アドレス。
TRANSACTION 16		Slave	データを送信します。
B0h	xxh		NXT_LDQ SPIコマンド・バイト(次のロード・キューを書き込みます)。
TRANSACTION 17		Slave	ALERT_RXを読み出します。
11h	xxh		受信バッファ・エラーがないかチェックします。
xxh	00h		エラーがなければ続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。

デバイスが正常に初期化された後、ホスト・マイクロコントローラは通常動作で続行できます。ホストは、MAX17851からの自動キーアライブ生成を無効化し、代わりにデバイスにアラート・パケット生成を設定できます。アラート・パケット設定シーケンスを表22に示します。

最終的なMAX17851設定

MAX17851が正常に初期化され、エラーがない場合、ホストはMAX17851からの自動キーアライブ生成を無効化し、デバイスに自動ALERTPACKET生成を設定できます。

設定シーケンスを表19に示します。

表19. マスタ/スレーブのUARTアラート・パケット設定

SPI DIN	SPI DOUT	ターゲット・デバイス	説明
TRANSACTION 1		Master	CONFIG_GEN2を設定します。
64h	xxh		TX_PREAMBLESを設定します。
10h	xxh		プリアンブル送信モードを無効化します。
TRANSACTION 2		Master and Slave	CONFIG_GEN3を設定します。
66h	xxh		ALRTPCKT_TIMINGを設定します。
08h	xxh		アラート・パケットの時間設定を1.28msにします。
TRANSACTION 3		Master	CONFIG_GEN4マスタを設定します。
68h	xxh		CO_ALRTPCKTENを設定します。
B8h	xxh		コマンド動作モードでのアラート・パケットの時間を設定します。 スリープ・モードまたはBMS安全モニタリング・モードの時間設定は、SLP_SCAN_DLYおよびSAFEMON_SCAN_DLYで指定されます。
TRANSACTION 4		Slave	CONFIG_GEN4スレーブを設定します。
68h	xxh		CO_ALRTPCKTENを設定します。
88h	xxh		
TRANSACTION 5		Master and Slave	CONFIG_SLPを設定します。
74h	xxh		SLP_SCAN_DLY、SLP_ALRTPCKTENを設定します。
3Fh	xxh		SLP_SCAN_DLYを1秒に設定します。SLP_ALRTPCKTENは、SLP_SCAN_DLYで設定された遅延を用いるために有効化します。SLP_CBNTFYをオフに設定します(ユーザにより選択)。

シングルUART動作

MAX17851は、シングル・デ이지チェーン・システム向けにシングルUART設定で設定できます。データは常にホストからデ이지チェーン経路に流れ、その後、デ이지チェーン・デバイスからホストにループバックします。

デバイスをシングルUARTモードに設定するには、まずホストがデバイスをシングルUART設定に設定します (MS_EN = 0b10)。

図21にシングルUARTのハードウェア構成を示します。初期パワーアップ時、またはリセット時、ホストは、最初のデバイス設定のために、シングルUARTのMAX17851の初期化のシーケンスに従うことができます。

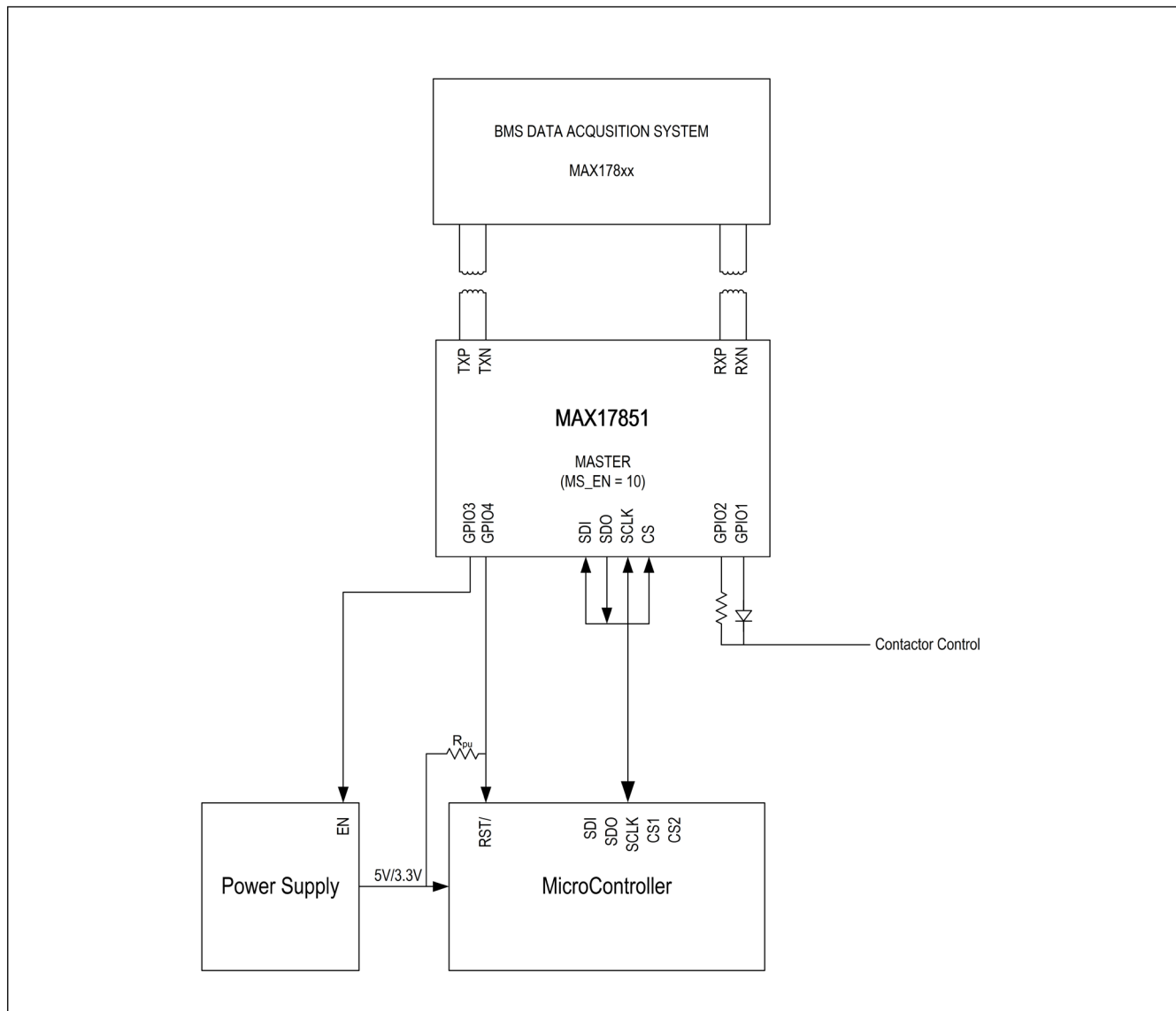


図21. シングルUART

シングルUARTのMAX17851の初期化

表20に示す例は、MAX17851をシングルUART設定に設定するためのシーケンスの詳細を表しています。このシーケンスに加え、ホストはレジスタ・データをリードバックすることによって、設定レジスタへの全ての書き込みを常に検証できます。これにより検証のレイヤがもう1つ加わることになります。

表20. シングルUARTデバイス設定シーケンス

SPI DIN	SPI DOUT	説明
TRANSACTION 1		CONFIG_GEN0を設定します。
60h	xxh	デバイス・カウントを書き込みます。
07h	xxh	デバイス・カウントを7に設定します(実際のデジタイゼーション・カウントに応じて変更)。
TRANSACTION 2		CONFIG_GEN1を設定します。
62h	xxh	ボー・レートを設定します。
30h	xxh	ボー・レート = 2Mbpsに設定します。
TRANSACTION 3		CONFIG_GEN4を設定します。
68h	xxh	MS_ENを設定します。
28h	xxh	マスタをシングルUARTで有効化します。
TRANSACTION 4		CONFIG_GEN5を設定します。
6Ah	xxh	ALRTPCKT_DBNCを設定します。
08h	xxh	エラーがフラグ通知されるまでにBMS安全モニタリング・モードまたはスリープ・モードの2連続エラーを有効化します。
TRANSACTION 5		CONFIG_SAFEMON0を設定します。
6Ch	xxh	GPIOREC_DLYを設定します。
5Ah	xxh	GPIO1、2のアサートとGPIO3、4のアサートの間に1秒の遅延を設定します。これにより、BMS安全モニタリング・モードでGPIO安全対策を実行します。
TRANSACTION 6		CONFIG_SAFEMON1を設定します。
6Eh	xxh	CONT_TIMER_DLYを設定します。
FEh	xxh	通信エラー(断線または永続的な通信フォルト)の発生時、信号をオープン・コンタクトに伝達するまでに1032分の遅延を設けます。
TRANSACTION 7		CONFIG_SAFEMON2を設定します。
70h	xxh	SAFEMON_SCAN_DLYを設定します。
01h	xxh	BMSデータ・アクイジション・システムの1秒間のフォルト・ポーリングを有効化します。
TRANSACTION 8		CONFIG_SAFEMON3を設定します。
72h	xxh	設定します。
0Fh	xxh	BMS安全モニタリング・モードに初めて入るときにGPIO1、GPIO2、GPIO3、GPIO4をマスクします。GPIO安全対策を実行する時間を用意します。
TRANSACTION 9		CONFIG_SLPを設定します。
74h	xxh	SLP_SCAN_DLY、SLP_ALRTPCKTEN、SLP_CBNTFYを設定します。
1Fh	xxh	スリープ・モードになりマイクロコントローラへの割込みによる応答が行われるときの、BMSデータ・アクイジション・システムの1秒間の自動ポーリングを有効化します。
TRANSACTION 10		CONFIG_COMMを設定します。
76h	xxh	COMM_RTRY、COMM_TO_DLYを設定します。
02h	xxh	daisy-chain = 7の場合 READALLの時間設定は251μsです。COMM_TO_DLYはメッセージの時間設定より30%長くすることを推奨します。
TRANSACTION 11		GPIO12を設定します。
80h	xxh	GPIO1_CFGおよびGPIO2_CFGを設定します。
66h	xxh	NMOS入力用のコンタクト・ドライバを有効化します(BMS安全モニタリング・モードのみ)。
TRANSACTION 12		GPIO34を設定します。
82h	xxh	GPIO3_CFGおよびGPIO4_CFGを設定します。
46h	xxh	BMS安全モニタリング・モードでのマスタ/スレーブの通信トランザクションを有効化します。マイクロコントローラのアクティブ・ハイ・リセットを有効化します。

表20. シングルUARTデバイス設定シーケンス (続き)

SPI DIN	SPI DOUT	説明
TRANSACTION 13 ALRTEN_OPSTATEを設定します。		
28h	xxh	SLP_ALRTEN、SLP_STATUS_ERR_ALRTEN、SAFEMON_ALRTEN、SAFEMON_GPIO12_ALRTEN、SAFEMON_STATUS_ERR_ALRTEN、SAFEMON_CONFIG_ERR_ALRTENを設定します。
3Fh	xxh	設定されたスリープ・モードおよびBMS安全モニタリング・モードのアラートに対するアラート・ピンのアサートを有効化します。
TRANSACTION 14 ALRTEN_LSSM_BYTEを設定します。		
24h	xxh	RX_READY_ALRTENを設定します。
80h	xxh	LSSMステータス・バイトのRX_READYアラートに対するアラート・ピンのアサートを有効化します。
TRANSACTION 15 ALRTEN_RXを設定します。		
20h	xxh	RX_STOP_ALRTENを設定します。
02h	xxh	RX_STOPに対するアラート・ピンのアサートを有効化します。
TRANSACTION 16 ALRTEN_WDを設定します。		
2Ch	xxh	WD_TO_ERR_ALRTENを設定します。
10h	xxh	ウォッチドッグ・タイムアウトに対するアラート・ピンのアサートを有効化します。 これにより、ホスト・マイクロコントローラとMAX17851の間の同期が可能になります。
TRANSACTION 17 受信バッファ・エラーがないかチェックします。		
11h	xxh	ALERT_RXレジスタのフラグを読み出します。
xxh	00h	エラーがなければ続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。

MAX17851のデバイス設定が終了した後のシステムの設定方法の詳細については、[シングルUARTのBMSデータ・アクイジション・システムの初期化](#)のセクションを参照してください。

シングルUARTのBMSデータ・アクイジション・システムの初期化

表21で説明する例は、2つのUARTデバイスの初期化シーケンスの概要を示すものです。

表21. シングルUARTのBMSデジタイゼーションの初期化シーケンス

SPI DIN	SPI DOUT	説明
TRANSACTION 1 CONFIG_GEN3を設定します。		
66h	xxh	ALRTPCKT_TIMINGを設定します。
05h	xxh	キープアライブ時間を160μsに設定します。
TRANSACTION 2 ALRTEN_RXを設定します。		
20h	xxh	RX_ERR_ALRTENおよびRX_OVRFLW_ERR_ALRTENを設定します。
88h	xxh	RX_ERR_ALRTENビットおよびRX_OVRFLW_ERR_ALRTENビット (ALERTピン・イネーブル・ビット)を有効化します。
TRANSACTION 3 CONFIG_GEN2を設定します。		
64h	xxh	TX_PREAMBLESおよびTX_QUEUE送信を設定します。
30h	xxh	プリアンプルの送信を有効化します (これによりデジタイゼーション・デバイスがウェイクアップします)。
TRANSACTION 4 STATUS_RXを読み出します。		
01h	xxh	STATUSを読み出します。
xxh	21h	RX_STATUS = 21hの場合、続行します。それ以外の場合、真またはタイムアウトになるまでトランザクションを繰り返します。
TRANSACTION 5 CONFIG_GEN2を設定します。		
64h	xxh	TX_PREAMBLESおよびTX_QUEUE送信を設定します。
10h	xxh	プリアンプルの送信を無効化します。
TRANSACTION 6 CLR_RXBUF		

表20. シングルUARTデバイス設定シーケンス (続き)

SPI DIN	SPI DOUT	説明
42h	xxh	受信バッファをクリアしリセットします。
00h	xxh	受信バッファをクリアしリセットします。
TRANSACTION 7		CLR_TXBUF
40h	xxh	送信バッファをクリアしリセットします。
00h	xxh	送信バッファをクリアしリセットします。
TRANSACTION 8		HELLOALLコマンド・シーケンスをロード・キューにロードします。
C0h	xxh	LD_Q SPIコマンド・バイトを書き込みます (ロード・キューを書き込みます)。
03h	xxh	メッセージ長。
57h	xxh	HELLOALLコマンド・バイト。
00h	xxh	レジスタ・アドレス (0x00)。
00h	xxh	HELLOALLの初期化アドレス。
TRANSACTION 9		RX_RD_MSGを読み出します (オプション)。
C2h	xxh	ロード・キューの内容を検証します。
00h	xxh	ポインタを読み出し対象のロード・キューに変更します。
TRANSACTION 10		バッファの内容を読み出します (オプション)。
C1h	xxh	ロード・キューのアドレスを読み出します。
xxh	03h	メッセージ長。
xxh	57h	HELLOALLコマンド・バイト。
xxh	00h	レジスタ・アドレス。
xxh	00h	HELLOALLの初期化アドレス。
TRANSACTION 11		HELLOALLシーケンスを送信します。
B0h	xxh	NXT_LDQ SPIコマンド・バイト (次のロード・キューを書き込みます)。
TRANSACTION 12		STATUS_RXを読み出します。
01h	xxh	RX_STOPステータス・ビットをポーリングします。
xxh	x2h	RX_Status[1]が真の場合、続行します。偽の場合、真になるまでトランザクションを繰り返します。
TRANSACTION 13		RX_RD_NXT_MSGを読み出します。
93h	xxh	受信バッファを処理します。 デ이지チェーンを通じて伝搬しMAX17851に返されたHELLOALLメッセージを読み出します。 ホストはデバイス・カウントを検証する必要があります。
xxh	57h	コマンド・バイト (HELLOALL) を送信します。
xxh	00h	アドレス = 00hを送信します。
xxh	02h	受信アドレス = 02h (2デバイス)
xxh	84h	LSSMバイト。
TRANSACTION 14		ALERT_RXを読み出します。
11h	xxh	受信バッファ・エラーがないかチェックします。
xxh	00h	エラーがない場合、続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。

MAX17851の更なる設定の詳細については、[最終的なMAX17851設定](#)のセクションを参照してください。

最終的なMAX17851設定

デバイスが正常に初期化された後、エラーがない場合、ホスト・マイクロコントローラは通常動作で続行できます。ホストは、MAX17851からの自動キープアライブ生成を無効化し、デバイスにアラート・パケット生成を設定することもできます。アラート・パケット設定シーケンスを[表22](#)に示します。

表22. アラート・パケット設定

SPI DIN	SPI DOUT	説明
TRANSACTION 1		CONFIG_GEN2を設定します。
64h	xxh	TX_PREAMBLESを設定します。
10h	xxh	プリアンブル送信モードを無効化します。
TRANSACTION 2		CONFIG_GEN3を設定します。
66h	xxh	ALRTPCKT_TIMINGを設定します。
08h	xxh	アラート・パケットの時間設定を1.28msにします。
TRANSACTION 3		CONFIG_GEN4を設定します。
68h	xxh	CO_ALRTPCKTENを設定します。
A8h	xxh	コマンド動作モードでのアラート・パケットの時間を設定します。 スリープ・モードまたはBMS安全モニタリング・モードの時間設定は、SLP_SCAN_DLYおよびSAFEMON_SCAN_DLYで指定されます。
TRANSACTION 4		CONFIG_SLPを設定します。
74h	xxh	SLP_SCAN_DLY、SLP_ALRTPCKTENを設定します。
3Fh	xxh	SLP_SCAN_DLYを1秒に設定します。 SLP_ALRTPCKTENは、SLP_SCAN_DLYで設定された遅延を用いるために有効化します。SLP_CBNTFYをオフに設定します(ユーザにより選択)。

デジチェーン設定用の設定メモリ・シーケンス

次の表で、BMS安全モニタリング・モードで適切な動作をするようにMAX17853またはMAX17854を設定するための標準的な手順を説明します。この表の重要なスレッシュホールド設定値は全て、BMS安全モニタリング・モードのBMSデータ・アクイジション・システムによる自動ポーリング動作のための設定例として示されています。スレッシュホールド、タイマー、セル・バランシングなどの設定は、設定メモリのプログラミングおよび送信のための手順を示すための例にすぎない点に注意してください。実際の設定値は、システム要件によって、また、デバイスをBMS安全モニタリング・モードでどのように設定する必要があるかによって異なります。このモードでは、BMSデータ・アクイジション・システムが重要なパラメータを測定し、フォルト情報があればそれをMAX17851デバイスに送信します。

表23. 設定メモリのシーケンス

SPI TRANSACTION	SPI DESCRIPTION	DESCRIPTION OF UART COMMAND
D0	Address Configuration Memory	
60	Write Config Memory QUEUE0_BLOCK0_RA1	Write 0x00FF: All auxiliary inputs configured for Ratiometric mode.
60	Write Config Memory QUEUE0_BLOCK0_RA2	Redundant Write 0x00FF: All auxiliary inputs configured for Ratiometric mode.
FF	Write Config Memory QUEUE0_BLOCK0_DATA LSB	
00	Write Config Memory QUEUE0_BLOCK0_DATA MSB	
64	Write Config Memory QUEUE0_BLOCK1_RA1	Write: 0x7FFF: Cells and block measurement enabled.
64	Write Config Memory QUEUE0_BLOCK1_RA2	Redundant write 0x7FFF.

表23. 設定メモリのシーケンス (続き)

FF	Write Config Memory QUEUE0_BLOCK1_DATA LSB	
7F	Write Config Memory QUEUE0_BLOCK1_DATA MSB	
65	Write Config Memory QUEUE0_BLOCK2_RA1	Write: 0x003F: Auxiliary measurement enabled.
65	Write Config Memory QUEUE0_BLOCK2_RA2	Redundant write 0x003F.
3F	Write Config Memory QUEUE0_BLOCK2_DATA LSB	
00	Write Config Memory QUEUE0_BLOCK2_DATA MSB	
6B	Write Config Memory QUEUE0_BLOCK3_RA1	Write: 0x0012: Disable the HVMUX CTST. Enable the built in diagnostic (user selected). In this case, selection is made of die temperature and an ADC reference diagnostic.
6B	Write Config Memory QUEUE0_BLOCK3_RA2	Redundant write: 0x0012:
12	Write Config Memory QUEUE0_BLOCK3_DATA LSB	
00	Write Config Memory QUEUE0_BLOCK3_DATA MSB	
6C	Write Config Memory QUEUE0_BLOCK4_RA1	Write: 0x0000: Disable the cell input diagnostic current sources.
6C	Write Config Memory QUEUE0_BLOCK4_RA2	Redundant write 0x0000
00	Write Config Memory QUEUE0_BLOCK4_DATA LSB	
00	Write Config Memory QUEUE0_BLOCK4_DATA MSB	
6D	Write Config Memory QUEUE0_BLOCK5_RA1	Write: 0x0000: Disable the auxiliary input diagnostic current sources.
6D	Write Config Memory QUEUE0_BLOCK5_RA2	Redundant write 0x0000
00	Write Config Memory QUEUE0_BLOCK5_DATA LSB	
00	Write Config Memory QUEUE0_BLOCK5_DATA MSB	
6E	Write Config Memory QUEUE0_BLOCK6_RA1	Write: 0x0000: Disable the auxiliary input application of diagnostic current sources.
6F	Write Config Memory QUEUE0_BLOCK6_RA2	Write 0x0000: All balancing switches OFF for auto polling operation only.
00	Write Config Memory QUEUE0_BLOCK6_DATA LSB	
00	Write Config Memory QUEUE0_BLOCK6_DATA MSB	
	PEC Calculation for Block0-6 = 0xAD	
	RESERVED BYTE	

表23. 設定メモリのシーケンス (続き)

20	Write Config Memory QUEUE1_BLOCK7_RA1	Write 0xE664: Setting the threshold for cell Over Voltage fault (user defined).
20	Write Config Memory QUEUE1_BLOCK7_RA2	Redundant write 0xE664: Setting the threshold for cell Over Voltage fault (user defined).
64	Write Config Memory QUEUE1_BLOCK7_DATA LSB	
E6	Write Config Memory QUEUE1_BLOCK7_DATA MSB	
22	Write Config Memory QUEUE1_BLOCK8_RA1	Write 0x8CCC: Setting the threshold for cell Under Voltage fault (user defined).
22	Write Config Memory QUEUE1_BLOCK8_RA2	Redundant Write 0x8CCC: Setting the threshold for cell Under Voltage fault (user defined).
CC	Write Config Memory QUEUE1_BLOCK8_DATA LSB	
8C	Write Config Memory QUEUE1_BLOCK8_DATA MSB	
1A	Write Config Memory QUEUE1_BLOCK9_RA1	Write 0x7FFF: Enables the Over Voltage alert for cell and BLOCK input.
1A	Write Config Memory QUEUE1_BLOCK9_RA2	Redundant write 0x7FFF.
FF	Write Config Memory QUEUE1_BLOCK9_DATA LSB	
7F	Write Config Memory QUEUE1_BLOCK9_DATA MSB	
1B	Write Config Memory QUEUE1_BLOCK10_RA1	Write 0x7FFF: Enables the Under Voltage alert for cell and BLOCK input.
1B	Write Config Memory QUEUE1_BLOCK10_RA2	Redundant write 0x7FFF.
FF	Write Config Memory QUEUE1_BLOCK10_DATA LSB	
7F	Write Config Memory QUEUE1_BLOCK10_DATA MSB	
31	Write Config Memory QUEUE1_BLOCK11_RA1	Write 0xE8BC: Setting the threshold for auxiliary Over Voltage fault (user defined).
31	Write Config Memory QUEUE1_BLOCK11_RA2	Redundant write 0xE8BC: Setting the threshold for auxiliary Over Voltage fault (user defined).
BC	Write Config Memory QUEUE1_BLOCK11_DATA LSB	
E8	Write Config Memory QUEUE1_BLOCK11_DATA MSB	
33	Write Config Memory QUEUE1_BLOCK12_RA1	Write 0x26C8: Setting the threshold for Auxiliary Under Voltage fault (user defined)
33	Write Config Memory QUEUE1_BLOCK12_RA2	Redundant Write 0x26C8: Setting the threshold for Auxiliary Under Voltage fault (user defined)
C8	Write Config Memory QUEUE1_BLOCK12_DATA LSB	
26	Write Config Memory QUEUE1_BLOCK12_DATA MSB	

表23. 設定メモリのシーケンス (続き)

1C	Write Config Memory QUEUE1_BLOCK13_RA1	Write 0x003F: Enables the auxiliary Over Voltage alert.
1C	Write Config Memory QUEUE1_BLOCK13_RA2	Redundant write 0x003F.
3F	Write Config Memory QUEUE1_BLOCK13_DATA LSB	
00	Write Config Memory QUEUE1_BLOCK13_DATA MSB	
	PEC Calculation for Block7-13 = 0x8C	
	RESERVED BYTE	
1D	Write Config Memory QUEUE2_BLOCK14_RA1	Write 0x003F: Enables the auxiliary Under Voltage alert.
1D	Write Config Memory QUEUE2_BLOCK14_RA2	Redundant write 0x003F.
3F	Write Config Memory QUEUE2_BLOCK14_DATA LSB	
00	Write Config Memory QUEUE2_BLOCK14_DATA MSB	
1F	Write Config Memory QUEUE2_BLOCK15_RA1	Write 0xD708: Cell Over Voltage alert clear threshold set.
1F	Write Config Memory QUEUE2_BLOCK15_RA2	Redundant write 0xD708: Cell Over Voltage alert clear threshold set.
D7	Write Config Memory QUEUE2_BLOCK15_DATA LSB	
08	Write Config Memory QUEUE2_BLOCK15_DATA MSB	
21	Write Config Memory QUEUE2_BLOCK16_RA1	Write 0x9998: Cell Under Voltage alert clear threshold set (user defined).
21	Write Config Memory QUEUE2_BLOCK16_RA2	Redundant write 0xD708: Cell Under Voltage alert clear threshold set (user defined).
98	Write Config Memory QUEUE2_BLOCK16_DATA LSB	
99	Write Config Memory QUEUE2_BLOCK16_DATA MSB	
30	Write Config Memory QUEUE2_BLOCK17_RA1	Write 0xD554: Auxiliary Over Voltage alert clear threshold set (user defined).
30	Write Config Memory QUEUE2_BLOCK17_RA2	Redundant Write 0xD554: Auxiliary Over Voltage alert clear threshold set (user defined).
54	Write Config Memory QUEUE2_BLOCK17_DATA LSB	
D5	Write Config Memory QUEUE2_BLOCK17_DATA MSB	
32	Write Config Memory QUEUE2_BLOCK18_RA1	Write 0x4D94: Auxiliary Under Voltage alert clear threshold set (user defined).
32	Write Config Memory QUEUE2_BLOCK18_RA2	Redundant write 0x4D94: Auxiliary Under Voltage alert clear threshold set (user defined).
94	Write Config Memory QUEUE2_BLOCK18_DATA LSB	

表23. 設定メモリのシーケンス (続き)

4D	Write Config Memory QUEUE2_BLOCK18_DATAMSB	
70	Write Config Memory QUEUE2_BLOCK19_RA1	Write 0x03FF: This configures the expiration time for the cell balancing operation and is set to 0x3FF for the indefinite operation or in other word no expiry. It should be noted that auto polling only exits based on the timer or a timeout criteria.
70	Write Config Memory QUEUE2_BLOCK19_RA2	Redundant write 0x03FF:
FF	Write Config Memory QUEUE2_BLOCK19_DATA LSB	
03	Write Config Memory QUEUE2_BLOCK19_DATAMSB	
80	Write Config Memory QUEUE2_BLOCK20_RA1	Write 0x1233: A cell balancing control register that initiates and controls the different cell balancing modes and operation. This configuration is setting the CBMODE of the daisy-chain device in auto group by sec. This command effectively starts the auto polling after the daisy-chain is configured.
80	Write Config Memory QUEUE2_BLOCK20_RA2	Redundant write 0x1233:
12	Write Config Memory QUEUE2_BLOCK20_DATA LSB	
33	Write Config Memory QUEUE2_BLOCK20_DATAMSB	
	PEC Calculation for Block14-20 = 0xC3	
	RESERVED BYTE	
Write Config Memory QUEUE0_BLOCK2_DATAMSB		

ウォッチドッグ設定

ウォッチドッグはデフォルトで無効化されているため (WD_EN = '0')、ホスト・マイクロコントローラは、スタートアップ後にこれを設定して有効化しなくてはなりません。ホスト・マイクロコントローラが無応答になった場合、ウォッチドッグはタイムアウトし、BMS安全モニタリング・モードに入ります。

MAX17851のウォッチドッグ機能は、次のシーケンスを用いて有効化と設定ができます。

表24. ウォッチドッグ設定

SPI DIN	SPI DOUT	説明
TRANSACTION 1		CONFIG_WD2を設定します。
88h	xxh	WD_DBNCを設定します。
07h	xxh	BMS安全モニタリング・モードに入るまでに必要なウォッチドッグ違反回数を設定します。
TRANSACTION 2		WD_WD0を設定します。
84h	xxh	WD_OPNおよびWD_CLOを設定します。
EFh	xxh	ウォッチドッグ・タイムアウト未満の時間にWD_OPNおよびWD_CLOを有効化します。
TRANSACTION 3		WD1を設定します。
86h	xxh	WD_1UDおよびWD_DIVを設定します。
FFh	xxh	延長ウォッチドッグ・ウィンドウおよびウォッチドッグ・クロック分周器を設定します。
TRANSACTION 4		WD2を設定します。
88h	xxh	WD_ENを設定します。

表24. ウォッチドッグ設定(続き)

SPI DIN	SPI DOUT	説明
87h	xxh	ウォッチドッグを有効化します。
TRANSACTION 5		ALERT_WDをモニタします。
1Ch	xxh	ウォッチドッグ機能が有効になり、ウォッチドッグ・アラート・レジスタがモニタされます。

UART書き込みおよび読出しのためのトランザクション・シーケンス

表25に示す例では、ホストが2つのUARTスレーブ・デバイスと通信を行い、以下を実行します。

- WRITEALLコマンド・シーケンスを用いて全てのスレーブ・デバイス用のデバイス・レジスタ・アドレス0x64に7FFFhの値を書き込む。
- READALLコマンド・シーケンスを用いて全てのスレーブ・デバイス用のデバイス・レジスタ・アドレス0x64から7FFFhの値をリードバックする。

この例では、スレーブ・デバイスがアライブ・カウンタを有効化して設定されていることを仮定しています。これら2つのコマンド・シーケンスを実行するために、ホストは表25に示すSPIトランザクションを実行します。

表25. UART書き込みおよび読出しのためのトランザクション・シーケンス

SPI DIN	SPI DOUT	説明
TRANSACTION 1		WRITEALLコマンド・シーケンスをロード・キューにロードします。
C0h	xxh	LDQ SPIコマンド・バイト。
06h	xxh	メッセージ長 = 6。
02h	xxh	WRITEALLコマンド・バイト。
64h	xxh	デバイスのレジスタ・アドレス。
FFh	xxh	書き込み対象のレジスタ・データのLSバイト。
7Fh	xxh	書き込み対象のレジスタ・データのMSバイト。
24h	xxh	02h、64h、7Fh、FFhのPECバイト。
00h	xxh	アライブ・カウンタ・バイト(シード値 = 0)。
TRANSACTION 2		送信キューからWRITEALLシーケンスの送信を開始します。
B0h	xxh	NXT_LDQ SPIコマンド・バイト。
TRANSACTION 3		メッセージが受信バッファに受信されたかどうかを調べます。
01h	xxh	RX_STATUSレジスタを読み出します。
xxh	x2h	RX_STOP(ビット1)が真ならば続行します。偽の場合は、真またはタイムアウトになるまで繰り返します。
TRANSACTION 4		受信バッファを讀出して、送信されたWRITEALLメッセージを検証します。
93h	xxh	RX_RD_NXT_MSG SPI
xxh	02h	送信されたコマンド・バイト(WRITEALL)。
xxh	64h	送信されたアドレス。
xxh	FFh	送信されたLSバイト。
xxh	7Fh	送信されたMSバイト。
xxh	02h	アライブ・カウンタ・バイト(アライブ・カウンタが有効化されている場合、送信されたシード値に2を加えた値)。
xxh	84h	LSSM_BYTE
xxh	ECh	PECの計算値。
TRANSACTION 5		受信バッファ・エラーがないかチェックします。
01h	xxh	RX_STATUSレジスタを読み出します。
xxh	00h	RX_ERR、RX_OVRFLW_ERRをチェックします。エラーがなければ続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。

表25. UART書込みおよび読出しのためのトランザクション・シーケンス (続き)

SPI DIN	SPI DOUT	説明
TRANSACTION 6		
C0h	xxh	NXT_LDQ SPIコマンド・バイト。
09h	xxh	メッセージ長。
03h	xxh	READALLコマンド・バイト。
64h	xxh	レジスタ・アドレス。
00h	xxh	データチェック・バイト(シード値 = 00h)。
A6h	xxh	03h、64h、00hのバイトのPECバイト。
00h	xxh	アライブ・カウンタ・バイト(シード値 = 00h)。
TRANSACTION 7		
B0h	xxh	NXT_LDQ SPIコマンド・バイト。
TRANSACTION 8		
メッセージが受信バッファに受信されたかどうかを調べます。		
01h	xxh	RX_Statusレジスタを読み出します。
xxh	x2h	RX_STOP(ビット1)が真ならば続行します。偽の場合は、真またはタイムアウトになるまで繰り返します。
TRANSACTION 9		
受信バッファを読み出し、デバイスのレジスタ・データがWRITEALLシーケンス中に書き込まれたものであることを検証します。		
93h	xxh	RX_RD_NXT_MSG SPIコマンド・バイト。
xxh	03h	送信されたコマンド・バイト(READALL)。
xxh	64h	送信されたレジスタ・アドレス。
xxh	FFh	デバイス1のLSバイト。
xxh	7Fh	デバイス1のMSバイト。
xxh	FFh	デバイス0のLSバイト。
xxh	7Fh	デバイス0のMSバイト。
xxh	00h	データチェック・バイト(全てのステータス・ビットがクリアされた場合、00h)。
xxh	02h	アライブ・カウンタ・バイト(アライブ・カウンタが有効化されている場合、送信されたシード値に2を加えた値)。
xxh	84h	LSSM_BYTE
xxh	D5h	PEC(それ以前の9バイトについてのもの)。
TRANSACTION 10		
受信バッファ・エラーがないかチェックします。		
01h	xxh	RX_STATUSレジスタを読み出します。
xxh	00h	RX_ERR、RX_OVRFLW_ERRをチェックします。エラーがなければ続行します。エラーがある場合はクリアし、エラー・ルーチンに移行します。

補助的なエラー・チェック機能

MAX17851は、内部ハードウェアによる検証と自動UARTメッセージ検証を実行します。LSSMステータス・バイトを通じ、最新のステータスが全てのトランザクションに付加されます。

必要に応じ、ホストは、補助的なエラー・チェックおよび検証を実行し、MAX17851とBMSデータ・アキュイジション・システムの完全性を確実なものにできます。これらのチェックには、以下のようなものがあります。

- 受信した書込みデータの検証(マッチング値、バイト数)
- 受信した読出しデータの検証(許容値、範囲、バイト数)
- 受信したPEC、データチェック、アライブ・カウンタ・バイトの検証
- 接続されているデジタイゼーション・デバイスのFMEAレジスタの検証

破損したプリアンブル・キャラクタ

受信したプリアンブルが破損している場合、そのメッセージは受信バッファには書き込まれません。この状態は通信タイムアウト・タイマーが時間切れになることによって検出され、COMM_ERRフラグがセットされます。

複数の同じメッセージが送信される場合（例えば、ホストがデジチェーン・デバイス・レジスタをポーリングしている場合など）では、この状態は固有のアライブ・カウンタ・シードの送信と検証によっても検出できます。自動アライブ・カウンタが有効化されている場合、このプロセスは自動的に行われ、エラーは、ALIVECOUNT_ERRでレポートされます。手動アライブ・カウンタが有効化されている場合、ホストは、アライブ・カウンタ・シードを手動でインクリメントし検証しなくてはなりません。アライブ・カウンタ・バイトはPEC後に送信されます。従って、PECはアライブ・カウンタ値の影響は受けません。

破損メッセージの内容

マンチェスター、パリティ、PECにエラーがある場合、それは、メッセージのデータが破損している可能性があることを示しています。UARTメッセージを受信すること、ホストは、受信メッセージで使用する全てのエラーチェック・データについて、適切な計算を行う必要があります。このデータの例は以下のとおりです。

- データチェック・バイト: スレーブ・デバイスから提供されたエラー・ステータス。スレーブ・デバイスのデータシートの記述に従い、スレーブ・デバイス・データの読出し値について送信と返信が行われます。
- LSSMステータス・バイト: MAX17851から提供され、各受信メッセージに付加されたエラー・ステータス。
- PEC: 各受信メッセージを保護するCRC-8パケット・エラーチェック・バイト。
- アライブ・カウンタ・バイト: 送信されたメッセージに回答しているデバイスの数を検証するために用いられます。スレーブ・デバイスのデータシートの記述に従い、全てのメッセージと共に送受信できます。
- RX_ERR_ALERTビット: マンチェスター・エラーやパリティ・エラーは、受信バッファの読出しトランザクションの過程で、メッセージ内の少なくとも1つの受信キャラクタで検出されます。

注: マンチェスター・エラーおよびパリティ・エラーは、UARTのデータ・ストリームのどこにおいても発生する可能性があるため、RX_ERR_ALERTで示されるエラーは、必ずしもスレーブ・デバイスによって返されるエラー・チェック・バイトに反映されるわけではありません。そのため、ホストがこのフラグをチェックしクリアするのは、受信バッファの各メッセージを読み出した後である必要があります、決してそれより前であってはなりません。ホストは、RX_ERR_ALERTENビットをセットして、そのようなイベントが通知されるようにする必要があります。

ストップ・キャラクタの破損または喪失

ストップ・キャラクタが破損している、または喪失していても、データは失われません。それは、メッセージが、後続の有効なストップ・キャラクタ（キープアライブ・モードでは自動的に送信）、フレーム・タイムアウト、次のUARTメッセージのプリアンブルのいずれかによって、引き続きフレーム化されるためです。破損したストップ・キャラクタは、データ・キャラクタと解釈される可能性があり、そのため受信バッファに格納される可能性があります。この場合、最終的に有効なストップ・キャラクタが次のプリアンブルの前に受信されると、メッセージ長が1バイト分だけ超過します。ホストは、受信メッセージ長を計算し、それを予定メッセージ長と比較することによって、この状態をチェックする必要があります。

次のメッセージを読み出す前に、ホストは、以下の全てが真であることも検証できます。

1. メッセージの最終バイトがヌル・バイト (00h) である。
2. RX_LAST_BYTEビットがセットされている。
3. RX_BYTE_ERRビットがクリアされている。

意図しないプリアンブル

メッセージの途中で意図しないプリアンブルがあると、受信バッファに意図しないメッセージが生じます。意図しないプリアンブルがメッセージ内の破損データ・キャラクタの結果である場合、そのメッセージは本来よりも早く打ち切られ、2番目の意図しないメッセージが形成されます。このイベントは、メッセージ内の受信バイト数を予定されている数と比較することによって検出できます。

意図しないストップ・キャラクタ

意図しないストップ・キャラクタがあると、メッセージが本来よりも早く打ち切られます。これは、メッセージ内の受信バイト数を予定されている数と比較することによって検出できます。

フォルト処理のガイドライン

表26. フォルト処理のガイドライン

ERROR BIT	REGISTER	BIT POSITION	ERROR BIT DESCRIPTION	PRIMARY ACTION FOR INTERMITTENT ERRORS	SECONDARY ACTION FOR PERSISTENT ERRORS

表26. フォルト処理のガイドライン (続き)

ALRTPCKT_STATUS_ERR	STATUS_LSSM_BYTE	6	A safety critical STATUS register error within one or more of the daisy-chain units is reported within the ALRTPCKT message.	1. Read ALRTPCKT_RD_MSG to determine daisy-chain error(s) and device location(s) of error 2. Diagnose cause of error(s) and follow required actions dictated by system safety requirements.	Follow required actions dictated by system safety requirements.
COMM_ERR	STATUS_LSSM_BYTE	5	A communication error is observed with corrupted, inserted, or omitted data within the UART message.	Resend communication command.	Report communication failures and consider terminating normal system behavior.
ALRTPCKT_COMM_ERR	STATUS_LSSM_BYTE	4	A communication error is observed in the auto-generated ALERTPACKET message which corrupted, inserted, or omitted data.	1. Increase COMM_RTRY counter to decrease sensitivity to communication channel noise. 2. Decrease the frequency of issuance of the ALERTPACKET_TIMING. Normal operation is still permissible.	Verify that communication to the daisy-chain is still possible. If errors within user defined message, re-initialize the daisy-chain assuming a wire break. Report communication failures and consider terminating normal system behavior.
ALRTPCKT_HW_ERR	STATUS_LSSM_BYTE	4	A hardware error is reported within the ALERTPACKET buffer.	Manually query to the daisy-chain STATUS register and compare to the ALERTPACKET STATUS register. If the STATUS registers match, the error effected the redundant buffer only. Normal operation is still permissible.	Manually query to the daisy-chain STATUS register and compare to the ALERTPACKET STATUS register. If the STATUS registers match, the error effected the redundant buffer only. Normal operation is still permissible.

表26. フォルト処理のガイドライン (続き)

COMM_MSMTCH_ERR	STATUS_LSSM_BYTE	3	An unexpected communication packet is received that differs from what is transmitted.	1. Reset the LSSM buffer, TX buffer, and RX buffer. 2. Resend communication command.	Report communication failures and consider terminating normal system behavior.
ALIVECOUNT_ERR	STATUS_LSSM_BYTE	1	The rolling Alive Counter returned value does not match the seed incremented by the number of BMS daisy-chained units.	1. Verify the integrity of the returned data message (message is free of error if COMM_ERR and COMM_MSMTCH_ERR are a logical 0). 2. Continue to transmit communication packets and monitored returned value. Normal operation is still permissible.	Disable rolling Alive Counter. Normal operation is still permissible.
HW_ERR	STATUS_LSSM_BYTE[0]	0	A hardware error is reported, which may affect device operation. Errors include supply voltages that exceed specified operating range, oscillator drift errors that may effect communication timing, corrupt device trim, corrupt memory (stuck at or transient) and internal register errors.	Issue a SWPOR and re-initialize the device. or Verify configuration registers and re-initialize if required, and resend communication command.	Report communication failures and consider terminating normal system behavior.
GPIO_ERR	STATUS_GEN	4	The GPIO output level does not match the intended programmed value.	Perform SAFEMON GPIO verification diagnostic	Report communication failures and consider terminating normal system behavior.
ALRTRST	ALERT_OPSTATE	7	Supply voltages tripped the device POR, causing a reset.	Reinitialize the daisy-chain.	Report communication failures and consider terminating normal system behavior.

表26. フォルト処理のガイドライン (続き)

DEV_COUNT_ERR	STATUS_GEN	6	The HELLOALL returns a device address (DA) that does not match the programmed DEV_COUNT bitfield.	Reinitialize the daisy-chain.	Report communication failures and consider terminating normal system behavior.
WD_ERR	STATUS_GEN	5	The watchdog response is not configured appropriately.	Resend watchdog command.	If system microcontroller remains functional with valid communication, SWPOR can be issued, and the system must be re-initialized. Report failure to system and consider terminating normal system behavior.
DATAPATH_ERR	STATUS_GEN	3	A hardware error occurs during Datapath and Configuration Memory Verification.	1. Verify the data and PEC of the Configuration Memory queue 4 and the data and PEC of queue 1 -3. 2. Perform Configuration Memory and Datapath diagnostic using VER_CONFIG.	Report communication failures and consider terminating normal system behavior.
SPI_ERR	STATUS_GEN	6	A SPI transaction is rejected due to device operation within Sleep or BMS Safe Monitoring mode.	Configure device while operating in Commanded Operation mode.	Configure device while operating in Commanded Operation mode.
SLP_STATUS_ERR	STATUS_OPSTATE	4	An error within one or more of the daisy-chain units is reported during Sleep mode operation with SLP_ALRTPCKT_EN = 1.	1. Read ALRTPCKT_RD_MSG to determine daisy-chain error(s) and device location(s) of error. 2. Diagnose cause of error(s) and follow required actions dictated by system safety requirements.	Follow required actions dictated by system safety requirements.

表26. フォルト処理のガイドライン (続き)

SAFEMON_STATUS_ERR	STATUS_OPSTATE	1	An error within one or more of the daisy-chain units is reported during BMS Safe Monitoring mode.	Critical system failure experienced and system forced into safe state.
--------------------	----------------	---	---	--

スリープ・モードの設定

MAX17851のスリープ・モードは、ホスト・マイクロコントローラが低電力モードになっている間に、デジチェーン・デバイスのセーフティ・クリティカルなパラメータを自律的にモニタするよう設計されています。

スリープ・モードに入るには、SLP_ENコマンド・レジスタのSLP_ENビットをアサートします。メモリ設定検証処理あるいは設定ロード処理がアクティブになっている場合、またはBMS安全モニタリング・モードになっている場合には、スリープ・モードに入ることはできません。スリープ・モードが終了すると、SLP_ENも自動クリアされます。

MAX17851がコマンド動作モードからスリープ・モードに移移すると、CO_ALRTPACKETおよびALRTPCKT_TIMINGに設定されている自動生成アラート・パケットは無効化されます。スリープ・モードの場合、アラート・パケットはSLP_ALRTPCKTENレジスタで有効化され、アラート・パケットの頻度はSLP_SCAN_DLYレジスタで設定されます。

MAX17851がスリープ・モードになっている間、デジチェーン・デバイスは、自動ポーリング／セル・バランシング動作を行うよう設定できます。ホストは、CONFIG_SLPLレジスタのSLP_CBNTFYビットを用いてセル・バランシングのタイマー満了時間をプログラムできます。SLP_CBNTFYタイマー時間が満了すると、デバイスはスリープ・モードを自動的に終了します。ALERT_OPSTATEレジスタのSLP_ALRTビットがフラグ通知を行い、ホストをウェイクアップするようALERTピンがアサートします。

TX_AUTO機能を用いたハードウェア・イン・ザ・ループ(HIL)テスト

MAX17851は、TX_AUTO機能を用いたハードウェア・イン・ザ・ループ(HIL)テストをサポートしています。

2つのMAX17851がマスタ／スレーブ・アプリケーション回路に配置され、マスタのTX/RXラインがデジチェーンからバイパスされてスレーブのRX/TXラインに接続されている場合、任意のシーケンスのマスタのトランザクションに(TXバッファの最大サイズに応じて)自動的に応答するようスレーブをプログラムできます。

これを行うには、スレーブ・デバイスのTX_AUTOビットを有効化し、LD_Qコマンドを介してそれを目的の応答でプログラミングします。TX_AUTOが有効化されると、MAX17851は、プリアンプルの受信時に送信バッファの内容を自動的に送信します。

このモードでは、MAX17851は図22に示すように設定されます。

HILは、ハードウェア検証を目的としており、SAFEMON動作やスリープ動作の間には機能しません。ホスト・マイクロコントローラは、データ送信または破損の複数の問題(破損データ、破損PEC、データチェック・エラーなど)を発生させこれを特定することができます。これを用いることで、多数の通信不良に対して想定されるマスタ応答をテストできます。

スレーブ・デバイスが自動送信モード(TX_AUTO = 1)に設定された後、マスタ・デバイスはデータをスレーブ・デバイスに送信できます。有効なプリアンプルを受信すると、スレーブ・デバイスは、既にロード・キューにあるデータを、マスタ・デバイスに送り返します。ホストは、データを評価し、正しく機能しているかどうか判定します。

以下のハードウェア・イン・ザ・ループ・テスト・シーケンスに示す例は、TX_AUTO = 1に設定した後に2つのMAX17851デバイスをテスト用に設定するシーケンスの概略を説明するものです。

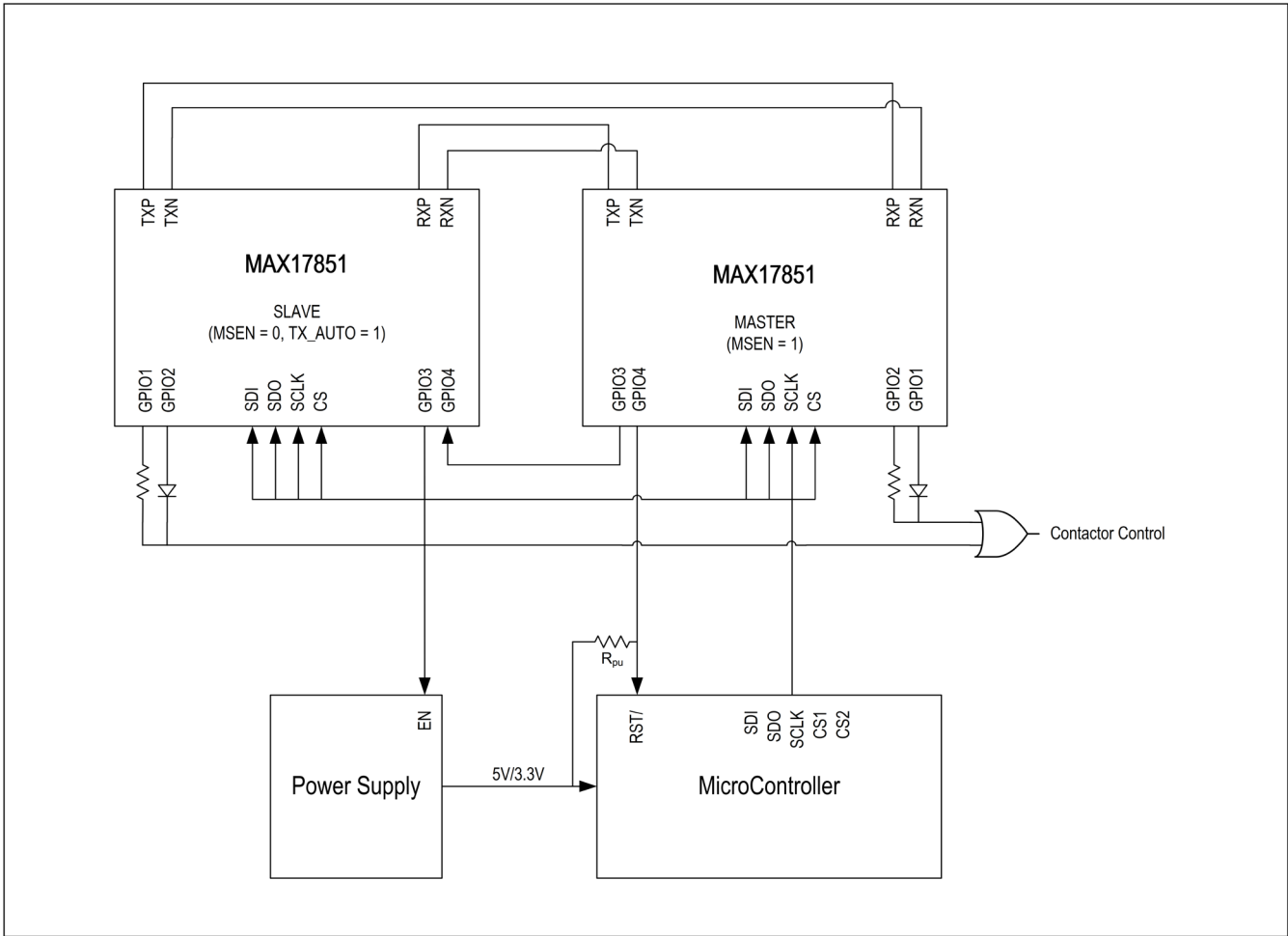


図22. ハードウェア・イン・ザ・ループのテスト・セットアップ

表27. ハードウェア・イン・ザ・ループのテスト・シーケンス

DIN	DOUT	説明
TRANSACTION 1		
20h	xxh	RX_ErrorおよびRX_Overflowに対するRx割込みフラグを有効化します。
88h	xxh	アラート・イネーブル・レジスタに書き込みます。
88h	xxh	RX_ERR_ALRTENビットおよびRX_OVRFLW_ERR_ALRTENビット (ALERTピン・イネーブル・ビット) をセットします。
TRANSACTION 2		
42h	xxh	受信バッファをクリアします。
00h	xxh	受信バッファをクリアします。
TRANSACTION 3		
40h	xxh	送信バッファをクリアします。
00h	xxh	送信バッファをクリアします。
TRANSACTION 4		
CSをスレーブ・デバイス用にセットします。		
TRANSACTION 5		
スレーブ・デバイスのTX_AUTOをセットします。		

表27. ハードウェア・イン・ザ・ループのテスト・シーケンス(続き)

DIN	DOUT	説明
66h	xxh	スレーブ・デバイスのTX_AUTOを設定します。
40h	xxh	スレーブ・デバイスの自動送信モードを有効化します。
TRANSACTION 6		スレーブ・デバイスのロード・キューにREADALLコマンド・シーケンスをロードします。
C0h	xxh	WR_LD_Q SPIコマンド・バイト。
06h	xxh	メッセージ長 = 6。
03h	xxh	READALLコマンド・バイト。
00h	xxh	デバイスのレジスタ・アドレス。
00h	xxh	LSBデータ。
00h	xxh	MSBデータ。
00h	xxh	データチェック・バイト。
F0h	xxh	PEC
TRANSACTION 7		CSをマスタ・デバイス用にセットします。
TRANSACTION 8		マスタ・デバイスのロード・キューにREADALLコマンド・シーケンスをロードします。
C0h	xxh	WR_LD_Q SPIコマンド・バイト。
06h	xxh	メッセージ長 = 6。
03h	xxh	READALLコマンド・バイト。
00h	xxh	デバイスのレジスタ・アドレス。
00h	xxh	データチェック・バイト。
58h	xxh	PEC
TRANSACTION 9		ヌル・メッセージが受信されるまで待機します(RX_EMPTYステータス・ビットをポーリングします)。
01h	xxh	RX_Statusレジスタを読み出します。
xxh	11h	
TRANSACTION 10		ロード・キューを送信します。
B0h	xxh	
TRANSACTION 11		RX_STOPをポーリングします。
01h	xxh	
xxh	x2h	
TRANSACTION 12		RXバッファを読み出します。
93h	xxh	READALLコマンド。
xxh	03h	コマンド・バイト。
xxh	00h	レジスタ・アドレス。
xxh	00h	LSBデータ。
xxh	00h	MSBデータ。
xxh	00h	データチェック・バイト。
xxh	84h	LSSMバイト。
xxh	70h	PEC
TRANSACTION 13		STATUS_RXを読み出します。
01h	xxh	
xxh	11h	エラーがなければ、通常動作を続行します。エラーがある場合、ホストはエラーをチェックする必要があります。

同様のシーケンスを用いて、ユーザも他のシステム・フォルト(プリアンブルなし、破損データ、破損レジスタ・アドレスなど)を検証できます。

UARTの物理レイヤ

UARTのシングルエンド・モード動作

デフォルトでは、UARTポートは差動通信用に設定されています。ホストがSINGLE_ENDED設定ビットをセットすることで、シングルエンド動作にできます。このモードでは、入力スレッショルドを負側にシフトしてゼロ差動電圧がロジック1になるようにすることで、UARTがシングルエンド信号を受信できます。RXP入力はグランドに接続され、RXN入力は反転信号を受信します。このモードでは、TXポートは差動モードの場合と同じ動作をします。

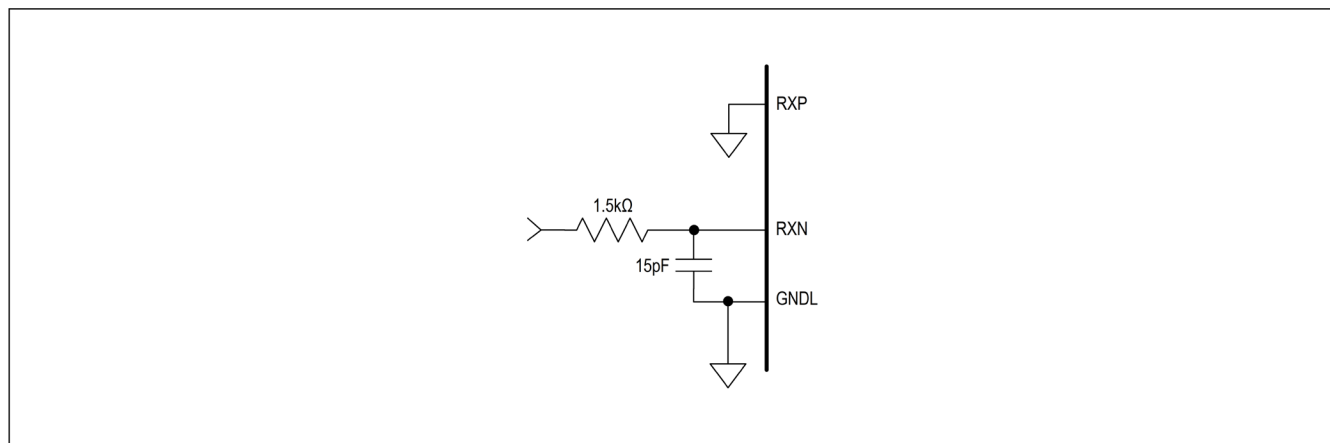


図23. シングルエンド・モード

UARTのトランス・カップリング

UART信号は、DCバランスされた信号であるため、トランス・カップリングが可能です。UARTのトランスとスレーブ・デバイスのレシーバの間に絶縁トランスを配置することで、スレーブ・デバイスが異なる電圧レベルで動作している場合にコモン・モード絶縁が可能です。信号トランスのセンタータップを使用すると、[図24](#)に示すように、ノードをローカル・グランドにACカップリングすることによって、同相モード除去性能を向上できます。1次側と2次側の寄生カップリングを通過できるコモン・モード電流はグランドにシャントされて、非常に効果的な同相ノイズ・フィルタを形成します。

データが送信されず(アイドル状態)TX_HI_Zレジスタ・ビットが0の場合、トランスミッタは、両方の出力をロジック・ロー・レベルに下げ、トランスの巻線を通じて電流が流れるのを防ぎます。

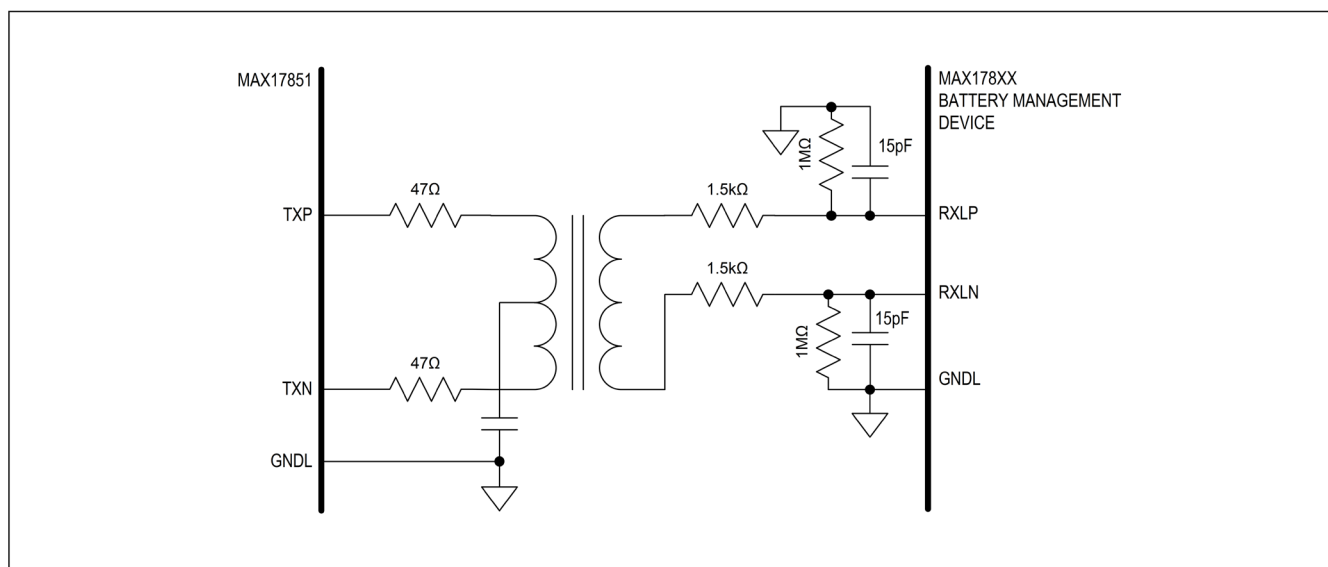


図24. UART信号のトランス・カップリング

最小トランス磁化インダクタンスを480μHとすることを推奨します。

UARTの補足的ESD保護

UARTポートには、接触放電に関するIEC 61000-4-2要求を満たすために補足的な保護が必要になることがあります。±8kVの保護レベルを満たすための推奨回路を図25と図26に示します。保護部品は、PCBの信号入力点にできるだけ近付けて配置する必要があります。ダイオードは、外部コネクタにできるだけ近付けて配置する必要があります。

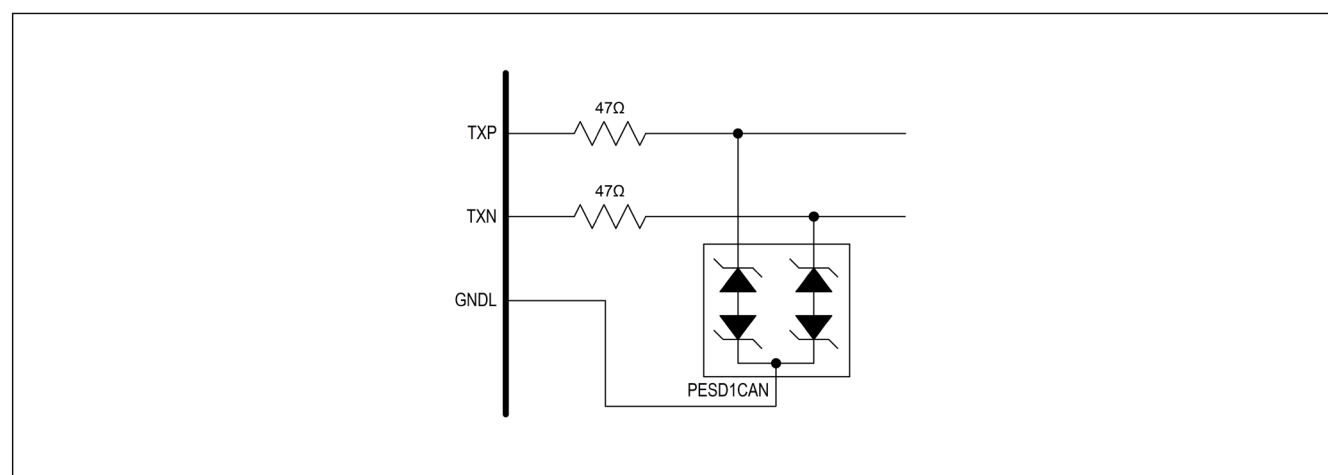


図25. UARTトランスミットの補足的ESD保護

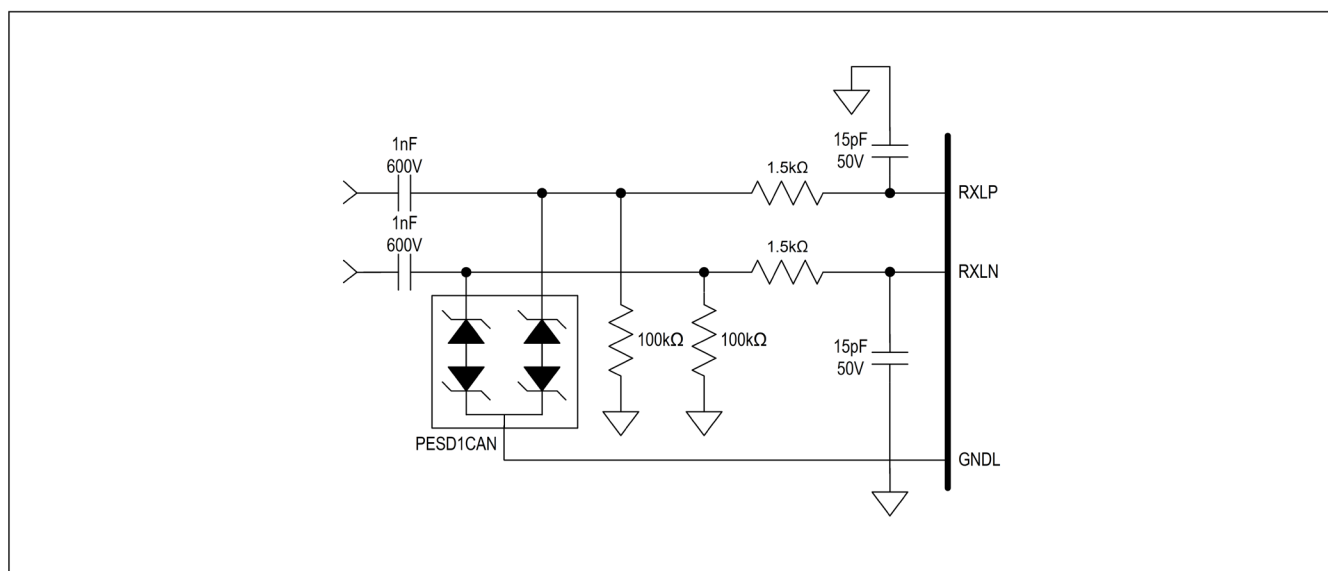


図26. UARTレシーバの補足的ESD保護(容量性カップリングと共に表示)

電源に関する考慮事項

MAX17851には2つのレギュレータ電源があります。 V_{DDL1} は、内部発振器およびロジックに給電する安定化電源です。 V_{DDL2} は、UARTインターフェースに給電する安定化電源です。

以下の図27に示すように、 V_{DDL2} のレギュレーションをバイパスし、3.3Vまたは5Vの電源から直接給電することもできます。

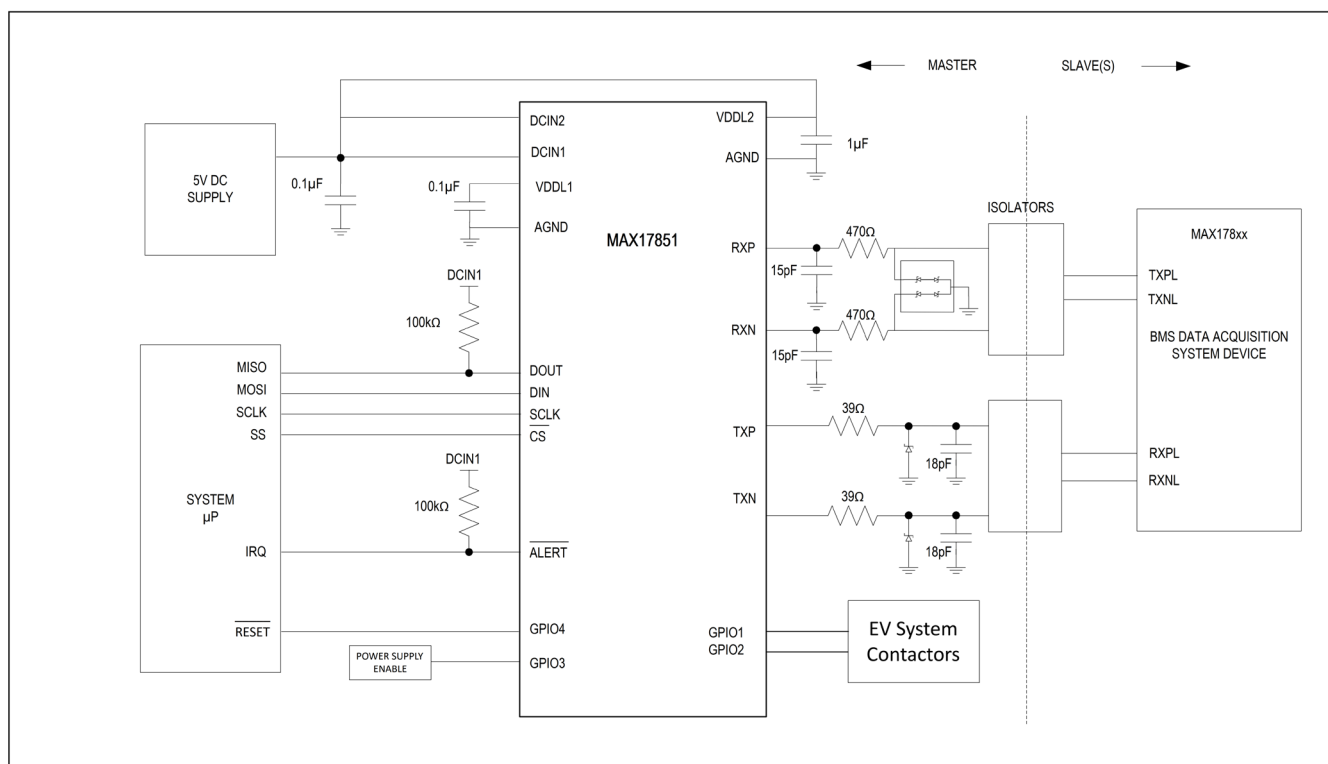


図27. 電源に関する考慮事項

PCBレイアウトに関する推奨事項

最大限の精度性能と高い環境耐性を実現するには、慎重なPCBレイアウトが極めて重要です。

1. デカップリング・コンデンサを、MAX17851と同じレイヤの上で各ピンの近くに配置します。コンデンサはグランド・リターンを共用しないようにし、ビアを使いAGND内部層に直接接続する必要があります。
2. UARTのRXポートおよびTXポートは100Ωの差動インピーダンスとなるよう配線する必要があります。グランド・リターンをAGNDプレーンに直接ビア接続して、MAX17851のできるだけ近くにUART ESD保護を配置するよう推奨します。これは、デバイス性能に影響を与える可能性のある他のノードにトランジェント・イベントがカップリングする前に、それらのイベントをクランプするためです。
3. SPI動作は、MAX17851と同じグランド・プレーンに配置されたシステム・マイクロコントローラによって駆動されます。パターン長に応じて、オーバーシュートやアンダーシュートが絶対最大動作条件を侵すことのないよう、オプションのソース終端が必要となる場合があります。このソース終端は、デバイスの送信ピンの付近に配置する必要があります。

レイアウト例(シルクスクリーン)

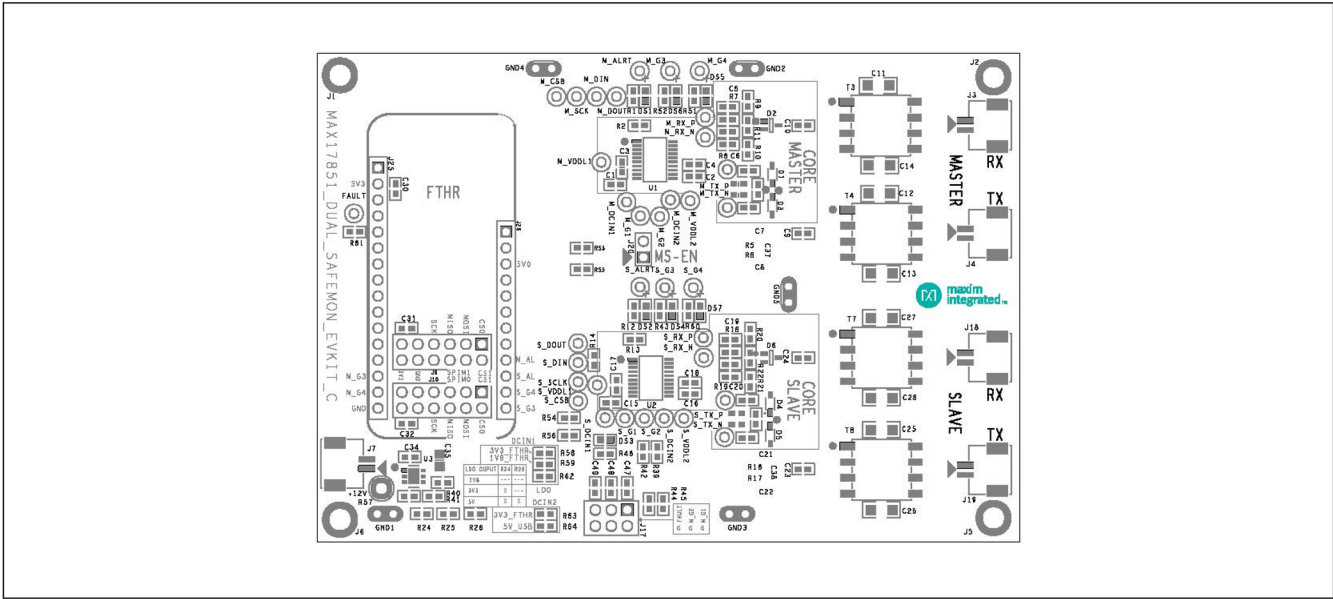


図28. レイアウト例:最上層のシルクスクリーン

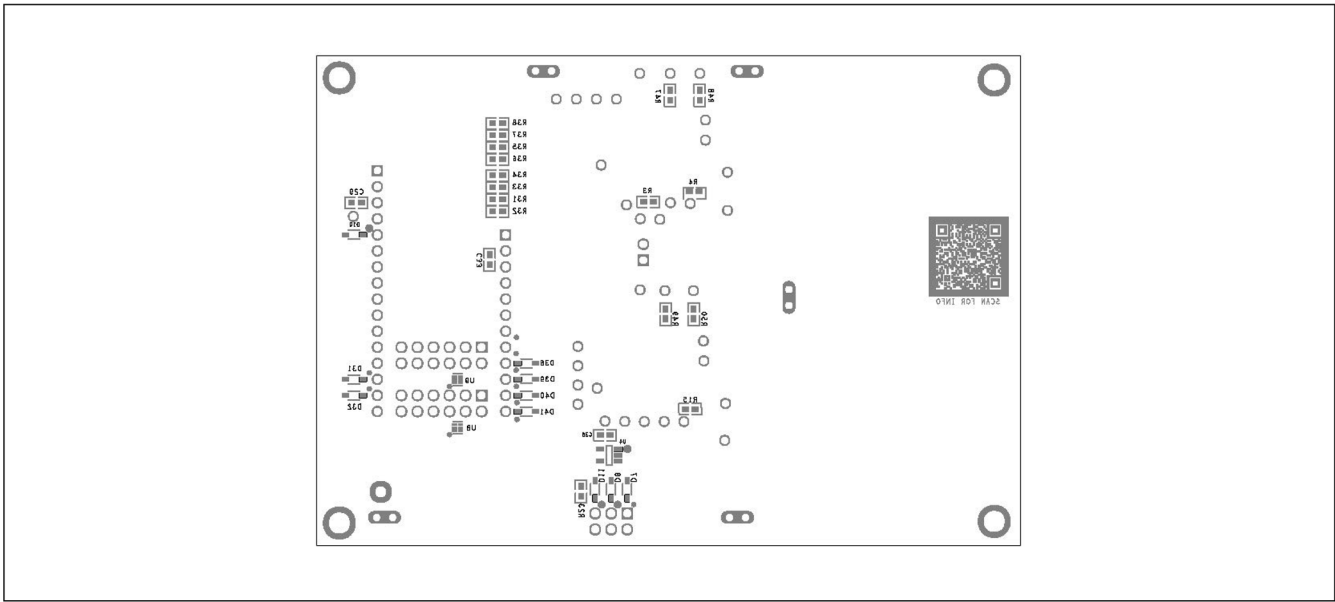


図29. レイアウト例:最下層のシルクスクリーン

レイアウト例(金属)

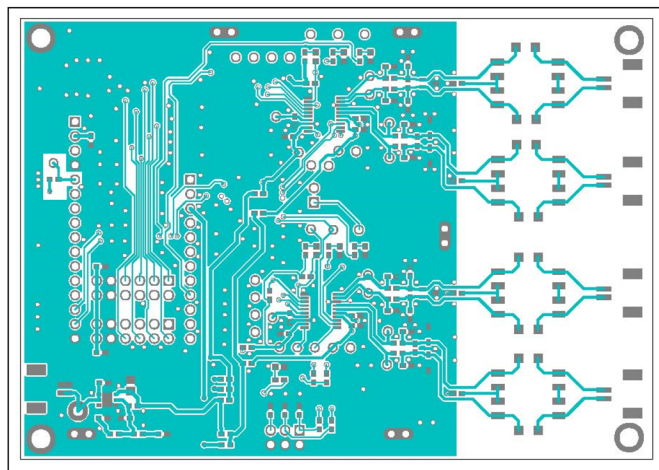


図30. レイアウト例:最上層の金属

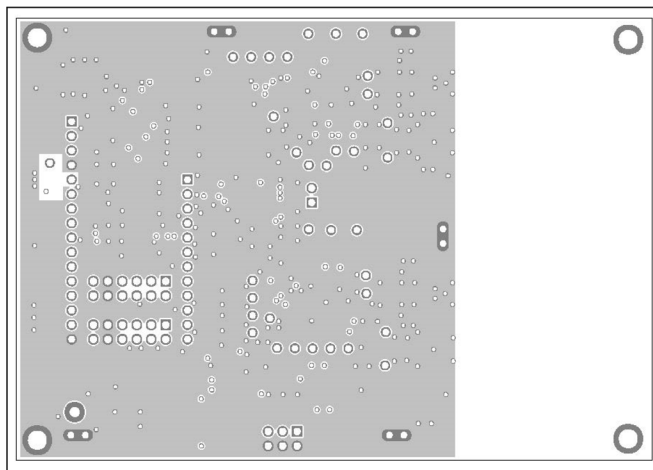


図31. レイアウト例:2層目の金属

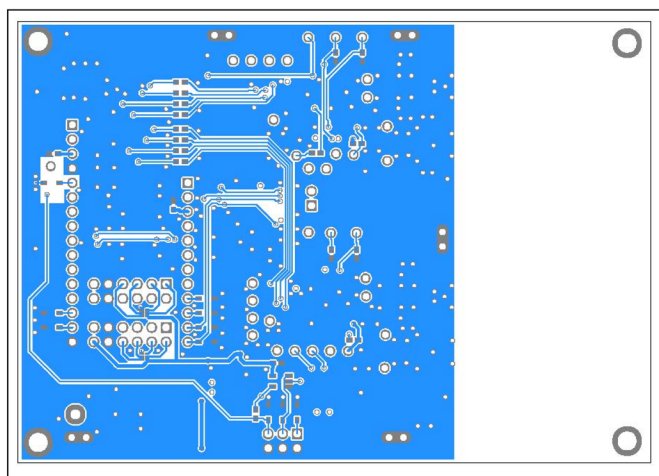


図32. レイアウト例:最下層の金属

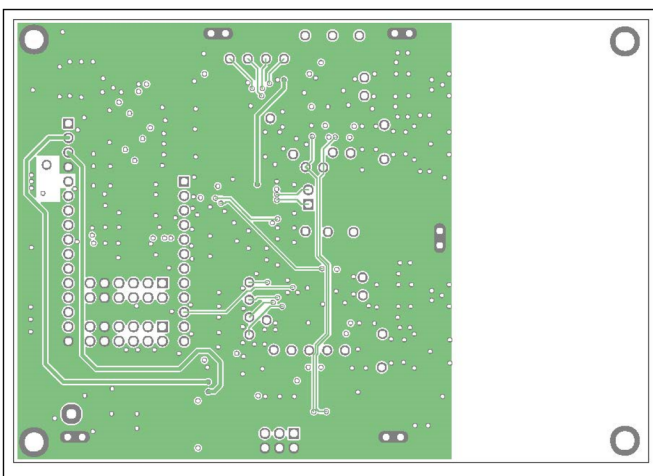
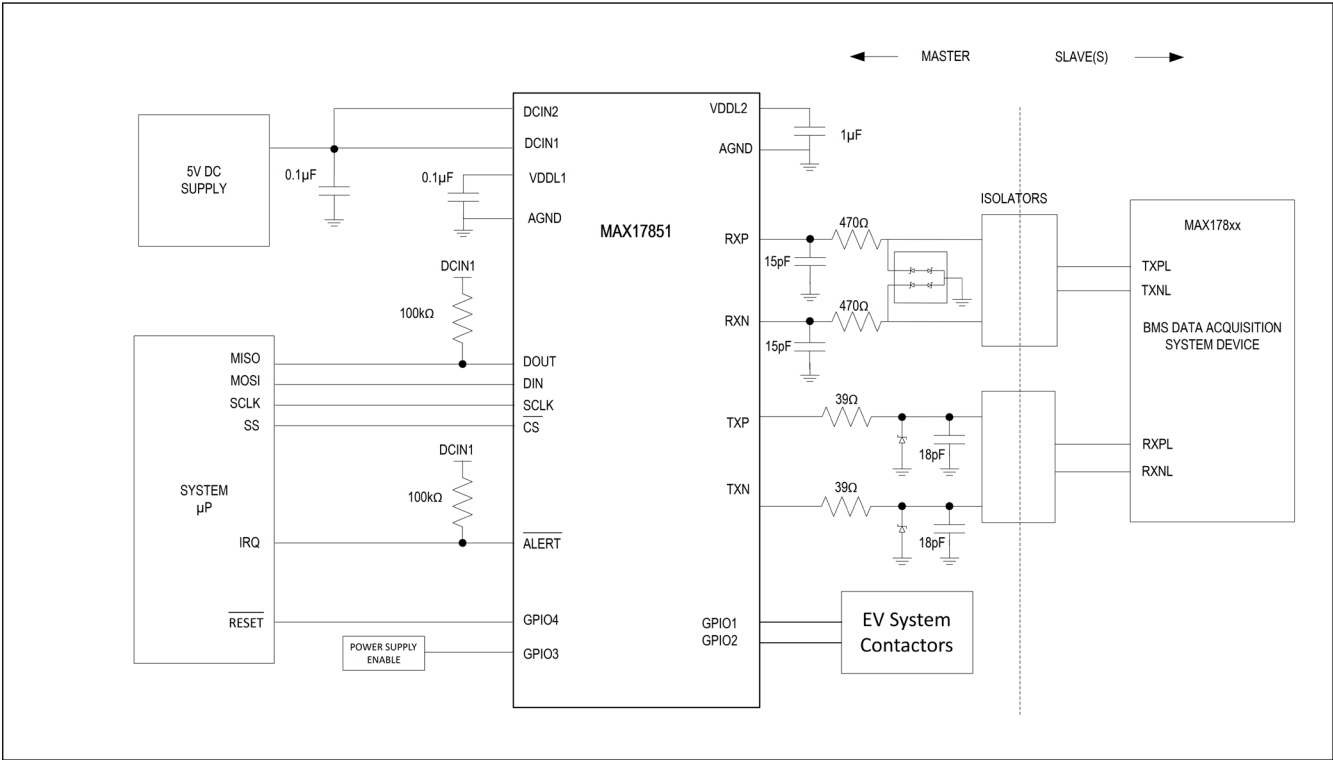


図33. レイアウト例:3層目の金属

標準アプリケーション回路

MAX17851のアプリケーション回路



型番

PART NUMBER	TEMP RANGE	PIN-PACKAGE
MAX17851AUP/V+	-40°C to +125°C	TSSOP U20+7C
MAX17851AUP/V+T	-40°C to +125°C	TSSOP U20+7C

+は鉛(Pb)フリー／RoHS準拠パッケージを表します。
T = テープ&リール。

改訂履歴

版数	改訂日	説明	改訂ページ
0	08/21	初版発行	—

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 25 頁、上から 6 行目

【誤】

「安全対策診断状態では、BMS 安全モニタリング制御ロジック、ウォッチドッグ・ロジック、MAX17851 の自律的な BMS 安全モニタリング・モード動作の範囲外の GPIOが正しく機能しているかどうかを、ホストが検証できます。」

【正】

「安全対策診断状態では、BMS 安全モニタリング制御ロジック、ウォッチドッグ・ロジック、MAX17851 の自律的な BMS 安全モニタリング・モード動作の GPIO の出力が正しく機能しているかどうかを、ホストが検証できます。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 34 頁、ウォッチドッグの項、上から 3 行目

【誤】

「・・・カウンタがインクリメントします (CONFIG_WD2[3:0] の WD_DBNC ビット)。」

【正】

「・・・カウンタがインクリメントします (CONFIG_WD2[2:0] の WD_DBNC ビット)。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 44 頁、最下段の行

【誤】

「UART バッファは、UART の各バッファのサイズと構成を示します。」

【正】

「図 17、図 18 とその説明は、UART の各バッファのサイズと構成を示します。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 52 頁、メッセージの読み出しの項、上から 2 行目

【誤】

「● RX_RD_NXT_MSG : RX_RD_PTR の場所（次の未読メッセージ）で読み出しを開始します。」

【正】

「● RX_RD_NXT_MSG : RX_RD NXT_PTR の場所（次の未読メッセージ）で読み出しを開始します。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを
記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 67 頁、表内、ビットフィールド「HFOSC_HW_ERRB」の説明欄、上から 3
行目

【誤】

「このステータスは、低周波発振器の 4 サイクル (64kHz) ごとに更新されます。」

【正】

「このステータスは、低周波発振器の 4 サイクル (8kHz) ごとに更新されます。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 97 頁、表内 ビットフィールド「ALRTPCKT_TIMING」の説明欄、下から 3 個目の注

【誤】

「注：推奨するアラート・パケットのタイミング・レートは 1.28ms (デフォルト) です。」

【正】

「注：推奨するアラート・パケットのタイミング・レートは 1.28ms です。この設定のデフォルトは 1111 です。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 115 頁、上の表内 ビットフィールド「CONFIG_BYTE_PTR」の説明欄

【誤】

「ONFIG_QUEUE_PTR に 'd0～'d2 の値が含まれている場合、'd30 以上'の無効な値を書き込むと、その書込みは無視されます。」
す。」

【正】

「ONFIG_QUEUE_PTR に 'd0～'d2 の値が含まれている場合、'd3 以上'の無効な値を書き込むと、その書込みは無視されます。」

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。
この正誤表は、2025 年 9 月 16 日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。
なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2025 年 9 月 16 日

製品名： MAX17851

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 131 頁、中程、「デイジーチェーン設定用の設定メモリ・シーケンス」の項

【誤】

「次の表で、BMS 安全モニタリング・モードで適切な動作をするように MAX17853 または MAX17854 を設定するための標準的な手順を説明します。」

【正】

「次の表で、BMS 安全モニタリング・モードで適切な動作をするように MAX17851 を設定するための標準的な手順を説明します。」