

卒業研究報告

題目

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

指導教員

山本 真行 講師

報告者

学生番号： 1050275

氏名： 湊 洋輔

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 序	2
1. 1. 本研究の目的	
1. 2. スプライトについて	
1. 3. 高知工科大学におけるスプライト観測	
2章 雷検出器	5
2. 1. 雷発生方向を探す方法	
2. 2. 雷検知器「LD-250 Lightning Detector」及び 「StormTracker-PCI」	
2. 3. 雷検知情報多機能表示ソフト「NEXSTORM」	
3章 VisualBasicによる自動追尾プログラム製作	8
3. 1. プログラムの概要	
3. 2. 雷位置情報の入力と算出	
3. 3. MScommによる制御命令の送信	
3. 4. 動作確認	
4章 評価	11
4. 1. 定期的動作の確認	
4. 2. システム全体の評価	
5章 まとめ	14
参考文献	15
謝辞	15
付録 (ソースコード)	16

1 章 序

1. 1. 本研究の目的

高知工科大学 山本真行研究室では、2005年1月に研究室脇の屋外に固定カメラを設置して、一定の方向の空を撮影して流星やスプライトの自動観測を毎晩実施している。

スプライトは地球高層大気においてまれに発生する発光現象であるが、観測自体が困難なためこれまで得られている情報が少なく毎晩のように撮影できる対象ではない。近年の研究により、スプライトは世界中のいたる所で落雷に伴い発生している可能性が高いことが判明してきた。観測確率を上げるには雷雲上空にカメラを向ける必要があるが、固定カメラによる一定方向の自動観測では必ずしも雷発生位置の上空を向いておらず観測時間が「もったいない」ことを意味している。本研究では、既設の高感度カメラを雷検知器と連動したシステムとして、雷が頻繁に発生している方角に360°旋回可能なカメラに改良することによりスプライトの検出確率を上げることを目的とする。

1. 2. スプライトについて

スプライト (sprite: 妖精の意) と呼ばれる雷雲上空での発光現象は、15年ほど前に偶然発見された。これまでの観測により、スプライトは高度40–90 kmで瞬間的に発光し、水平方向には10–50 kmの空間的広がりをもっていることが明らかになってきている。

雷放電に伴いスプライトが発生するメカニズムについては研究途上ではあるが、雷放電の直後、雷雲と電離層間には非常に強い下向きの電場が発生し、この電場の大きさが高度40–90 kmでの絶縁破壊電場を越えると、絶縁破壊に至り、電子との衝突によってエネルギーを得た大気分子が励起され発光すると考えられている。この際、数ms~数10msのごく短い時間発せられる光がスプライトとして観測される。スプライトの色は主に赤色が強く、高度が低いほど青色が強くなるなどスプライトの素性は最近の研究により徐々に明らかになってきている。

しかし、スプライト研究の歴史は浅く、さらに関連する未知の現象がまだ多く存在している可能性が高いと考えられており、自動観測においてはその検出確率を上げる工夫が重要である。

- 「絶縁破壊」とは、大空気中にある値以上の電圧を加えれば、絶縁状態が破壊されて瞬間的に導体として作用し電流が流れることである。通常の落雷の場合にも同様のメカニズムが働くと考えられている。

1. 3. 高知工科大学におけるスプライト観測

高知工科大学 電子・光システム工学科（高知県香美郡土佐山田町： $33^{\circ}.72\text{ N}$, $133^{\circ}.62\text{ E}$ ）では、2005年1月より東方向にカメラを向けてスプライトの観測を実施している。視野角が $30^{\circ} \times 22^{\circ}$ であるので、このカメラに写る範囲は全天のごく一部に限られている。仰角を低く設置しているため、遠くは約700km先の茨城県沖のスプライトまで撮影された。この結果、1年間の観測で得られたスプライトは12例であった。



図 1. 1

高知工科大学におけるスプライト検出画像の例（2005年10月7日 19:26頃）

図1・1にスプライト検出画像の例を示す。右上に移っている線状および複雑な形状の光がスプライトであり、その真下の山陰に光っている光が雷に照らされた雷雲と考えられる。動画から切り出した1/30秒の静止画でありビデオフレームの1フレーム(1/30秒)のみに検出されていたことから継続時間は33ms以内である。

撮影には、高知工科大学 研究 A 棟南側の2階屋根に設置された高感度 CCD カメラ (Watec 製 Neptune-100, CBC 製 8 mm f/0.8 レンズ) とタイムシフト動体監視ソフトウェア UFOCapture (SonotaCo 氏製作のシェアウェア、EX1.85 バージョン) が使用された。視野中心は方位角 82 度 (北から東まわり) で、スプライトは、東方向の仰角 20 度付近に現れた。

2 章 雷検出器

2. 1. 雷発生方向を探す方法

前章にて説明したように、スプライトを撮影するためにはまず雷がどの方向で発生しているか知る必要がある。落雷の際には広い周波数領域で電波が発生することが判っており、この問題の解決方法としては雷電波の利用が考えられる。本研究では、UFOCapture を使って雷やスプライトの観測を進めているグループ” SonotaCo Network “のホームページ (<http://sonotaco.sub.jp/etc/Lightning/Storm.html>) に紹介されている雷検知器を使って、雷の発生方向について情報を得ることとした。

2. 2. 雷検知器「LD-250 Lightning Detector」及び「StormTracker-PCI」

本実験で用いた米国 Boltek 社製の雷検知器は二種類あり、まず最新の「LD-250 Lightning Detector」という USB 接続による外付けの検知器を用いた。実際に Boltek 社に注文すると二週間ほどで届いた。

海外の製品なので説明書などは全て英語で書かれていたが、SonotaCo 氏のホームページ内の日本語の説明により補いつつ解決し問題なく使えるようになった。雷検知器を簡単に説明すると、アンテナと受信部からの信号をパソコンで解析して、発生した雷の方位と距離を 1 地点からの観測でリアルタイム判定するものだ。「LD-250 Lightning Detector」と「StormTracker-PCI」には大きな差はないが、本研究では PCI 版の方を使うことにした。PCI 版はパソコン (PCI バス) に装着するため、後でパソコン移動させたりするとき便利なことと、後述する雷検知情報多機能表示ソフトで使用すると方位、距離以外に雷の極性や種類 (落雷か雲放電か) まで判定できるためである。特にスプライトの観測では極性との相関を考慮することが大切である。

2. 3. 雷検知情報多機能表示ソフト「NEXSTORM」

雷検知情報多機能表示ソフト AstroGenic 社製 NEXSTORM (図 2. 1 に動作画面の例を示す) を使用すると雷電波の方位角、距離以外に雷の極性や種類 (地上への落雷か雲間放電か) までの判定結果が表示できる。そして、これらのデータが SonotaCo 氏製作のプログラム NSList (図 2. 2 に例を示す) によって雷の方位、距離のデータが CSV 形式のテキストファイルとして日付毎のファイルに出力され、イベント毎に更新される。NEXSTORM とデータ出力プログラム NSList によって、カメラを動かすために必要な方位角の情報を別のソフトから取り込むことができるようになる。これらのソフトの連携によって、スプライトの撮影に最も重要な雷雲の方位角の情報が入手できる。

なお、現在 SonotaCo Network では、全国 5 地点に設置された雷検知器の組み合わせをインターネット上で実現して、より正確な雷出現位置のリアルタイム表示実験が進行中である。

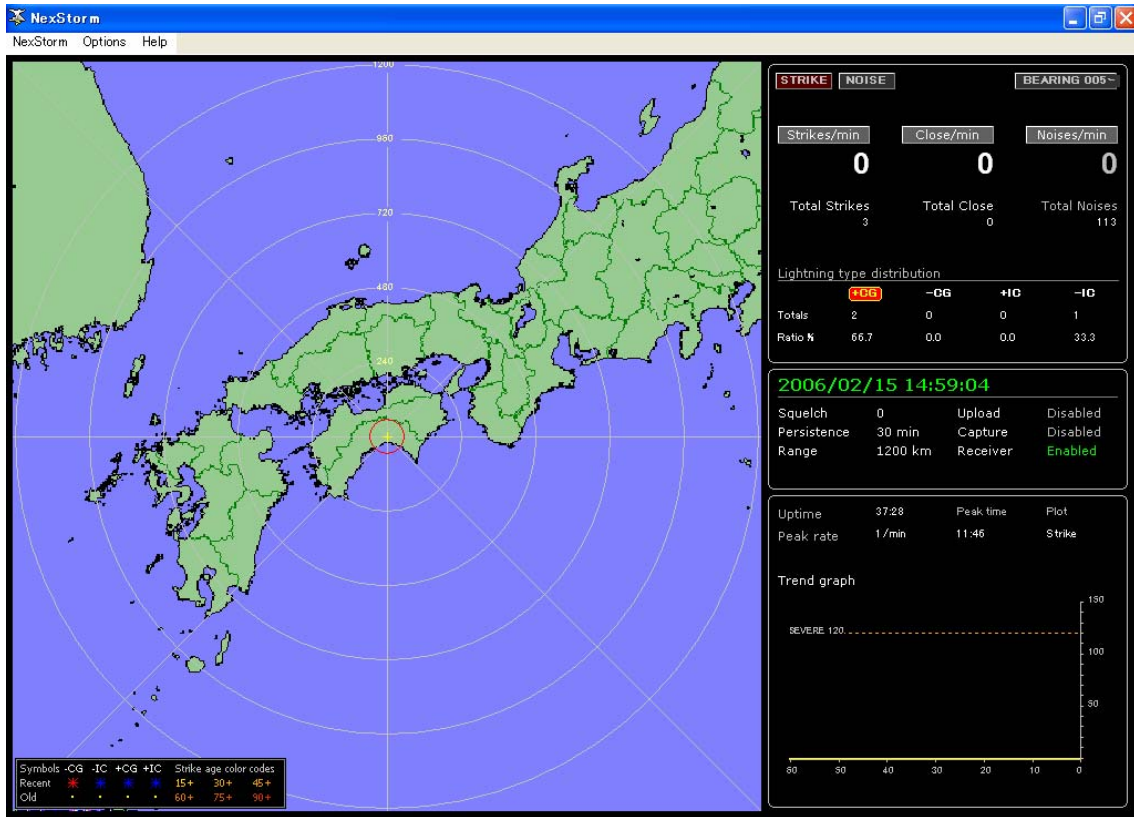


図 2. 1 NEXSTORM

The screenshot shows the NSList software interface, titled "NSList : NS IPC Listing Tool V1.4 by SonotaCo". It includes checkboxes for "Include Noise" (unchecked), "Log CSV file" (checked), and "Dist(mile)" (unchecked). The "Last heart beat" is 15 41 10. Below is a table of lightning strike data:

D	H	M	S	Dir(deg)	Dist(km)	Type
15	11	45	34	5.3	593	+CG
15	09	59	50	339.5	652	+CG
15	04	53	49	27.4	178	-IC
14	20	06	41	334.8	646	+CG
14	18	52	22	335.7	592	+CG
14	17	38	25	336.3	662	+CG
14	17	12	02	335.7	642	+CG
14	16	58	41	341.3	660	+CG
14	15	23	42	313.6	230	+IC
14	15	23	27	317.5	241	+IC
14	15	23	22	317.2	96	-IC
14	15	23	20	315.2	96	-IC

図 2. 2 NSList

3章 VisualBasic による自動追尾プログラム製作

3. 1、プログラムの概要

雷検知情報多機能表示ソフト NEXSTORM で取得したデータは、雷電波の検出日時及び方位角などの情報が含まれる。本研究ではこれらの情報の主に方位角を用いてカメラを動かすことにした。まず、雷検知器からの雷電波のデータはデータ出力プログラム NSList により、年月日のファイル名（例えば“L20060208.CSV”）の CSV 形式のテキストファイルとして次のよう出力される。

```
2006,02,08,08,44,46," -CG",11.1,701
2006,02,08,09,34,48," -CG",11.6,651
2006,02,08,15,30,08," +IC",302.2,349
2006,02,08,15,43,03," -IC",104.2,479
2006,02,08,17,15,50," +IC",287.3,370
2006,02,08,17,15,57," +CG",294.4,370
2006,02,08,18,37,13," -IC",221.7,116
2006,02,08,19,17,09," +IC",48.1,159
2006,02,08,19,33,48," +IC",295.8,409
2006,02,08,19,48,55," -CG",304.6,567
2006,02,08,23,00,46," -IC",287.2,119
```

すなわち、雷電波を検出するごとに、年、月、日、時、分、秒、雷の種類と極性、方位角（度）、距離（km）の値が出力される。ここでは＋は極性を、CG は雲から地上への落雷、IC は雲間放電による電波であることを表している。

次にこれらの数値をどのように自動追尾プログラムに生かすかを考える。雷は、常に同じ場所で発生するものとは限らないが比較的長時間、一定地域の上空に雷雲が存在する。最も発生頻度の高い方位角にカメラを動かすことが有効であると考えられる。ここでカメラの方位角は一定時間（ここでは10分）一定にすることに決めた。その後タイマーにて10分ごとに直前の1時間の平均値を計算しカメラ制御に用いる。カメラの仰角は一定角度（ここでは約10度）に固定することとした。これはスプライトが高度40－90kmで瞬間的に発光する現象であり、遠方の雷雲上空を晴れた場所から狙うことになるためである。ソフトを起動してから1時間に得られた方位角のデータを平均して値を決めカメラ制御のための情報として送信することにした。

3. 2、雷位置情報の入力と算出

雷の位置情報は前節で説明したようにテキストデータとして数字で得られるので1時間ごとに現れるデータを平均してその値を求める。

算出方法としては、2006,02,08,23,00,46,"-IC",287.2,119 の様に記録されているデータから VisualBasic のプログラムによって方位角のデータだけを取り出して List1 に保存し、そのデータを平均する。平均したデータは List2 に保存し、表示しそのデータを10分ごとに VisualBasic におけるシリアル通信命令である MSComm によって送信するというものである。

このプログラムを実行しているときに雷の情報が無い場合も考えられる。特に1日の開始時からしばらく雷が検出されない場合 CSV 形式テキストファイル自体が存在せずエラーが起きるといった不具合が生じたので、この問題を解決するために、プログラム自体で自動的に23:59になると次の日の仮のデータを作る仕様にした。これによりエラーは回避され、雷情報が無くてもカメラは直前のデータ位置から方向を変えずに撮影できるようになった。その後、雷検出により新たな情報がきたら NSList によりデータファイルが上書きされるため問題はない。

3. 3、MSComm による制御命令の送信

MSComm (MSComm : Microsoft Communication Control)とは、Visual Basic4.0 以降に内蔵された OCX 型のカスタムコントロールで、この OCX を使えば Visual Basic のフォームに MSComm コントロールを貼り付けるだけで簡単にシリアルポートへの送受信プログラムを作ることができる。

コミュニケーション コントロールは、標準的なシリアルポートを利用するためのインターフェイスで、(1)～(4)の機能が利用できる。

(1)シリアル ポートへの接続

16個の内の指定したポートへ MSComm を接続する。

(2)コマンドの出力

RTS、CTS などシリアルインターフェイスであり、として用意されている各種の制御信号の出力制御や、状態の入力を行うことができる。

(3)データの送受信

シリアル通信で行うデータの送受信を行う。

(4)状態変化や通信誤りの確認

各種の制御信号の状態の変化や、データ通信中にエラーが発生したときにイベントを発生させる割り込み方式や、状態を読み出して確認する(ポーリング方式)こともできる。

3. 4. 動作確認

本プログラムでは、CSV ファイルからのデータ自動入力と MSComm によるデータ送信が正しく実施され、10分ごとに平均した方位角データを RS-232C ケーブルを通じて外部に送信できることを確認した。

下の図 3. 1 に開発したプログラムの起動画面を示す。

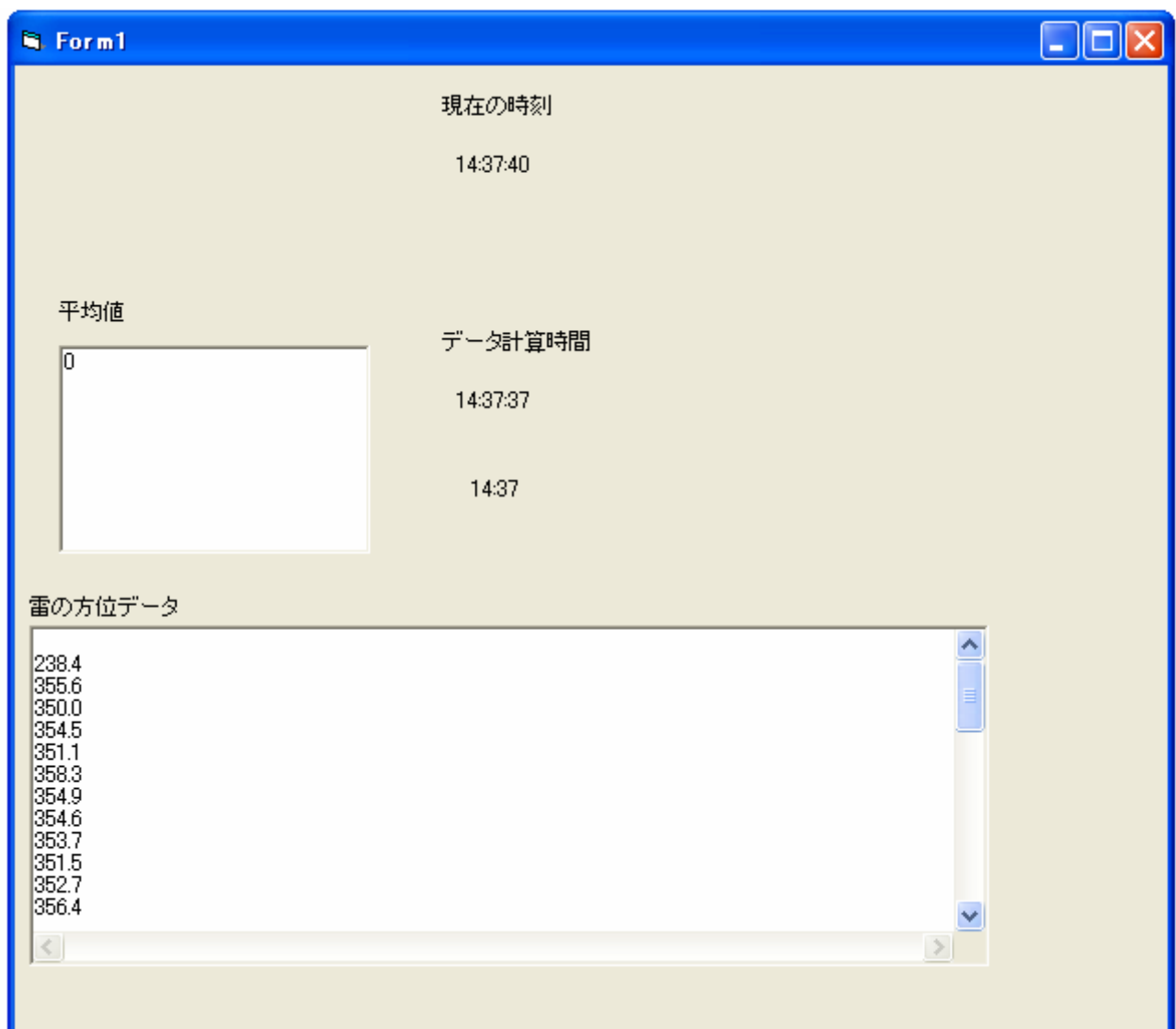


図 3. 1 起動画面

4 章 評価

4. 1、定期的動作の確認

前節でも説明したようにソフトを起動してから 1 時間毎に得られた方位角のデータを平均して値を決めカメラ制御のための情報として送信することに成功した。日付けの変わり目の動作についても問題なく動くことが確認できた。

方位角データの受信チェックには、山下泰樹氏の卒業研究（タイトル：DA ボードを用いた流星自動追跡装置の基礎開発）にて製作のプログラム上及び汎用通信ソフトであるハイパーターミナルにて確認した。

4. 2、システム全体の評価

下の図4. 1は本研究で扱ったシステム全体の構成図となっている。

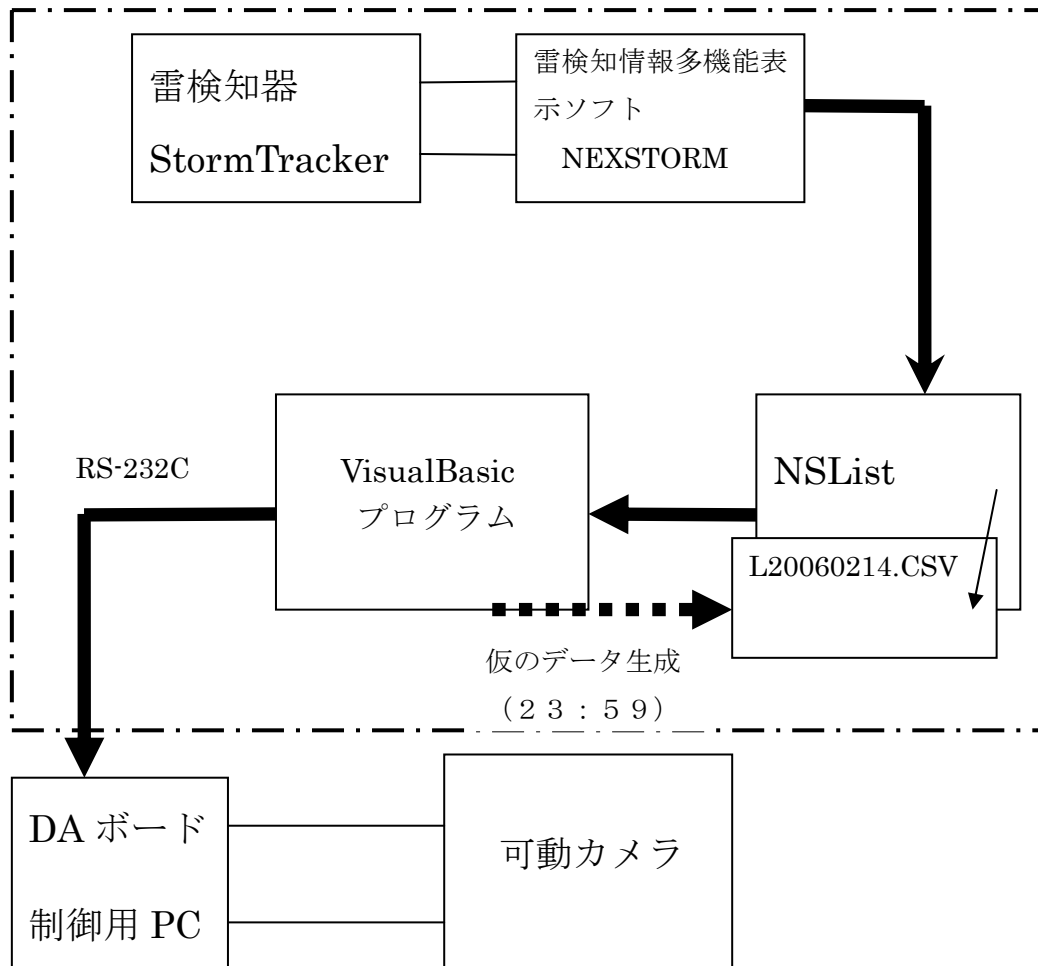


図4. 1 システム全体の構成図

今回開発したソフトウェアにより、

- ・ 雷検知器及び表示ソフトから出力される方位角のデータだけを取り出すこと
- ・ ソフト起動時からタイマーにて10分ごとに、直前の1時間の方位角データを用いた平均値を算出すること
- ・ 方位角平均値データを10分ごとに VisualBasic におけるシリアル通信命令である MSComm によってカメラ制御側に送信すること

の3項目が達成され、毎日24時間稼動できる可動式スプライト観測システムの基礎プログラムの開発に成功した。雷検知器の検出性能に左右されるものの、雷方向に一定時間ごとにカメラを向けるという当初の目的の基礎が築かれた。今後、研究室における様々な基礎開発成果の応用により複雑な観測システムが構築できる。

5章 まとめ

大学の勉強の中で一番苦手であり避けてきたプログラム製作が卒業研究での課題になったことは予想外であった。本研究目的である可動式スプライト撮影システムの基礎開発で一番重要な部分がソフトウェア開発だったからである。その内容としては、24時間連続動作する必要性があり仕様通り一定時間毎に直前雷検出方向の値を算出して自動的に指示を与えること、ならびに日付変更などの際の自動処理が必要で、そのデータを10分ごとに VisualBasic におけるシリアル通信命令である MSComm によってカメラ制御側に送信するというものであった。これらを自分で考えておきながら、今までプログラム自体を避けてきていたこともありとても嫌気がした。しかし実際に製作を行っていくうちに、解らないことはプログラムの記述方法だけであることが分かった。頭の中で何をしなければならぬということが解れば次はどういった処理が必要かという発想が生まれるようになり、解からないなりに時間をかけて完成させることができた。今まで自分自身を振り返ると、時間をかけてまで苦手なことを行う経験をしなかったが、時間をかけて何かを達成することの充実感や達成感を得ることができた。また本研究により、スプライトという発光現象を知ることもでき、さらに苦手であったプログラムを少しではあるが扱い方を知ることができ良い経験になったと思う。

参考文献

- UFOCaptureStormTracking
(<http://sonotaco.sub.jp/etc/Lightning/Storm.html>)
- SonotaCo Lightning and Sprite
(<http://hp.vector.co.jp/authors/VA034934/Lightning/index.html>)
- VisualBasic
(<http://homepage1.nifty.com/rucio/main/main.htm>)
- VisualBasic 逆引き大全 500の極意 株式会社 秀和システム
- VisualBasic6.0 300の技 技術評論社
- Visual Basic プログラミング 1001 Tips オーム社

謝辞

常に自分の思いどおりいかなければ不満ばかりを言っていた私を見捨てることなく力を貸してくれた、指導教員の山本真行講師には言葉にならないぐらい感謝しています。また、研究室の院生、4年生にも迷惑ばかりかけ本当に悪いことをしたと思っています。皆様の心の広さに感謝しています。本研究が無事に終わったことは、本当に山本真行研究室に所属する皆様、学外でいつもお世話になっている皆様の協力によるものです。

ありがとうございました。

付録 (ソースコード)

```
Private Sub Timer1_Timer()  
  
    '// 全体の定義 '//  
    Dim strFileName As String  
    Dim intFileNo As Integer  
    Dim strRecBuff As String  
    Dim lngLineCounter As Integer  
    Dim n As Integer  
  
    Dim Azimuth As String  
  
    Dim strArray() As String  
    Dim intCounter As Integer  
    Dim AzimuthArray(1000) As String  
  
    Dim a As Integer  
    Dim Average As Long  
    Dim AverageArray As Long  
  
    Dim c As Integer  
    c = 0  
    Dim Hour(1000) As String  
  
    Dim X As Integer
```

```
'// タイマーによるプログラム開始 ///
```

```
Dim year As String
```

```
Dim month As String
```

```
Dim Day As String
```

```
Dim Flag As Integer
```

```
Flag = 0
```

```
Dim Y As String
```

```
Dim strFilename2 As String
```

```
'Dim Scanfile As String
```

```
Dim gyou, houi As Double
```

```
Dim Ib0, Ib1 As Integer
```

```
Dim Ib(1) As Byte
```

```
Dim z As Integer
```

```

year = Mid(Date, 1, 4)
month = Mid(Date, 6, 2)
Day = Mid(Date, 9, 2)
Day = Format(Day, "00")

// 0時になったときは前日の日付けにする //
If Mid(Time, 1, 1) = "00" Then
Day = Day - 1
Day = Format(Day, "00")
Flag = 1
Else
Flag = 0
End If

strFileName = "C:\¥NSList¥" & "L" & year & month & Day & ".csv"
intFileNo = 1
// CSV ファイルの読み込み //
Open strFileName For Input As intFileNo

// ファイルの最初から最後まで読み込む //
Do Until EOF(intFileNo)
Line Input #intFileNo, strRecBuff

strArray = Split(strRecBuff, ",")
// 方位角と時間を取り出す //
AzimuthArray(n) = strArray(7)
Hour(n) = strArray(3)

n = n + 1

Loop

```

```
'// 読み込んだデータを並べて List1 に表示 '//  
For m = 0 To n - 1  
Azimuth = Azimuth & vbCrLf & AzimuthArray(m)  
Next  
  
List1.Text = Azimuth  
  
Close intFileNo  
  
'// 翌日に入った時の処理 '//  
Y = Mid(Time, 2, 1)  
If Y = ":" Then  
    Y = Mid(Time, 1, 1) - 1  
Else  
    Y = Mid(Time, 1, 2) - 1  
End If  
If Y = -1 Then  
    Y = 23  
End If  
Y = Format(Y, "00")
```

```

'// 23:59 がくると先に翌日の CSV ファイルをつくる '//
Label1.Caption = Left(Time, 5)

If Left(Time, 5) = "23:59" Then

    strFilename2 = "C:\¥NSList¥" & "L" & year & month & Day + 1 & ".csv"

    Set objFileSystem = CreateObject("Scripting.FileSystemObject")

    Set objFile = objFileSystem.CreateTextFile(strFilename2, False)

    objFile.Close

End If

For a = 0 To n - 1

If Hour(a) <> Y Then

    AzimuthArray(a) = 0

    c = c + 1

End If

```

Average = Average + AzimuthArray(a)

Next

X = n - c

If X = 0 Then

X = 1

End If

// 方位角のデータの平均処理と表示 //

List2.Text = Average / X

```
'// MSComm によるシリアル通信設定  //
```

```
MSComm1.CommPort = 1
```

```
MSComm1.Handshaking = MSCommLib.HandshakeConstants.comNone
```

```
MSComm1.RTSEnable = False
```

```
MSComm1.RThreshold = 1
```

```
MSComm1.SThreshold = 1
```

```
MSComm1.InputMode=MSCommLib.InputModeConstants.comInputModeBinary
```

```
MSComm1.PortOpen = True
```

```

gyou = 0

houi = Average / X

Ib0 = gyou
Ib1 = houi

    '// 2 byte に詰め込む '//
Ib(0) = gyou
z = houi

If z > 255 Then
    z = z - 256
    Ib(1) = z
    Ib(0) = Ib(0) * 2 + 1
Else
    Ib(1) = z
    Ib(0) = Ib(0) * 2
End If
 '// MSComm により方位角を送信 '//
 '// 2 byte を送信 '//
If MSComm1.PortOpen = True Then

    MSComm1.Output = Ib

    MSComm1.PortOpen = False

End If

Label2.Caption = Time

End Sub

```



```
Private Sub Form_Load()  
    '// タイマーインターバルの設定 (10分)  '//  
    Timer1.Interval = 600000  
  
    Timer2.Interval = 500  
    Enabled = True  
  
End Sub
```

```
Private Sub Timer2_Timer()  
    '// 現在時刻の表示  '//  
    Label3.Caption = Time  
  
End Sub
```