**無手順Ｃ通信（別売り）**

**◎ＳＲＩＮ、ＳＲＯＴ命令**

  無手順通信のためアスキー・コード（７ＦＨ以下）のみを送ることが出来ます。

　この命令を使用したシリアル通信はフォーマットの指定は有りません。

*但し、０ＤＨ（ｷｬﾘｯｼﾞ･ﾘﾀｰﾝ）で受信は終りとし、次からまた新たな受信とします。*

*０ＡＨは常に無視されます。特に０ＤＨ、０ＡＨ（ﾗｲﾝﾌｨｰﾄﾞ）の場合は、、その次からの受信になります。*

*０Ｄはカウント数に含めます。０Ａはカウントにも含めません。（送信の場合はこの０ＤＨに特別な意味はありません）。*

　ボーレートは＜ＣＰＵ設定＞中で９６００ｂｐｓと１９２００ｂｐｓのどちらかを選択できます。

　同じ設定枠中の送信バイト、受信バイトは無関係で、両方＝０でかまいません。

　シリアル通信には、ＲＸのＲＣ７ピンとＴＸのＲＣ６ピンを使用します。

　　８ビットを１単位として送ります。　１ストップビット、パリティ無しです。

　下記の受信命令の第一オペランドで指定する受信バイト数受信が済めば、その次から新たな受信となります。

　受信命令　**ＳＲＩＮ**

例

　ＳＲＩＮ　Ｋ３０　Ｄ１８　Ｍ１８０

　　　　　　    |       |         |\_\_\_受信終了後オンする接点番号

　　　　　　    |       | 　　　（　＝＞１　ﾊﾞｯﾌｧに新ﾃﾞｰﾀが入った。ＣＰＵ内部側ﾃﾞ自動的に

　　　　　　    |       |　　　　　＝＞０　ﾊﾞｯﾌｧを読込んだので更新可の通知　ﾕｰｻﾞ側ﾃﾞ）

　　　　　　    |       |\_\_\_\_\_\_\_\_\_\_\_\_\_受信データの先頭格納Ｄメモリー番号

　　　　　　    |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_受信バイト数（最大１２０）

　ＳＲＩＮの命令の頭の接点がオフで電気が来てないと、指定先の内容は更新されません。

　しかし、リングバッファへの取り込みと、上記Ｋ３０の３０バイト毎に、上記Ｍ１８０をオンすることを続けます。

　このことはＭ１８０のオン／オフに拘わらず実行されます。

　註）受信内容ははＤメモリー列へ下位，上位の順で８ビットずつ、ＩＮＰＵＴされます

　　　上例でＫ３０とした場合、３０バイトの受信データがＤ１８～Ｄ３２の１５ワードへ送られます。

　　　　　　　Ｍ１８０（Ｂ接点）

　｜－－－｜／｜－－－－－－［ＳＲＩＮ　Ｋ３０　Ｄ１８　Ｍ１８０］－｜

　上記回路だと、３０バイトの受信ごとにＭ１８０をオンし（これによってＤ１８にデータを送り込んだ後にＳＲＩＮ命令自体がオフし）、別行の回路でＭ１８０をオフすることによって（ＳＲＩＮ命令がオンし）、次の３０バイトを指定先（この場合Ｄ１８）にリングバッファから送り込みます（すでに次の３０バイトが来ていればそれを送り、来ていなければ３０バイトになるのを待ちます）。

　上記、Ｍ１８０、Ｋ３０、Ｄ１８は各々一つの例です。

　どうしても追いつかない場合（そんなことは起きにくいことですが）、頭にＭ１８０の代わりにＭ９０３６（常時オン）を入れることによって、常に最新のデータが、指定先に送り込まれることになります。

　●０Ｄ（ゼロＤ）を区切りとして、新たなカウントを始めます。

　　このことによって、リングバッファの頭を決定することができます。

　送信命令　**ＳＲＯＴ**

例

　ＳＲＯＴ　Ｋ３０　Ｄ１８　Ｍ１８０

　　　　　　    |       |         |\_\_\_送信終了後オフする接点番号

　　　　　　    |       |　　　　（　＝＞０　送り終わった信号。ＣＰＵ内部側ﾃﾞ自動的に

　　　　　　    |       |　　　　　　＝＞１　送るべきﾃﾞｰﾀが揃ったら、１とする。ﾕｰｻﾞ側ﾃﾞ）

　　　　　　    |       |\_\_\_\_\_\_\_\_\_\_\_\_\_送信データの先頭Ｄメモリー番号

　　　　　　    |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_送信バイト数（最大１２０）

　上記３つのオペランドは先頭文字以外の値いはすべて任意です。

　註）指定Ｄメモリー列はすべて下位，上位の順で８ビット（通常アスキーに対応）＊２＝１６ビット送られます。

　　　上例でＫ３０とした場合、Ｄ１８～Ｄ３２の１５ワードが送られます。

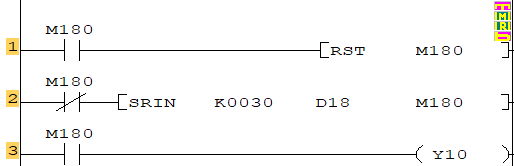
                Ｍ２０

　｜－－－｜｜－－－－－－［ＳＲＯＴ　Ｋ３０　Ｄ１８　Ｍ２０］－｜

　上記回路のようにすれば、他の回路でＭ２０をオンすれば、一回だけ通信することができます。

　◎通信内容はアスキーでなくても可です（０～FFｈ）であれば良い。

基本的な受信プログラム



Ｄ10

K0030：指定したデバイスを強制ONするバイト数。

D10：M180がONになると、受信したデータ３０バイトまたはCR（０ｘ０D）の一つ前までをD10以降のデバイスに格納する。格納は、下位バイト上位バイトの順となる。

M180：連枝のシステムが、３０バイト受信するかCR（０ｘ０D)を受信すると強制ON

※ＳＲＩＮの条件は、受信中常にＯＮにしておいてください。ＯＦＦになるとＤへの格納が行われません。

※受信したデータを処理する場合、Ｍ１８０がＯＮの条件を用いて処理を行ってください。また、処理が終わったらＭ１８０をユーザー側でＯＦＦにしてください。Ｍ１８０がＯＮのままだと、受信処理が行われません。

ＧＮＳＳ　通信データ

今回、「ＧＮＳＳ（ＧＰＳ・ＧＬＯＮＡＳＳ・ＱＺＳＳ）受信機キット　１ＰＰＳ出力　みちびき３機対応　アンテナセット付キット」を購入し動作確認を行いました。

<http://akizukidenshi.com/catalog/g/gK-13850/>

送信されたデータは下記の通りです。

$GPGGA,085849.000,3459.4648,N,13542.2783,E,1,13,0.80,44.3,M,34.5,M,,\*55

$GNGLL,3459.4648,N,13542.2783,E,085849.000,A,A\*45

$GNGSA,A,3,22,11,193,18,08,30,01,03,17,,,,1.14,0.80,0.81\*24

$GNGSA,A,3,72,65,87,86,,,,,,,,,1.14,0.80,0.81\*1E

$GPGSV,4,1,14,28,64,290,12,03,61,127,20,01,56,032,20,22,56,080,32\*70

$GPGSV,4,2,14,11,48,055,20,193,40,162,29,17,32,308,14,18,31,051,21\*40

$GPGSV,4,3,14,08,22,101,29,30,12,226,20,19,12,297,18,06,05,247,16\*77

$GPGSV,4,4,14,23,01,148,24,43,,,\*40

$GLGSV,2,1,06,87,49,008,31,72,39,096,28,88,33,304,,65,24,148,36\*68

$GLGSV,2,2,06,71,21,041,,86,15,068,26\*63

$GNRMC,085849.000,A,3459.4648,N,13542.2783,E,0.38,51.13,010519,,,A\*43

$GPVTG,51.13,T,,M,0.38,N,0.71,K,A\*06

$GPZDA,085849.000,01,05,2019,,\*50



33

38

37

32

32

34

35

33

31

38

34

36

34

39

35

34

33

30

30

30

39

34

38

35

2E

2C

2C

4E

2C

2E

2E

2E

38

30

2Ｃ

41

47

47

50

47

24

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

33

34

35

39

2E

34

36

31

37

39

34

38

32

32

34

35

33

31

45

4E

2E

2C

2C

2C

2C

4C

4C

4E

47

47

24

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

下位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

上位

$GNGLL,3459.4648,N,13542.2783,E,085849.000,A,A\*45

応答パケットを、ＧＮＧＬＬを前提として、座標を考えると。

受信したデータは、下位ー上位、下位ー上位の順に格納されていく。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D13 | | D14 | | D15 | | D16 | | D17 | | D18 | | D19 | | D20 | | D21 | | D22 | |
| 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 | 下 | 上 |
| ， | ３ | ４ | ５ | ９ | ． | ４ | ６ | ４ | ８ | ， | N | ， | １ | ３ | ５ | ４ | ２ | ． | ２ |
| 2c | 33 | 34 | 35 | 39 | 2e | 34 | 36 | 34 | 38 | 2c | 4e | 2c | 31 | 33 | 35 | 34 | 32 | 2e | 32 |

受信したデータを数字に変換する方法

　アスキーコードで、格納されているデータを数字に変換する方法があります。アスキーコード（８ビット）のうち下位４ビットが数字となります。

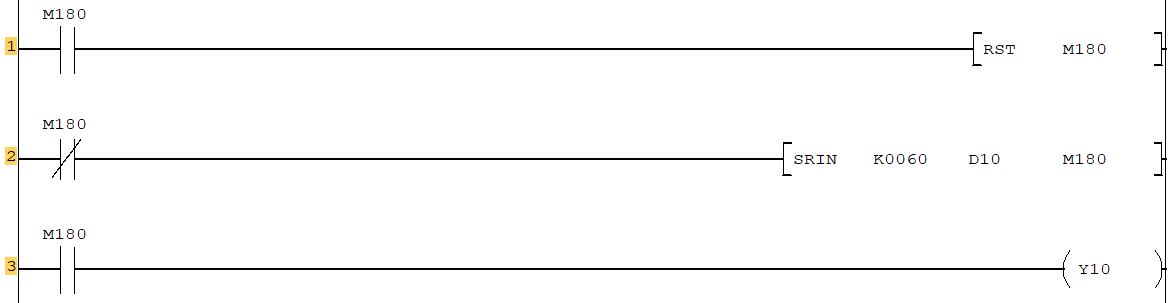
(手順）

　ビットシフトした値をアスキーコードから数字を生成（WAND命令でマスクする）

　受信データを適切な位置になるようにビットシフト（SFR命令８ビット右シフトまたはそのまま）

　数字に１０進数の桁をかける（＊　命令、HA[10]、H64[100]、H3E8[1000]）

　計算した結果を足し合わせる



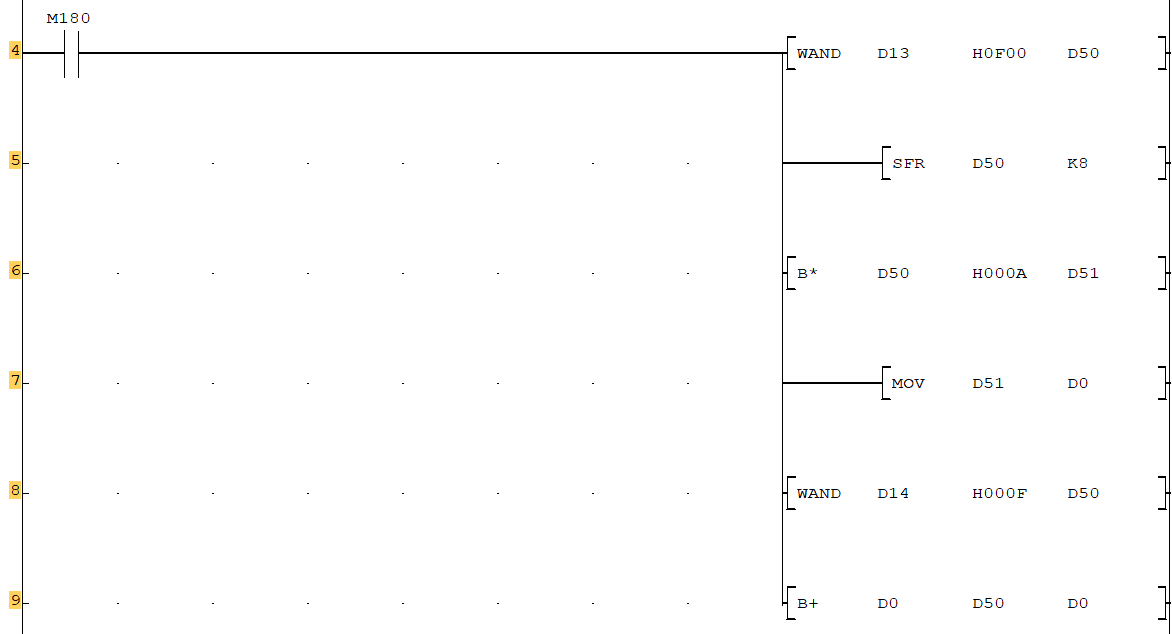
データレジスタ　２バイト・１６ビット

文字コード　→　数字

GPSのパケットから緯度を計算

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| データレジスタ | バイト位置 | 文字 | 文字コード | |  |
| D13 | 下 | , | 2C |  |  |
| 上 | 3 | 33 | 10位 | 度数 |
| Ｄ14 | 下 | 8 | 38 | 1位 |
| 上 | 4 | 34 | 10位 | 分数 |
| Ｄ15 | 下 | 4 | 34 | 1位 |
| 上 | . | 2E |  |
| Ｄ16 | 下 | 5 | 35 | 小数点１位 |
| 上 | 2 | 32 | 小数点２位 |
| Ｄ17 | 下 | 3 | 33 | 小数点３位 |
| 上 | 3 | 33 | 小数点４位 |
| Ｄ18 | 下 | , | 2C |  |  |
| 上 | N | 4E |  |  |



・度数を求める。

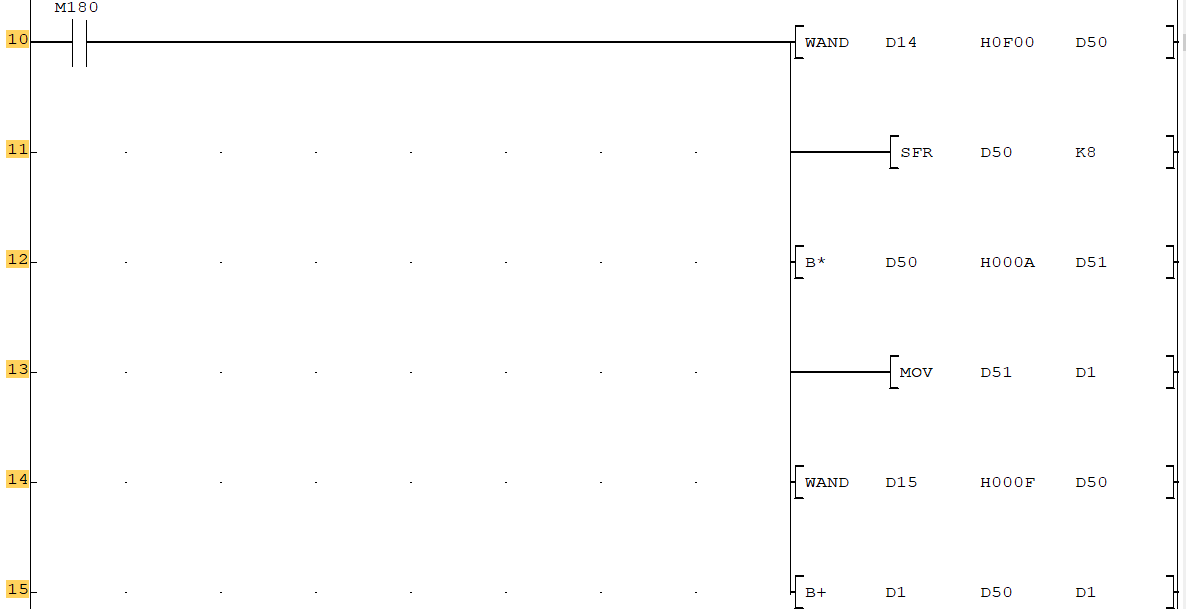
　D13の上位1バイトの値を4ビットのみ取り出す

　D13の上位8ビットを下位8ビットにする

　上の値に10をかける

　D14の値を下位４ビットのみ取り出す

　上の値と10をかけた値を足し合わせる　・・・・　度数が判明



・分を求める。

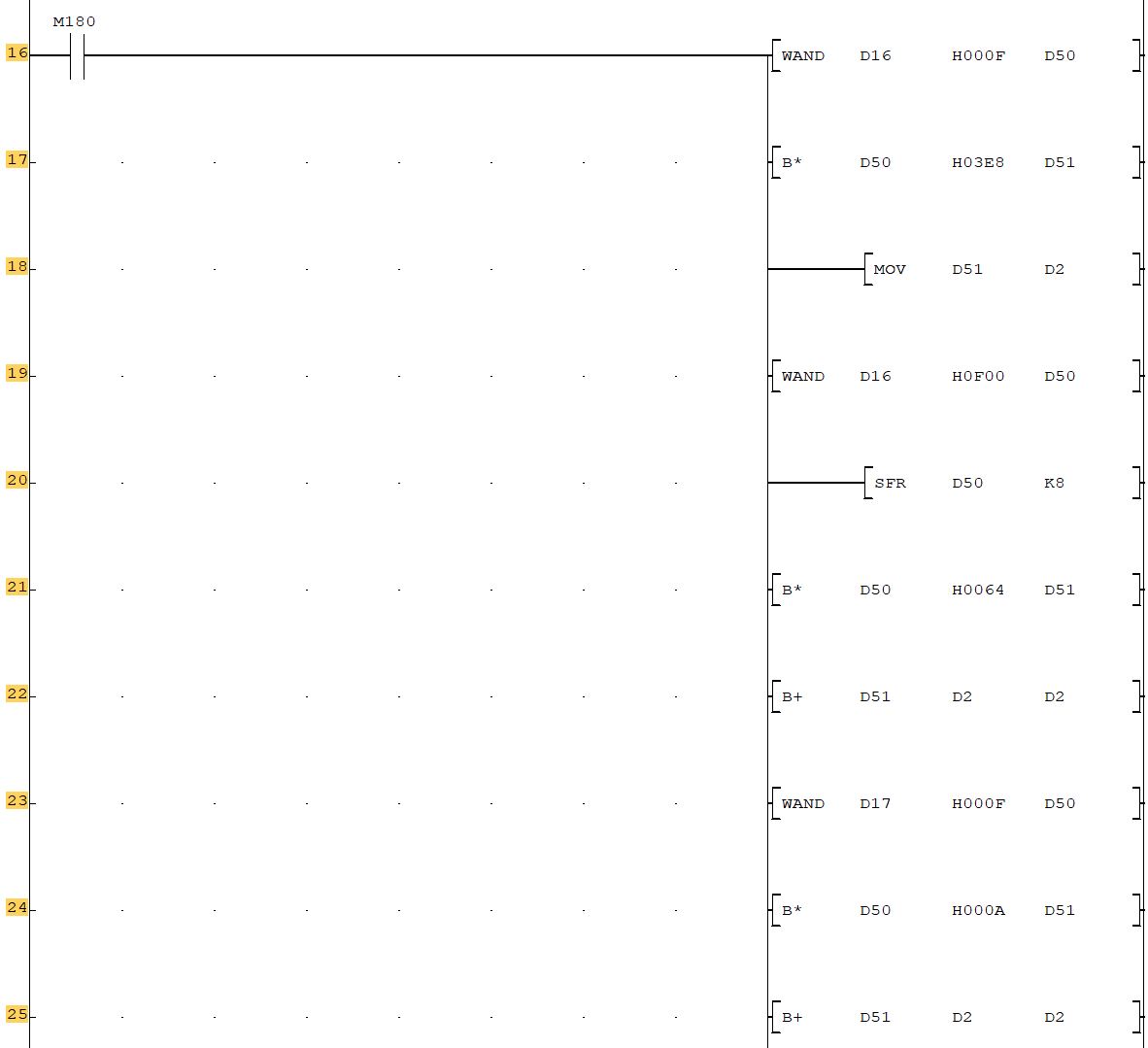
　D14の値を上位4ビットのみ取り出す

　D14の上位8ビットを下位8ビットにする

　上の値に10をかける

　D15の下位4ビットのみ取り出す

　上の値と10をかけた値を足し合わせる　・・・・　分数が判明

・秒を求める。

　D16の値の下位4ビットのみ取り出す

　上の値に1000を掛ける

　D16の値の上位4ビットのみ取り出す

　D16の上位8ビットを下位8ビットにする

　D16の下位4ビットのみ取り出す

　上の値に100を掛ける

　100をかけた値と最初の値を足し合わせる。

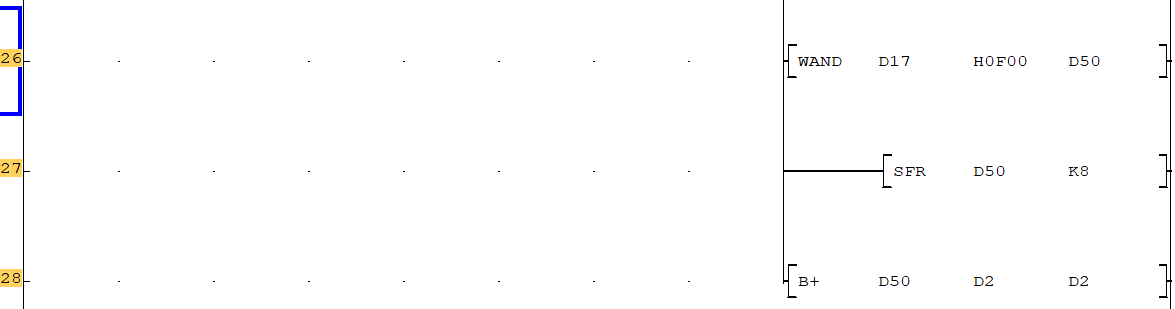
　D17の下位4ビットのみ取り出す

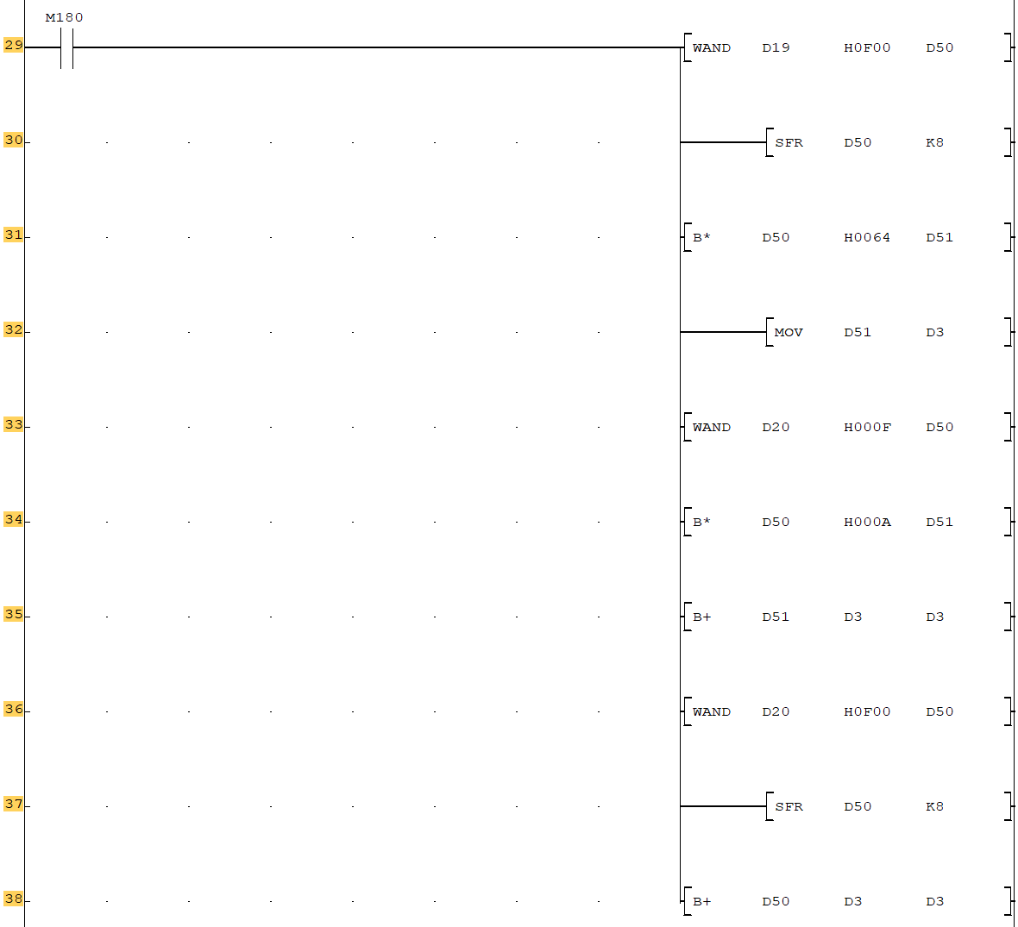
　上の値に10をかける

　足し合わせた値と10をかけた値を足し合わせる

　D17の上位4ビットのみ取り出す

　D17の上位8ビットを下位8ビットにする

　足し合わせた値と10をかけた値を足し合わせる

・度数を求める。

　D19の上位1バイトの値を4ビットのみ取り出す

　D19の上位8ビットを下位8ビットにする

　上の値に10をかける

　D20の値を下位４ビットのみ取り出す

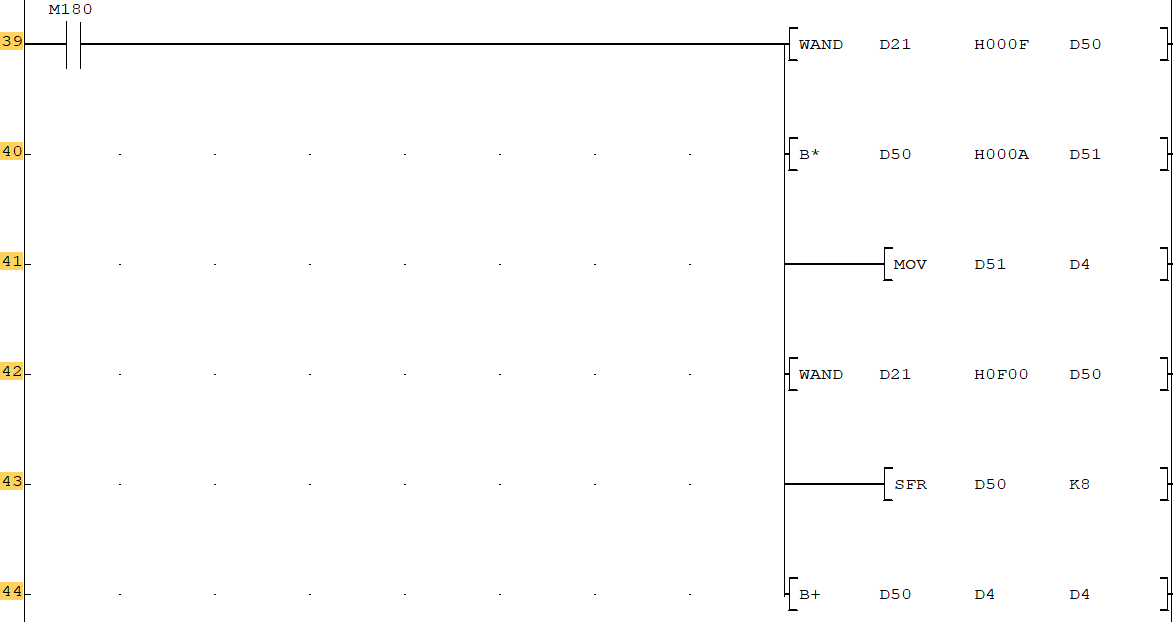
　上の値と10をかけた値を足し合わせる　・・・・　度数が判明

・分を求める。

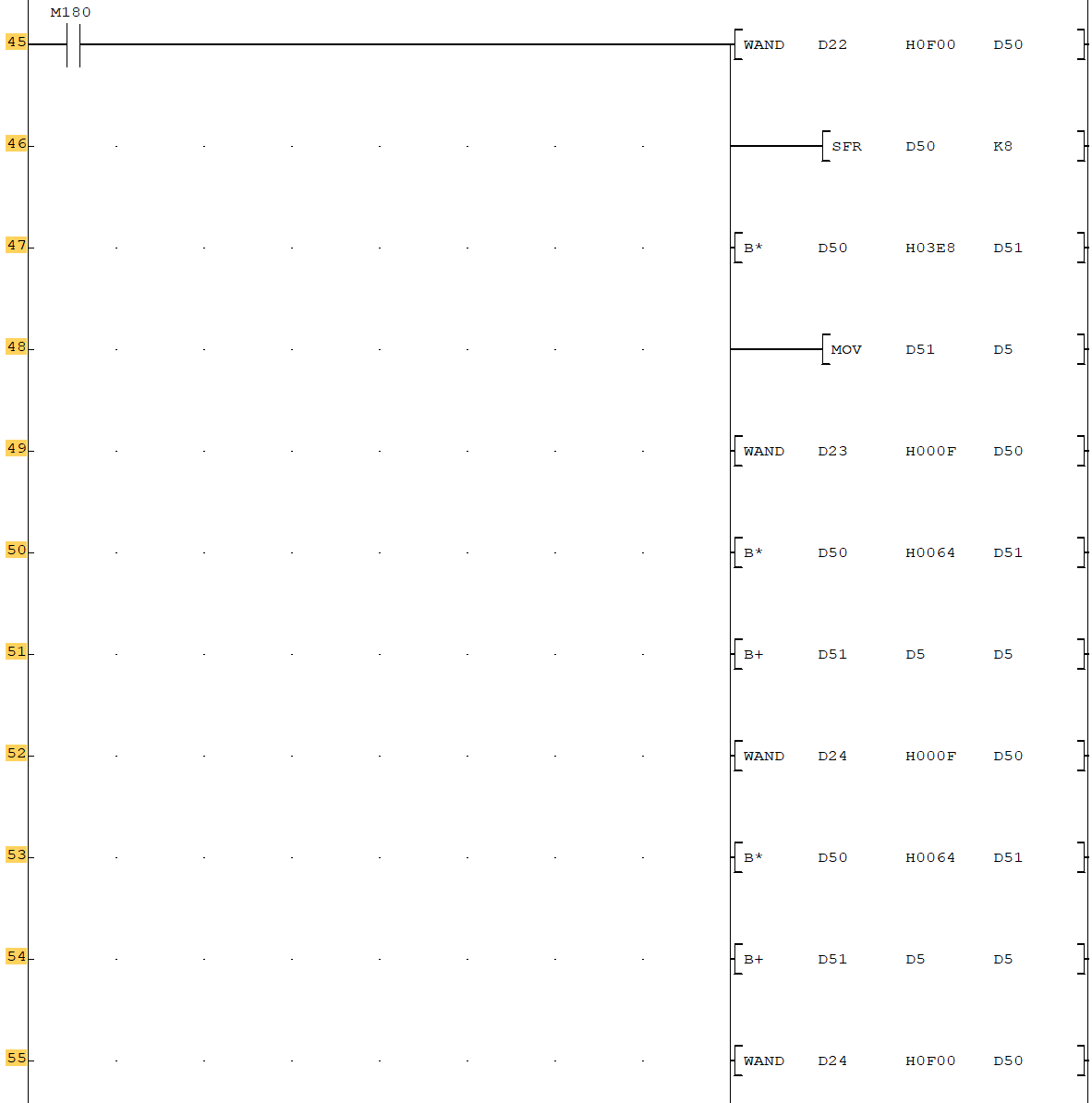
　D20の値を上位4ビットのみ取り出す

　D20の上位8ビットを下位8ビットにする

　上の値に10をかける

　D21の下位4ビットのみ取り出す

　上の値と10をかけた値を足し合わせる　・・・・　分数が判明

・秒を求める。

　D22の値の下位4ビットのみ取り出す

　上の値に1000を掛ける

　D22の値の上位4ビットのみ取り出す

　D22の上位8ビットを下位8ビットにする

　D22の下位4ビットのみ取り出す

　上の値に100を掛ける

　100をかけた値と最初の値を足し合わせる。

　D23の下位4ビットのみ取り出す

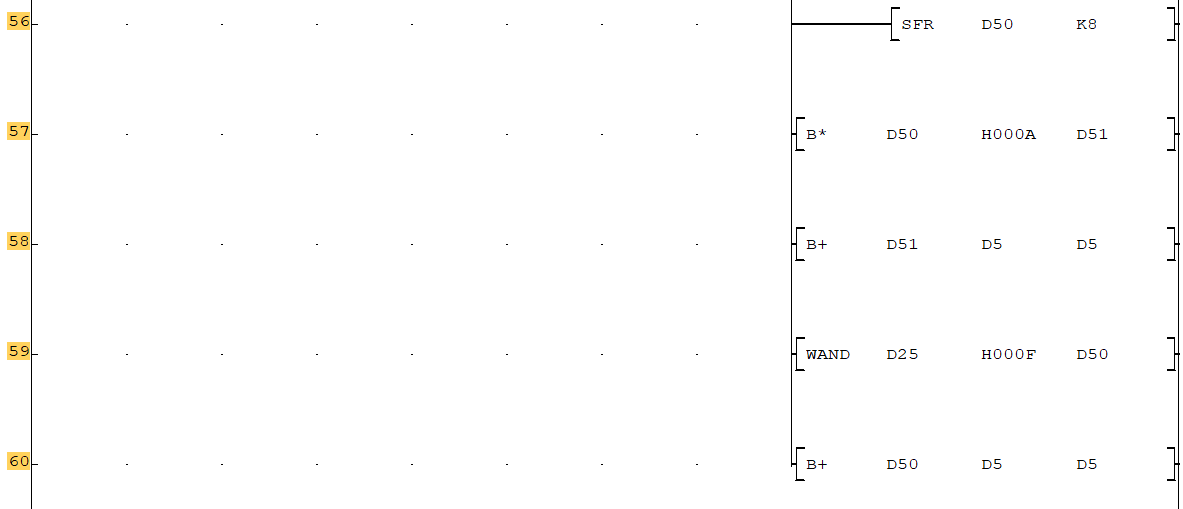
　上の値に10をかける

　足し合わせた値と10をかけた値を足し合わせる

　D23の上位4ビットのみ取り出す

　D23の上位8ビットを下位8ビットにする

　足し合わせた値と10をかけた値を足し合わせる



|  |  |  |  |
| --- | --- | --- | --- |
|  | 緯度 | 経度 |  |
| 度数 | Ｄ0 | Ｄ3 |  |
| 分数 | Ｄ1 | Ｄ4 |  |
| 秒数 | Ｄ2 | Ｄ5 |  |

|  |  |
| --- | --- |
| 一時変数 |  |
| Ｄ50 | 文字を数値に変換 |
| Ｄ51 | 数値を倍数に変換（１倍、１０倍、１００倍、１０００倍など） |
| Ｄ52 |  |

アスキーコード表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 上位１バイト | | | | | | | | | | | | | | | |
|  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|  | 0 |  |  | sp | 0 | @ | P | ` | p |  |  |  |  |  |  |  |  |
| 1 |  |  | ! | 1 | A | Q | a | q |  |  |  |  |  |  |  |  |
| 2 |  |  | “ | 2 | B | R | b | r |  |  |  |  |  |  |  |  |
| 3 |  |  | # | 3 | C | S | c | s |  |  |  |  |  |  |  |  |
| 4 |  |  | $ | 4 | D | T | d | t |  |  |  |  |  |  |  |  |
| 5 |  |  | % | 5 | E | U | e | u |  |  |  |  |  |  |  |  |
| 6 |  |  | & | 6 | F | V | f | v |  |  |  |  |  |  |  |  |
| 7 |  |  | ‘ | 7 | G | W | g | w |  |  |  |  |  |  |  |  |
| 8 |  |  | ( | 8 | H | X | h | x |  |  |  |  |  |  |  |  |
| 9 |  |  | ) | 9 | I | Y | i | y |  |  |  |  |  |  |  |  |
| A | lf |  | \* | : | J | Z | j | Z |  |  |  |  |  |  |  |  |
| B |  |  | + | ; | K | [ | k | { |  |  |  |  |  |  |  |  |
| C |  |  | , | < | L | \ | l | | |  |  |  |  |  |  |  |  |
| D | cr |  | - | = | M | ] | m | } |  |  |  |  |  |  |  |  |
| E |  |  | . | > | N | ^ | n | ~ |  |  |  |  |  |  |  |  |
| F |  |  | / | ? | O | \_ | o |  |  |  |  |  |  |  |  |  |