卒 業 研 究 報 告

題　目

PIC プロセッサを用いた流星自動追跡装置の製作

指 導 教 員

山本 真行　講師

報 告 者

学籍番号： 1060197
氏名： 大岡　裕志

平成 18 年 2 月 21 日

高知工科大学　電子・光システム工学科

第1章　序論

1.1 背景

近年、日本のアマチュア天文観測者による高度な流星観測が活発に行われている。それは夜空を見上げて流星を数えるような古典的な方法ではない。高感度 CCD カメラと高機能ソフトウェア、大容量記憶装置を備えた PC を使用した自動観測である。このような自動観測が活発化した背景には、PC の高機能化と低価格化が進んだ事、そしてこれらの性能を余す事なく活用出来る安価かつ優秀なソフトウェアが開発された事があると言える。このように高度な自動観測の普及によって、これまで見過ごされていた自然現象が多くの観測者によって観測されはじめ、非常に稀な現象を捉えたり、統計的手法による研究へと発展させたりすることが可能となった。今後より多くの現象を観測するために、観測装置の更なる高性能化が求められている。

1.2 概要

高知工科大学電子・光システム工学科 山本真行研究室では、定点固定カメラによる自動流星観測を 2005 年 1 月より行い、流星や高層大気の放電現象であるスプライトの観測を継続している。また、日本全国の 10 地点以上でも流星観測者等が同様の固定カメラで観測している。固定カメラは広範囲の観測により、多くの流星を捉えることが可能な反面、注目すべき部分の詳細な画像を得る事は出来ない。そこで、本研究は高空間分解能画像を得る事が出来る望遠カメラで流星を追尾させる装置の製作を行う。制御には小型で安価、プログラムの作成も比較的容易な PIC プロセッサ(後述)を使用する。なお、カメラを動かす際の位置情報は、現在設置してある固定広視野カメラ、又は新たに設置する全天カメラからの画像情報を元にソフトウェアで解析し、時々刻々シリアル通信で PIC プロセッサに伝達される。このソフトウェアについては「流星自動追跡装置における連続画像処理プログラムの開発」(榮田、高知工科大学卒業研究報告、2006)を参照されたい。また、本研究と同様な研究を PIC プロセッサではなく PC を使用して行う研究は「DA ボードを用いた流星自動追跡装置の基礎開発」(山下、高知工科大学卒業研究報告、2006)に記されているので参照願いたい。

1.3 目的

本研究は、夜空に突然現れては瞬く間に消えてしまう流星を捕捉してなお追尾し続ける装置を製作するもので、一瞬で流星を捕捉する高速動作と流星の角速度に追随する高精度な制御が要求される。完成すれば、流星観測以外にも様々な応用分野が見込まれる。具体的には鳥獣の観測や高速移動物の監視である。本卒業研究では、PIC プロセッサによる 2 軸モータ制御部分の基礎開発を行うとともに、電子回路及び組み込みプログラミング技術に習熟する事を目的とする。本研究が、近い将来の研究室での発展を経て多くの観測者の研究・観測並びに科学の発展に役立つ事を期待する。

第2章　装置の製作

2.1 装置の構成

本研究を含む全体の構成は図 2.1 のようになる。

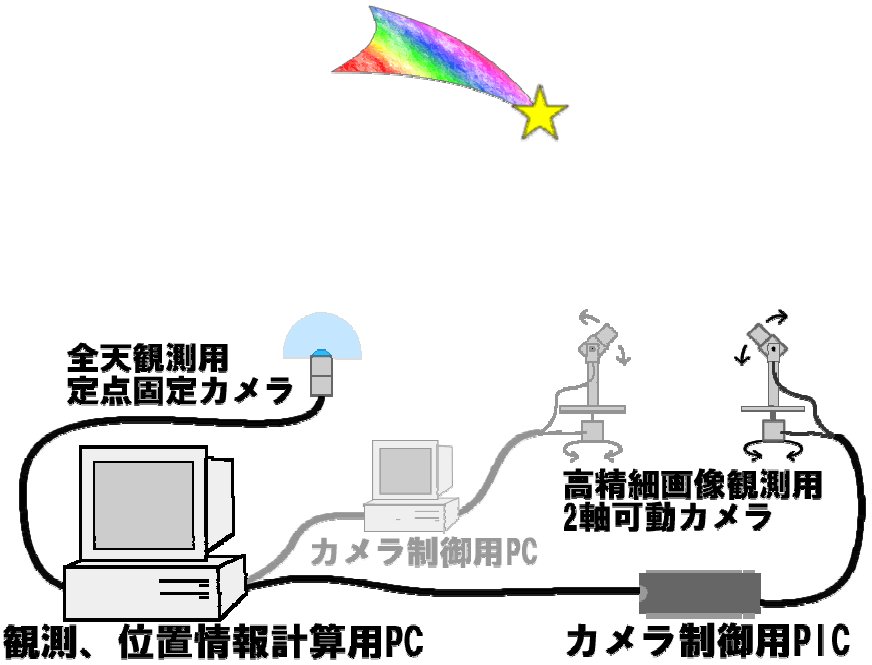

図 2.1　全体の構成図

本論文では図 2.1 のカメラ制御用 PIC プロセッサを搭載した回路の製作と 2 軸可動カメラ架台の製作、位置情報の受信、それらを総括する PIC プロセッサのプログラムについて述べる。

2.2 構成部品

本研究で製作する部分の概略図を図 2.2 に示す。

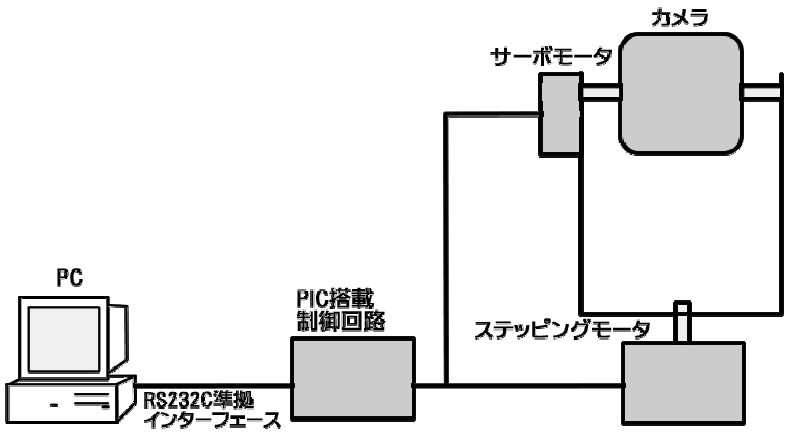

図 2.2　装置の概略図

観測用 PC で解析された位置情報は、RS232C 準拠のシリアルインターフェースで受信し、モータを制御する回路には PIC プロセッサを使用する。仰角はカメラの片側の軸に取り付けた DC サーボモータで駆動し、方位角はカメラ架台の下部に取り付けたステッピングモータで駆動する。

2.2.1 PIC プロセッサ

本研究で製作する装置は屋外に設置するので可能な限り小型、かつ軽量であることが望ましい。そのためにはカメラとモータを支持する台はもちろん、制御回路を載せる基板も小型である必要がある。そこで、制御回路には CPU、RAM、I/O 制御等を一つの IC に内蔵したワンチップマイコンを使用する。

ワンチップマイコンの代表的なものとして、日立製作所の H8 や米ザイログ社の Z80 等が挙げられるが、本研究では web や書籍での情報の多さ、開発の容易さを考慮し米マイクロチップテクノロジー社の PIC(Peripheral Interface Controller)を選んだ。

PIC プロセッサにはピン数やメモリ容量、A/D コンバータやシリアルコントローラ等の内蔵機能の有無によって、多くの種類に分けられる。その中で今回は書籍も多く、一番扱いやすい PIC プロセッサの 1 つとされている PIC16F84A(図 2.3)を使用した。

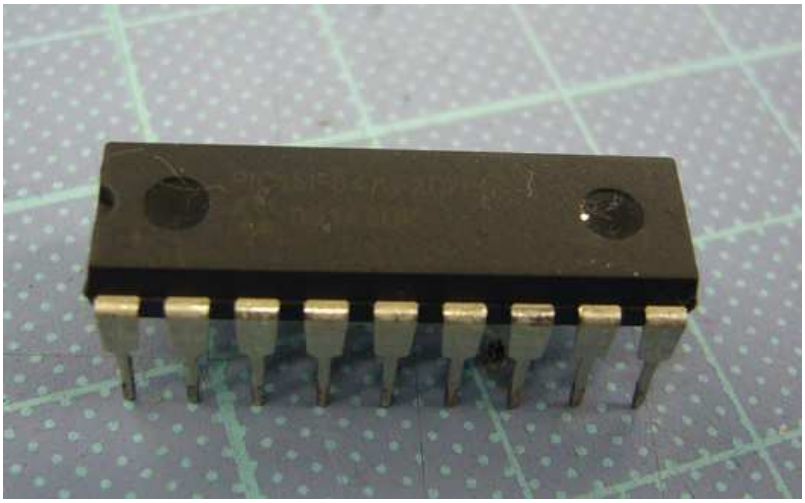

図 2.3　本研究で用いた PIC プロセッサ PIC16F84A

2.2.2　ステッピングモータ

ステッピングモータは、X、$\bar{X}$、Y、$\bar{Y}$それぞれの極に順番に電流を流す事でロータを回転させる事ができ、これを励磁と呼ぶ。励磁の方法にはいくつかの種類があるが、本研究ではステップ角と回転速度を考慮し 1-2 相励磁が最適と判断し、これを使用した。1-2 相励磁は仕様のステップ数の倍のステップ数が必要な反面、倍の精度でステッピングモータを駆動できる。実際に送信するパルスを図 2.4 に示す。

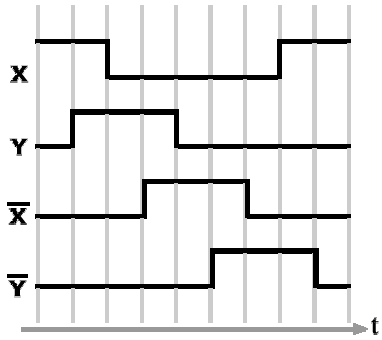

図 2.4　1-2 相励磁方式で正転の場合のパルス

使用するモータはトルクと回転速度を考慮し、十分な性能を持ったモータを選ばなければならないが、本研究では装置の基礎を作ることを目的としているので性能は考慮せず安価に入手可能なステッピングモータ(多摩川精機社製 TS3103N124)を選んだ。外観と仕様を図 2.5 と表 2.1 に示す。例えば、秋月電子にて入手可能である。

図 2.5　本研究で用いたステッピングモータ 多摩川精機社製 TS3103N124

表 2.1　ステッピングモータ(TS3103N124)の仕様

ステップ数	200(1.8deg/Step)
コイル抵抗	80[Ω/相]
コイル電流	140[mA](12V) 280[mA](24V)
保持トルク	6.5[kg･cm]
外形寸法	Φ56.4 x H53[mm]
フランジ	56.4 x 56.4[mm]
重さ	500[g]
軸径	9[mm]
軸長	18[mm]

(秋月電子(http://akizukidenshi.com/catalog/items2.php?p=1&q="P-00232")より引用)

2.2.3　DC サーボモータ

DC サーボモータには、DC モータ、減速機構、制御基板、ポテンショメータが入っており、信号線に加える 20ms 周期のパルス幅で任意の角度に回転する。例えば、図 2.6 のように 1.5ms のパルス信号を加えるとサーボモータは中間位置に停止する。本研究で使用した Micro2BB/MG の場合、1.5ms を中心として前後に 0.9ms の範囲でパルスを与えれば、前後に 90 度以内の任意の位置に回転する。

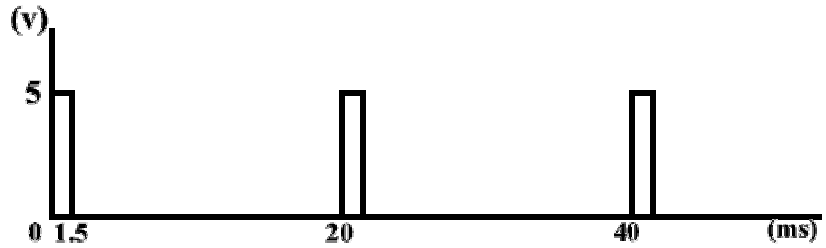

図 2.6　サーボモータを制御する信号

本研究におけるカメラの仰角制御に適切と思われるモータをカタログやインターネット上の情報を基にトルクと回転速度を考慮し選定した。外観と仕様を図 2.7 と表 2.2 に示す。用いたサーボモータは GWS 社製 Micro2BB/MG であり、例えば浅草ギ研等にて入手できる。

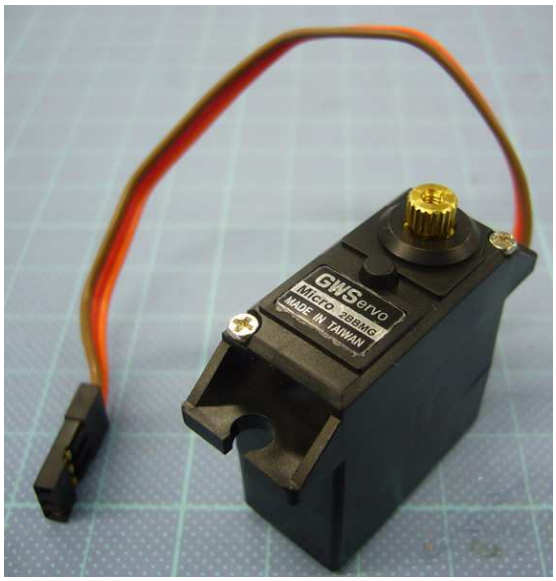

図 2.7　本研究で用いた GWS 社製 Mirco2BB/MG

表 2.2　DC サーボモータ (Micro2BB/MG)の仕様

トルク	5.4[kg･cm](4.8V) 6.4[kg･cm](6.0V)
スピード	0.17[sec/60°](4.8V) 0.14[sec/60°](6.0V)
ギア	フルメタル
方式	アナログ
サイズ	28.0 x 14.0 x 29.8[mm]
重量	28[g]
信号線の被覆の色と意味	橙:制御信号、赤:電源、茶:GND

(浅草ギ研(http://www.robotsfx.com/robot/RCServo.html)より引用)

2.2.4　観測用カメラ

観測用のカメラとしては流星等の微光天体の映像を得るために適していると近年話題の WATEC 社製のモノクロリモコン式カメラ Neptune100(略称 WAT-100N)を使用した。外観と仕様を図 2.8 と表 2.3 に示す。

図 2.8　WATEC 社製 Neptune100

表 2.3　観測用カメラ(Neptune100)の仕様

有効画素数	38 万画素
解像度	570[TV 本]
最低被写体照度	0.001[lx]
SN 比	50[dB]
重量	125[g]

(WATEC(http://www.watec.net/japan/bw/wat_100n.html)より引用)

2.3 回路構成

制御回路を載せる基板は、RS232C シリアル通信ケーブルを接続する D-Sub 9 ピンコネクタ(メス)と DC ジャックを簡単に実装でき、IC を十分取り付けられる大きさのものとして、サンハヤト製 ICB-93-CK(図 2.9)を使用した。回路は RS-232C レベルと TTL レベルを変換するインターフェース IC と今回プログラミングする PIC プロセッサ、及び電源回路、ノイズフィルタ、ステッピングモータドライブ回路から成る。

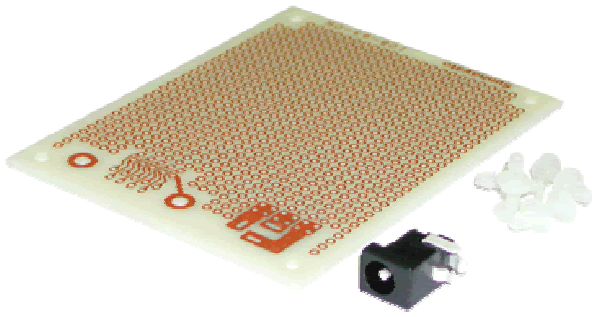

図 2.9　サンハヤト製 ICB-93-CK

(サンハヤト(https://www.sunhayato.co.jp/products/details.php?u=95&id=07002)より引用)

2.3.1　回路図

本研究で作成した回路図を図 2.10 に示す。サーボモータのような機構部品はノイズ源となることが多いので、デリケートな半導体部品とは別電源にすることが望ましい。しかし、本研究では基板の小型化を優先するので同じ電源から供給した。この際サーボモータから発生するノイズの低減対策として 0.1μF のコンデンサを接続した。観測用カメラとステッピングモータ、その駆動用 IC には汎用 AC アダプタから供給される DC12V を使用する。DC サーボモータと PIC プロセッサ及び RS232C インターフェース IC は、3 端子レギュレーターを介して生成された DC5V にて駆動される。

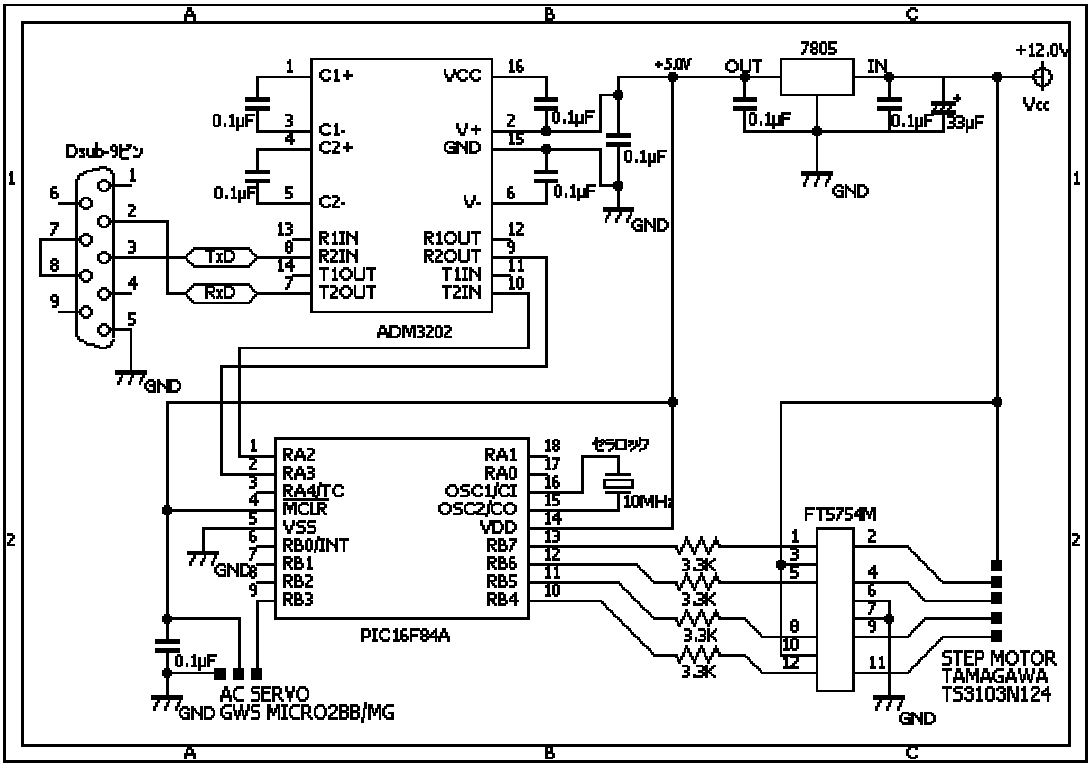

図 2.10　制御回路の回路図

2.3.2　構成部品

制御基板を構成する主要部品を表 2.4 に示す。表で示した IC 等の他は、10 数個のコンデンサ、抵抗によって構成されている比較的シンプルな省スペース回路が実現できた。製作した基板を図 2.11 に示す。

表 2.4　制御回路の構成部品

型番	素子の名称	メーカー
7805	3 端子レギュレーター	不明
ADM3202AN	RS232C インターフェース IC	Analog Devices
FT5754M	NPN パワーダーリントン トランジスタアレイ	FUJITSU
PIC16F84A	PIC プロセッサ	Microchip Technology

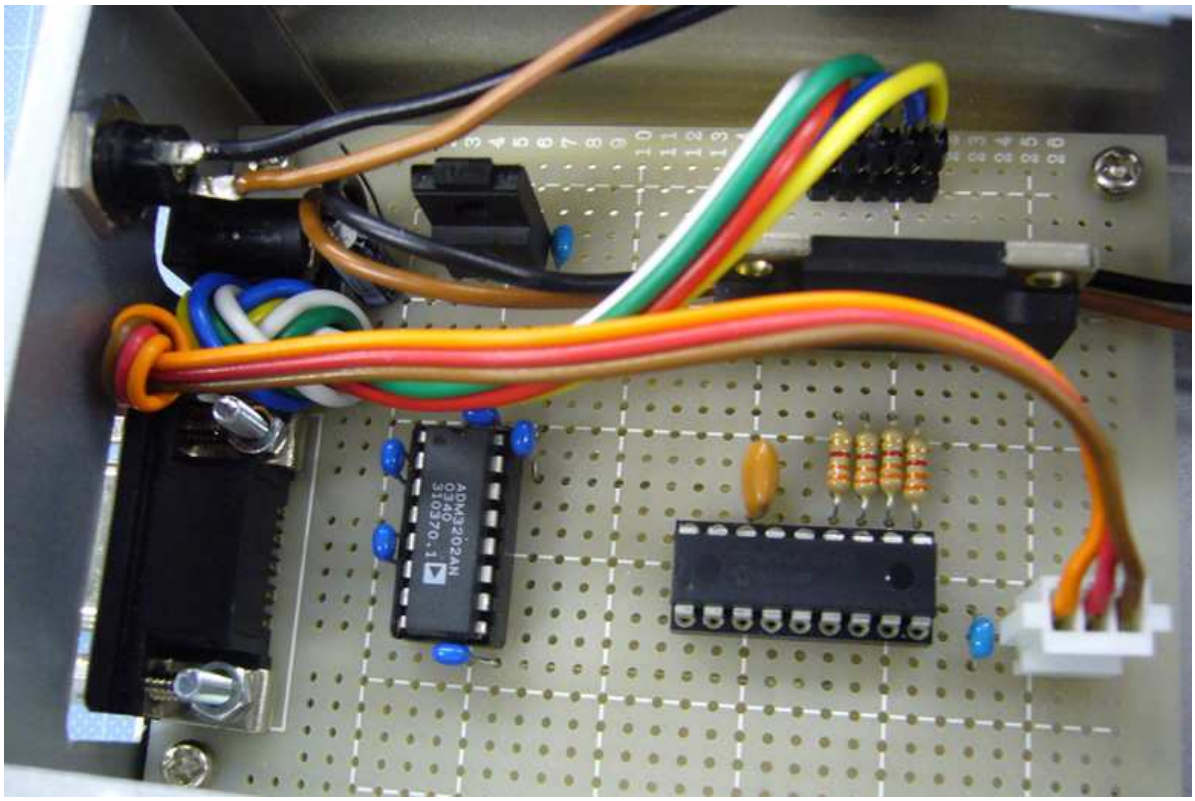

図 2.11　製作した基板

2.4 カメラ架台

高速で動作するカメラを支える必要があるので､強度を優先しつつもサーボモータ、ステッピングモータの負荷を減らすために極力シンプルな構成を心がけた。屋外での使用を前提とするため、主にステンレス部品を使用して作製した。ケース部分は、φ300mm のアクリル製透明半球とアルミ製の鍋を用い、カメラの回転に対し若干のクリアランスを確保する設計とした。完成した架台を図 2.12 に、ケースに収納した様子を図 2.13 に示す。

図 2.12　製作したカメラ架台

図 2.13　製作したケース(右下の液晶モニタは確認用)

第3章　制御プログラム

3.1 PIC プログラム開発環境

PIC プログラムの開発環境には米マイクロチップテクノロジー社より無償で提供されている MPLAB を使用した。コンパイラは MPLAB 付属のアセンブラ(MPASM)を使用した。MPLAB は PIC プログラム作成環境における一般的な環境である。PC 上での MPLAB の操作画面の様子を図 3.1 に示す。MPLAB の使用方法や FAQ などは、電子工作の実験室(http://www.picfun.com/)などが参考になる。特にアセンブラでのプログラム開発においては、命令ごとに細かなコメント行をつけておく必要がある。PIC へ書き込む際にはコンフィギュレーションワードの設定にも留意しなくてはいけない。

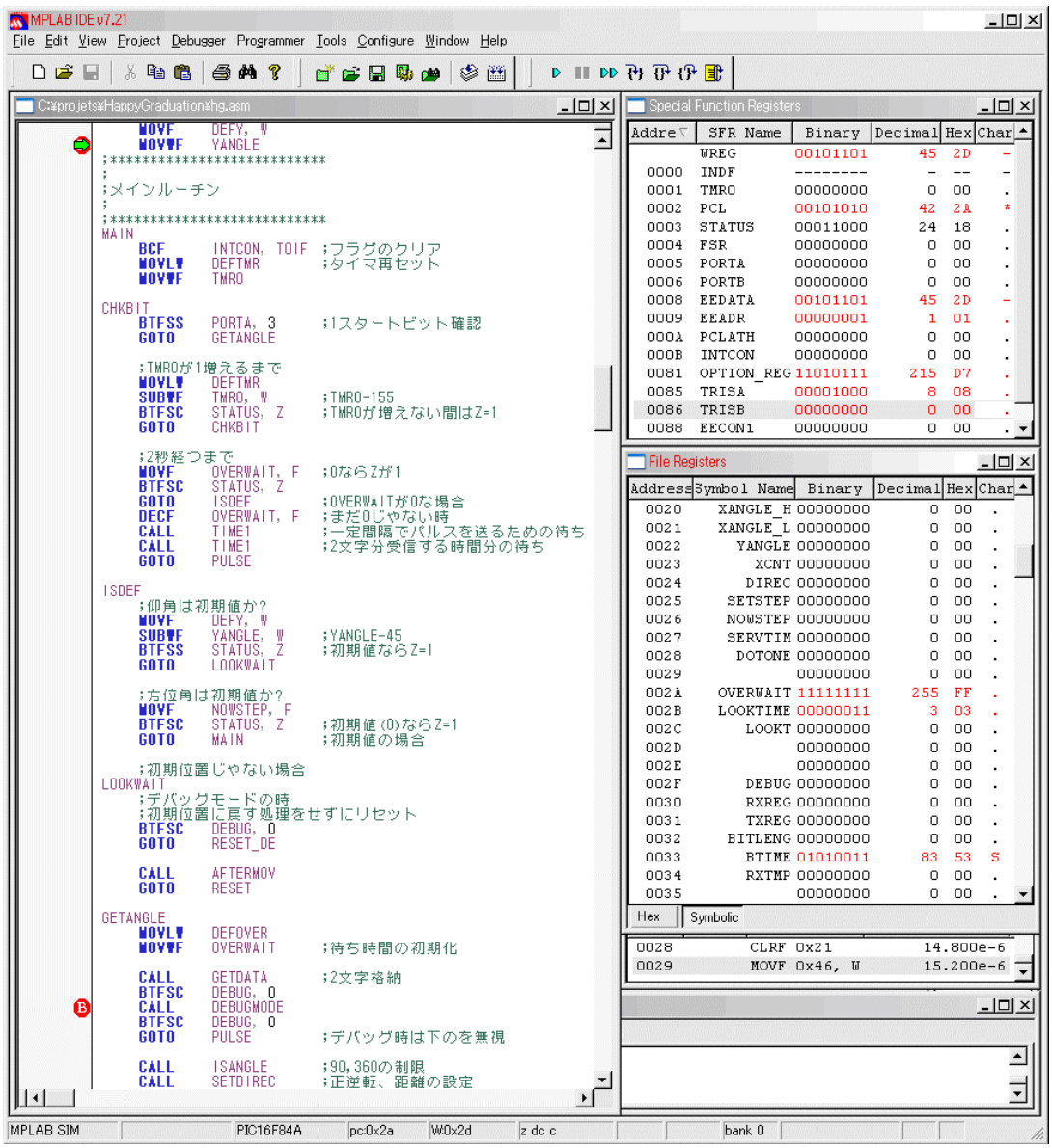

図 3.1　MPLAB で開発中の画面

3.2 シリアル通信

位置情報の通信は RS232C 準拠のシリアルインターフェースを使用した。方位角 360 度と仰角 90 度を 1 度の精度で制御するためには、それぞれ 9bit(=512)と 7bit(=128)必要になるので、2 文字(byte)分 16bit の通信を行う必要がある。通信仕様としては、表 3.1 に示すように一般的なものを使用した。

表 3.1　通信仕様

通信速度	9600bps
スタートビット	1bit
ストップビット	1bit
パリティビット	無し
データ長	8bit
フロー制御	無し
方位角	9bit
仰角	7bit

方位角は 1 文字(8bit)では収まらないので、開発チーム内での議論の結果、仰角を 1bit 左にシフトし、空いた最下位ビットに方位角の最上位ビットを付加して送信する仕様とした。シリアル通信を採用したことにより PC 間のデータ通信に使える他、PIC 側でシリアル通信インターフェース IC を用いることにより PC-PIC 間でも共通に利用できるフォーマットとなった。また、シリアル通信の電圧範囲は-12V から 12V であり、他の通信方式に比べて高く、遠距離を安定して通信できるメリットがある。送信例を図 3.2 に示す。

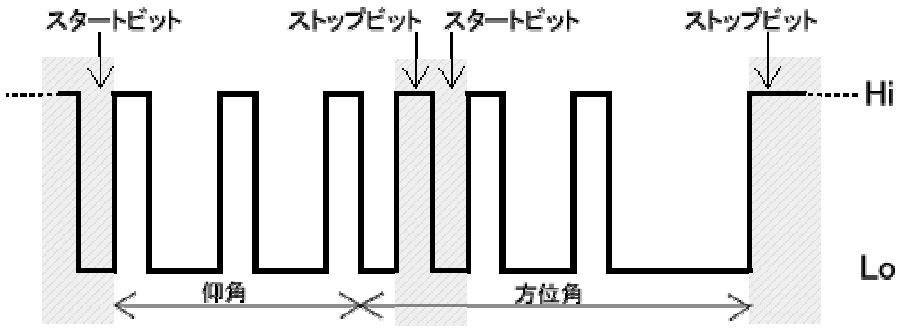

図 3.2　位置情報の送信パルス例(仰角 73 度$(1001001)_2$、方位角 144 度$(010010000)_2$)

3.3 フローチャート

本プログラムは起動後、仰角を初期位置に設定した後、スタートビットが送信されるのを待つ。スタートビット受信後は図 3.3 のタイムフロー図の流れのように処理を

行う。スタートビット認識後 2 文字連続で受信し、適宜処理を行いステッピングモータにパルスを送り、続けてサーボモータにパルスを送る。サーボモータにパルスを送った後は PIC 内臓のタイマで正確に 11 文字目まで待つ。これにより連続受信に対応できる。

ステッピングモータには 1 ステップで 0.9° 動くパルスが 10.4ms ごとに送られる。サーボモータには先述したように 20ms ごとに任意の幅のパルスを送るので、10.4ms のループの 2 回に 1 回ごとに送信する。その結果 20.8ms 周期の送信になるが、問題なく動作する事を確認した。

本プログラムでは 10.4ms 経過し、位置情報が受信できない場合に変数 OVERWAIT をデクリメントし、0 になるまでループする。受信できている間はデクリメントされない。変数 OVERWAIT は 255 に設定しているので、位置情報が途絶えてから約 2.6 秒間このループが繰り返される。

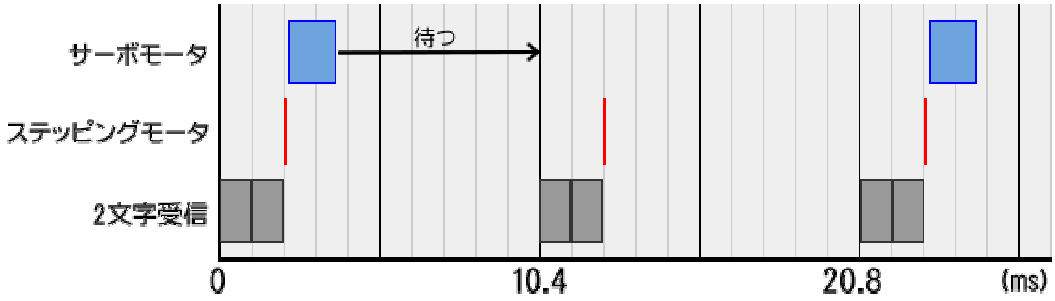

図 3.3　プログラムのタイムフロー図

今回使用したプログラムの内メインルーチン部分のフローチャートを図 3.4 に示す。太い線の部分がスタートビット待ちの待機ループである。関数等の詳細な解説は巻末の付録を参照願いたい。

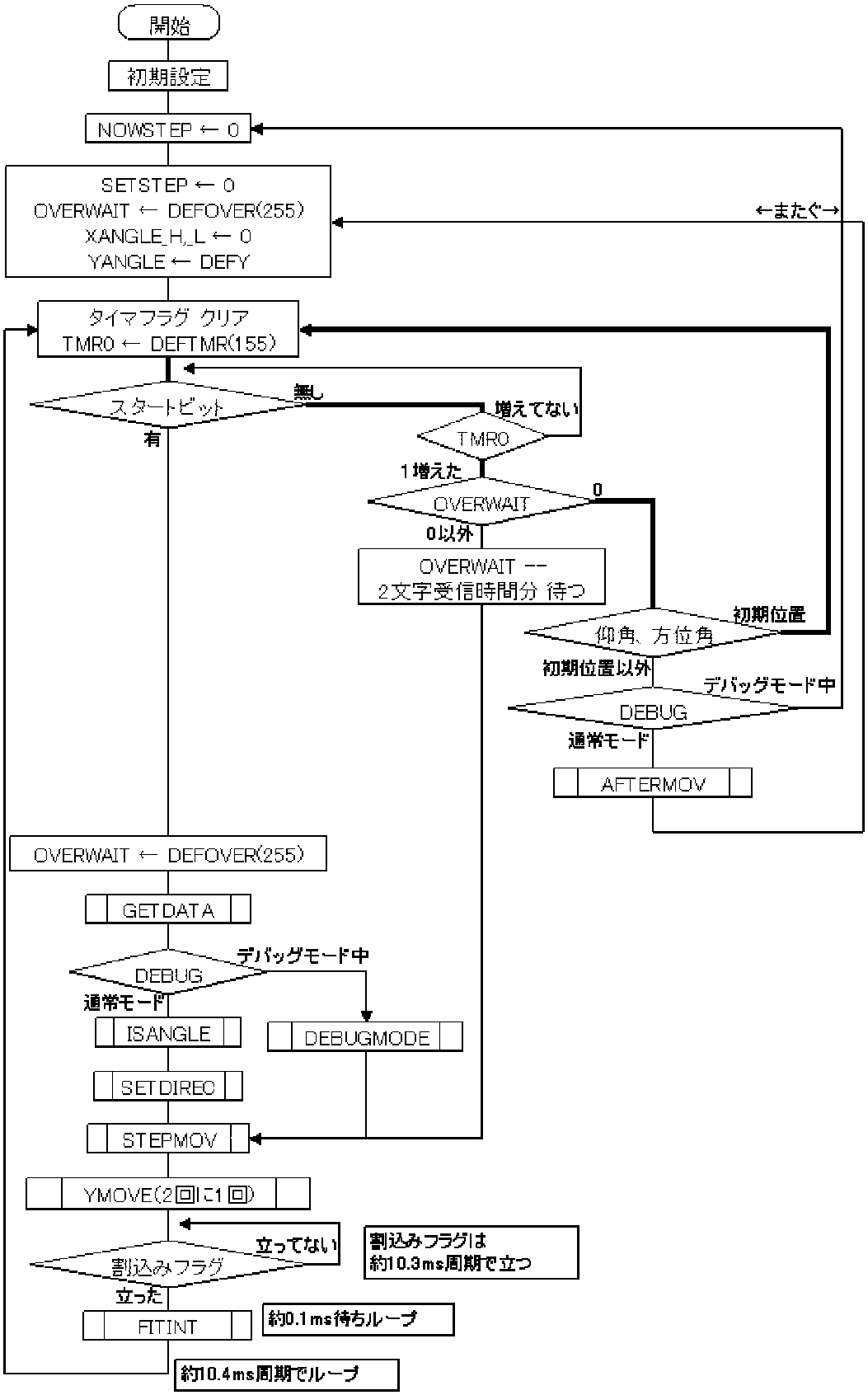

図 3.4　メインルーチンのフローチャート

3.4 使用方法とデバッグ

本装置は、仕様どおりのデータを送信可能なアプリケーションからの位置情報に基づいた動作が可能である。しかし、設置後の連続稼働においては、ノイズによる誤動作や､観測対象の変化によってプログラムの微調整が必要とされることも予想される。このため本装置にはデバッグモードを搭載した。デバッグモードの操作方法を表 3.2 に示す。

表 3.2　デバッグモードの操作方法

1byte 目(16 進数)	2byte 目(16 進数)	処理内容
77	77	デバッグモード開始
00	00～FF	正転、任意の角度
11	00～FF	逆転、任意の角度
22	00～FF	モータ可動後の待機時間(秒)
33	00～5A	仰角の初期位置
57	57	デバッグモード終了

なお、デバッグモード時は入力したコマンドと処理結果がコールバックされる。開始命令と終了命令はコマンドだけコールバックされる。また、本装置が入力可能な状態になると常に「39 (16 進数)」が表示される。モータ可動後の待機時間と仰角の初期位置は PIC 内臓の EEPROM に書き込む仕様としているので､装置の電源を入れ直しても、その記憶は保持される。データシートによると EEPROM の書き換え回数は 10,000,000 回、40 年間保障されているので、事実上無制限に保持される。本モードはカメラを設置する際に方位角 0 度を北に向ける際にも用いられる。

第4章　評価

4.1 動作速度性能と考察

仰角の初期位置を 90 度、方位角は現在向いている方向を 0 度として本装置の性能を評価する。PC から 2 文字の位置情報を送信し、カメラが停止するまでの所要時間をストップウォッチで計測した結果を表 4.1 に示す。

表 4.1　カメラの動作所要時間

送信した位置情報[度]	所要時間[s]
仰角 90 方位角 40	0.00 0.44
仰角 20 方位角 180	0.19 0.00
仰角 45 方位角 300	0.12 0.66

流星は平均的には約 1 秒間観測できる。全天カメラで流星を観測して位置情報を計算し、送信するまで 0.1 秒必要だったとすると、1 秒間出現している流星を 0.5 秒以上観測できる範囲は、移動所要時間が 0.4 秒以下の範囲である。仰角を稼働させるサーボモータは 0.2 秒で 90 度稼働できるので、仰角はどの角度でも問題ない。律速は方位角を稼働させるステッピングモータで、0.4 秒で稼働できる角度は初期位置から 36 度の範囲、即ち 0 度と 180 度を中心とした、その前後 36 度の範囲である。

低速長経路流星の場合は 10 秒程度に及ぶ例もある。このような場合には全天の範囲で追従できる速度がある。サーボモータに比べてステッピングモータが低速なので、初期位置からならば方位角が 10 度移動する間に仰角はどの角度にも到達できる。ステッピングモータが律速であり、動作時間の上では水平軸方向も大型のサーボモータに切り替える必要性がある。

4.2 今後の課題

プログラムの設計段階では位置情報の連続受信に対応させており、シミュレータでは連続受信も可能であった。しかし、実装した際に原因不明のタイミングのずれが生じ、連続で位置情報が送信されると正確にスタートビットを認識できず、誤った位置情報として認識される。

方位角を稼働させるステッピングモータの回転速度は遅く、最も回転する 90 度の場合だと、回転が終わるまで 1 秒程度必要である。全天を効率よく観測するためには、高速で回転するステッピングモータを使用する必要がある。サーボモータと同等かそれ以上の回転速度があれば十分と考えられる。なお、その際には初動トルクに気をつける必要がある。

ステッピングモータの動作中は約 10ms ごとにパルスを送るので、100Hz で振動することになる。このステッピングモータには仰角回転用のサーボモータと支持機構及びカメラが搭載されているため重量があり 100Hz の振動は機構部品が緩む原因となっている。本研究で使用した 1-2 相励磁はパルスで駆動しているため振動も激しいが、マイクロステップ駆動は電流を制御して励磁するため、より滑らかな駆動が可能となり、機構部品の緩みも減少すると思われる。

本研究を始めるにあたって、PIC プロセッサで一番スタンダードな PIC16F84A を選定し、更にマイコンの動作を詳しく理解するために C のコンパイラも用意しなかった。この PIC には後述する様なデバイスが内蔵されていないので、本研究のように複雑な動作をさせるプログラムの作成には非常に時間がかかった。PIC の上位機種にはシリアル通信が容易に出来る USART 等の通信モジュールやサーボモータの制御に使える CCP(Capture/Compare/PWM)等のデバイスを内蔵したものがある。例えば PIC16F88 なら PIC16F84A と同じピン数で、戦術の機能を内蔵し、タイマ割りこみに使用できるタイマも PIC16F84A が 8bit 一つなのに対して 8bit が 2 つ、16bit が 1 つ内蔵されている。この様な PIC プロセッサを使用すれば、本研究で作製したプログラムと同等かそれ以上の機能を持ち、メンテナンス性も高いプログラムが容易に作製できるだろう。

本研究では屋外に観測装置を設置するまでには至らなかったので、風雨や日光に耐えられる材質を考慮し、機密性や装置の直射日光への対策を考慮する必要がある。

本装置は流星観測用に製作したが、この装置がインターネット等で遠隔操作できれば流星観測以外にも様々な応用分野が見込まれる。

以上の課題は、高知工科大学 電子・光システム工学科における今後の課題として解決されることを期待する。

第5章　結論

本研究の目的は、夜空に瞬く間に現れては消え行く流星を鮮明な拡大画像で撮影する野心的なものであった。現時点で完成した装置はそのプロトタイプをなす機能を備えているが、実用性の面では速度面で不足があり約 0.1 秒の動作範囲ではカメラの初期位置付近の流星しか捉えることが出来ないと予測される。装置の検証では、実験室での性能評価にとどまり実際の観測装置の設置にはいたらなかった。しかし、隕石落下などの場合比較的長経路の低速流星となる場合も多く、本装置の速度で追従できる範囲にある。

他方、組込み系ソフトウェアの開発に関する基礎を学ぶという観点では、本研究を通じて理論値と実測値の違いを始め、メモリ容量や動作速度の限界等、組み込み装置全般の製作にかかわる様々な要素を学ぶことが出来た。本研究における経験は、今後著者の人生における糧となり、豊かな生涯を送るための礎となることは間違いない。

謝辞

本研究を完遂するに当たり丁寧な助言を賜りました高知工科大学 電子･光システム工学科 山本真行講師に心から感謝いたします。電子回路や観測装置の製作時にノイズや振動等、豊富な経験に基づく適切なアドバイスをして下さり大変感謝しております。ありがとうございました。

装置を製作する際に様々な助言と助力をして下さった同学科岩下研究室の多木勝彦氏と綿森研究室の車載仁氏、矢野研究室の桑田聡子氏に心から感謝いたします。また、カメラ架台の製作に関してご助力頂いた同研究室の湊 洋輔氏に感謝いたします。

3 年後期だけの所属でしたが、高村禎二元客員助教授には大変感謝しています。事あるごとに新しいお話をして下さり、視野を広げることが出来ました。元高村研究室のメンバー、特に 4 年次も同じ研究室に所属した川﨑肇氏、ならびに 4 年からの付き合いになりましたが同研究室の皆様に深い感謝を致します。楽しい学生生活を送ることが出来たのは皆様のおかげです。ありがとうございました。

最後に、楽しく幸せな学生生活を送ることが出来たのは、色々な面で支えてくれた両親と家族のおかげです。心から感謝いたします。

参考文献

[1] 光永法明、後田敏　署　「はじめての PIC アセンブラ入門」(CQ 出版社)

[2] 河西直史、鶴見恵一、山本健一　署　「PIC マイコンによるメカトロニクス入門」(CQ 出版社)

[3] 後閑哲也　署　「たのしくできる PIC 電子工作」(東京電機大学出版局)

[4] 後閑哲也　「電子工作の実験室」(http://www.picfun.com/)(1/31)

付録　ソースコード

```
;****************************************
;
; シリアル通信でカメラを動かす
;
; PIC16F84A 10MHz
;
;****** 概要 ****************************
; 割込み周期は10msに近づけておくと
;   ・毎回ステップモータへのパルス
;   ・2回に1回サーボモータへのパルス
; を送れるので都合が良い
; 今回の通信仕様だと10文字受信する時間が10[ms]に近い
;  受信10文字 (1/9600)*100*10 = 10.4167[ms]
; これに合わせてTMR0割り込みが入る長さを調整する
;  割込み1回  1/(10M/4)*256 = 0.1024[ms]
;  10.4167[ms]/0.1024[ms] = 101.72[ms]
; よって割込み回数は101回
;
; しかし、このままでは割り込みが発生する度に誤差が
; 割込み1回ごとの誤差は
;  10.4167[ms] - 10.3424[ms] = 74.3[us]
; この差を埋めるために FITINT を呼ぶ
;
; その後、スタートビットをTMR0が1増えるまでの間待ち
; GETDATAで角度と方位に代入しモータにパルスを送る
;
; スタートビットが来ない場合は保持している角度が
; 初期位置のままでない時だけ、数秒待ち初期位置に戻す。
; 初期位置の場合はスタートビットを待つ。
;
;****** 通信仕様(外部) ******************
; ･RS-232C通信仕様
;      通信速度            9600bps
;      スタートビット      1bit
;      ストップビット      1bit
;      データ長            8bit
;      パリティビット      無し
;      フロー制御          無し
;
; ･通信で使用するポート
;      PORTA RA2 出力  →(RxD)
;      PORTA RA3 入力  ←(TxD)
;
; ･送信サブルーチン(PUTCHAR)
;      引数       TXREG
;      戻り値     無し
;
; ･受信サブルーチン(GETCHAR)
;      引数       無し
;      戻り値     RXREG
;      スタートビット受信確認後呼び出す
;
;****** 通信仕様(内部) ******************
; ･方位角 0度-360度 9bit(512)
;      XANGLE_H, XANGLE_L
; ･仰角    0度-90度 7bit(128)
;      YANGLE
;
; ･仰角、方位角の順に送信
;      仰角を1bit左にシフトし、
;      方位角の最上位bitを仰角の後ろに追加し1文字として送る

;****** Y軸仕様 *************************
```

```
; ･サーボモータ(GWS MICRO-MG)の仕様
;      初期位置              45度
;      稼働範囲              大体180度
;      パルス幅               0.6ms-2.4Mms 1.5msでニュートラル
;      パラメータ            YANGLE 0-180
;      ※これに下駄(+60)を足し10us倍してパルス幅とする
;      ※ステッピングが高速な場合は、0-90のほうが早い
;      ※遅いから工夫して180度まで回す！
;
; ･制御に使用するポート
;      PORTB RB3
;
; ･サブルーチン(YMOVE)
;      YANGLEに角度を入れて呼び出す
;
;****** X軸仕様 **************************
; ･ステッピングモータ(秋月の安いやつ)の仕様
;      初期位置              90度  パルス終了後2s待って初期位置に
;      稼働範囲              360度
;      パラメータ            上位1bit:XANGLE_H 下位8bit:_L 0-360
;      ※例       300度 XANGLE_H:0000 0001
;                       XANGLE_L:0010 1100
;
; ･制御に使用するポート
;      PORTB RB4-RB7で
;
; ･サブルーチン(XMOVE)
;      XANGLE_H,_Lに角度を入れて呼び出す
;
;****************************************   *
      LIST                  P=PIC16F84A
      INCLUDE   "P16F84A.INC"
      __CONFIG _HS_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

;****************************
;変数宣言
;****************************
XANGLE_H        EQU       0x20              ;X軸の一番上の
XANGLE_L        EQU       0x21              ;X軸の方位
YANGLE          EQU       0x22              ;Y軸の角度
XCNT            EQU       0x23              ;ステップモータ駆動用
DIREC           EQU       0x24              ;ステップモータ0=正転 1=逆転
SETSTEP         EQU       0x25              ;到達すべき角度
NOWSTEP         EQU       0x26              ;現在の角度
SERVTIM         EQU       0x27              ;サーボに割込みを2回に1回入れるため
DOTONE          EQU       0x28              ;角度を1.1倍する時の。

OVERWAIT        EQU       0x2A              ;受信後モータを稼働させる時間
LOOKTIME        EQU       0x2B              ;向いた後保持する秒数
LOOKT           EQU       0x2C

DEBUG           EQU       0x2F              ;デバッグモード0:OFF 1:ON

RXREG           EQU       0x30              ;受信データバッファ
TXREG           EQU       0x31              ;送信データバッファ
BITLENG         EQU       0x32              ;送受信データ長さ
BTIME           EQU       0x33              ;パルス幅調整用
RXTMP           EQU       0x34              ;2文字連続受信するためのテンポラリ


CNT1            EQU       0x40              ;タイマ用カウンタ
CNT2            EQU       0x41
CNT3            EQU       0x42
CNT_NS          EQU       0x43
CNT_B           EQU       0x44              ;パルス幅待ち用
CNT_F           EQU       0x45              ;転送速度とクロックの隙間を埋める
```

```
DEFY            EQU     0x46            ;仰角初期位置
DEFTMR          EQU     D'155'          ;割込み回数 256-101
DEFOVER         EQU     D'255'          ;データ受信後、カメラ固定で待つ時間
                                        ;割込み時間*255=2.55s

        ORG     0x0
        GOTO    START

        ORG     0x4
        RETFIE

;*************************
;初期設定
;*************************
START
        BCF     STATUS, RP0             ;バンク0に
        CLRF    INTCON                  ;割込み禁止
        CLRF    PORTA                   ;入出力の設定前にポートを初期化
        CLRF    PORTB

        ;入出力、割込みプリスケーラ設定
        BSF     STATUS, RP0             ;バンク1に
        CLRF    TRISA                   ;全て出力
        CLRF    TRISB                   ;全て出力
        BSF     TRISA,3                 ;RA3だけ入力
        MOVLW   B'11010111'             ;プリスケーラは(1/256)に
        MOVWF   OPTION_REG
        BCF     STATUS, RP0

        ;パルス幅
        ; RS232C通信速度設定用変数
        ; BTIME = ((周波数(Hz)/転送速度(bps))/4 - 10)/ 3
        ; 9600bps@10MHz = 83
        MOVLW   D'83'
        MOVWF   BTIME

        ;ステップカウンタ初期化
        CLRF    XCNT
        CLRF    SERVTIM

        ;デバッグモードOFF
        CLRF    DEBUG

        ;EEPROMから読み込み
        ;向いた後待つ時間
        MOVLW   0x00
        MOVWF   EEADR                   ;読み込みアドレス
        BSF     STATUS, RP0             ;バンク1に
        BSF     EECON1, RD              ;読み込み
        BCF     STATUS, RP0             ;バンク0に
        MOVF    EEDATA, W
        MOVWF   LOOKTIME
        ;仰角
        MOVLW   0x01
        MOVWF   EEADR                   ;読み込みアドレス
        BSF     STATUS, RP0             ;バンク1に
        BSF     EECON1, RD              ;読み込み
        BCF     STATUS, RP0             ;バンク0に
        MOVF    EEDATA, W
        MOVWF   DEFY
        ;初期設定終わり

RESET_DE
        CLRF    NOWSTEP                 ;ステップモータ現在地

RESET
        CLRF    SETSTEP                 ;ステップモータ目的地
```

```
        ;データ未受信後待機時間(割込み*OVERWAIT)
        MOVLW     DEFOVER                 ;文字が来なくなっても
        MOVWF     OVERWAIT                ;2sはパルスを送り続ける

        ;方位角、仰角の初期値
        CLRF      XANGLE_H
        CLRF      XANGLE_L
        MOVF      DEFY, W
        MOVWF     YANGLE
;***************************
;
;メインルーチン
;
;***************************
MAIN
        BCF       INTCON, T0IF            ;フラグのクリア
        MOVLW     DEFTMR                  ;タイマ再セット
        MOVWF     TMR0

CHKBIT
        BTFSS     PORTA, 3                ;1スタートビット確認
        GOTO      GETANGLE

        ;TMR0が1増えるまで
        MOVLW     DEFTMR
        SUBWF     TMR0, W                 ;TMR0-155
        BTFSC     STATUS, Z               ;TMR0が増えない間はZ=1
        GOTO      CHKBIT

        ;2秒経つまで
        MOVF      OVERWAIT, F             ;0ならZが1
        BTFSC     STATUS, Z
        GOTO      ISDEF                   ;OVERWAITが0な場合
        DECF      OVERWAIT, F             ;まだ0じゃない時
        CALL      TIME1                   ;一定間隔でパルスを送るための待ち
        CALL      TIME1                   ;2文字分受信する時間分の待ち
        GOTO      PULSE

ISDEF
        ;仰角は初期値か?
        MOVF      DEFY, W
        SUBWF     YANGLE, W               ;YANGLE-45
        BTFSS     STATUS, Z               ;初期値ならZ=1
        GOTO      LOOKWAIT

        ;方位角は初期値か?
        MOVF      NOWSTEP, F
        BTFSC     STATUS, Z               ;初期値(0)ならZ=1
        GOTO      MAIN                    ;初期値の場合

        ;初期位置じゃない場合
LOOKWAIT
        ;デバッグモードの時
        ;初期位置に戻す処理をせずにリセット
        BTFSC     DEBUG, 0
        GOTO      RESET_DE

        CALL      AFTERMOV
        GOTO      RESET

GETANGLE
        MOVLW     DEFOVER
        MOVWF     OVERWAIT                ;待ち時間の初期化

        CALL      GETDATA                 ;2文字格納
        BTFSC     DEBUG, 0
```

```
        CALL    DEBUGMODE
        BTFSC   DEBUG, 0
        GOTO    PULSE                   ;デバッグ時は下のを無視

        CALL    ISANGLE                 ;90,360の制限
        CALL    SETDIREC                ;正逆転、距離の設定

PULSE
        ;パルス送信
        CALL    STEPMOV                 ;ステップモータを回す
        BTFSC   SERVTIM, 0              ;2回に1回しかサーボに送らない
        GOTO    MOVSERV
        BSF     SERVTIM, 0
        GOTO    WAITIN
MOVSERV
        CALL    YMOVE                   ;サーボモータ駆動
        BCF     SERVTIM, 0

WAITIN
        ;それぞれが0なら入力可とみなして39を表示
        MOVF    OVERWAIT, F
        BTFSS   STATUS, Z
        GOTO    WAITING
        MOVF    SETSTEP, F
        BTFSS   STATUS, Z
        GOTO    WAITING
                MOVLW   H'39'           ;thx
                MOVWF   TXREG           ;
                CALL    PUTCHAR         ;
WAITING
        BTFSS   INTCON, T0IF            ;割込み待ち
        GOTO    $-1
        CALL    FITINT                  ;74usぐらい待つ
        GOTO    MAIN

;*****************************
;
;サブルーチン
;
;*****************************
;*****************************
;カメラ移動した後処理
;*****************************
AFTERMOV:
        ;何秒か待つ
        MOVF    LOOKTIME, W
        MOVWF   LOOKT
LOOKSEC
        MOVF    LOOKT, F
        BTFSC   STATUS, Z
        GOTO    RVSDIREC
                MOVF    LOOKT, W        ;残り秒数カウント
                MOVWF   TXREG           ;
                CALL    PUTCHAR         ;
        CALL    TIME1S
        DECF    LOOKT, F
        GOTO    LOOKSEC
RVSDIREC
        BTFSS   DIREC, 0                ;ステップの正逆反転
        GOTO    SET1
        BCF     DIREC, 0
        GOTO    AFTERRET
SET1
        BSF     DIREC, 0
        GOTO    AFTERRET                ;初期値に戻す
AFTERRET
        RETURN
```

```
;**************************
;方位角(0-360)、仰角(0-90)の制限
;超えてたら最大値に
;**************************
ISANGLE:
      MOVF      XANGLE_H, W
      SUBLW     D'1'                    ;1-XANGLE_H
      BTFSS     STATUS, C               ;_Hが2以上ならC=0
      GOTO      ISMAX                   ;_Hが2以上
      BTFSS     STATUS, Z               ;_Hが1ならZ=1
      GOTO      ISY                     ;_Hが0

      ;_Hが1,_Lが104超えてたら、104に
      MOVF      XANGLE_L, W
      SUBLW     D'104'
      BTFSC     STATUS, C
      GOTO      ISY
      MOVLW     D'104'
      MOVWF     XANGLE_L

ISMAX
      ;_Hが2以上,最大値の360に
      MOVLW     D'1'
      MOVWF     XANGLE_H
      MOVLW     D'104'
      MOVWF     XANGLE_L
      GOTO      ISY

ISY
      MOVF      YANGLE, W
      SUBLW     D'90'                   ;90-YANGLE
      BTFSC     STATUS, C               ;マイナスならC=0
      GOTO      ISRET
      MOVLW     D'90'
      MOVWF     YANGLE                  ;90度を超えたら、90に

ISRET
      RETURN

;**************************
;XANGLEによってステップモータのパラメータを決める
;DIREC 0:正 1:逆
;SETSTEP ステップモータに送るステップ数
;SETSET以降は回転の効率化処理,SETSTEP,YANGLE,DIREC
;**************************
SETDIREC
      BTFSC     XANGLE_H, 0
      GOTO      RVSHIGH                 ;256-359:逆転
      MOVLW     D'180'
      SUBWF     XANGLE_L, W             ;XANGLE_L-180
      BTFSS     STATUS, C               ;正値ならCは1
      GOTO      FWDSET                  ;  0-179:正転
      GOTO      RVSLOW                  ;180-255:逆転

      ;正転
FWDSET
      BCF       DIREC, 0
      MOVF      XANGLE_L, W
      GOTO      SETSET

      ;逆転
RVSHIGH
      BSF       DIREC, 0
      MOVF      XANGLE_L, W
      SUBLW     D'104'                  ;(360-256)-(XANGLE-256)
      GOTO      SETSET
```

```
RVSLOW
      BSF       DIREC, 0
      MOVF      XANGLE_L, W
      SUBLW     D'255'
      ADDLW     D'105'                  ;255-XANGLE_L+105 (; ;)1byte..
      GOTO      SETSET

SETSET
      MOVWF     SETSTEP
      SUBLW     D'90'                   ;90-SETSTEP
      BTFSC     STATUS,C                ;マイナスならC=0
      GOTO      CONVERT

      ;SETSTEP=180-SETSTEP
      MOVF      SETSTEP, W
      SUBLW     D'180'                  ;180-SETSTEP
      MOVWF     SETSTEP

      ;YANGLE=180-YANGLE
      MOVF      YANGLE, W
      SUBLW     D'180'                  ;180-YANGLE
      MOVWF     YANGLE

      ;DIREC反転
      BTFSS     DIREC, 0                ;ステップの正逆反転
      GOTO      SETSET1
      BCF       DIREC, 0
      GOTO      CONVERT
SETSET1
      BSF       DIREC, 0
      GOTO      CONVERT

CONVERT
      ;角度からステップ数への変換(1.1倍)
      MOVLW     D'10'
      MOVWF     DOTONE
ADDSET
      ;10カウントするごとに1追加
      SUBWF     SETSTEP, W              ;SETSTEP - DOTONE
      BTFSS     STATUS, C               ;上のが負になるとCは0
      RETURN

      MOVF      DOTONE, W
      ADDLW     D'10'
      MOVWF     DOTONE                  ;DOTONE+=10
      INCF      SETSTEP, F              ;SETSTEP++

      GOTO      ADDSET

;**************************
;ステップをどっち向きに回すか
;現在位置と到達予定位置等を比較
;**************************
STEPMOV:
      MOVF      SETSTEP, F              ;
      BTFSC     STATUS, Z               ;if(setstep != 0){
      GOTO      RVSSTEP                 ;  if((setstep-nowstep) != 0){
FWDSTEP                                 ;    nowstep++;
      MOVF      NOWSTEP, W              ;    xmove();
      SUBWF     SETSTEP, W              ;  }
      BTFSC     STATUS, Z               ;}else{
      GOTO      NOTSTEP                 ;  if(nowstep != 0){
      INCF      NOWSTEP, F              ;    nowstep--;
      GOTO      DOSTEP                  ;    xmove();
RVSSTEP                                 ;  }
      MOVF      NOWSTEP, F              ;}
```

```
      BTFSC     STATUS, Z
      GOTO      NOTSTEP
      DECF      NOWSTEP, F
DOSTEP
      CALL      XMOVE                   ;ステップモータ駆動
      GOTO      RETSTEP

NOTSTEP
      ;デバッグモード時は待たない
      BTFSS     DEBUG, 0
      GOTO      RETSTEP
      CLRF      OVERWAIT                ;もう待たない
RETSTEP
      RETURN

;***************************
;X軸モーター制御(ステッピングモータ)
;正転、逆転に1ステップ送るだけ
;***************************
XMOVE:
      BTFSS     DIREC, 0
      GOTO      FWD                     ;DIREC=0 正転
      GOTO      RVS                     ;DIREC=1 逆転

FWD   ;正転
      MOVF      XCNT, W                 ;IF(W == 0)XCNT=7
      BTFSC     STATUS, Z
      GOTO      SET7
      DECF      XCNT, F                 ;XCNT--
      GOTO      STEPSTEP
SET7  MOVLW     D'7'
      MOVWF     XCNT
      GOTO      STEPSTEP

RVS   ;逆転
      MOVF      XCNT, W
      SUBLW     D'7'                    ;IF((7-W) == 0)XCNT=0
      BTFSC     STATUS, Z
      GOTO      SET0
      INCF      XCNT, F                 ;XCNT++
      GOTO      STEPSTEP
SET0  CLRF      XCNT
      GOTO      STEPSTEP

STEPSTEP
      ;パルス出力
      MOVF      XCNT, W
      CALL      TABLE
      MOVWF     PORTB
      RETURN

TABLE                                   ;PCLがアドレスの255をまたがないように注意
      ADDWF     PCL, F                  ;W(0-7)を参照して返り値を決める
      RETLW     B'10010000'             ;上位4ビットが有効
      RETLW     B'00010000'
      RETLW     B'00110000'
      RETLW     B'00100000'
      RETLW     B'01100000'
      RETLW     B'01000000'
      RETLW     B'11000000'
      RETLW     B'10000000'

;***************************
;Y軸モーター制御(サーボモータ)
;     YANGLE(絶対値 0-180まで可)
;***************************
YMOVE:
```

```
        ;180までの制限(モータ保護)
        MOVF    YANGLE, W
        SUBLW   D'180'                  ;180-YANGLE
        BTFSC   STATUS, C               ;マイナスならC=0
        GOTO    YPULSE
        MOVLW   D'180'
        MOVWF   YANGLE                  ;180度を超えたら180に

YPULSE
        MOVF    YANGLE, W               ;下駄
        ADDLW   D'69'                   ;昔は60

        BSF     PORTB, 3                ;ON
        CALL    TIMENS                  ;幅を10us倍する
        BCF     PORTB, 3                ;OFF
        RETURN

;**************************
;2文字受信して7bitと9bitに格納
;"ww" : デバッグモードON
;"WW" : デバッグモードOFF
;**************************
GETDATA:
        ;2文字分データ受信
        CALL    GETCHAR                 ;1文字目受信
        MOVF    RXREG, W                ;急ぐためテンポラリに退避
        MOVWF   RXTMP
        CALL    TIMEWID                 ;2文字目のスタートビット分待つ
        CALL    GETCHAR                 ;スタートビット以後を読み込む

        ;デバッグモードに入るかどうか
        BTFSC   DEBUG, 0
        GOTO    ENDCHK
        ;"ww"でON
        MOVLW   H'77'
        SUBWF   RXTMP, W
        BTFSS   STATUS, Z
        GOTO    INDATA
        MOVLW   H'77'
        SUBWF   RXREG, W
        BTFSS   STATUS, Z
        GOTO    INDATA
        BSF     DEBUG, 0                ;デバッグモードON

        ;デバッグモードに入る時
        ;方位角、仰角の初期化
        CLRF    XANGLE_H
        CLRF    XANGLE_L
        CLRF    SERVTIM
        MOVF    DEFY, W
        MOVWF   YANGLE
                MOVLW   H'77'           ;デバッグ用
                MOVWF   TXREG           ;
                CALL    PUTCHAR         ;
        GOTO    GETEND
ENDCHK
        ;"WW"でOFF
        MOVLW   H'57'
        SUBWF   RXTMP, W
        BTFSS   STATUS, Z
        GOTO    GETEND
        MOVLW   H'57'
        SUBWF   RXREG, W
        BTFSS   STATUS, Z
        GOTO    GETEND
        BCF     DEBUG, 0                ;デバッグモードOFF
                MOVLW   H'57'           ;デバッグ用
```

```
                MOVWF     TXREG          ;
                CALL      PUTCHAR        ;
        GOTO      GETEND

INDATA
        ;データ格納
        ;方位角の下位8bit
        MOVF      RXREG, W               ;方位角の下8bit格納
        MOVWF     XANGLE_L

        ;方位角の上位1bit、仰角格納
        BCF       STATUS, C              ;方位角の9bit目初期化
        RRF       RXTMP, W               ;YANGLEに入れるため,W

        BTFSS     STATUS, C
        BCF       XANGLE_H, 0            ;方位角の上1bitを0にする
        BTFSC     STATUS, C              ;0の場合スキップ
        BSF       XANGLE_H, 0

        MOVWF     YANGLE
GETEND
        RETURN

;****************************
;1バイト受信
;       スタートビット受信後呼ばれる
;       ストップビットを調べろ！
;****************************
GETCHAR:
        MOVLW     D'9'                   ;ビットカウンタ初期値
        MOVWF     BITLENG                ;初期値セット
        CLRF      RXREG                  ;バッファクリア

RECEIVE
        BCF       STATUS, C              ;RRFでCフラグを使うため初期化
        BTFSC     PORTA, 3               ;受信ビットが0ならスキップ
        BSF       STATUS, C              ;1の時1にセット
        RRF       RXREG,F                ;RXREGに1ビット書き込み、シフト
        CALL      TIMEWID                ;待つ

        DECFSZ    BITLENG, F             ;9回(DATA+STOP)まわす
        GOTO      RECEIVE
        RETURN

;****************************
;1バイト送信
;****************************
PUTCHAR:
        MOVLW     D'8'                   ;データ長設定
        MOVWF     BITLENG
        BCF       PORTA,2                ;スタートビット送信+待ち
        CALL      TIMEWID

TRANSMIT
        RRF       TXREG, F               ;最下位ビットをCに移して他はシフト

        BTFSS     STATUS, C              ;1ならスキップ
        BCF       PORTA, 2               ;0を出力
        BTFSC     STATUS, C              ;0ならスキップ
        BSF       PORTA, 2               ;1を出力
        CALL      TIMEWID                ;待ち

        DECFSZ    BITLENG, F             ;8回す
        GOTO      TRANSMIT

        BSF       PORTA, 2               ;ストップビット送信+待ち
        CALL      TIMEWID
```

```
      RETURN

;****************************
;  タイマー
;     TIMEWID   :  102msec :RS232Cパルス待ち 9600BPS@10MHz
;     TIMENS    : W*10usec :PWM(W = 60-240)
;     FITINT    : 74.3usec :通信速度とTMR0の差を埋める
;****************************
TIMEWID:
      MOVF      BTIME, W                ;BTIMEはあらかじめ定義
      MOVWF     CNT_B
      DECFSZ    CNT_B, F
      GOTO      $-1                     ;2+BTIME*3-1
      RETURN                            ;BTIME*3+1+(CALL)

TIMENS:                                 ;W * 10us 1500usが真ん中
      MOVWF     CNT_NS
LOOP_NS
      GOTO      $+1
      GOTO      $+1
      GOTO      $+1
      GOTO      $+1
      GOTO      $+1

      GOTO      $+1
      GOTO      $+1
      GOTO      $+1
      GOTO      $+1
      GOTO      $+1

      GOTO      $+1

      DECFSZ    CNT_NS, F
      GOTO      LOOP_NS                 ;1+(22+3)*W-1
      RETURN                            ;W*25+2+2(CALL)

FITINT:                                 ;185サイクル
      MOVLW     D'57'                   ;74.3u 大体181サイクル
      MOVWF     CNT_F
      DECFSZ    CNT_F, F
      GOTO      $-1                     ;2+3*60-1

      RETURN                            ;181+1+2(CALL)

;****************************
;     TIME10    :  10usec
;     TIME1     :   1msec
;     TIME1S    :    1sec
;****************************
TIME10:                                 ;10us 25サイクル (STRICT)
      MOVLW     D'7'
      MOVWF     CNT1
T_LP1
      DECFSZ    CNT1, F
      GOTO      T_LP1                   ;2+3*7-1
      RETURN                            ;22+1+2(CALL)

TIME1:                                  ;1ms 2500サイクル (ABOUT)
      MOVLW     D'89'
      MOVWF     CNT2
T_LP2
      CALL      TIME10
      DECFSZ    CNT2, F
      GOTO      T_LP2                   ;2+(25+3)*89-1 = 2493
      RETURN                            ;2493+2+2(CALL) = 2497

TIME1S:                                 ;1S 2500000サイクル(ABOUT)
```

```
        MOVLW     D'250'
        MOVWF     CNT3
T_LP3
        CALL      TIME1
        CALL      TIME1
        CALL      TIME1
        CALL      TIME1
        DECFSZ    CNT3, F
        GOTO      T_LP3                   ;2+(10000+3)*250-1 = 2500751
        RETURN                            ;2500751+1+2(CALL)

;***************************
;デバッグモード
;0x77 0x77 : ONとOFFを切り替え
;0x00 0xXX : XX度正転
;0x11 0xYY : YY度逆転
;0x22 0xZZ : カメラを向けた後ZZ秒待つ
;***************************
DEBUGMODE:

        ;正転
        MOVLW     H'00'
        SUBWF     RXTMP, W
        BTFSS     STATUS, Z
        GOTO      DEBUG1
        BCF       DIREC, 0                ;正転セット
        MOVF      RXREG, W
        MOVWF     SETSTEP
                  MOVLW     H'00'         ;デバッグ用
                  MOVWF     TXREG         ;
                  CALL      PUTCHAR       ;
        GOTO      ENDDEBUG

DEBUG1
        ;逆転
        MOVLW     H'11'
        SUBWF     RXTMP, W
        BTFSS     STATUS, Z
        GOTO      DEBUG2
        BSF       DIREC, 0                ;逆転セット
        MOVF      RXREG, W
        MOVWF     SETSTEP
                  MOVLW     H'11'         ;デバッグ用
                  MOVWF     TXREG         ;
                  CALL      PUTCHAR       ;
        GOTO      ENDDEBUG

DEBUG2
        ;待ち時間
        MOVLW     H'22'
        SUBWF     RXTMP, W
        BTFSS     STATUS, Z
        GOTO      DEBUG3
        MOVF      RXREG, W
        MOVWF     LOOKTIME
        MOVWF     EEDATA

        ;EEPROM書き込み
        MOVLW     0x00
        MOVWF     EEADR
        BSF       STATUS, RP0             ;バンク1に
        BSF       EECON1, WREN            ;EEPROM書き込み有効
        MOVLW     H'55'
        MOVWF     EECON2
        MOVLW     H'AA'
        MOVWF     EECON2
        BSF       EECON1, WR              ;書き込み開始データ
```

```
        BTFSS     EECON1, EEIF            ;書き込み終わるまでまつ
        GOTO      $-1
        CLRF      EECON1                  ;
        BCF       STATUS, RP0             ;バンク0に戻る
                  MOVLW     H'22'         ;デバッグ用
                  MOVWF     TXREG         ;
                  CALL      PUTCHAR       ;
                  MOVF      EEDATA,W      ;デバッグ用
                  MOVWF     TXREG         ;
                  CALL      PUTCHAR       ;

DEBUG3
        ;仰角(0-90度)
        MOVLW     H'33'
        SUBWF     RXTMP, W
        BTFSS     STATUS, Z
        GOTO      ENDDEBUG
        MOVF      RXREG, W

        SUBLW     D'90'                   ;90-YANGLE
        BTFSC     STATUS, C               ;マイナスならC=0
        GOTO      DE3N
        MOVLW     D'90'                   ;90度を超えたら、90に
        GOTO      DE3W
DE3N
        MOVF      RXREG, W
DE3W
        MOVWF     DEFY
        MOVWF     EEDATA
        ;EEPROM書き込み
        MOVLW     0x01
        MOVWF     EEADR
        BSF       STATUS, RP0
        BSF       EECON1, WREN            ;EEPROM書き込み有効
        MOVLW     H'55'
        MOVWF     EECON2
        MOVLW     H'AA'
        MOVWF     EECON2
        BSF       EECON1, WR              ;書き込み開始データ
        BTFSS     EECON1, EEIF            ;書き込み終わるまでまつ
        GOTO      $-1
        CLRF      EECON1                  ;
        BCF       STATUS, RP0             ;バンク0に戻る
                  MOVLW     H'33'         ;デバッグ用
                  MOVWF     TXREG         ;
                  CALL      PUTCHAR       ;
                  MOVF      EEDATA,W      ;デバッグ用
                  MOVWF     TXREG         ;
                  CALL      PUTCHAR       ;

ENDDEBUG
        RETURN

        ;PICに書き込む時にEEPROMに書き込む
        ORG                 0x2100
        DE                  D'1'          ;LOOKTIME初期値
        DE                  D'90'         ;YANGLE初期値

        END
```