

# 卒業研究報告

題 目

DA ボードを用いた流星自動追跡装置の基礎開発

---

指 導 教 員

山本 真行 講師

---

報 告 者

学籍番号： 1060277

氏名： 山下 泰樹

---

平成 18 年 2 月 21 日

高知工科大学 電子・光システム工学科

## 目次

1 章 はじめに	・・・ 2
1-1.背景	
1-2.目的	
2 章 装置の仕様	・・・ 4
2-1.DA ボード	
2-2.ステッピングモータ	
2-3. RS-232C シリアル通信	
2-4.ドライブ回路	
2-5.カメラ	
3 章 システムの構成	・・・ 10
3-1.全体の構成	
3-2.プログラム	
3-3.出力波形のデータに関して	
4 章 評価	・・・ 16
5 章 まとめ	・・・ 17
謝辞	・・・ 18
参考文献	・・・ 18
付録	・・・ 19

# 1 章 はじめに

## 1-1.背景

近年コンピュータの高機能化によって専用の機器を使用しなくてもさまざまなことが簡単にできるようになった。生活の中にも携帯電話、プリンター、冷蔵庫、車などから人工衛星までコンピュータによって制御されている。このような時代背景の中で、制御は無くてはならない技術の一つである。

夜中に突然現れて瞬く間に消える流星はこれまで自動撮影が困難な対象であったが、デジタルカメラ等に用いられる CCD(電荷結合素子)に代表される映像素子の技術発展により、比較的安価な天文用高感度ビデオカメラ Watec Neptune-100(WAT-100N)等が登場し、比較的暗い流星の撮影も可能となった。また、タイムシフト動体検出ソフトウェア UFOCapture(SonotaCo 氏製作のシェアウェア)が、2003 年に登場したことにより、流星の現れた瞬間にトリガをかけることで、前後の一部始終をハードディスクに映像ファイルとして記録できるようになった。現在全国 10 地点以上にて毎晩自動流星観測が行われている。流星観測は電子・光技術の進展に伴い飛躍的に発展している分野の 1 つである。

## 1-2.目的

上述のような状況ではあるが、WAT-100NとUFOCaptureによる自動流星観測では観測確率を上げるため視野角の広いカメラを用いることが多く、撮影された流星の詳細な変化を捉えることは困難であった。高分解能の流星撮影のためには、全天カメラによって位置を特定し、かつリアルタイムで位置情報を用いた制御を行って、拡大撮影カメラによる追跡を行う方法が考えられる。実際に鈴木智氏による実践例が報告されている(火球(横浜): <http://www18.tok2.com/home/satoshi2440841/>)。

本研究では、モータの制御を通して自身のプログラミング技術の向上をするとともに、流星自動追跡装置の基礎技術の確立をテーマとした。画像解析用のパーソナルコンピュータ(PC)から流星の位置情報を受信し、そのデータを元に制御用 PC と DA ボードを使用しステッピングモータを動作させ、カメラ架台を動かして視野方向を制御するソフトの開発を目的とした。なお画像解析用 PC における連続画像処理プログラムについては「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 明日香 2005 年度高知工科大学卒業研究報告)を、PIC による制御開発に関しては「PIC プロセッサを用いた流星自動追跡装置の製作」(大岡 裕志 2005 年度高知工科大学卒業研究報告)を参照されたい。

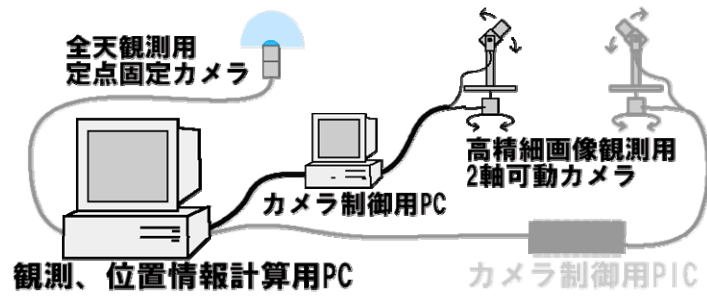
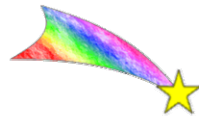


図 1-1 流星自動追跡装置システム全体図

## 2 章 装置の仕様

### 2-1.DA ボード

D/A コンバータを内蔵したインタフェースボードのことを DA ボードと呼ぶ。D/A コンバータとはデジタル・アナログ変換器でありデジタルの値をアナログ量に変換する装置のことを言う。今回使用したボードは、Interface 社製 PCI-3523A という AD/DA ボードである。主な仕様としては、12 ビット DA 変換機能を持ったボードで DA 変換セトリングタイムは 1 チャンネルあたり  $10\mu\text{s}$ 、出力チャンネルは 4 チャンネル、各チャンネルの出力レンジは  $\pm 10\text{V}$  の範囲で設定が可能である。汎用入出力、複数枚同期アナログ出力の機能も備えている。ハードウェアの仕様詳細については表 2-1 に、外観は図 2-1 に示す。

今回使用したピンは 19 番から 26 番を使用した。ピンについては図 2-2、図 2-3 に示す。



図 2-1 DA ボード Interface 社製 PCI-3523A

表 2-1 DA ボード仕様(Interface 社 USER'S MANUAL より抜粋)

項目	仕様
出力仕様	出力電流:5mA 以下 負荷容量:450pF 以下
出力チャンネル数	4 チャンネル
出力レンジ	バイポーラ:±10V
出力インピーダンス	1Ω (TYP)
出力保護	アナログ出力信号はグランド、±15V 電源に対して短絡保護される
分解能	12 ビット
DA 変換器	使用素子:DAC7725U 相当品
相対精度	±1LSB(MAX)(25°C)
誤差	±0.4%(MAX)(0~50°C)
セトリングタイム	10μs(1 チャンネルあたり)
絶縁方式	非絶縁
タイマ	基本クロック:8MHz 用途:タイマ割り込み、変換タイミング発生
電源仕様	DC+5V:0.5A(TYP)
対応バススロット	PCI ローカルバス(Rev.2.1 以上)
使用条件	気温:0~50°C 湿度:20~90%(非結露)
仕様コネクタ	FAP-5001-1202-0BF(山一電機)
ボードサイズ	スーパーショートサイズ

## 2-2.ステッピングモータ

ステッピングモータとは各端子(X, X̄, Y, Ȳ)に順番にパルス信号をあたえることで決められたステップ単位で動作するモータである。特徴としては、回転角度と回転速度を与えるパルス信号の回数と周期によって決定されるため、フィードバック制御なしで意図した角度に容易に動かすことができる。通電した状態では非常に大きな静動トルクが得られる。通電していない状態でも静動トルクがあり、パルスを止めるとその位置で確実に止まるので正確な制御ができる。直流ブラシモータのように機械的磨耗の心配が少なくほとんど保持を必要としない。しかし、トルクはあまり大きくなく、安定して高速で回転することがあまり得意ではない、あたえるパルス幅があまりにも短いと回転しなくなるという欠点もある。

ステッピングモータを回転させるための励磁の方法はいくつかある。まず一番わかりやすく簡単な方法は図 2-2 の様にX→Y→X̄→Ȳ→X...の順番に電流す方法で、ひとつのパルス

に対しワンステップ回転する。励磁の順番を逆にすれば逆回転可能である。しかし、この方法はトルクが小さく安定性があまりよくないので実際にはあまり使われない。今回使用した励磁の方法は図 2-5 の様なパルスをあてた。このようにすることで動作が安定する。

本研究で今回使用したステッピングモータ(多摩川精機製 TS3103N124)の仕様を表 2-2 に示す。

表 2-2 ステッピングモータ仕様(多摩川精機製 TS3103N124)

ステップ数	200
駆動電圧	12V
コイル抵抗	86Ω/相
コイル電流	140mA

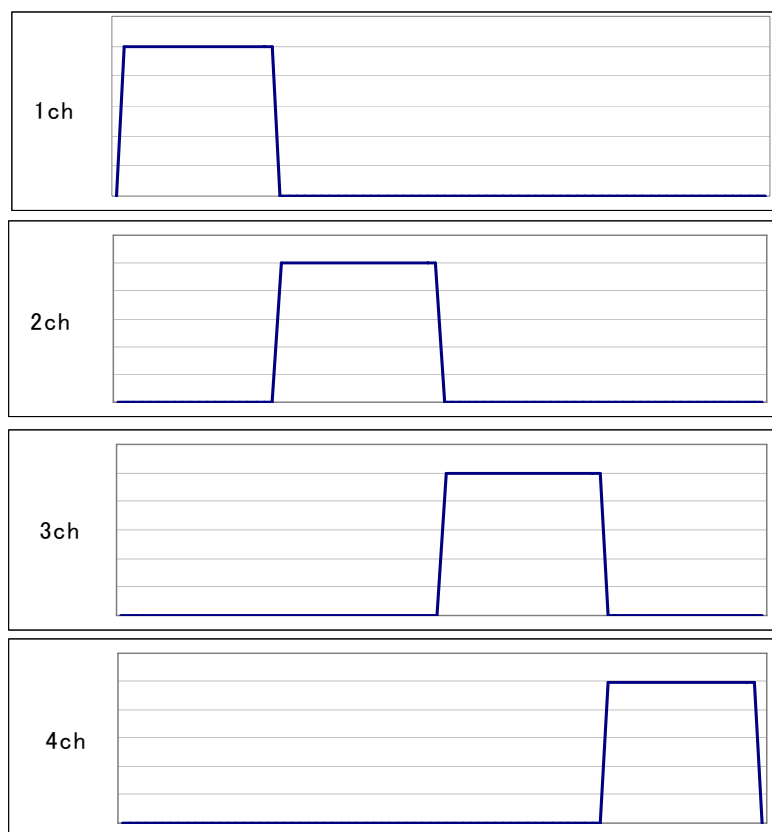


図 2-2 基本的な波形

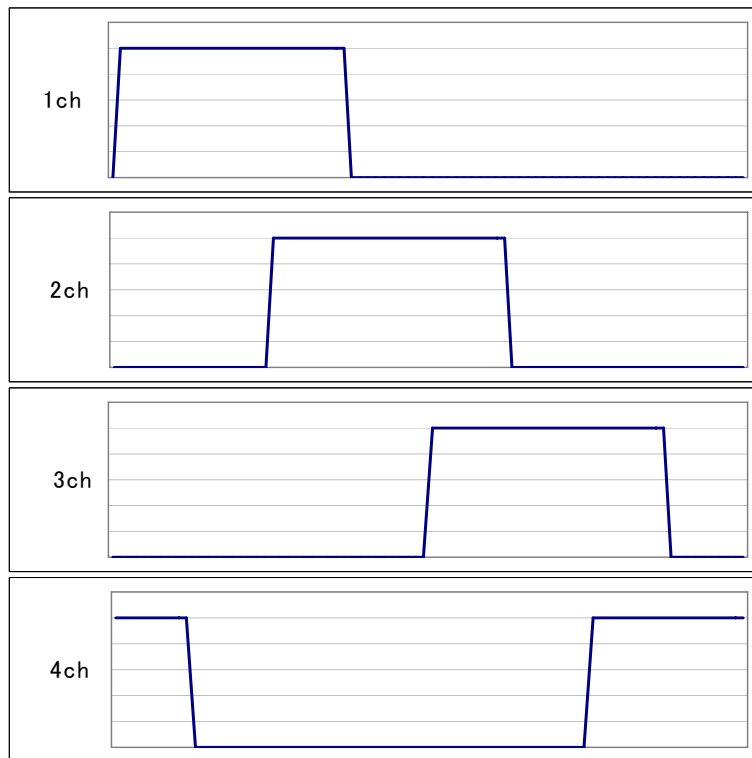


図 2-3 実際の実出力波形

### 2-3.RS-232C シリアル通信

RS-232C はシリアル通信規格の一つでほとんどのパーソナルコンピュータに標準で搭載されている規格である。ケーブルの最大長は約 15m、最高通信速度は 112.5kbps。であり、比較的高電圧の信号を送信するためノイズにも強い。今回の実験では表 2-3 の様な設定で通信を行った。

表 2-3 シリアル通信の設定

通信速度	9600bps
パリティ	なし
データ長	8 ビット
ストップビット長	1 ビット

### 2-4.ドライブ回路について

ドライブ回路は秋月電子の PIC ステッピングモータドライバキット(K-154)のドライブ回路を流用した。回路は非常にシンプルでトランジスタ 4 つとモータの逆起電力から回路を保護するダイオードでできている。

PIC の 6 番から 9 番(図 2-6)に相当する端子に DA ボードからの信号を入力した。DA ボードの 19 番ピンからの出力をドライブ回路の 6 番に、DA ボードの 21 番ピンからの出力をドライブ回路の 8 番に、DA ボードの 23 番ピンからの出力をドライブ回路の 7 番に、DA ボードの 25 番ピンからの出力をドライブ回路の 9 番に入力した。DA ボードの 19,21,23,25

番ピンはアナログ出力の 1~4ch に相当する DA 部に関する出力端子を表 2-6 に、AD/DA ボード全体のピンアサインを図 2-5 に、回路図を図 2-4 に示す。

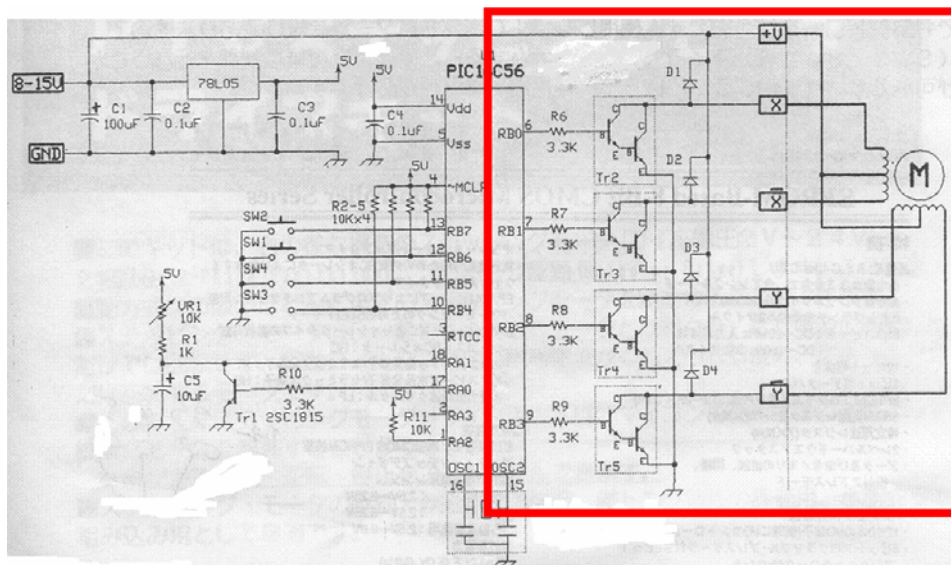


図 2-4 ドライブ回路の回路図

AIN1	1	2	COM1
AIN2	3	4	COM2
AIN3	5	6	COM3
AIN4	7	8	COM4
AIN5	9	10	COM5
AIN6	11	12	COM6
AIN7	13	14	COM7
AIN8	15	16	COM8
NC	17	18	NC
VOUT1	19	20	VCOM1
VOUT2	21	22	VCOM2
VOUT3	23	24	VCOM3
VOUT4	25	26	VCOM4
NC	27	28	NC
IN1	29	30	NC
IN2	31	32	NC
OUT1	33	34	NC
OUT2	35	36	NC
NC	37	38	NC
EXINT IN2	39	40	NC
EXINT IN1	41	42	DCOM
EXTRG IN	43	44	NC
NC	45	46	NC
NC	47	48	NC
NC	49	50	NC

NC : ボード内部で未接続

図 2-5 全端子(Interface 社 USER'S MANUAL より)

信号名	ピン番号	チャンネル	機能	用途	仕様
VOUT1	19	CH1	アナログ出力信号	外部アナログ信号の出力端子	出力インピーダンス: 1Ω (TYP) 出力電流: 5mA以下 負荷容量: 450pF以下
VOUT2	21	CH2			
VOUT3	23	CH3			
VOUT4	25	CH4			
VCOM1	20	CH1	アナログ出力コモン		
VCOM2	22	CH2			
VCOM3	24	CH3			
VCOM4	26	CH4			

図 2-6 出力端子(Interface 社 USER'S MANUAL より)

## 2-5.高感度 CCD カメラ

観測用のカメラは Watec 社製 Neptune-100(WAT-100N)を使用した。WAT-100N は天体用超高感度モノクロ CCD カメラで白黒ではあるが通常肉眼では見えにくい 5 等星ほどの明るさの星も見ることができるカメラで微光天体の映像撮影に適している。WAT-100N の外観と仕様を図 2-7 と表 2-4 に示す。レンズは CBC 製 8mm, f/0.8 が装着されている。



図 2-7 観測用カメラ(Watec 社 Neptune-100N 及び CBC 製 8mm, f/0.8 レンズ)

表 2-4 観測用カメラ(WAT-100N)の仕様

項目	仕様
有効画素数	38 万画素
解像度	570TV 本
最低被写体照度	0.001 lx. @F1.4

## 3章 システム構成

### 3-1.全体の構成

本研究で作成したシステム全体の構成を図 3-1 に示す。RS-232C を使用してデータを受信した後、受信データを元に必要な長さのパルスを生成し DA ボードのアナログ出力ポートより出力する。出力パルスはモータドライブ回路に入る。最後にモータドライブ回路からモータへパルスが入力されモータが動作する。

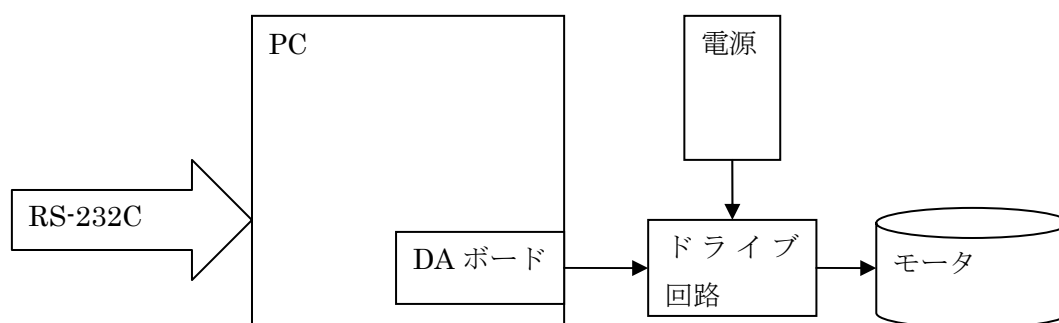


図 3-1 システム全体の構成

### 3-2.プログラムについて

プログラムの開発言語は VisualBasic6.0 を使用した。Visual Basic は直接 I/O やメモリにアクセス出来ない為、DA ボードを直接制御する事が出来ない。そこでボードに対しての制御を行う DLL プログラムから必要な関数を呼び出し、これを介してボードの制御を行うことになる。DLL はボード供給元の Interface 社からダウンロードしインストールして使用した。

本研究にて開発したプログラムには 3 つのモード、即ち、マニュアルモード、流星自動モード、サンダー自動モードを用意した。マニュアルモードは図 3-2 の①のテキストボックスに入力した値を図 3-2 の②の「回す」ボタンを押すことによって入力値に対応したモータの回転動作をさせるモードである。このモードでは RS-232C からの入力が必要なく、デバッグの他、主に初期設定時のカメラ視野決定に用いることを想定している。マニュアルモードでのフローチャートを図 3-3 に示す。

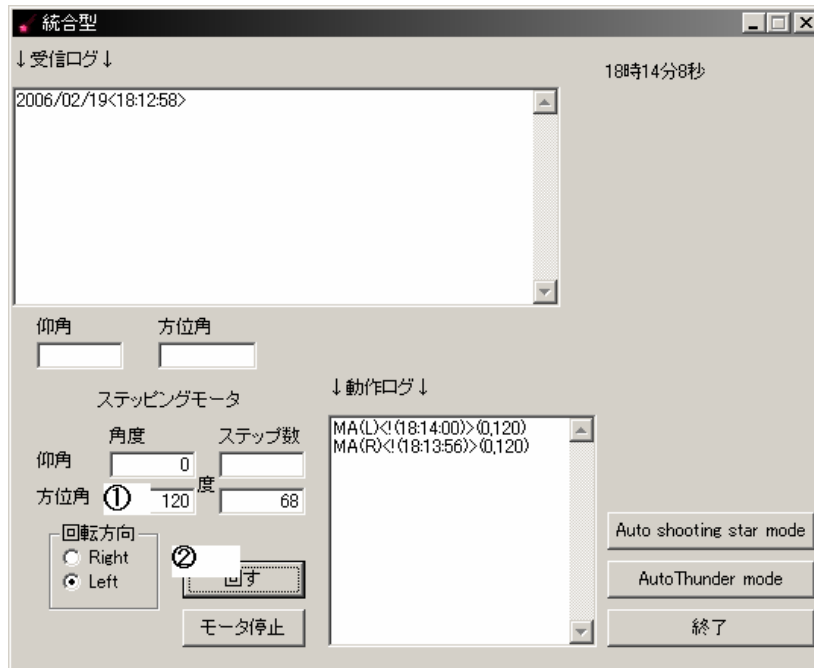


図 3-2 操作画面

流星及びサンダーの自動モードではRS-232Cを介して方位角、仰角のデータを受信する。受信するデータは8ビットバイナリデータ×2の2バイトデータとして受信される。しかし受信された値は仰角方位角ではなく、仰角7ビットと方位角9ビットのデータに変換しなければならない。これは1度精度の方位角(0~359°)と仰角(0~90°)を効率よく送受信する仕様としたためである。この変換は、概念的には仰角側にあるLSBを方位角のMSBにしてやれば完了する。プログラムのいうとデータ変換はまず受信したデータの仰角側の数値を2で割ったとき余りが出れば方位角に256を足す。あまりが出なければ方位角はそのままのデータとする。仰角は余りが出た場合は1を引いてから2で割る。余りが出なかった場合はそのまま2で割る。データの変換はこれで完了する。データ変換について表3-1と表3-2に示す。

表 3-1 受信データについて

仰角	方位角	仰角/方位角(最大値)
8bit	8bit	255/255
7bit	9bit	127/511 (0~90)/(0~359)

表 3-2 受信データ変換例

	仰角	方位角
受信したデータ	10011101(157)	00110111(55)
仰角の LSB を方位角の最上位に移動する	1001110,100110111	
角度データ	1001110(78)	100110111(311)

流星自動モードでのシリアル通信の受信には VisualBasic の MSComm という命令を使う。まず RS-232C ポートを開きデータが送られてくるのを待つ状態にする。データが送られてくるとまず 2 バイトのデータを仰角方位角にする。そのデータを元にステップ数を算出する。そして波形を生成して出力しモータを動作させる。モータを動作させてから 5 秒後に元の位置に戻るようモータを動作させる。そしてまた待ち状態に戻る。自動モードはこの繰り返しの処理を行う。ここで 5 秒という値は流星の継続時間とモータの動作時間を考慮した値である。流星自動モードのフローチャートを図 3-4 に示す。

サンダー自動モードは、流星自動モードを仕様変更したモードである。流星自動モードとサンダー自動モードの違いとしては、流星自動モードは動作後 5 秒で元の位置に戻るが、サンダー自動モードは次のデータが来るまで元の位置には戻らない。プログラム上は一度戻さずに前の位置から次の位置へ回転することは簡単だが、元の位置に戻さずに回転するとカメラのケーブルを絡ませる可能性があるため必ず一度初期位置に戻る使用とした。サンダーモードでは、雷雲の上空の高層大気に発生する放電現象である「スプライト」を自動観測することを念頭においている。雷雲の位置は雷電波検知器により測定できる。また雷雲は通常同じ方角にしばらく存在することから、ここでは 10 分毎に指令がくることを想定した。サンダー自動モードでのフローチャートを図 3-5 に示す。サンダー自動モードで連携する送信ソフトの開発については「雷検知器を用いた可動式スプライト撮影システムの基礎開発」(湊 洋輔 高知工科大学 2005 年度 卒業研究報告)を参照されたい。

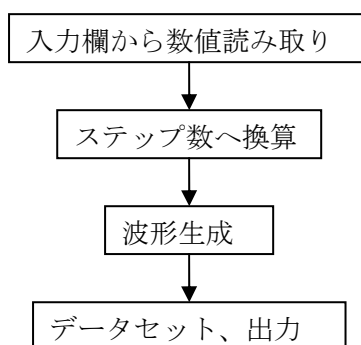


図 3-3 マニュアルモードのフローチャート

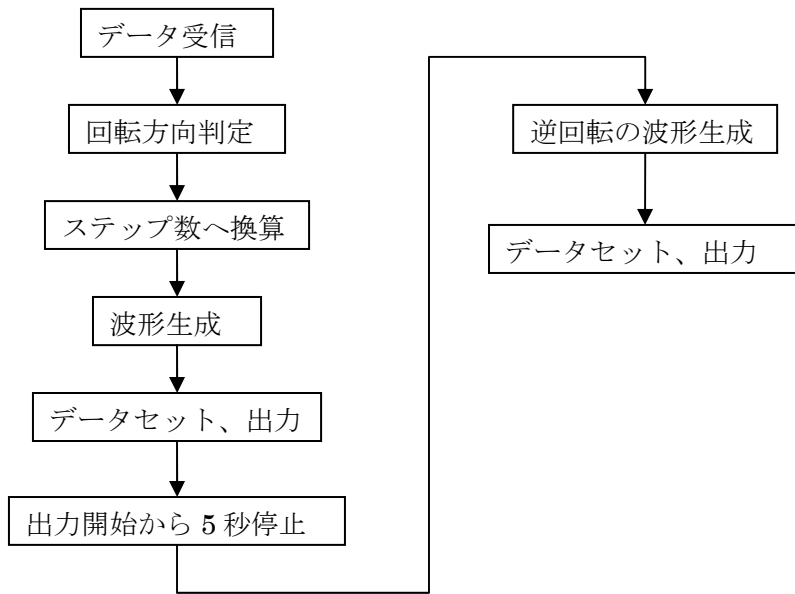


図 3-4 自動流星モードのフローチャート

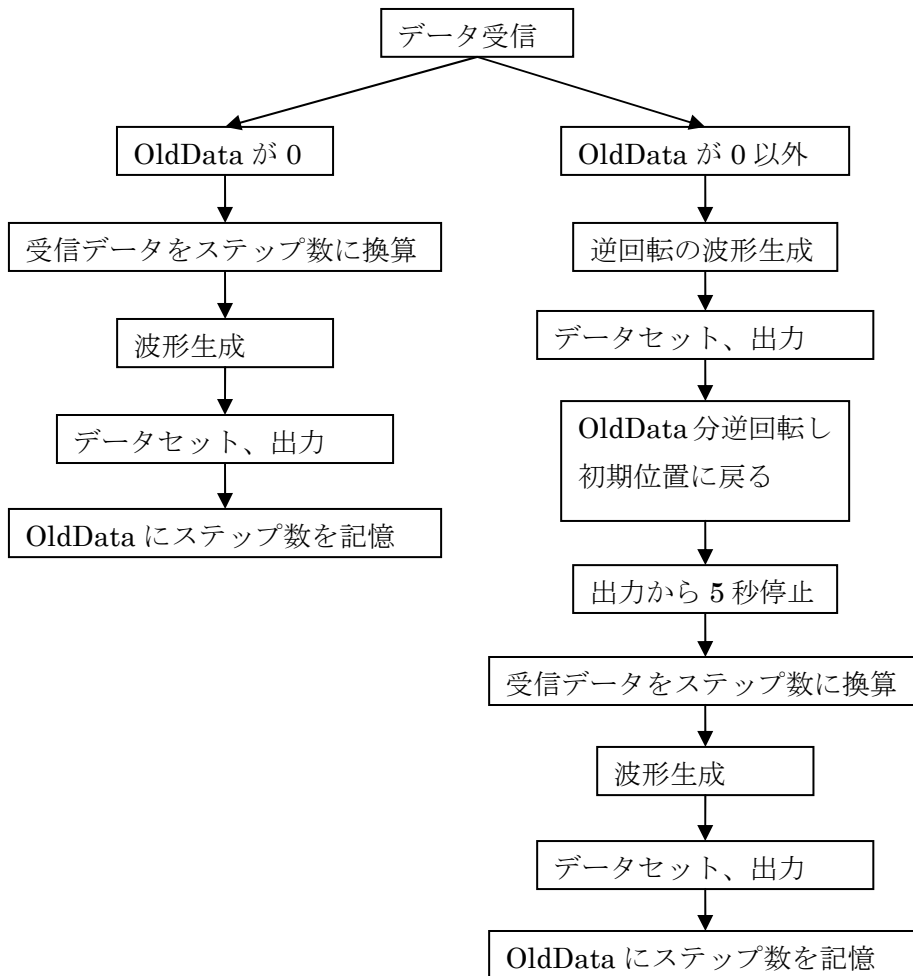


図 3-5 サンダー自動モードのフローチャート

### 3-3.出力波形のデータに関して

ステッピングモータ制御パルスとして 4ch 同時に出力している波形のデータ構造は一つの配列に入っている。1 件目の 1 チャンネル、1 件目の 2 チャンネル、1 件目の 3 チャンネル、1 件目の 4 チャンネル、2 件目の 1 チャンネル、2 件目の 2 チャンネル…の順にあらかじめ用意された配列として格納しておく。格納するデータ配列の様子を表 3-3 に示す。

表 3-3 DA 出力用データ構造

1ch 1 件目のデータ	2ch 1 件目のデータ	3ch 1 件目のデータ	4ch 1 件目のデータ	1ch 2 件目のデータ	2ch 2 件目のデータ	3ch 2 件目のデータ	4ch 2 件目のデータ	1ch 3 件目のデータ	2ch 3 件目のデータ	3ch 3 件目のデータ	4ch 3 件目のデータ
↓				↓				↓			
1 件目のデータ				2 件目のデータ				3 件目のデータ			

## 4 章 評価

### 4-1.評価方法

開発した装置の動作確認としてははじめにマニュアルモードによる動作で、正転、逆転とも正常に動作することが確認できた。

榮田氏の製作した方位角・仰角送信ソフト「素敵デバッグ」を使用してデータを送信し、受信状況を画面上で確認しモータの動作が正常であることを確認した。動作確認試験の様子を図 4-1 に示す。

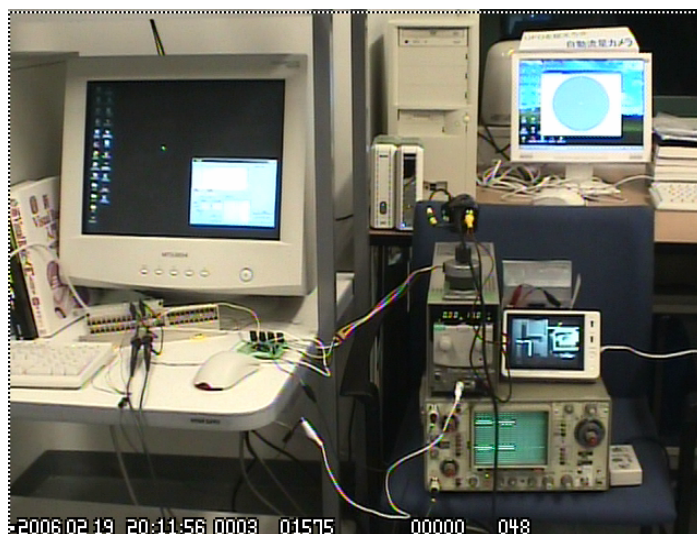


図 4-1 動作の様子を撮影したもの

### 4-2.問題点

動作中にモータが意図しない動きをすることが時折認められた。これはカメラ後ろ側に接続されたケーブルが原因であることが判明した。実験時にはケーブルが引っかかりモータの負荷となることがあったため、設置に当たっては配線の処理に気をつける必要がある。

モータは所定の動作をしたが一番遠い位置の 180 度まで動かせると約 1 秒かかる。なお、当初は 0~359° を正転のみで対応したが、181° ~359° は反転させる改良を行った。流星の発光継続時間は概ね数秒間以上である。より高速に動作させれば流星の出現直後からの映像を撮影することができる。現在使用しているステッピングモータの仕様ではこれ以上の速度は出せないため、高速流星への対応としてはより高速な回転が可能なモータに変える必要がある。

また、現在のプログラムの仕様上ステッピングモータを 4 ステップ単位つまり 7.2 度刻みでしか制御できていない。この問題を解決するためには現在のプログラムの波形生成部分をほぼすべて書き換えなければならない。

## 5章 まとめ

流星自動追跡装置の基礎技術となる DA ボードを用いたステッピングモータ制御のプログラムが開発できた。また VisualBasic を学習し DA ボード対応のソフトウェア開発の基礎を習得できた。今回は方位角方向の回転のみではあったが、シリアル通信を介した指令に対して動作する自動プログラムが構築された。2 軸制御化やより高度なサーボモータの制御に関してもパルス幅を変えれば本技術の延長で実現できる。本ソフトウェアでは流星自動モードだけでなく雷やスプライトの観測に応じたサンダー自動モードも合わせて実装し他ソフトとの組み合わせによるシステム化に道を開く要素技術である。

## 謝辞

本研究、論文作成を行うにあたりご指導ご鞭撻いただいた高知工科大学 電子・光システム工学科 山本 真行講師に厚くお礼申し上げます。

同研究室で一緒に学んだ、大岡 裕志さん、川崎 肇さん、榮田 明日香さん、末信 和子さん、濱口 美子さん、湊 洋輔さん、森岡 信繁さん 1年間本当にありがとうございました。本プログラムソースコードの記述に関して、Interface 社の WEB サイトを通じて助言をいただきました。ありがとうございました。

## 参考文献

- ・チュートリアル Visual Basic による DA 入門書、インターフェイス社
- ・USER'S MANUAL PCI-3523A、インターフェイス社
- ・VisualBasicVer.6.0 初級プログラミング入門[下]、河西 朝雄 著、技術評論社
- ・自動計測システムのための VisualBasic6 入門、金藤 仁 著、技術評論社
- ・新 VisualBasic 入門 ビギナー編 Ver.6.0 対応版、林 晴比古 著、ソフトバンク パブリッシング
- ・新 VisualBasic 入門 シニア編 Ver.6.0 対応版、林 晴比古 著、ソフトバンク パブリッシング
- ・ブレーキ付 DC モーターの回転数制御システムの開発、津野田 雄三、平成 16 年度高知工科大学 卒業研究報告
- ・DC モーター過渡現象解析用高速自動測定システム開発、北地 敦明、平成 15 年度高知工科大学 卒業研究報告

## 付録

### プログラムソースコード

'通信関係

Option Explicit

Public Com, Ba, Db, Stb As Integer

Public Par As String

Public pH, pM, pS, pD, Wt As Integer

Dim Buffer As Variant

Dim D(1) As Integer

Dim Ib() As Byte

Dim C As Integer

'モータ関係

Dim nRet As Long                   '関数戻り値

Dim lpszName As String           'デバイス名

Dim hDeviceHandle As Long       'デバイスハンドル

Dim hChannel As Long            'チャンネル番号

Dim Config As DASMPREQ         'DASMPREQ 構造体

'モータ動作

Dim Ang1 As Integer

Dim Rep, RepOld, THRep, THRepOld As Integer

Dim Ms(1) As Integer

Dim Smpldata(0 To 799) As Integer

Dim temp As Long

Dim nCount As Integer

Dim wMax As Long

Dim wMid As Integer

Dim wMin As Integer

Dim V5 As Long

Dim Rev As Integer

'タイマ関係

Dim Wa As Long

'止める

Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

```
Private Sub Command2_Click()  
    If MSComm1.PortOpen = True Then MSComm1.PortOpen = False  
  
    '終了処理  
    nRet = DaClose(hDeviceHandle)  
  
End  
  
End Sub
```

```
Private Sub Form_Load()  
    Text1.Text = Date & "<" & Time & ">"
```

```
    Tim1.Enabled = True
```

```
    '初期設定
```

```
        MSComm1.CommPort = 1  
        MSComm1.Settings = "9600,n,8,1"  
        MSComm1.Handshaking = comNone  
        MSComm1.RTSEnable = False  
        MSComm1.RThreshold = 1  
        MSComm1.SThreshold = 1  
        MSComm1.InputMode = comInputModeBinary
```

```
        MSComm2.CommPort = 1  
        MSComm2.Settings = "9600,n,8,1"  
        MSComm2.Handshaking = comNone  
        MSComm2.RTSEnable = False  
        MSComm2.RThreshold = 1  
        MSComm2.SThreshold = 1  
        MSComm2.InputMode = comInputModeBinary
```

```
    'ボードの初期化処理
```

```

lpszName = "FBIDA1"
hDeviceHandle = DaOpen(lpszName)
If hDeviceHandle = -1 Then
MsgBox "デバイスの初期化に失敗しました"
End
End If
'バッファ内の出力データをクリアします
nRet = DaClearSamplingData(hDeviceHandle)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "バッファのクリアに失敗しました"
Exit Sub
End If
'アナログ出力設定情報読み出し
nRet = DaGetSamplingConfig(hDeviceHandle, Config)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "アナログ出力設定情報取得に失敗しました"
Exit Sub
End If

End Sub

```

```

Private Sub Tim1_Timer()
pH = Hour(Time)
pM = Minute(Time)
pS = Second(Time)

Label3.Caption = pH & "時" & pM & "分" & pS & "秒"
End Sub

```

```

Private Sub Command1_Click()
'開始停止ボタン
MSComm1.PortOpen = Not MSComm1.PortOpen
If MSComm1.PortOpen = True Then

```

```

        Command1.Caption = "Auto shooting star mode Stop"
    Else
        Command1.Caption = "Auto shooting star mode Run"
    End If

End Sub

Private Sub MSComm1_OnComm()

'受信とか
'自動 mode !

    Select Case MSComm1.CommEvent
        Case comEvCD
        Case comEvCTS
        Case comEvDSR
        Case comEvRing
        Case comEvReceive

            Ib = MSComm1.Input
            Text1.Text = "<" & Time & ">" & "あいて" & "(" & Ib(0) & "," & Ib(1) &
" )" & Chr(&HD) & Chr(&HA) & Text1

            C = Ib(0) Mod 2
            If C = 1 Then
                D(1) = Ib(1) + 256
                D(0) = (Ib(0) - 1) / 2
            Else
                D(1) = Ib(1)
                D(0) = Ib(0) / 2
            End If
            Text3.Text = D(0)
            Text4.Text = D(1)
            Text5.Text = "<!((" & Time & ")>キタ! (" & D(0) & "," & D(1) & ") " &
Chr(&HD) & Chr(&HA) & Text5

```

Case comEvSend

Case comEvEOF

'エラーダイアログ

Case comEventBreak

MsgBox "中断信号受信", vbExclamation, "通信エラー"

Case comEventFrame

MsgBox "フレームエラー", vbExclamation, "通信エラー"

Case comEventOverrun

MsgBox "オーバーランエラー", vbExclamation, "通信エラー"

Case comEventRxOver

MsgBox "受信バッファオーバー", vbExclamation, "通信エラー"

Case comEventRxParity

MsgBox "受信パリティエラー", vbExclamation, "通信エラー"

Case comEventTxFull

MsgBox "送信バッファフル", vbExclamation, "通信エラー"

Case comEventDCB

MsgBox "DCB 取得エラー", vbExclamation, "通信エラー"

Case Else

MsgBox "未定義エラー発生", vbExclamation, "通信エラー"

End Select

'自動 modeVer.

'無理やり停止(Err 無視

nRet = DaStopSampling(hDeviceHandle)

nRet = DaClearSamplingData(hDeviceHandle)

'計算エリア

Ms(0) = D(0)

Ms(1) = D(1)

Rep = Ms(1) ¥ 7.2

If Rep = 0 Then

Exit Sub

End If

\*\*\* 出力データを形成します \*\*\*

wMax = &HFFF

wMin = &H0

,

wMid = (wMax + wMin) \ 2

V5 = (wMax - wMid) \ 2 + wMid

'回転方向判定

If Ms(1) <= 180 Then

For nCount = 0 To 799 Step 4

If 0 <= nCount And nCount < 100 Then

SmplData(nCount) = V5

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = V5

End If

If 100 <= nCount And nCount < 200 Then

SmplData(nCount) = wMid

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = V5

End If

If 200 <= nCount And nCount < 300 Then

SmplData(nCount) = wMid

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = V5

SmplData(nCount + 3) = V5

End If

```
If 300 <= nCount And nCount < 400 Then
  SmplData(nCount) = wMid
  SmplData(nCount + 1) = wMid
  SmplData(nCount + 2) = V5
  SmplData(nCount + 3) = wMid
End If
```

```
If 400 <= nCount And nCount < 500 Then
  SmplData(nCount) = wMid
  SmplData(nCount + 1) = V5
  SmplData(nCount + 2) = V5
  SmplData(nCount + 3) = wMid
End If
```

```
If 500 <= nCount And nCount < 600 Then
  SmplData(nCount) = wMid
  SmplData(nCount + 1) = V5
  SmplData(nCount + 2) = wMid
  SmplData(nCount + 3) = wMid
End If
```

```
If 600 <= nCount And nCount < 700 Then
  SmplData(nCount) = V5
  SmplData(nCount + 1) = V5
  SmplData(nCount + 2) = wMid
  SmplData(nCount + 3) = wMid
End If
```

```
If 700 <= nCount And nCount < 800 Then
  SmplData(nCount) = V5
  SmplData(nCount + 1) = wMid
  SmplData(nCount + 2) = wMid
  SmplData(nCount + 3) = wMid
End If
```

```
If 792 <= nCount Then
  SmplData(nCount) = wMid
  SmplData(nCount + 1) = wMid
  SmplData(nCount + 2) = wMid
  SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
Else
```

```
'180 以上
```

```
Rev = 180 - (Ms(1) - 180)
```

```
Rep = Rev ¥ 7.2
```

```
For nCount = 0 To 799 Step 4
```

```
If 0 <= nCount And nCount < 100 Then
```

```
  SmplData(nCount) = V5
```

```
  SmplData(nCount + 1) = wMid
```

```
  SmplData(nCount + 2) = wMid
```

```
  SmplData(nCount + 3) = V5
```

```
End If
```

```
If 100 <= nCount And nCount < 200 Then
```

```
  SmplData(nCount) = V5
```

```
  SmplData(nCount + 1) = wMid
```

```
  SmplData(nCount + 2) = wMid
```

```
  SmplData(nCount + 3) = wMid
```

```
End If
```

```
If 200 <= nCount And nCount < 300 Then
```

```
  SmplData(nCount) = V5
```

```
  SmplData(nCount + 1) = V5
```

```
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 300 <= nCount And nCount < 400 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 400 <= nCount And nCount < 500 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 500 <= nCount And nCount < 600 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 600 <= nCount And nCount < 700 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = V5  
End If
```

```
If 700 <= nCount And nCount < 800 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid
```

```
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 792 <= nCount Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
End If
```

```
'ボードの設定
```

```
nRet = DaSetBoardConfig(hDeviceHandle, 800, 0, 0, 0)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "ボードの設定に失敗しました"
```

```
Exit Sub
```

```
End If
```

```
' 出力設定
```

```
Config.ulSmplRepeat = Rep
```

```
Config.fSmplFreq = 5000
```

```
Config.ulChCount = 4
```

```
Config.SmplChReq(0).ulChNo = 1
```

```
Config.SmplChReq(0).ulRange = DA_10V
```

```
Config.SmplChReq(1).ulChNo = 2
```

```
Config.SmplChReq(1).ulRange = DA_10V
```

```
Config.SmplChReq(2).ulChNo = 3
```

```

Config.SmplChReq(2).ulRange = DA_10V
Config.SmplChReq(3).ulChNo = 4
Config.SmplChReq(3).ulRange = DA_10V

' アナログ出力更新設定
nRet = DaSetSamplingConfig(hDeviceHandle, Config)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "アナログ出力更新設定に失敗しました"
Exit Sub
End If
'バッファ内の出力データをクリアします
nRet = DaClearSamplingData(hDeviceHandle)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "バッファのクリアに失敗しました"
Exit Sub
End If
'出力するデータをバッファにセットします
nRet = DaSetSamplingData(hDeviceHandle, SmplData(0), 200)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "出力データの登録に失敗しました"
Exit Sub
End If

'For nCount = 0 To 799
'   Text9.Text = Text1 & Chr(&HD) & Chr(&HA) & SmplData(nCount)
'Next nCount

'アナログ出力を開始します
nRet = DaStartSampling(hDeviceHandle, FLAG_ASYNC)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "アナログ出力に失敗しました"
End If

```

'待ち

'10s 時を止めるコマンド

'Text5.Text = "元の位置に戻る" & Text5

Call Sleep(5000)

'回転方向判定

If Ms(1) <= 180 Then

'↓ 逆回転の波形

For nCount = 0 To 799 Step 4

If 0 <= nCount And nCount < 100 Then

SmplData(nCount) = V5

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = V5

End If

If 100 <= nCount And nCount < 200 Then

SmplData(nCount) = V5

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = wMid

End If

If 200 <= nCount And nCount < 300 Then

SmplData(nCount) = V5

SmplData(nCount + 1) = V5

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = wMid

End If

If 300 <= nCount And nCount < 400 Then

SmplData(nCount) = wMid

```
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 400 <= nCount And nCount < 500 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 500 <= nCount And nCount < 600 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 600 <= nCount And nCount < 700 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = V5  
End If
```

```
If 700 <= nCount And nCount < 800 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = V5  
End If
```

```
If 792 <= nCount Then
```

```
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
Else
```

```
'180 以上
```

```
For nCount = 0 To 799 Step 4
If 0 <= nCount And nCount < 100 Then
SmplData(nCount) = V5
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 100 <= nCount And nCount < 200 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 200 <= nCount And nCount < 300 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = V5
SmplData(nCount + 3) = V5
End If
```

```
If 300 <= nCount And nCount < 400 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
```

```
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 400 <= nCount And nCount < 500 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 500 <= nCount And nCount < 600 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 600 <= nCount And nCount < 700 Then  
SmplData(nCount) = V5  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 700 <= nCount And nCount < 800 Then  
SmplData(nCount) = V5  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 792 <= nCount Then  
SmplData(nCount) = wMid
```

```
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
End If
```

```
'バッファ内の出力データをクリアします
```

```
nRet = DaClearSamplingData(hDeviceHandle)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "バッファのクリアに失敗しました"
```

```
Exit Sub
```

```
End If
```

```
'出力するデータをバッファにセットします
```

```
nRet = DaSetSamplingData(hDeviceHandle, SmplData(0), 200)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "出力データの登録に失敗しました"
```

```
Exit Sub
```

```
End If
```

```
'アナログ出力を開始します
```

```
nRet = DaStartSampling(hDeviceHandle, FLAG_ASYNC)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "アナログ出力に失敗しました"
```

```
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
'回すボタン
```

\*\*\* 出力データを形成します \*\*\*

wMax = &HFFF

wMin = &H0

wMid = (wMax + wMin) ¥ 2

V5 = (wMax - wMid) ¥ 2 + wMid

'無理やり停止(Err 無視

nRet = DaStopSampling(hDeviceHandle)

'角度から

Ang1 = Text2

If Ang1 >= 360 Then

    Ang1 = Ang1 Mod 360

End If

Rep = Ang1 ¥ 7.2

Text6 = Rep \* 4

If Rep = 0 Then

    Exit Sub

End If

If Option1 = True Then

    For nCount = 0 To 799 Step 4

        If 0 <= nCount And nCount < 100 Then

            SmplData(nCount) = V5

            SmplData(nCount + 1) = wMid

            SmplData(nCount + 2) = wMid

            SmplData(nCount + 3) = V5

        End If

        If 100 <= nCount And nCount < 200 Then

            SmplData(nCount) = wMid

```
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = V5  
End If
```

```
If 200 <= nCount And nCount < 300 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = V5  
End If
```

```
If 300 <= nCount And nCount < 400 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 400 <= nCount And nCount < 500 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 500 <= nCount And nCount < 600 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 600 <= nCount And nCount < 700 Then  
SmplData(nCount) = V5  
SmplData(nCount + 1) = V5
```

```
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
If 700 <= nCount And nCount < 800 Then
SmplData(nCount) = V5
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
If 792 <= nCount Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
Text5.Text = "MA(R)<!( " & Time & ")>( " & Text7 & ", " & Text2 & ") " & Chr(&HD) &
Chr(&HA) & Text5
```

```
Else
```

```
For nCount = 0 To 799 Step 4
If 0 <= nCount And nCount < 100 Then
SmplData(nCount) = V5
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 100 <= nCount And nCount < 200 Then
SmplData(nCount) = V5
```

```
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 200 <= nCount And nCount < 300 Then  
SmplData(nCount) = V5  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 300 <= nCount And nCount < 400 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 400 <= nCount And nCount < 500 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 500 <= nCount And nCount < 600 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 600 <= nCount And nCount < 700 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid
```

```
SmplData(nCount + 2) = V5
SmplData(nCount + 3) = V5
End If
```

```
If 700 <= nCount And nCount < 800 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 792 <= nCount Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
Text5.Text = "MA(L)<!((" & Time & ")>(" & Text7 & "," & Text2 & ")" & Chr(&HD) &
Chr(&HA) & Text5
```

```
End If
```

```
'Text9.Text = wMax & Chr(&HD) & Chr(&HA) & wMid & Chr(&HD) & Chr(&HA) &
wMin
```

```
'ボードの設定
```

```
nRet = DaSetBoardConfig(hDeviceHandle, 800, 0, 0, 0)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "ボードの設定に失敗しました"
```

```

Exit Sub
End If
' 出力設定
Config.ulSmplRepeat = Rep
Config.fSmplFreq = 5000
Config.ulChCount = 4
Config.SmplChReq(0).ulChNo = 1
Config.SmplChReq(0).ulRange = DA_10V
Config.SmplChReq(1).ulChNo = 2
Config.SmplChReq(1).ulRange = DA_10V
Config.SmplChReq(2).ulChNo = 3
Config.SmplChReq(2).ulRange = DA_10V
Config.SmplChReq(3).ulChNo = 4
Config.SmplChReq(3).ulRange = DA_10V

' アナログ出力更新設定
nRet = DaSetSamplingConfig(hDeviceHandle, Config)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "アナログ出力更新設定に失敗しました"
Exit Sub
End If
'バッファ内の出力データをクリアします
nRet = DaClearSamplingData(hDeviceHandle)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "バッファのクリアに失敗しました"
Exit Sub
End If
'出力するデータをバッファにセットします
nRet = DaSetSamplingData(hDeviceHandle, SmplData(0), 200)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "出力データの登録に失敗しました"
Exit Sub
End If

'For nCount = 0 To 799

```

```
'    Text9.Text = Text1 & Chr(&HD) & Chr(&HA) & SmplData(nCount)
'Next nCount
```

```
'アナログ出力を開始します
```

```
nRet = DaStartSampling(hDeviceHandle, FLAG_ASYNC)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "アナログ出力に失敗しました"
```

```
End If
```

```
End Sub
```

```
Private Sub TheWorld_Click()
```

```
Call Sleep(10000)
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
'強制停止
```

```
'アナログ出力停止
```

```
nRet = DaStopSampling(hDeviceHandle)
```

```
'バッファ内の出力データをクリアします
```

```
nRet = DaClearSamplingData(hDeviceHandle)
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
'開始停止ボタン
```

```
    MSComm2.PortOpen = Not MSComm2.PortOpen
```

```
    If MSComm2.PortOpen = True Then
```

```
        Command6.Caption = "AutoThunder mode Stop"
```

```
    Else
```

```
        Command6.Caption = "AutoThunder mode Run"
```

```
    End If
```

```
End Sub
```

```
Private Sub MSComm2_OnComm()
```

```
'受信
```

```
'TH Auto mode !
```

```
    Select Case MSComm2.CommEvent
```

```
        Case comEvCD
```

```
        Case comEvCTS
```

```
        Case comEvDSR
```

```
        Case comEvRing
```

```
        Case comEvReceive
```

```
            Ib = MSComm2.Input
```

```
            Text1.Text = "TH<" & Time & ">" & "あいて" & "(" & Ib(0) & "," & Ib(1) & ")" & Chr(&HD) & Chr(&HA) & Text1
```

```
            C = Ib(0) Mod 2
```

```
                If C = 1 Then
```

```
                    D(1) = Ib(1) + 256
```

```
                    D(0) = (Ib(0) - 1) / 2
```

```
                Else
```

```
                    D(1) = Ib(1)
```

```
                    D(0) = Ib(0) / 2
```

```
                End If
```

```
            Text3.Text = D(0)
```

```
            Text4.Text = D(1)
```

```
            Text5.Text = "TH<!((" & Time & ")>キタ! (" & D(0) & "," & D(1) & ")" & Chr(&HD) & Chr(&HA) & Text5
```

```
        Case comEvSend
```

```
        Case comEvEOF
```

```
'エラーダイアログ
```

```
    Case comEventBreak
```

```
        MsgBox "中断信号受信", vbExclamation, "通信エラー"
```

```
    Case comEventFrame
```

```

        MsgBox "フレームエラー", vbExclamation, "通信エラー"
    Case comEventOverrun
        MsgBox "オーバーランエラー", vbExclamation, "通信エラー"
    Case comEventRxOver
        MsgBox "受信バッファオーバー", vbExclamation, "通信エラー"
    Case comEventRxParity
        MsgBox "受信パリティエラー", vbExclamation, "通信エラー"
    Case comEventTxFull
        MsgBox "送信バッファフル", vbExclamation, "通信エラー"
    Case comEventDCB
        MsgBox "DCB 取得エラー", vbExclamation, "通信エラー"
    Case Else
        MsgBox "未定義エラー発生", vbExclamation, "通信エラー"

End Select

```

'モータ回転のプログラム自動 modeVer.

'無理やり停止(Err 無視

nRet = DaStopSampling(hDeviceHandle)

nRet = DaClearSamplingData(hDeviceHandle)

'計算エリア

Ms(0) = D(0)

Ms(1) = D(1)

THRep = Ms(1) ¥ 7.2

'THRepOld = 25

\*\*\* 出力データを形成します \*\*\*

wMax = &HFFF

wMin = &H0

wMid = (wMax + wMin) ¥ 2

V5 = (wMax - wMid) ¥ 2 + wMid

If THRepOld = 0 Then

GoTo FA

End If

For nCount = 0 To 799 Step 4

If 0 <= nCount And nCount < 100 Then

SmplData(nCount) = V5

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = V5

End If

If 100 <= nCount And nCount < 200 Then

SmplData(nCount) = wMid

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = wMid

SmplData(nCount + 3) = V5

End If

If 200 <= nCount And nCount < 300 Then

SmplData(nCount) = wMid

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = V5

SmplData(nCount + 3) = V5

End If

If 300 <= nCount And nCount < 400 Then

SmplData(nCount) = wMid

SmplData(nCount + 1) = wMid

SmplData(nCount + 2) = V5

SmplData(nCount + 3) = wMid

End If

If 400 <= nCount And nCount < 500 Then

```
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = V5  
SmplData(nCount + 3) = wMid  
End If
```

```
If 500 <= nCount And nCount < 600 Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 600 <= nCount And nCount < 700 Then  
SmplData(nCount) = V5  
SmplData(nCount + 1) = V5  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 700 <= nCount And nCount < 800 Then  
SmplData(nCount) = V5  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

```
If 792 <= nCount Then  
SmplData(nCount) = wMid  
SmplData(nCount + 1) = wMid  
SmplData(nCount + 2) = wMid  
SmplData(nCount + 3) = wMid  
End If
```

Next nCount

'ボードの設定

nRet = DaSetBoardConfig(hDeviceHandle, 800, 0, 0, 0)

If nRet <> DA\_ERROR\_SUCCESS Then

MsgBox "ボードの設定に失敗しました"

Exit Sub

End If

' 出力設定

Config.ulSmplRepeat = THRepOld

Config.fSmplFreq = 5000

Config.ulChCount = 4

Config.SmplChReq(0).ulChNo = 1

Config.SmplChReq(0).ulRange = DA\_10V

Config.SmplChReq(1).ulChNo = 2

Config.SmplChReq(1).ulRange = DA\_10V

Config.SmplChReq(2).ulChNo = 3

Config.SmplChReq(2).ulRange = DA\_10V

Config.SmplChReq(3).ulChNo = 4

Config.SmplChReq(3).ulRange = DA\_10V

' アナログ出力更新設定

nRet = DaSetSamplingConfig(hDeviceHandle, Config)

If nRet <> DA\_ERROR\_SUCCESS Then

MsgBox "アナログ出力更新設定に失敗しました"

Exit Sub

End If

'バッファ内の出力データをクリアします

nRet = DaClearSamplingData(hDeviceHandle)

If nRet <> DA\_ERROR\_SUCCESS Then

MsgBox "バッファのクリアに失敗しました"

Exit Sub

End If

```
'出力するデータをバッファにセットします
nRet = DaSetSamplingData(hDeviceHandle, SmplData(0), 200)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "出力データの登録に失敗しました"
Exit Sub
End If
```

```
'アナログ出力を開始します
nRet = DaStartSampling(hDeviceHandle, FLAG_ASYNC)
If nRet <> DA_ERROR_SUCCESS Then
MsgBox "アナログ出力に失敗しました"
End If
```

```
'待ち
Call Sleep(4000)
```

FA:

```
THRepOld = THRep
```

```
If THRep = 0 Then
Exit Sub
End If
```

```
'↓ 逆回転の波形
For nCount = 0 To 799 Step 4
If 0 <= nCount And nCount < 100 Then
SmplData(nCount) = V5
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 100 <= nCount And nCount < 200 Then
SmplData(nCount) = V5
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
If 200 <= nCount And nCount < 300 Then
SmplData(nCount) = V5
SmplData(nCount + 1) = V5
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
If 300 <= nCount And nCount < 400 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = V5
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
If 400 <= nCount And nCount < 500 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = V5
SmplData(nCount + 2) = V5
SmplData(nCount + 3) = wMid
End If
```

```
If 500 <= nCount And nCount < 600 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = V5
SmplData(nCount + 3) = wMid
End If
```

```
If 600 <= nCount And nCount < 700 Then
```

```
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = V5
SmplData(nCount + 3) = V5
End If
```

```
If 700 <= nCount And nCount < 800 Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = V5
End If
```

```
If 792 <= nCount Then
SmplData(nCount) = wMid
SmplData(nCount + 1) = wMid
SmplData(nCount + 2) = wMid
SmplData(nCount + 3) = wMid
End If
```

```
Next nCount
```

```
'ボードの設定
```

```
nRet = DaSetBoardConfig(hDeviceHandle, 800, 0, 0, 0)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "ボードの設定に失敗しました"
```

```
Exit Sub
```

```
End If
```

```
' 出力設定
```

```
Config.ulSmplRepeat = THRep
```

```
Config.fSmplFreq = 5000
```

```
Config.ulChCount = 4
```

```
Config.SmplChReq(0).ulChNo = 1
```

```
Config.SmplChReq(0).ulRange = DA_10V
```

```
Config.SmplChReq(1).ulChNo = 2
```

```
Config.SmplChReq(1).ulRange = DA_10V
Config.SmplChReq(2).ulChNo = 3
Config.SmplChReq(2).ulRange = DA_10V
Config.SmplChReq(3).ulChNo = 4
Config.SmplChReq(3).ulRange = DA_10V
```

```
' アナログ出力更新設定
```

```
nRet = DaSetSamplingConfig(hDeviceHandle, Config)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "アナログ出力更新設定に失敗しました"
```

```
Exit Sub
```

```
End If
```

```
'バッファ内の出力データをクリアします
```

```
nRet = DaClearSamplingData(hDeviceHandle)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "バッファのクリアに失敗しました"
```

```
Exit Sub
```

```
End If
```

```
'出力するデータをバッファにセットします
```

```
nRet = DaSetSamplingData(hDeviceHandle, SmplData(0), 200)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "出力データの登録に失敗しました"
```

```
Exit Sub
```

```
End If
```

```
'アナログ出力を開始します
```

```
nRet = DaStartSampling(hDeviceHandle, FLAG_ASYNC)
```

```
If nRet <> DA_ERROR_SUCCESS Then
```

```
MsgBox "アナログ出力に失敗しました"
```

```
End If
```

```
End Sub
```