

SD330自動同調システム ソフトウェア説明書

MPU編(R8C／38A)

2016年4月1日

Narichan

1、プログラムの概要

ホスト(PC)からのコマンドによって動作を開始します。常にコマンド待ちがアイドリングです。ハードウェア説明書にあるようにDDS、UART(USB)、モータードライブ回路、リレーコントロールなどを制御します。

DDSはAD9850を使用しています。周波数の指定と発振、停止を行います。

DDSの出力はリターンロスブリッジ(RL)に繋がれています。

RLの出力としてブリッジのバランス状態をAD8307でdBmに変換されMPUのAN3に入力されます。

UARTはPCとのコマンド受信とその返信を行います。SWRの帯域のデータはMPUのRAM(10KB)に保存され、このUARTでPCに送信されます。通信は高速で約1Mbpsで通信します。

モータードライブ回路は回転方向の指定(正転、逆転、ブレーキ)と高速、低速の指定をします。

リレーコントロールはRLブリッジがアンテナに繋がるようにリレーを切り替えます。通常アンテナはリグに繋がっています。

SWRの測定は、まずリレーを切り替え、DDSにてRF信号をRLに送ります。出力をAD変換してその値(0から255)をPCに返すか、メモリーに保存します。PCに返すのは自動または手動で同調させる時、メモリーに保存するのは周波数帯域のデータを測定するときです。

MPUの使用端子と機能はハードウェア説明書を参照ください。

2、使用メモリー

使用メモリー	サイズ	備考
ROM	48856	0x4000-0xfed7
RAM	10240	0x400-0x2bff
フラッシュメモリ	4096	未使用 0x3000-0x3fff
ベクターテーブル	40	0xfed8-0xffff
最大使用ユーザスタック		
最大使用割り込みスタック		

3、ソフトウェア説明

3. 1、関数

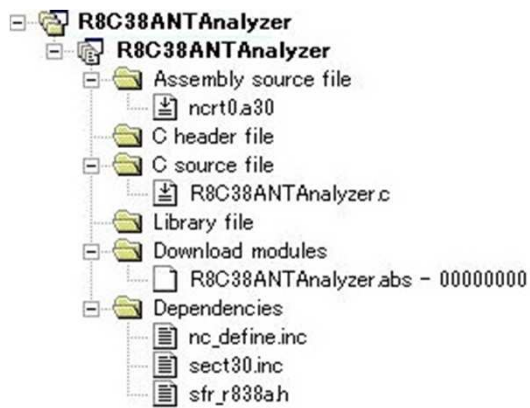
- (1) void main(void);
 概要 メイン関数
 機能説明 システムのメインルーチン
- (2) void init(void);
 概要 システムクロック設定・I/O初期設定処理
 機能説明
- (3) void pwm_stop(void);
 概要 モーターストップ処理
 機能説明
- (4) void pwm_start_U(void);
 概要 モーター高速正転開始処理
 機能説明
- (5) void pwm_start_D(void);
 概要 モーター高速逆転開始処理
 機能説明
- (6) void pwm_start_U_Low(void);
 概要 モーター低速正転開始処理
 機能説明
- (7) void pwm_start_D_Low(void);
 概要 モーター低速逆転開始処理
 機能説明
- (8) void timer(unsigned long timer_set);
 概要 10mSシステムタイマー処理
 機能説明
- (9) int get_uart1(unsigned char *s);
 概要 UART受信処理
 機能説明

- (10) int put_uart1(unsigned char r);
 概要 UART送信処理
 機能説明
- (11) void Send_UART_s(char *put_string);
 概要 UART文字列送信処理
 機能説明
- (12) void DDS_DATA_put(unsigned long freq_d);
 概要 DDS指定周波数設定処理
 機能説明
- (13) void DDS_RESET_(void);
 概要 DDS周波数ゼロ出力処理
 機能説明
- (14) void DDS_INIT(void);
 概要 DDSシリアル通信設定処理
 機能説明
- (15) int DDS_FSET_(void);
 概要 帯域内スイープ周波数設定処理
 機能説明
- (16) void all_f_seep(void);
 概要 HF全帯域スイープ処理
 機能説明
- (17) void min_f_seep(void);
 概要 指定周波数・帯域設定処理
 機能説明
- (18) void DDS_VOLT_put(unsigned long freq_d);
 概要 DA変換電圧出力処理
 機能説明
- (19) unsigned char Low_pass_filter(int a,unsigned char One_before,unsigned char present_value);
 概要 デジタルLPF処理
 機能説明
- (20) unsigned char cnv_uart(unsigned char r);
 概要 特定キャラクタ変換処理
 機能説明

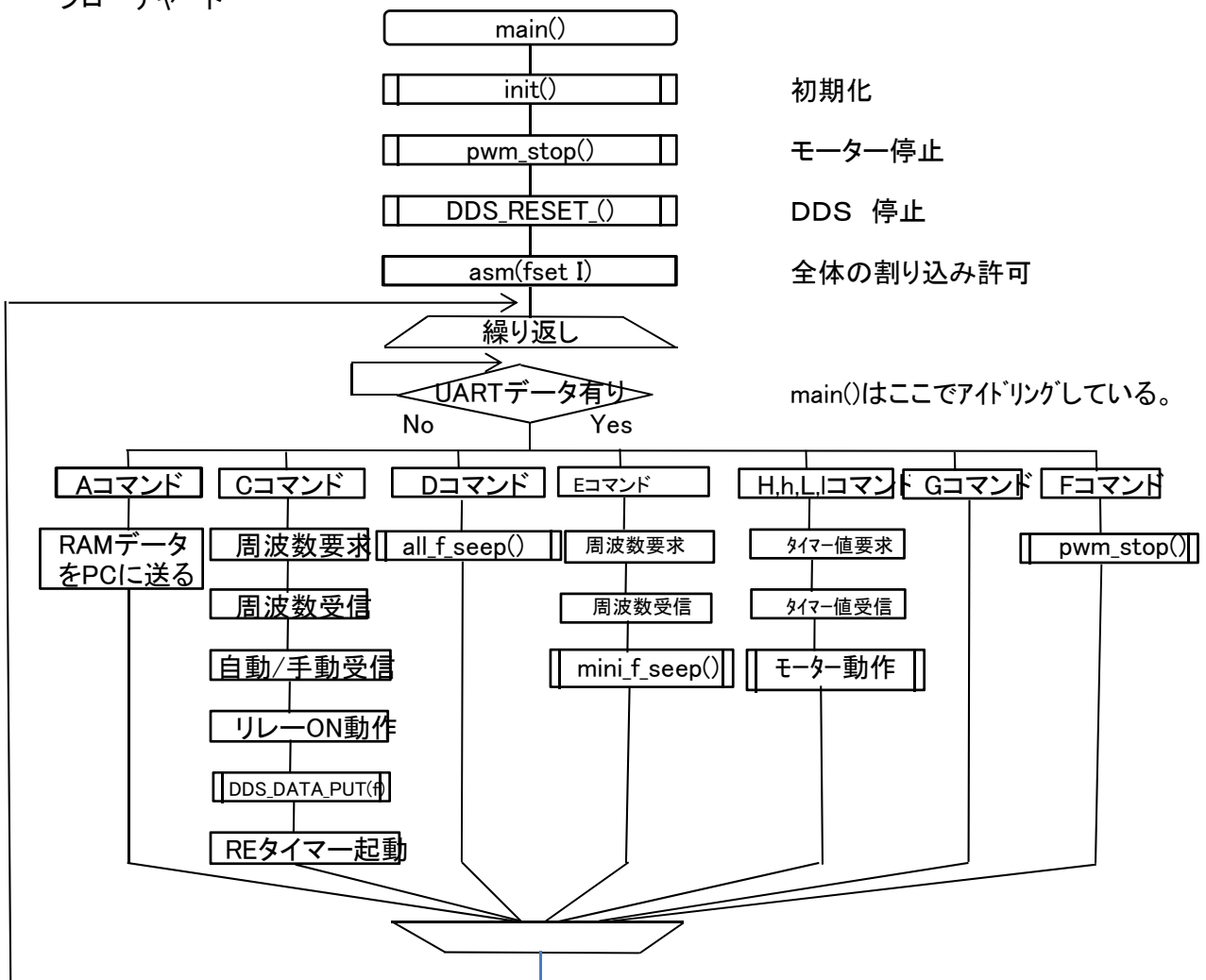
3. 2、割り込みルーチン

- (1) void intTRE(void)
 ベクター vect=10
 概要 タイマーRE処理
 機能説明 SWR測定時 5. 12mSごとにAD変換開始または512mSごとにAD値をUART送信
- (2) void intTRB(void)
 ベクター vect=24
 概要 タイマーRB処理
 機能説明 常時 10mSごとにカウントアップ WAIT(void timer(unsigned long timer_set)関数用
 モーターのオーバーラン用タイマー
- (3) void intTRA(void)
 ベクター vect=22
 概要 タイマーRA処理
 機能説明 帯域SWR測定時 10 μ Sごとにカウントアップ 10 μ SでAD変換開始、30 μ Sで次の周波数
- (4) void intAD_E(void)
 ベクター vect=14
 概要 AD変換処理
 機能説明 AD変換完了時データーをメモリに格納、またはSWR最低を検出する

3. 3R8Cセクション構成



3. 4 メイン関数 フローチャート



PCとのコマンド一覧

```

//*****
// PCからのコマンド一覧(cmd_timeにセット)
// 'A':メモリデータ送信
// 'B':現在の同調周波数送信 Bコマンドは未使用
// 'C':指定周波数まで自動同調
// 'C':指定周波数の手動同調(SWR連続計測)
// 'D':全域のSWRデータ測定後メモリ
// 'E':指定周波数から1MHzSWR測定後メモリ
// 'F':モーター停止
// 'G':Cコマンド停止
// 'H,h,L,l':モーター移動開始
//                                     H:高速fアップ
//                                     h:高速fダウン
//                                     L:低速fアップ
//                                     l:低速fダウン
// 'A':自動チューニングモード 'F'返信の応答周波数4BYTEに続く1BYTE
// 'M':手動チューニングモード 'F'返信の応答周波数4BYTEに続く1BYTE
// PCへ送出する返信
// 'O':OK送信
// 'N':NG送信 コマンド実行中はN送出後次のコマンド拒否
// 'F':C,Eコマンドで周波数データ(4バイト)要求
//
// 'S':Cコマンドで一旦停止を知らせる
// 'R':Cコマンド移動完了を知らせる
// 'T':Cコマンド再スタートを知らせる
// 'V': 'H,h,L,l'コマンドでタイムアウト時間 要求
// 'X':モーターがタイムアウト停止を知らせる
// 内部コマンド
// 'Z':起動後に全域のSWR値測定後メモリ
// 0x00:現在アイドル中コマンド待ち
//*****

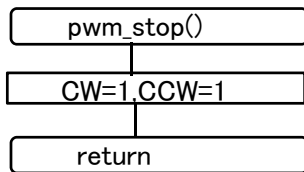
```

3. 5 システムクロック設定・I/O初期設定処理 フローチャート



UART1のクロックは外部発振器に寄っている。
 $14745.6\text{kHz} / 16 / 1 = 912.6\text{KHz}$

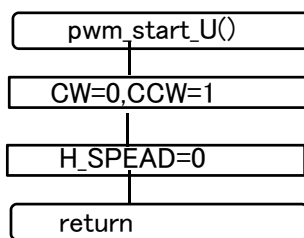
3. 6 モーターストップ処理 フローチャート



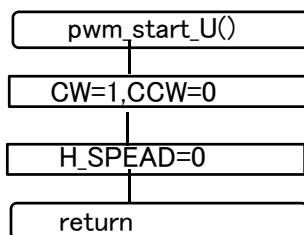
モーター動作ファンクション

CW	CCW	H_SPEAD	モード
1	1	—	ブレーキ
0	1	0	高速正転
1	0	0	高速逆転
0	1	1	低速正転
1	0	1	低速逆転

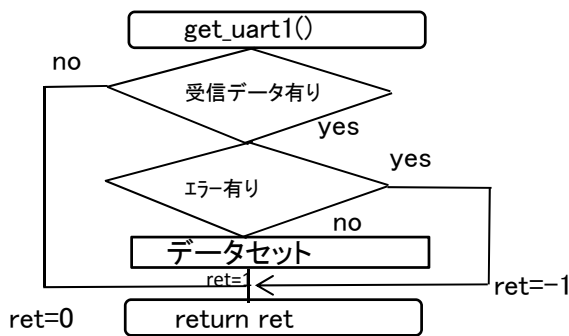
3. 7 モーター高速正転開始処理 フローチャート



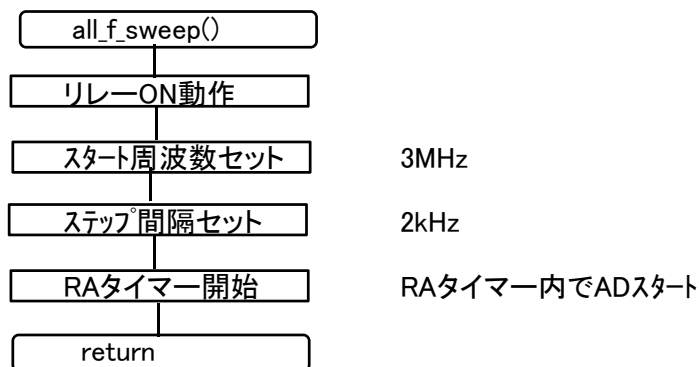
3. 8 モーター高速逆転開始処理 フローチャート



3. 9 UART受信処理 フローチャート

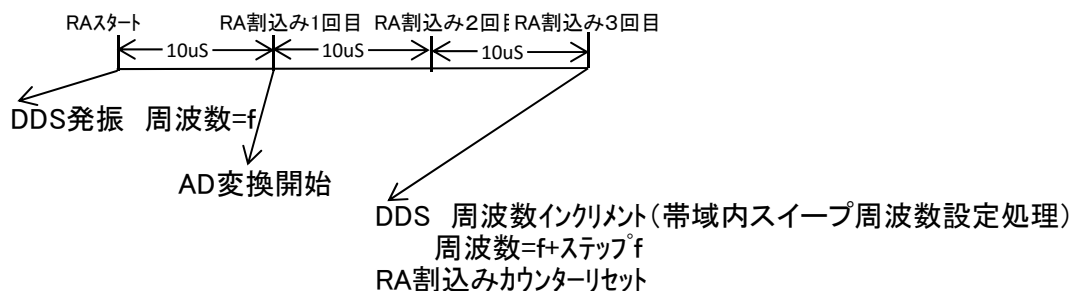


3. 10 HF全帯域スイープ処理 フローチャート



RAMメモリアドレス	周波数	ステップf
1～6000	3MHz～15MHz	2kHz
6001～8000	15MHz～25MHz	5KHz
8000～8600	25MHz～31MHz	10KHz

動作概要



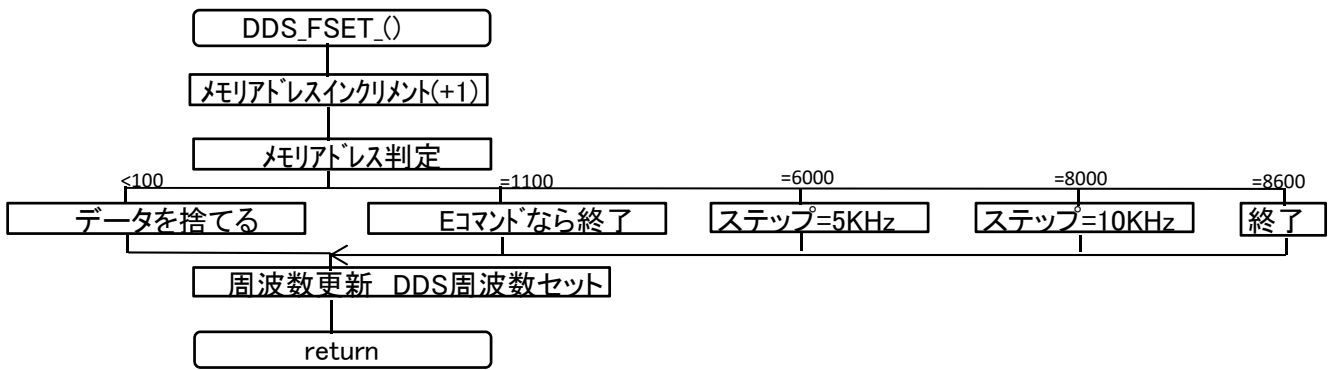
DDSを発振させ10μS後にAD変換開始する。20μS後に次の周波数を発振させる。
 AD変換は7μSで補正が5μSなので余裕をみて20μS見ている。
 周波数は15MHzまで2KHzステップ、25MHzまで5KHzステップ、31MHzまで10KHzステップとした。
 RAMは10kBYTEなので8600データ(8.6KBYTE)とした。
 値はAD変換値のみで周波数はPC側でメモリアドレスから逆算する。
 3MHzから31MHzまで0.3秒でデータを取得します。リレー動作を含めて1秒かかります。
 RAMへの書き込みはAD割り込みルーチン内で実行します。

3. 11 指定周波数・帯域設定処理

3. 10 HF全帯域スイープ処理と同じルーチンです。

スタート周波数はEコマンドで受信した周波数、ステップは1KHzで取得データは1000個(1MHz分)です。

3. 12 帯域内スイープ周波数設定処理 フローチャート

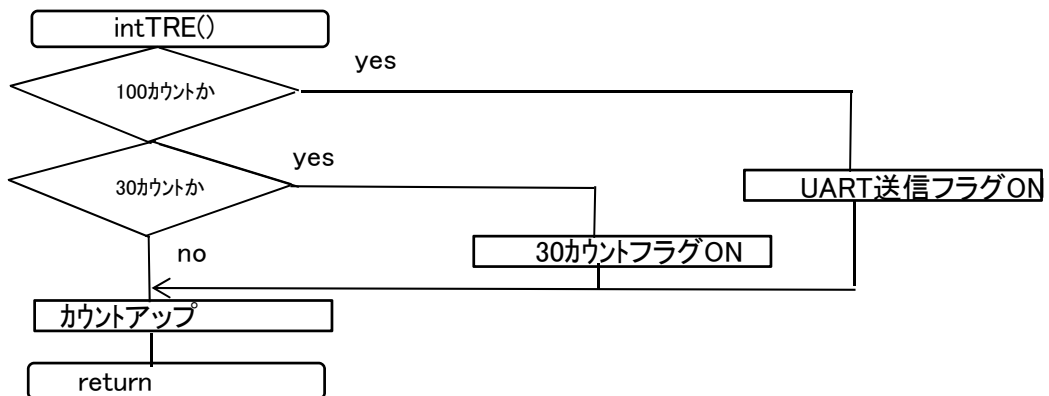


DDSが発振してリターンロスブリッジ(RLB)が値を返すが発振開始時RLBからの電圧が正常値に立ち上がるまで30周波数くらいかかる。そのため最初の100データは捨てている。

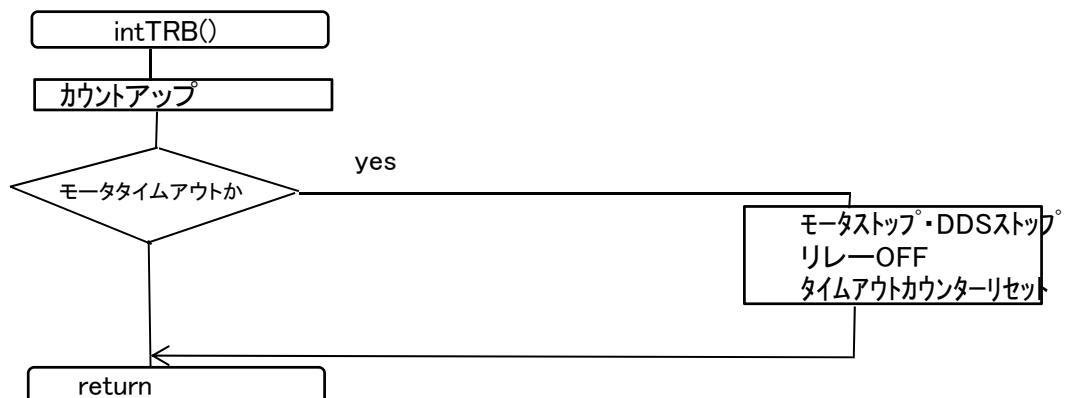
E指定周波数

Eコマンドの場合指定周波数から1MHz分(1kHzステップで1000個)取得するので1100取得して最初の100は捨てている。そのためPCからの周波数は100KHz下がった周波数を指定してくる。

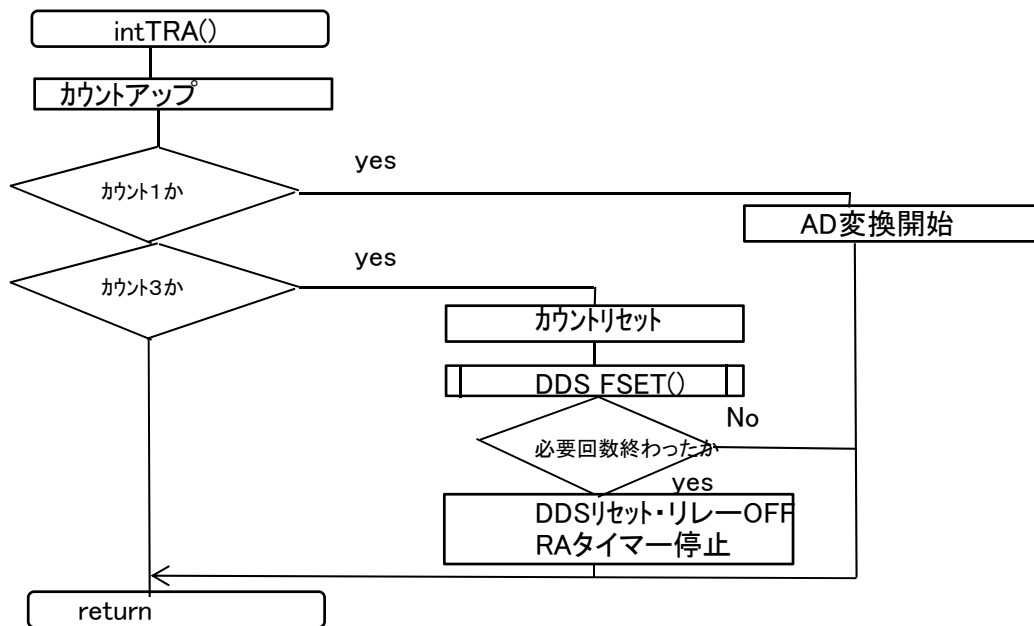
3. 21、割り込みルーチン タイマーRE処理 フローチャート



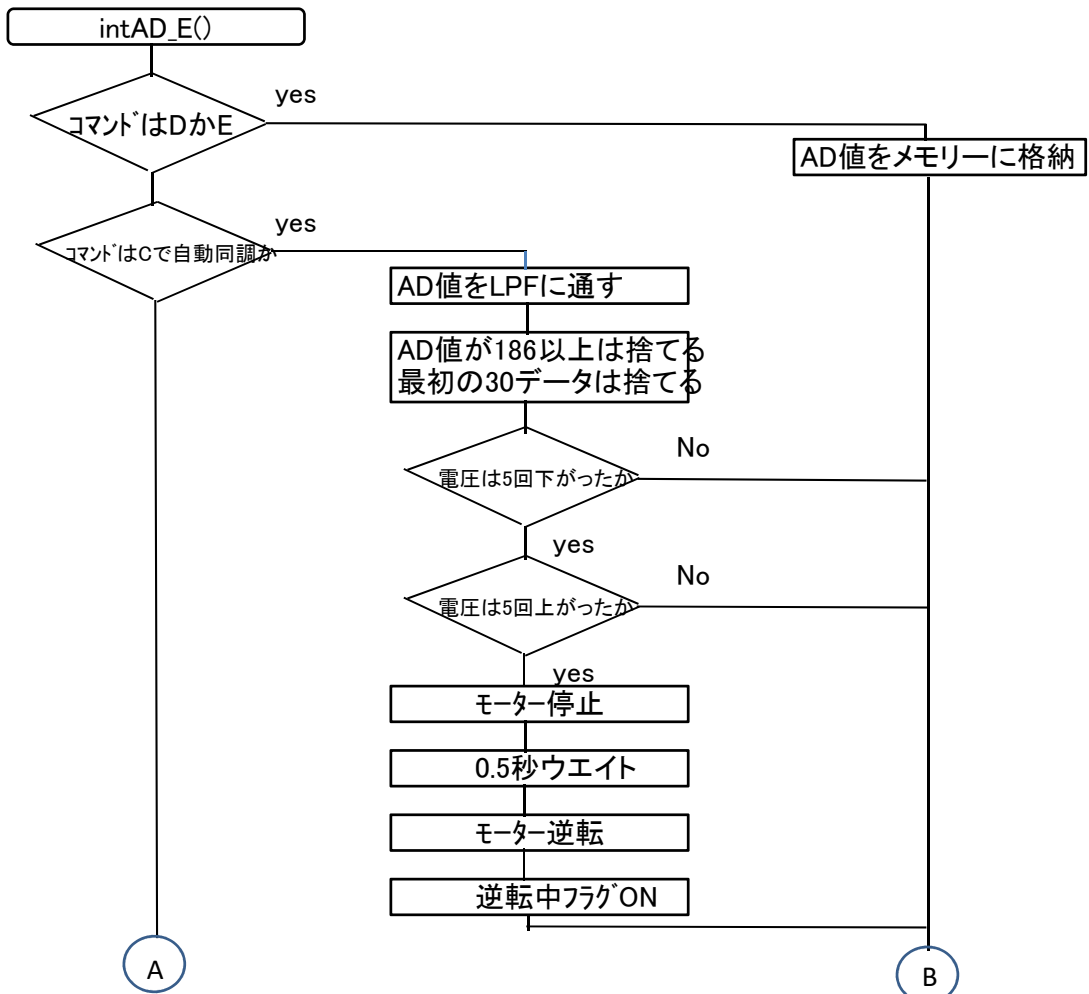
3. 22 割り込みルーチン タタイマーRB処理 フローチャート

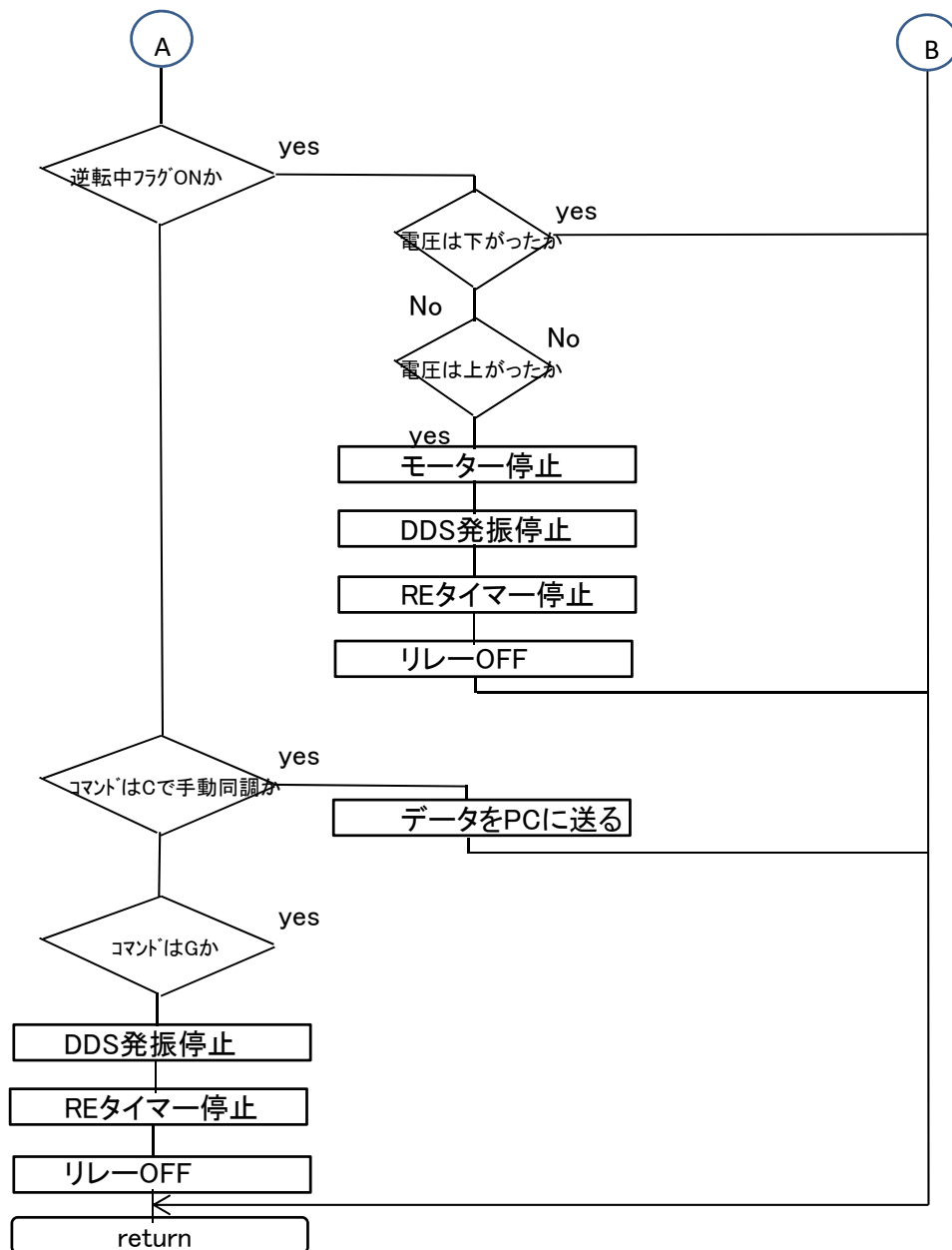


3. 23 割り込みルーチン タイマーRA処理 フローチャート



3. 24 割り込みルーチン AD変換処理 フローチャート





動作概要

このAD変換がこのシステムの要になります。

AD変換の開始はタイマー割り込みの中で行われます。HF全域のSWR測定時と指定帯域のSWR測定時はRAタイマーによっています。RAタイマーは常時は停止していますがDコマンドEコマンドを受信すると開始します。30 μ Sごとに周波数を変えAD変換を行います。

この時のAD変換値はRAMに書き込んで終わりです。その後Aコマンド受信でPCに送信します。

指定周波数のSWR測定時はREタイマー割り込みの中で行われます。REタイマーは常時は停止しています。Cコマンドを受信すると開始します。こちらは5.12mSごとに割り込みます。

手動同調の時はREタイマーで512mS(100カウント)ごとに測定データをPCに送信して終わりです。RAMには保存しません。

AD変換が割り込みで開始しますので取得した値がどのような状態で取得されたか判断しないといけません。上記のDコマンド、EコマンドCコマンドの手動時は状態を知らなくとも問題ありません。しかしCコマンドの自動同調時は、SWRが下がって行ってるのか、または同調点を過ぎて上がっているのか知らないと次の動作ができません。その観点から考察すると自動同調のAD変換は割り込みを使わず、ルーチンの中でループで値を取得したほうがすっきりするかも知れません。特にマルチタスクは必要ないのでそのほうが良かった気がします。

Cコマンド自動同調時の動作

AD変換は5.12mSごとに行われます。AD値186以上とはSWRが10以上。これは無視します。LPF通過後も値が振れたときのために5回連続して下がることを確認します。同様に上がり始めの判定も5回連続して上がることを確認します。その後逆転します。回転方向が変わるので貫通電流が流れるのを防ぐためウェイトを入れます。最低を検出したらフラグを立てて逆転に備えます。(ループ処理でなく割り込みで処理しているためです。)逆転が始まって最低を検出するのは1回でも上がればOKとしています。

前記、5回上がることを検出しますが1回でも上がれば上がり始とすれば逆転は必要ありません。しかしモーターの速度が最高速ですので必ずオーバーランします。周波数が高くて同調がブロードならそれでも良いかもしれませんが。低い周波数で2mのアンテナではクリティカル過ぎて同調しません。やむなく逆転させることとなりました。ちなみに逆転時はモーターの速度は最低速度です。最高速度の印加電圧が10.6Vで最低が1.9Vです。最低時はオーバーランが無視できるほど低速です。