

制御 PC 電源瞬断ソリューション(RTC)



株式会社マイクロネット

<http://www.mnc.co.jp>

TEL: +81(0)299-90-1733

FAX: +81(0)299-92-8557




目次

1	概要	4
2	瞬断イベント処理	4
3	制御対象を安全状態にする	5
4	上位システムに通知する対処	6
5	メモリ上のデータを退避する対処	6
6	瞬断トリガーツール起動パラメータ設定	7
7	改訂履歴	7

※本ドキュメントの内容は予告なく変更される可能性があります。

また、本ドキュメントの無断転載・使用を固く禁じます。

本書で使用するマークについて

	ノート: 操作方法や手順等の補足情報や注釈を説明しています。
	情報: 製品を利用する上で有効な豆知識となる説明をしています。
	警告: 製品仕様上注意が必要な事象について説明しています。

Windows、Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

INtime は、米国 TenAsys Corporation の登録商標です。

TenAsys®, INtime®, eVM® and iRMX® are registered trademarks in USA of the TenAsys Corporation.

その他、本書に記載されている会社名、商品名は、各社の商標または登録商標です。

本書の内容を無断で転載することは禁止されています。

本書の内容に関しては、予告なしに変更することがあります。あらかじめご了承ください。

※本ドキュメントの内容は予告なく変更される可能性があります。

また、本ドキュメントの無断転載・使用を固く禁じます。

1 概要

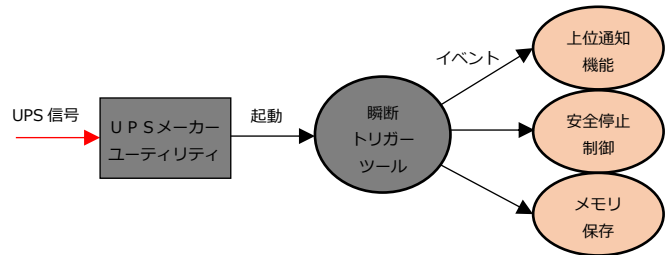
本ドキュメントは、電源瞬断により RT-C 言語コントローラを正しく終了処理を行えない事で制御機器の危険動作やデータ破損などの対策方法について記述しています。



本ドキュメントをお読みいただく前に「制御 PC 電源瞬断ソリューション(共通)」ドキュメントをお読みください

2 瞬断イベント処理

電源瞬断により UPS ベンダーが提供している「UPS ユーティリティ」から弊社の「瞬断トリガーツール」起動する事により各処理にイベントを発行する事が出来ます。



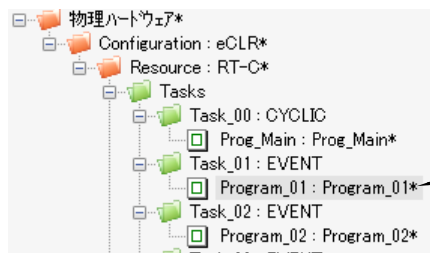
RT-C 言語コントローラはイベントタスクで瞬断トリガーツールからのイベントを待ちます。

イベントを受けたイベントタスクで電源瞬断に必要なイベント処理を行ってください。



通知するイベントタスク No は、瞬断トリガーツールの起動パラメータで変更する事が出来ます。

プログラミング例（※制御内容はシステムによって異なります）



瞬断トリガーツールからイベントタスクでイベント通知受ける

```
static short On_Process()
{
    Int16 inReturn = CRTC.STS_PRO;

    // 以下、ユーザが通常処理を記述します。
    // 制御対象を安全状態にする
    GoToSafe();
    // メモリデータの保存
    SaveMemoryToDisk();
    // 上位サーバへの通知
    ReportToServer();

    // ユーザが記述する通常処理はここまでです。

    return inReturn;
}
```

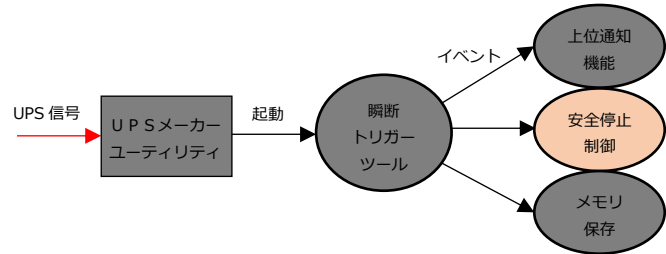
瞬断トリガーツールからのイベント通知を受けたイベントタスク動作するプログラム内で瞬断対処を行います。

処理はユーザーがシステムに合わせて作成します。

3 制御対象を安全状態にする

電源瞬断に必要なリアクションの1つとして「安全状態にする」があります。

瞬断によりUPSから信号を受け制御機器を安全状態（動作を停止させる、釣り上げている物を下す、ブレーキをかけておくなど）にする必要があります。



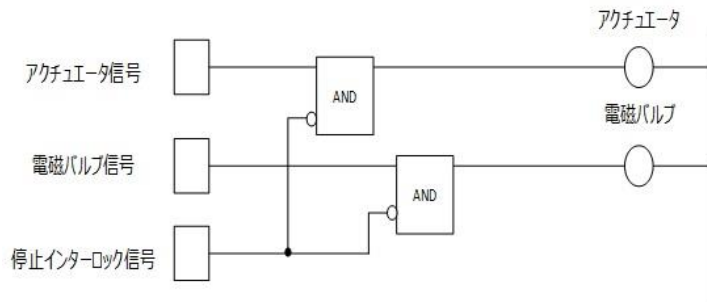
例：安全状態にする

1. 制御信号出力

- 1) アクチュエータ信号の OFF
- 2) 電磁バルブ信号の OFF
- 3) 停止インターロック信号の ON

2. モーター制御

- 1) 安全位置に移動させる
- 2) 電磁ブレーキをかけて保持する



プログラミング例（※制御内容はシステムによって異なります）

```
static void GoToSafe()
{
    // アクチュエータ制御信号をOFF
    pcibus.OutBYTE(actuator, 0x00);
    // 電磁バルブ制御信号をOFF
    pcibus.OutBYTE(valve, 0x00);
    // 停止インターロック信号をON
    pcibus.OutBYTE(safetylock, 0x01);
}
```

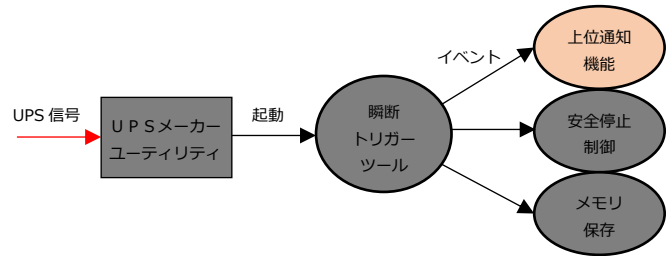


イベントタスクは1回だけ実行されます。
1回のシーケンスで終わらない場合は、イベントタスクで終了イベント通知が来たことをグローバル変数に保持し別のサイクリックタスクで処理を行ってください。

4 上位システムに通知する対処

電源瞬断に必要なリアクションの1つとして「上位システムに通知」があります。

瞬断によりUPSより信号を受け制御PCをシャットダウンする事を上位システムに通知が必要です。



通信回線用ファンクション「RC232C ファンクション」「TCP/IP ファンクション」などを使用して上位に通知する事が出来ます。

プログラミング例（※制御内容はシステムによって異なります）

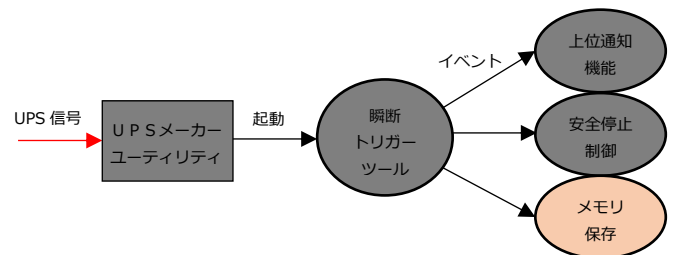
```
addrinfo hints = new addrinfo();
addrinfo* ai = null;
hints.ai_family = (int)SOCK_AF_INET;
hints.ai_socktype = (int)SOCK_TYPES.SOCK_DGRAM;
hints.ai_protocol = (int)SOCK_IPPROTO_IPPROTO_UDP;
hints.ai_flags = 0;
error = Sock.getaddrinfo(ServerBuff, PortBuff, hints, (addrinfo*)&ai);
sock = Sock.socket(ai->ai_family, (int)SOCK_TYPES.SOCK_DGRAM, 0);
i_status = Sock.connect(sock, ref *ai-> ai_addr, ai-> ai_addrlen);
bytes_sent = Sock.send(sock, bufptr, sendamt, 0);
Sock.freeaddrinfo(ai);
```

左の例はUDPで設定されているサーバにメッセージを送信する例です。

5 メモリ上のデータを退避する対処

電源瞬断に必要なリアクションの1つとして「メモリ保存」があります。

メモリ内のみに保持しているデータは制御PCシャットダウン時に消えてしまう為、保持が必要なメモリ内のデータをファイルに出力します。



プログラミング例（※制御内容はシステムによって異なります）

```
IecString80 FileName = new IecString80();
FileName.ctor();
FileName.s.Init("Data.csv");

IecString80 Buff = new IecString80();
Buff.ctor();
Buff.s.Init("Test Data");

File myFile = File.GetFreeFileSlot();
if (myFile.Open(FileName.s.GetIecString(), (ushort)FileMode.Open) == false)
    return;
int iPosi = myFile.Tell();
myFile.Write(Buff.s.GetIecString() + iPosi, 0, 10);
myFile.Close();
myFile = null;
```

6 瞬断トリガーツール起動パラメータ設定

瞬断トリガーツールから RT-C 言語コントローラにイベント通知するには、瞬断トリガーツールの起動パラメータを下記のように設定してください。

通知するイベントタスク No	瞬断トリガーツールの起動パラメータ
イベント 0	PFevent.exe NodeA/ProConOS/SEM_EVENT_00
イベント 1	PFevent.exe NodeA/ProConOS/SEM_EVENT_01
	⋮
イベント 14	PFevent.exe NodeA/ProConOS/SEM_EVENT_13
イベント 15	PFevent.exe NodeA/ProConOS/SEM_EVENT_15

7 改訂履歴

版数	発行日	改定内容
第 1 版	2020 年 7 月	初版発行