

No	該当シークエンス	エラー項目	該当エラー	それに対する例外処理	検証方法	結果 (CIB)	結果 (OBC)
	OBC起動しない	さ	OBC 故障	CIBが溶断シークエンス どちらも溶断が始まってしまう可能性あり →時間差でうまく調整する？ →EEPROMに溶断中かどうかを書き込む？ ありだと思う。溶断ステータスを未溶断、溶断中、溶断済みでわけるのはいいと思う。 井手			
	IO_PIN_50セット	あ	ioピン不良 OBC出力の不良は経験ない CIB側の入力不良が心配 井手		保留		
	EPSスイッチ全オフ	A	OBC→EPS I2Cerror	エラーが起きたらEPSキルをRXに送信？ →出来なかったらそのままでもいい？→応スイッチのリミットタイマーを打ち上げ前につけておく(30秒) ・リミットタイマーは初期運用後に停止できるよね？ 毎回30秒で止まってしまうのか心配 岩崎	リミットタイマー関数デバック済みのため、特に行わない リミットタイマーはSW10のみの実装予定		
	溶断ステータス読み込み (溶断前に必ず読み込み)	B	OBC→EEPROMI2Cerror	SDに溶断ステータス書き込み さらにエラーでRXにリセットコマンド →無限ループに入る可能性 RXリセットコマンドをやめたとしてどうするか？ →読み込めないで溶断開始 I2cエラーが出たらタスク停止、ioピン50を切ってCIBが溶断 ただし、溶断シークエンスに入る前の読み込みは片方が溶断済とわかれば通常運用へ	I2cエラーがでたと想定してiopinを切ってタスクを停止 →CIBが溶断シークエンスに入れるかを確認する	iopinを切ることで死亡判定 →TXにUART送信して2秒間隔で溶断できていることを確認(putcの文字間隔で確認)	
	通常溶断SW	A	OBC→EPS I2Cerror	エラーが起きたら通常溶断終了、冗長溶断へ →通常ができなければ冗長予定だがUART割り込みも失敗したらどうするのか？ →ioピンをきらない限りCIBの溶断にはならない→どちらも失敗する可能性を考慮する必要あり →一回失敗したら終わる処理ではなく、ループで何度もUARTを送る処理になってなかったっけ？ 井手	I2cエラーが出てそのままプログラムが流れるので特に検証を行わない		
	通常溶断SW	お	EPS 溶断SW	通常溶断不可→冗長溶断へ	I2cエラーが出てそのままプログラムが流れるので特に検証を行わない		
	OBCが通常溶断中	1	OBCのみ再起動	OBCのインシャライズでEPS全スイッチオフ RXがOBCの起動確認してEPSキル →スイッチのリミットのタイマーつける(30秒) ニクロム線の焼ききれる可能性があるが OBCだけオフになる可能性が低いため考慮しない	溶断中にOBCを再起動させEPSの溶断SWをオフに出来るかを確認	関数の機能確認済みだが別のデバックでおかし 関数修正終わり次第再確認	
	OBCが通常溶断中	2	OBCオフ		溶断中にOBCを再起動させEPSの溶断SWをオフに出来るかを確認 特に対策はないことになった		
	アップリンク待ち	1	OBCのみ再起動	何回溶断したかをEEPROMに保存、再起動する値を読んで再開	OBCを途中で再起動させ再度読み込みを行うかを検証		○
	溶断回数読み込み	B	OBC→EEPROMI2Cerror	エラーが出るのとからスタートして通常溶断になる →溶断ステータスと同様にタスク停止でCIBに頼む？ ・溶断するのが確定で回数で変わるのが溶断時間だけなら OBCで溶断してもいい気がする 岩崎	溶断回数読み込み時にエラーを起こして0から始まるかを確認する		○
	OBC冗長系コマンド送信中	1	OBCのみ再起動	TXがコマンド破棄 CRCではじけるはず	あえて途中で途切れたUARTを送信後OBCを再起動		
	OBC冗長系コマンド送信中	き	マルチプレクサ	冗長系溶断使用できない通常溶断へ			
	OBC冗長系コマンド送信中	C	OBC→TX UARTerror	冗長系溶断使用できない通常溶断へ	今のuartの回数って返り値ないから成功の判断をどうするかが問題だね 岩崎 IO_PINのHigh、Lowの組み合わせでチェックできない？ 井手		