

RENESAS TECHNICAL UPDATE

〒211-8668 神奈川県川崎市中原区下沼部 1753

ルネサス エレクトロニクス株式会社

問合せ窓口 <http://japan.renesas.com/inquiry>E-mail: csc@renesas.com

製品分類	MPU & MCU	発行番号	TN-SH7-A875A/J	Rev.	第1版
題名	SH7455 グループ、SH7456 グループ ユーザーズマニュアル ハードウェア編 正誤表 Rev. B		情報分類	ドキュメント訂正追加など	
適用製品	SH7455 グループ、SH7456 グループ	対象ロット等	関連資料	SH7455 グループ、SH7456 グループ ユーザーズマニュアルハードウェア編 Rev. 1.10 (R01UH0030JJ0110)	

2011 年 9 月 16 日に発行致しました「SH7455 グループ、SH7456 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10」に、内容の訂正がありましたので、お知らせします。

SH7455 グループ、SH7456 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10 をご利用の際は、添付の正誤表をご利用ください。

なお、添付の正誤表 (Rev.B) には、下記の同マニュアルの正誤表 (Rev.A)、および CAN モジュールに関するユーザーズマニュアルの誤記訂正の内容も記載しています。

- ・テクニカルアップデート TN-SH7-A827A/J: 正誤表 (Rev.A)
- ・テクニカルアップデート TN-SH7-A859A/J: CAN モジュールに関するユーザーズマニュアルの誤記訂正

添付資料：「SH7455 グループ、SH7456 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10」正誤表 Rev.B...9 枚

※下記において、赤字、下線の部分が追加/変更のある部分です。

Rev.	ページ	項目	内容	
Rev.A 追加	21-1	表 21.1 ATU-IIIS の概要	誤 正	表 21.1 ATU-IIIS の概要 の機能の説明を訂正 ・ <u>59</u> 本の割り込み要因を生成可能です。 ・ <u>64</u> 本の割り込み要因を生成可能です。
Rev.A 追加	38-29	表 38.28 DRI タイミング (特殊モード ON 時)	誤 <	

Rev.	ページ	項目		内容																								
Rev.B 追加	12-39	12.9.4 書き込み/消去中の リセット	追 加	12.9.4 書き込み/消去中のリセットに説明を追加 追加) <u>書き込み/消去処理中に、RESET#端子への"L"レベル入力によるハード ウェアリセット、電源切断、または FRESETR レジスタの FRESET ビッ トをセットして FCU リセットを実行すると、書き込み/消去対象領域の 全データは不定となります。不定となった領域を再度使用する場合は、 消去処理を行ってから使用してください。</u>																								
Rev.B 追加	15-54	15.3.18 (11) INT2B11 レジスタ： FlexRay モジュール の割り込み詳細要因	誤 正	15.3.18 (11) INT2B11 レジスタ：FlexRay モジュールの割り込み詳細要因の 説明の誤記を訂正 ・ <u>SH7546</u> グループは、FlexRay モジュールを内蔵していません。 ・ <u>SH7456</u> グループは、FlexRay モジュールを内蔵していません。																								
Rev.B 追加	18-10	表 18.16 レジスタ構成	誤 正	表 18.16 レジスタ構成のレジスタ名の誤記を訂正 ・ ポート DEF しきい値切替レジスタ ・ ポート GHJ しきい値切替レジスタ ・ ポート DEF <u>入力</u> しきい値切替レジスタ ・ ポート GHJ <u>入力</u> しきい値切替レジスタ																								
Rev.B 追加	24-10	24.3.3 RSPIi 端子制御 レジスタ (SPiPCR)	誤 正	24.3.3 RSPIi 端子制御レジスタ (SPiPCR) の MOIFE ビット説明の誤記を訂正 ・ MOIFE ビットが"0"の場合には、RSPIi は SSL ネゲート期間中に前回のシ リアル転送の最終 <u>データ</u> を MOSli 端子に出力します。 ・ 0：MOSli 出力値は前回転送の最終 <u>データ</u> ・ MOIFE ビットが"0"の場合には、RSPIi は SSL ネゲート期間中に前回のシ リアル転送の最終 <u>出力値</u> を MOSli 端子に出力します (<u>CPHA=0 の場合は不 定</u>)。 ・ 0：MOSli 出力値は前回転送の最終 <u>出力値</u> (<u>CPHA=0 の場合は不定</u>)																								
Rev.B 追加	24-26	24.3.13 RSPIi コマンドレジ スタ 0~3 (SPiCMD0 ~SPiCMD3)	誤 正	24.3.13 RSPIi コマンドレジスタ 0~3 (SPiCMD0~SPiCMD3) の SPNDEN ビッ ト説明の誤記を訂正 ・ SPNDEN ビットが"0"の場合には、RSPIi は次アクセス遅延を 1RSPCK にします。 ・ 0：次アクセス遅延は 1RSPCK ・ SPNDEN ビットが"0"の場合には、RSPIi は次アクセス遅延を 1RSPCK+ <u>2Pck</u> にします。 ・ 0：次アクセス遅延は 1RSPCK+ <u>2Pck</u>																								
Rev.B 追加	24-33	表 24.7 SSL ネゲート期間の MOSli 信号値の決定 方法	誤 正	表 24.7 SSL ネゲート期間の MOSli 信号値の決定方法の誤記を訂正 <table><tr><th>MOIFE</th><th>MOIFV</th><th>SSL ネゲート期間の MOSli 信号値</th></tr><tr><td>0</td><td>0、1</td><td>前回転送の最終 <u>データ</u></td></tr><tr><td>1</td><td>0</td><td>常に"L"</td></tr><tr><td>1</td><td>1</td><td>常に"H"</td></tr></table> <table><tr><th>MOIFE</th><th>MOIFV</th><th>SSL ネゲート期間の MOSli 信号値</th></tr><tr><td>0</td><td>0、1</td><td>前回転送の最終 <u>出力値</u>(<u>CPHA=0 の場合は不定</u>)</td></tr><tr><td>1</td><td>0</td><td>常に"L"</td></tr><tr><td>1</td><td>1</td><td>常に"H"</td></tr></table>	MOIFE	MOIFV	SSL ネゲート期間の MOSli 信号値	0	0、1	前回転送の最終 <u>データ</u>	1	0	常に"L"	1	1	常に"H"	MOIFE	MOIFV	SSL ネゲート期間の MOSli 信号値	0	0、1	前回転送の最終 <u>出力値</u> (<u>CPHA=0 の場合は不定</u>)	1	0	常に"L"	1	1	常に"H"
MOIFE	MOIFV	SSL ネゲート期間の MOSli 信号値																										
0	0、1	前回転送の最終 <u>データ</u>																										
1	0	常に"L"																										
1	1	常に"H"																										
MOIFE	MOIFV	SSL ネゲート期間の MOSli 信号値																										
0	0、1	前回転送の最終 <u>出力値</u> (<u>CPHA=0 の場合は不定</u>)																										
1	0	常に"L"																										
1	1	常に"H"																										

Rev.	ページ	項目	内容
Rev.B 追加	24-39	図 24.11 RSPI 転送フォーマット (CPHA="0")	図 24.11 RSPI 転送フォーマット (CPHA="0") の誤記を訂正
			図 24.12 RSPI 転送フォーマット (CPHA="1") の誤記を訂正

Rev.	ページ	項目		内容																																																
Rev.B 追加	24-55	24.4.9 (1) (g) 次アクセス遅延 (t3)	誤	24.4.9 (1) (g) 次アクセス遅延 (t3) の説明の誤記を訂正 ・ RSPi は、シリアル転送で参照する SPiCMD レジスタをポインタ制御によって決定し、選択した SPiCMD レジスタの SPNDEN ビットと SPiND レジスタを使用して、表 24.11 のようにシリアル転送時の RSPCK 遅延を決定します。																																																
			正	・ RSPi は、シリアル転送で参照する SPiCMD レジスタをポインタ制御によって決定し、選択した SPiCMD レジスタの SPNDEN ビットと SPiND レジスタを使用して、表 24.11 のようにシリアル転送時の 次アクセス 遅延を決定します。																																																
Rev.B 追加	26-7	表 26.3 レジスタ構成	誤	表 26.3 レジスタ構成のレジスタ名の誤記を訂正 ・ CAN チャネルサーチサポートレジスタ																																																
			正	・ CAN1 チャネルサーチサポートレジスタ																																																
Rev.B 追加	26-5	表 26.3 レジスタ構成	誤	CANi クロック選択レジスタ (CiCLKR) (i=0~3) のリセット後の値を訂正 <table><tr><th>レジスタ名</th><th>シンボル</th><th>リセット後の値</th><th>P4 領域 アドレス</th><th>アクセス サイズ</th><th>掲載 ページ</th></tr><tr><td>CAN0 クロック選択レジスタ</td><td>C0CLKR</td><td>H'00</td><td>H'FFFF 6847</td><td>8、16、32</td><td>26-16</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>CAN1 クロック選択レジスタ</td><td>C1CLKR</td><td>H'00</td><td>H'FFFF 7847</td><td>8、16、32</td><td>26-16</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>CAN2 クロック選択レジスタ</td><td>C2CLKR</td><td>H'00</td><td>H'FFFF 8847</td><td>8、16、32</td><td>26-16</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>CAN3 クロック選択レジスタ</td><td>C3CLKR</td><td>H'00</td><td>H'FFFF 9847</td><td>8、16、32</td><td>26-16</td></tr></table>	レジスタ名	シンボル	リセット後の値	P4 領域 アドレス	アクセス サイズ	掲載 ページ	CAN0 クロック選択レジスタ	C0CLKR	H'00	H'FFFF 6847	8、16、32	26-16	:	:	:	:	:	:	CAN1 クロック選択レジスタ	C1CLKR	H'00	H'FFFF 7847	8、16、32	26-16	:	:	:	:	:	:	CAN2 クロック選択レジスタ	C2CLKR	H'00	H'FFFF 8847	8、16、32	26-16	:	:	:	:	:	:	CAN3 クロック選択レジスタ	C3CLKR	H'00	H'FFFF 9847	8、16、32	26-16
			レジスタ名	シンボル	リセット後の値	P4 領域 アドレス	アクセス サイズ	掲載 ページ																																												
CAN0 クロック選択レジスタ	C0CLKR	H'00	H'FFFF 6847	8、16、32	26-16																																															
:	:	:	:	:	:																																															
CAN1 クロック選択レジスタ	C1CLKR	H'00	H'FFFF 7847	8、16、32	26-16																																															
:	:	:	:	:	:																																															
CAN2 クロック選択レジスタ	C2CLKR	H'00	H'FFFF 8847	8、16、32	26-16																																															
:	:	:	:	:	:																																															
CAN3 クロック選択レジスタ	C3CLKR	H'00	H'FFFF 9847	8、16、32	26-16																																															
			正	<table><tr><th>レジスタ名</th><th>シンボル</th><th>リセット後の値</th><th>P4 領域 アドレス</th><th>アクセス サイズ</th><th>掲載 ページ</th></tr><tr><td>CAN0 クロック選択レジスタ</td><td>C0CLKR</td><td>不定</td><td>H'FFFF 6847</td><td>8、16、32</td><td>26-16</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>CAN1 クロック選択レジスタ</td><td>C1CLKR</td><td>不定</td><td>H'FFFF 7847</td><td>8、16、32</td><td>26-16</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>CAN2 クロック選択レジスタ</td><td>C2CLKR</td><td>不定</td><td>H'FFFF 8847</td><td>8、16、32</td><td>26-16</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr><tr><td>CAN3 クロック選択レジスタ</td><td>C3CLKR</td><td>不定</td><td>H'FFFF 9847</td><td>8、16、32</td><td>26-16</td></tr></table>	レジスタ名	シンボル	リセット後の値	P4 領域 アドレス	アクセス サイズ	掲載 ページ	CAN0 クロック選択レジスタ	C0CLKR	不定	H'FFFF 6847	8、16、32	26-16	:	:	:	:	:	:	CAN1 クロック選択レジスタ	C1CLKR	不定	H'FFFF 7847	8、16、32	26-16	:	:	:	:	:	:	CAN2 クロック選択レジスタ	C2CLKR	不定	H'FFFF 8847	8、16、32	26-16	:	:	:	:	:	:	CAN3 クロック選択レジスタ	C3CLKR	不定	H'FFFF 9847	8、16、32	26-16
レジスタ名	シンボル	リセット後の値	P4 領域 アドレス	アクセス サイズ	掲載 ページ																																															
CAN0 クロック選択レジスタ	C0CLKR	不定	H'FFFF 6847	8、16、32	26-16																																															
:	:	:	:	:	:																																															
CAN1 クロック選択レジスタ	C1CLKR	不定	H'FFFF 7847	8、16、32	26-16																																															
:	:	:	:	:	:																																															
CAN2 クロック選択レジスタ	C2CLKR	不定	H'FFFF 8847	8、16、32	26-16																																															
:	:	:	:	:	:																																															
CAN3 クロック選択レジスタ	C3CLKR	不定	H'FFFF 9847	8、16、32	26-16																																															
Rev.B 追加	26-16	26.3.2 CANi クロック選択 レジスタ (CiCLKR) (i=0~3)	誤	26.3.2 CANi クロック選択レジスタ (CiCLKR) (i=0~3) のリセット後の値を訂正 ビット： <table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>CCLKS</td></tr></table> リセット後の値： <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> <リセット後の値： H'00> <table><tr><th>ビット</th><th>シンボル</th><th>リセット後の値</th><th>R</th><th>W</th><th>説 明</th></tr><tr><td>4</td><td>—</td><td>0</td><td>?</td><td>0</td><td>予約ビット 書く場合、“0”を書いてください。読んだ場合、その値は不定。</td></tr></table>	7	6	5	4	3	2	1	0	—	—	—	—	—	—	—	CCLKS	0	0	0	0	0	0	0	0	ビット	シンボル	リセット後の値	R	W	説 明	4	—	0	?	0	予約ビット 書く場合、“0”を書いてください。読んだ場合、その値は不定。												
			7	6	5	4	3	2	1	0																																										
—	—	—	—	—	—	—	CCLKS																																													
0	0	0	0	0	0	0	0																																													
ビット	シンボル	リセット後の値	R	W	説 明																																															
4	—	0	?	0	予約ビット 書く場合、“0”を書いてください。読んだ場合、その値は不定。																																															
			正	ビット： <table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>CCLKS</td></tr></table> リセット後の値： <table><tr><td>0</td><td>0</td><td>0</td><td>不定</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> <リセット後の値： 不定> <table><tr><th>ビット</th><th>シンボル</th><th>リセット後の値</th><th>R</th><th>W</th><th>説 明</th></tr><tr><td>4</td><td>—</td><td>不定</td><td>?</td><td>0</td><td>予約ビット 書く場合、“0”を書いてください。読んだ場合、その値は不定。</td></tr></table>	7	6	5	4	3	2	1	0	—	—	—	—	—	—	—	CCLKS	0	0	0	不定	0	0	0	0	ビット	シンボル	リセット後の値	R	W	説 明	4	—	不定	?	0	予約ビット 書く場合、“0”を書いてください。読んだ場合、その値は不定。												
7	6	5	4	3	2	1	0																																													
—	—	—	—	—	—	—	CCLKS																																													
0	0	0	不定	0	0	0	0																																													
ビット	シンボル	リセット後の値	R	W	説 明																																															
4	—	不定	?	0	予約ビット 書く場合、“0”を書いてください。読んだ場合、その値は不定。																																															

Rev.	ページ	項目		内容					
Rev.B 追加	26-43	26.3.11 CANi 受信 FIFO ポイ ンタ制御レジスタ (CiRFPCR) (i=0～3)	誤	26.3.11 CANi 受信 FIFO ポインタ制御レジスタ (CiRFPCR) (i=0～3) の CiRFPCR ビットの R のステータスを訂正					
				ビット	シンボル	リセット 後の値	R	W	説 明
			7～0	CiRFPCR	不定	<u>R</u>	W	CiRFPCR レジスタに"H'FF"を書き込 むと、受信 FIFO の CPU 側ポインタが 移動	
			正						
ビット	シンボル	リセット 後の値		R	W	説 明			
7～0	CiRFPCR	不定	<u>?</u>	W	CiRFPCR レジスタに"H'FF"を書き込 むと、受信 FIFO の CPU 側ポインタが 移動				
Rev.B 追加	26-47	26.3.13 CANi 送信 FIFO ポイ ンタ制御レジスタ (CiTFPCR) (i=0～3)	誤	26.3.13 CANi 送信 FIFO ポインタ制御レジスタ (CiTFPCR) (i=0～3) の CiTFPCR ビットの R のステータスを訂正					
				ビット	シンボル	リセット 後の値	R	W	説 明
			7～0	CiTFPCR	不定	<u>R</u>	W	CiTFPCR レジスタに"H'FF"を書き込 むと、CPU 側の送信 FIFO ポインタが 移動します。	
			正						
ビット	シンボル	リセット 後の値		R	W	説 明			
7～0	CiTFPCR	不定	<u>?</u>	W	CiTFPCR レジスタに"H'FF"を書き込 むと、CPU 側の送信 FIFO ポインタが 移動します。				
Rev.B 追加	26-58	26.3.20 CANi エラー割り込 み要因判定レジスタ (CiEIFR) (i=0～3)	誤	26.3.20 CANi エラー割り込み要因判定レジスタ (CiEIFR) (i=0～3) の BLIF ビット説明の誤記を訂正 (テクニカルアップデート TN-SH7-A859A/J: CAN モジュールに関するユー ザーズマニュアルの誤記訂正と内容は同じ)					
				ビット	シンボル	リセット後 の値	R	W	説 明
			7	BLIF	0	R	W	バスロック検出フラグ*1 CAN モジュールが CAN オペレーションモードの間、CAN バス上に 32 の 連続するドミナントビットを検出すると、BLIF ビットは"1"になります。 "1"になった後、次の <u>どちらかで再検出します。</u> • このビットを"1"から"0"にした後、レセシブビットを検出。 • このビットを"1"から"0"にした後、CANリセットモード <u>またはCAN Hail モード</u> に遷移し、再度CANオペレーションモードに遷移。 0 : バスロック未検出 1 : バスロック検出	
			正						
ビット	シンボル	リセット後 の値		R	W	説 明			
7	BLIF	0	R	W	バスロック検出フラグ*1 CAN モジュールが CAN オペレーションモードの間、CAN バス上に 32 の 連続するドミナントビットを検出すると、BLIF ビットは"1"になります。 "1"になった後、次の <u>いずれかの条件が成立するとバスロックを再検出でき るようになります。</u> • このビットを"1"から"0"にした後、レセシブビットを検出 (<u>バスロック解 消</u>)。 • このビットを"1"から"0"にした後、CANリセットモードに遷移し、再度 CANオペレーションモードに遷移 (<u>内部リセット</u>)。 0 : バスロック未検出 1 : バスロック検出				

Rev.	ページ	項目	内容
Rev.B 追加	26-74	図 26.9 CAN 動作モード間の遷移 (i=0~3)	図 26.9 CAN 動作モード間の遷移 (i=0~3) の誤記を訂正 (テクニカルアップデート TN-SH7-A859A/J: CAN モジュールに関するユーザーズマニュアルの誤記訂正と内容は同じ) CANM, SLPM, BOM, RBOC : CICTLRレジスタのビット 【注】*1 バスオフ状態からCAN Haltモードへの遷移タイミングは、BOMビットの設定に依存します。BOMビットが"01"のとき、状態遷移のタイミングはバスオフ状態への遷移直後になります。BOMビットが"10"のとき、状態遷移のタイミングはバスオフ状態終了時になります。BOMビットが"11"のとき、状態遷移のタイミングはCANMビットが"10" (CAN Haltモード) の設定時になります。 *2 SLPMビットは、CANリセットモードまたはCAN Haltモード時に変更してください。SLPMビットを書き換える場合は、本ビットのみ"0"または"1"にしてください。
			CANM, SLPM, BOM, RBOC : CICTLRレジスタのビット 【注】*1 バスオフ状態からCAN Haltモードへの遷移タイミングは、BOMビットの設定に依存します。BOMビットが"01"のとき、状態遷移のタイミングはバスオフ状態への遷移直後になります。BOMビットが"10"のとき、状態遷移のタイミングはバスオフ状態終了時になります。BOMビットが"11"のとき、状態遷移のタイミングはCANMビットを"10" (CAN Haltモード) にしたときになります。 *2 SLPMビットは、CANリセットモードまたはCAN Haltモード時に変更してください。SLPMビットを書き換える場合は、本ビットのみ"0"または"1"にしてください。 <u>*3 CANバスがドミナントレベルでロックされている間は、CAN Haltモードに遷移しません。その場合は、CANリセットモードに遷移してください。</u>

Rev.	ページ	項目	内容																
Rev.B 追加	26-76	表 26.9 CAN リセットモードと CAN Halt モードでの動作	表 26.9 CAN リセットモードと CAN Halt モードでの動作の誤記を訂正 (テクニカルアップデート TN-SH7-A859A/J: CAN モジュールに関するユーザーズマニュアルの誤記訂正と内容は同じ) <table border="1"> <thead> <tr> <th>モード</th><th>受信</th><th>送信</th><th>バスオフ</th></tr> </thead> <tbody> <tr> <td>CAN リセットモード (強制遷移) CANM="11"</td><td>CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移</td><td>CAN モジュールはメッセージ送信の終了を待たずに CAN リセットモードに遷移</td><td>CAN モジュールはバスオフ復帰の終了を待たずに CAN リセットモードに遷移</td></tr> <tr> <td>CAN リセットモード CANM="01"</td><td>CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移</td><td>CAN モジュールはメッセージ送信の終了を待って CAN リセットモードに遷移*1*4</td><td>CAN モジュールはバスオフ復帰の終了を待たずに CAN リセットモードに遷移</td></tr> <tr> <td>CAN Halt モード</td><td>CAN モジュールは受信メッセージの終了を待って CAN Halt モードに遷移*2*3</td><td>CAN モジュールはメッセージ送信の終了を待って CAN Halt モードに遷移*1*4</td><td> [BOM ビットが"00"の場合] CAN モジュールはバスオフ復帰後のみ、プログラムの Halt 要求を受け付ける [BOM ビットが"01"の場合] CAN モジュールはバスオフ復帰の終了を待たずに自動的に CAN Halt モードに遷移 (プログラムの Halt 要求とは無関係に) [BOM ビットが"10"の場合] CAN モジュールはバスオフ復帰の終了を待って自動的に CAN Halt モードに遷移 (プログラムの Halt 要求とは無関係に) [BOM ビットが"11"の場合] CAN モジュールはバスオフ中にプログラムによる Halt 要求があると、CAN Halt モードに遷移 (バスオフ復帰の終了を待たずに) </td></tr> </tbody> </table> 【注】 *1 いくつかのメッセージが送信要求されている場合、最初の送信完了後にモードを遷移します。サスペンドトランスミッション中に CAN リセットモードが要求されている状態では、バスアイドルになったとき、次の送信が終了したとき、または CAN モジュールが受信になったときに、モードを遷移します。 *2 CAN バスがドミナントレベルでロックされた場合、CIEIFR レジスタの BLIF ビットをモニタすると、プログラムはバスロック状態を検出できます。 *3 CAN Halt モードが要求された後、受信中に CAN バスエラーが発生すると、CAN Halt モードに遷移します。 *4 CAN リセットモードまたは CAN Halt モードが要求された後、送信中に CAN バスエラーまたは CAN アービトラレーションロストが発生すると、要求された CAN モードに遷移します。 【記号説明】 BOM ビット：CiCTRL レジスタのビット (i=0~3)	モード	受信	送信	バスオフ	CAN リセットモード (強制遷移) CANM="11"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待たずに CAN リセットモードに遷移	CAN モジュールはバスオフ復帰の終了を待たずに CAN リセットモードに遷移	CAN リセットモード CANM="01"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待って CAN リセットモードに遷移*1*4	CAN モジュールはバスオフ復帰の終了を待たずに CAN リセットモードに遷移	CAN Halt モード	CAN モジュールは受信メッセージの終了を待って CAN Halt モードに遷移*2*3	CAN モジュールはメッセージ送信の終了を待って CAN Halt モードに遷移*1*4	[BOM ビットが"00"の場合] CAN モジュールはバスオフ復帰後のみ、プログラムの Halt 要求を受け付ける [BOM ビットが"01"の場合] CAN モジュールはバスオフ復帰の終了を待たずに自動的に CAN Halt モードに遷移 (プログラムの Halt 要求とは無関係に) [BOM ビットが"10"の場合] CAN モジュールはバスオフ復帰の終了を待って自動的に CAN Halt モードに遷移 (プログラムの Halt 要求とは無関係に) [BOM ビットが"11"の場合] CAN モジュールはバスオフ中にプログラムによる Halt 要求があると、CAN Halt モードに遷移 (バスオフ復帰の終了を待たずに)
モード	受信	送信	バスオフ																
CAN リセットモード (強制遷移) CANM="11"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待たずに CAN リセットモードに遷移	CAN モジュールはバスオフ復帰の終了を待たずに CAN リセットモードに遷移																
CAN リセットモード CANM="01"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待って CAN リセットモードに遷移*1*4	CAN モジュールはバスオフ復帰の終了を待たずに CAN リセットモードに遷移																
CAN Halt モード	CAN モジュールは受信メッセージの終了を待って CAN Halt モードに遷移*2*3	CAN モジュールはメッセージ送信の終了を待って CAN Halt モードに遷移*1*4	[BOM ビットが"00"の場合] CAN モジュールはバスオフ復帰後のみ、プログラムの Halt 要求を受け付ける [BOM ビットが"01"の場合] CAN モジュールはバスオフ復帰の終了を待たずに自動的に CAN Halt モードに遷移 (プログラムの Halt 要求とは無関係に) [BOM ビットが"10"の場合] CAN モジュールはバスオフ復帰の終了を待って自動的に CAN Halt モードに遷移 (プログラムの Halt 要求とは無関係に) [BOM ビットが"11"の場合] CAN モジュールはバスオフ中にプログラムによる Halt 要求があると、CAN Halt モードに遷移 (バスオフ復帰の終了を待たずに)																

Rev.	ページ	項目	内容																
Rev.B 追加	26-78	表 26.9 CAN リセットモードと CAN Halt モードでの動作	<table border="1"> <thead> <tr> <th>モード</th><th>受信</th><th>送信</th><th>バスオフ</th></tr> </thead> <tbody> <tr> <td>CAN リセットモード (強制遷移) CANM="11"</td><td>CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移</td><td>CAN モジュールはメッセージ送信の終了を待たずに CAN リセットモードに遷移</td><td>CAN モジュールはバスオフ復帰を待たずに CAN リセットモードに遷移</td></tr> <tr> <td>CAN リセットモード CANM="01"</td><td>CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移</td><td>CAN モジュールはメッセージ送信の終了を待って CAN リセットモードに遷移^{*1*}</td><td>CAN モジュールはバスオフ復帰を待たずに CAN リセットモードに遷移</td></tr> <tr> <td>CAN Halt モード</td><td>CAN モジュールは受信メッセージの終了を待って CAN Halt モードに遷移^{*2*}</td><td>CAN モジュールはメッセージ送信の終了を待って CAN Halt モードに遷移^{*1*}</td><td> [BOM ビットが"00"の場合] CAN モジュールはバスオフ復帰後のみ、プログラムの Halt 要求を受け付ける [BOM ビットが"01"の場合] CAN モジュールはバスオフ復帰を待たずに自動的に CAN Halt モードに遷移（プログラムの Halt 要求とは無関係に） [BOM ビットが"10"の場合] CAN モジュールはバスオフ復帰を待って自動的に CAN Halt モードに遷移（プログラムの Halt 要求とは無関係に） [BOM ビットが"11"の場合] CAN モジュールはバスオフ中にプログラムによる Halt 要求があると、CAN Halt モードに遷移（バスオフ復帰を待たずに） </td></tr> </tbody> </table> 【注】 *1 いくつかのメッセージが送信要求されている場合、最初の送信完了後にモードを遷移します。サスペンドトランスミッション中に CAN リセットモードが要求されている状態では、バスアイドルになったとき、次の送信が終了したとき、または CAN モジュールが受信になったときに、モードを遷移します。 *2 CAN バスがドミナントレベルでロックされた場合、CIEIFR レジスタの BLIF ビットをモニタすると、プログラムはバスロックを検出できます。CAN バスがドミナントレベルでロックされている間は、CAN Halt モードに遷移しません。この場合は、CAN リセットモードに遷移してください。 *3 CAN Halt モードが要求された後、受信中に CAN バスエラーが発生すると、CAN Halt モードに遷移します（ただし、CAN バスがドミナントレベルでロックされている場合は CAN Halt モードに遷移しません）。 *4 CAN リセットモードまたは CAN Halt モードが要求された後、送信中に CAN バスエラーまたは CAN アービトラレーションロストが発生すると、要求された動作モードに遷移します（ただし、CAN バスがドミナントレベルでロックされている場合は CAN Halt モードに遷移しません）。 【記号説明】 BOM ビット：CICTLR レジスタのビット（i=0~3）	モード	受信	送信	バスオフ	CAN リセットモード (強制遷移) CANM="11"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待たずに CAN リセットモードに遷移	CAN モジュールはバスオフ復帰を待たずに CAN リセットモードに遷移	CAN リセットモード CANM="01"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待って CAN リセットモードに遷移 ^{*1*}	CAN モジュールはバスオフ復帰を待たずに CAN リセットモードに遷移	CAN Halt モード	CAN モジュールは受信メッセージの終了を待って CAN Halt モードに遷移 ^{*2*}	CAN モジュールはメッセージ送信の終了を待って CAN Halt モードに遷移 ^{*1*}	[BOM ビットが"00"の場合] CAN モジュールはバスオフ復帰後のみ、プログラムの Halt 要求を受け付ける [BOM ビットが"01"の場合] CAN モジュールはバスオフ復帰を待たずに自動的に CAN Halt モードに遷移（プログラムの Halt 要求とは無関係に） [BOM ビットが"10"の場合] CAN モジュールはバスオフ復帰を待って自動的に CAN Halt モードに遷移（プログラムの Halt 要求とは無関係に） [BOM ビットが"11"の場合] CAN モジュールはバスオフ中にプログラムによる Halt 要求があると、CAN Halt モードに遷移（バスオフ復帰を待たずに）
モード	受信	送信	バスオフ																
CAN リセットモード (強制遷移) CANM="11"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待たずに CAN リセットモードに遷移	CAN モジュールはバスオフ復帰を待たずに CAN リセットモードに遷移																
CAN リセットモード CANM="01"	CAN モジュールは受信メッセージの終了を待たずに CAN リセットモードに遷移	CAN モジュールはメッセージ送信の終了を待って CAN リセットモードに遷移 ^{*1*}	CAN モジュールはバスオフ復帰を待たずに CAN リセットモードに遷移																
CAN Halt モード	CAN モジュールは受信メッセージの終了を待って CAN Halt モードに遷移 ^{*2*}	CAN モジュールはメッセージ送信の終了を待って CAN Halt モードに遷移 ^{*1*}	[BOM ビットが"00"の場合] CAN モジュールはバスオフ復帰後のみ、プログラムの Halt 要求を受け付ける [BOM ビットが"01"の場合] CAN モジュールはバスオフ復帰を待たずに自動的に CAN Halt モードに遷移（プログラムの Halt 要求とは無関係に） [BOM ビットが"10"の場合] CAN モジュールはバスオフ復帰を待って自動的に CAN Halt モードに遷移（プログラムの Halt 要求とは無関係に） [BOM ビットが"11"の場合] CAN モジュールはバスオフ中にプログラムによる Halt 要求があると、CAN Halt モードに遷移（バスオフ復帰を待たずに）																
Rev.B 追加	28-11	28.3 レジスタの説明	28.3 レジスタの説明 DMA 転送要求許可ビットの説明を訂正 ・また、DMA の誤動作を防ぐため、DRI 取り込み許可（DRIIDCAPCNT.DCPEN ビット="1"）時は <u>DMA 転送</u> マスクから <u>DMA 転送許可</u> への書き換えのみとしてください。DRI 取り込み許可時に <u>DMA 転送許可</u> から <u>DMA 転送</u> マスクへの書き換えをすると正常に <u>DMA 要求</u> が処理されない場合がありますので行わないでください。 ・また、DMA の誤動作を防ぐため、DRI 取り込み許可（DRIIDCAPCNT.DCPEN ビット="1"）時は <u>DMA 転送要求</u> マスクから <u>DMA 転送要求許可</u> への書き換えのみとしてください。DRI 取り込み許可時に <u>DMA 転送要求許可</u> から <u>DMA 転送要求</u> マスクへの書き換えをすると正常に <u>DMA 転送要求</u> が処理されない場合がありますので行わないでください。																
Rev.B 追加	28-16	28.3.4 DRI0DIN DMA 転送許可レジスタ（DRI0DINDEN）	28.3.4 DRI0DIN DMA 転送許可レジスタ（DRI0DINDEN）の説明を訂正 ・DMA 転送 <u>禁止</u> マスク（禁止）設定と内部 DMA 転送要求が同時に発生した場合は、<u>DMA 転送</u> マスク（禁止）が優先されます。 また、DRI 取り込み許可（DRIIDCAPCNT.DCPEN ビット="1"）時における DRI0DINDEN レジスタの書き換えは、<u>転送</u> マスクから <u>転送許可</u> に書き換えることのみ可能です。DRI 取り込み許可時は、<u>転送許可</u> から <u>転送</u> マスクへの書き換えを行わないでください。 ・DMA 転送 <u>要求</u> マスク（禁止）設定と内部 DMA 転送要求が同時に発生した場合は、<u>DMA 転送要求</u> マスク（禁止）が優先されます。 また、DRI 取り込み許可（DRIIDCAPCNT.DCPEN ビット="1"）時における DRI0DINDEN レジスタの書き換えは、<u>DMA 転送要求</u> マスクから <u>DMA 転送要求許可</u> に書き換えることのみ可能です。DRI 取り込み許可時は、<u>DMA 転送要求許可</u> から <u>DMA 転送要求</u> マスクへの書き換えを行わないでください。																

Rev.	ページ	項目		内容																				
Rev.B 追加	28-21	28.3.8 DRI0DEC DMA 転送 許可レジスタ (DRI0DEC DEN)	誤	28.3.8 DRI0DEC DMA 転送許可レジスタ（DRI0DEC DEN）の説明を訂正 ・DMA 転送禁止マスク（禁止）設定と内部 DMA 転送要求が同時に発生した場合は、DMA 転送マスク（禁止）が優先されます。 また、DEC 動作許可（DRIiDECnCNT.DECnEN ビット="1"）時における DRI0DEC DEN レジスタの書き換えは、転送マスクから転送許可に書き換えることのみ可能です。DEC 動作許可時は、転送許可から転送マスクへの書き換えを行わないでください。																				
			正	・DMA 転送要求マスク（禁止）設定と内部 DMA 転送要求が同時に発生した場合は、DMA 転送要求マスク（禁止）が優先されます。 また、DEC 動作許可（DRIiDECnCNT.DECnEN ビット="1"）時における DRI0DEC DEN レジスタの書き換えは、DMA 転送要求マスクから DMA 転送要求許可に書き換えることのみ可能です。DEC 動作許可時は、DMA 転送要求許可から DMA 転送要求マスクへの書き換えを行わないでください。																				
Rev.B 追加	28-27	28.3.12 DRI0DMA 転送許可 レジスタ (DRI0TRMDEN)	誤	28.3.12 DRI0DMA 転送許可レジスタ（DRI0TRMDEN）の説明を訂正 ・DMA 転送禁止マスク（禁止）設定と内部 DMA 転送要求が同時に発生した場合は、DMA 転送マスク（禁止）が優先されます。また、DRI 取り込み許可（DRIiDCAPCNT.DCPEN ビット="1"）時における DRI0TRMDEN レジスタの書き換えは、転送マスクから転送許可に書き換えることのみ可能です。DRI 取り込み許可時は、転送許可から転送マスクへの書き換えを行わないでください。																				
			正	・DMA 転送要求マスク（禁止）設定と内部 DMA 転送要求が同時に発生した場合は、DMA 転送要求マスク（禁止）が優先されます。また、DRI 取り込み許可（DRIiDCAPCNT.DCPEN ビット="1"）時における DRI0TRMDEN レジスタの書き換えは、DMA 転送要求マスクから DMA 転送要求許可に書き換えることのみ可能です。DRI 取り込み許可時は、DMA 転送要求許可から DMA 転送要求マスクへの書き換えを行わないでください。																				
Rev.B 追加	32-13	32.4.1 FlexRay 動作制御レ ジスタ（FXROC）	誤	32.4.1 FlexRay 動作制御レジスタ（FXROC）の FBSEN ビット説明の誤記を訂正 ・FRNVMn																				
			正	・FRNVMn ・FRNMVn ・FRNMVn																				
Rev.B 追加	32-152	表 32.8 FlexRay モジュール の状態遷移条件	誤	表 32.8 FlexRay モジュールの状態遷移条件の誤記を訂正 <table><tr><th>T#</th><th>遷移条件</th><th>遷移前の状態</th><th>遷移後の状態</th></tr><tr><td>T1</td><td>リセット</td><td>すべての状態</td><td>DEFAULT CONFIG</td></tr><tr><td>T2</td><td>CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")</td><td>DEFAULT CONFIG</td><td>CONFIG</td></tr><tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr><tr><td>T15</td><td>CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")</td><td>HALT</td><td>DEFAULT CONFIG</td></tr></table>	T#	遷移条件	遷移前の状態	遷移後の状態	T1	リセット	すべての状態	DEFAULT CONFIG	T2	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	DEFAULT CONFIG	CONFIG	⋮	⋮	⋮	⋮	T15	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	HALT	DEFAULT CONFIG
			T#	遷移条件	遷移前の状態	遷移後の状態																		
T1	リセット	すべての状態	DEFAULT CONFIG																					
T2	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	DEFAULT CONFIG	CONFIG																					
⋮	⋮	⋮	⋮																					
T15	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	HALT	DEFAULT CONFIG																					
正	<table><tr><th>T#</th><th>遷移条件</th><th>遷移前の状態</th><th>遷移後の状態</th></tr><tr><td>T1</td><td>リセット</td><td>すべての状態</td><td>DEFAULT CONFIG</td></tr><tr><td>T2</td><td>CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")</td><td>DEFAULT CONFIG</td><td>CONFIG</td></tr><tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr><tr><td>T15</td><td>CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")</td><td>HALT</td><td>DEFAULT CONFIG</td></tr></table>	T#	遷移条件	遷移前の状態	遷移後の状態	T1	リセット	すべての状態	DEFAULT CONFIG	T2	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	DEFAULT CONFIG	CONFIG	⋮	⋮	⋮	⋮	T15	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	HALT	DEFAULT CONFIG			
T#	遷移条件	遷移前の状態	遷移後の状態																					
T1	リセット	すべての状態	DEFAULT CONFIG																					
T2	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	DEFAULT CONFIG	CONFIG																					
⋮	⋮	⋮	⋮																					
T15	CONFIG コマンド発行 (CMD3～CMD0 ビット="B'0001")	HALT	DEFAULT CONFIG																					

以上