

交通部中央氣象局委託研究計畫成果報告

船舶海象觀測系統(二) Observation System of Maritime Meteorology(II)

計畫類別: ☐國外 ☒國內

計畫編號: CWB 87-2M-07

執行期間: 86 年 7 月 1 日至 87 年 6 月 30 日

計畫主持人: 李賢文

國立臺灣海洋大學海洋科學系

中華民國八十七年六月三十日

前言：

為了促進商船參與海洋氣象資料之收集，我們需要建立一套全自動的船舶海洋氣象觀測系統，在不影響船員工作之下，將可以推廣鼓勵商船使用共同從事台灣海洋氣象觀測工作，增進海洋氣象預報之能力。

若要建立我國海洋氣象即時的資料，最佳方式為擁有專用的海洋氣象觀測船。但考量目前政府財政上的困難和氣象即時資料的重要性，運用民間船舶所觀測的海洋氣象資料作有系統的記錄，並加以自動化及同步傳輸至中央氣象局，應為目前較可行的方式。

但是目前民間船舶所觀測的海洋氣象資料，均為目測儀器值，資料的準確性不足，所以需要將其所觀測的海洋氣象資料電子化，和訂定統一的資料格式及傳輸流程，以利資料傳輸。

中央氣象局根據所收集的海洋氣象資料作預報研判，並適時提供海上航行船隻海洋氣象資訊，以確保航行安全。建立海洋氣象資料庫作為國家建設之參考，同時可與他國交換資料，促進國際合作。

我國目前鄰近的海域，除了若干島嶼設有氣象觀測站之外，海洋上並無足夠的氣象觀測點，收集即時的氣象觀測資料，因此缺乏精確的即時海域氣象資料，運用民間船舶觀測海洋氣象資料，可彌補海洋氣象預報責任區內觀測資料之不足。

依據【行政院第十四次科技顧問會議結論與建議處理辦法中，議題參之主題一，會議結論第四條第五點】，中央氣象局擬定「推動商船參與海洋氣象觀測草案」，以鼓勵商船參與海洋氣象觀測工作。以便商船一方面將所觀測的海洋氣象資料傳輸至中央氣象局作海洋氣象預報參考，另一方面中央氣象局將完整的海洋氣象預報以通訊網方式提供責任區內的商船，以確保航行安全。如此雙方面得以互惠互利的方式共同執行本項計畫。

為了達到上述「推動商船參與海洋氣象觀測」的目標，首先需要建立船舶海象自動觀測與記錄系統，如此商船才在船上有限的人力之下，執行海象觀測作業。承本計畫第一期研究之結果，船舶海象觀測與記錄系統已建立完成，第二期研究的目標在建立船舶海象資料自動回傳系統，以便商船一方面將所觀測的海洋氣象資料傳輸至中央氣象局作海洋氣象預報參考。

本計畫擇定國立臺灣海洋大學所管理的「海研二號」，建立船舶海洋氣象觀測自動系統，以供未來推動商船參與海洋氣象觀測之基礎。

方法：

擇定「海研二號」研究船，建立為海洋氣象觀測系統自動化的船舶研究船上現有的氣象觀測儀器、定位系統與電腦設備及性能如下：

1. 風速風向計：觀測風速風向，將電子信號送到風速風向平均電路板。但是船舶在航行中所觀測得風速風向為相對風速風向，並非實際真風速風向，須參考電羅經的船艏向及全球定位系統的船速，經由電腦程式計算後，才能獲得真風速風向。其計算公式如下：

$$W = V - U$$

其中 U 為船速向量（由電羅經與GPS定出），

V 為船上風速儀測定的風速向量，

W 為實際風速向量。

2. 氣溫計：測量海平面空氣溫度並將電子信號送到JUNCTION BOX。
3. 氣壓計：測量海平面大氣壓力並將電子信號送到JUNCTION BOX。
4. 相對濕度計：測量海平面空氣的相對濕度並將電子信號送到JUNCTION BOX。
5. 風速風向平均電路板：由於風速及風向並非固定不變，若單取一時點的觀測值誤差較大，因此取一時段的平均作為觀測值，此電

路板的功能除了將一段時間的風速風向平均外，還負責將電子信號送到JUNCTION BOX。

6. JUNCTION BOX：接收風速風向平均電路板、氣溫計、氣壓計及相對濕度計的電子信號，將電子信號送到AD-CONVERT。
7. AD-CONVERT-I：接收JUNCTION BOX送來的類比信號，轉換成數位信號後送入PC。
8. 電羅經：將船艏向的信號經COM2送入PC。
9. 全球衛星定位系統：將船位及船舶速度信號經COM1送入PC。
10. 個人電腦：負責記錄海象觀測資料及計算真風速風向。

為了建立自動化系統，必須將氣象觀測儀器記錄的資料予以數位化，並撰寫電腦程式，以控制及整合各項觀測資料，並予以存檔。

結果：

本計畫研究成果如下：

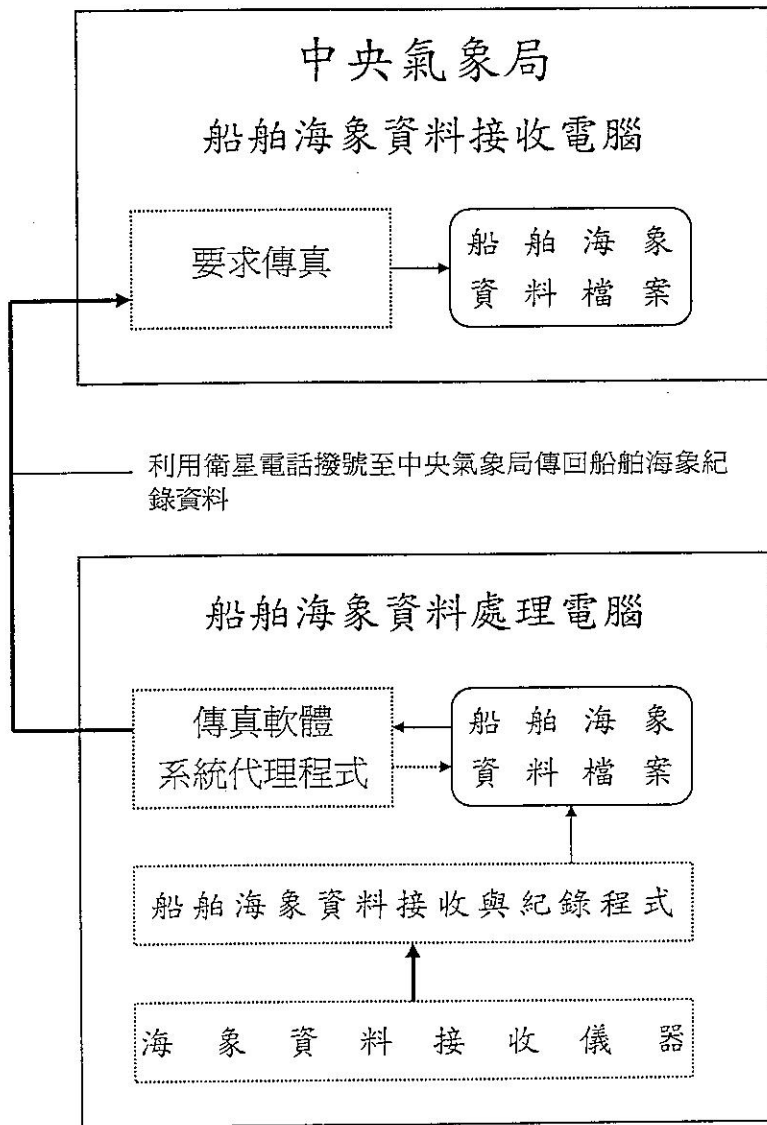
- 一、GPS 所提供的船位及航速，電羅經所提供的船艏向資料以接入 PC。
- 二、風速風向儀、氣壓計、溫度計、溼度計及日照值等所輸出的類比訊號亦轉成數位訊號並接入 PC。
- 三、整合一及二所提接入 PC 的數位資料再加上觀測之時間，撰寫船舶海象資料接收與紀錄程式，此程式已經完成，所紀錄的資料內容與格式如表一。
- 四、為使作業流程單純化及日後易於管理，本系統架設在 Window95 之下，並使用傳真軟體(AWSNTO32.EXE)將所接收的船舶海象資料，經由 MODEM 和衛星電話傳回中央氣象局接收站(本系統之架構如圖一所示)。資料傳輸處理程序分為兩部分，第一部份為將中央氣象局接收船舶傳回資料的電腦做為資料接收電腦。第二部份為將船舶

接收海象資料的電腦做為資料處理電腦，將船舶所接收的海象資料，藉助 Microsoft 的系統代理程式，於所設定的時間啟動傳真軟體(AWSNT032.EXE)撥號至中央氣象局，將電腦所記錄的海象資料，經由 Modem 傳真回中央氣象局接收站。受限於「海研二號」研究船現有海事衛星電話(IMARSAT)規格，不能使用資料傳輸模式(Data Mode)，只可使用傳真模式(Fax Mode)來傳送資料，故原先使用 PCAnywhere 遙控軟體所建立資料傳輸處理程序無法使用。

目前單機電腦對應多機電腦的一對多經由 MODEM 傳送資料的程序，已經測試完成。

五、傳送資料系統之架構如下：

船舶海洋氣象觀測自動系統



表一、目前接收資料之說明及範例

說明

欄名	紀錄資料	單位
A:	日期及時間	年年月月日日時時分分秒秒
B:	緯度	度度 分分.分分分
C:	經度	度度度 分分.分分分
D:	實際風向+	度(正北為 0 度, 正東為 90 度)
E:	實際風速+	節(海浬/小時)
F:	空氣溫度+	度(攝氏)
G:	大氣壓力+	毫巴
H:	相對溼度+	百分比
I:	日照輻射值*	Joules/meter/minute

範例

A	B	C	D	E	F	G	H	I
971001000100	25 08.823' N	121 47.295' E	72.0	12.2	23	1015.5	90.00	9.0
971001000200	25 08.825' N	121 47.294' E	74.0	8.6	23	1015.7	90.00	9.0
971001000300	25 08.822' N	121 47.293' E	74.0	7.1	23	1015.6	90.00	10.0
971001000400	25 08.823' N	121 47.290' E	64.0	9.6	23	1015.7	90.00	9.0
971001000500	25 08.826' N	121 47.289' E	73.0	8.6	23	1015.8	91.00	8.0
971001000600	25 08.831' N	121 47.296' E	74.0	10.1	23	1015.7	90.00	8.0
971001000700	25 08.834' N	121 47.307' E	69.0	11.0	23	1015.7	91.00	8.0
971001000800	25 08.832' N	121 47.309' E	70.0	8.4	23	1015.8	90.00	10.0
971001000900	25 08.831' N	121 47.304' E	62.0	10.4	23	1015.8	90.00	9.0
971001001001	25 08.830' N	121 47.301' E	75.0	8.5	23	1015.8	90.00	9.0
971001001101	25 08.826' N	121 47.306' E	68.0	11.1	23	1015.8	90.00	9.0

註: + 一般商船上裝有此項儀器, 但非自記式。

* 一般商船上沒有此項儀器。

結論：

本計畫利用「海研二號」研究船的氣象觀測儀器及電腦網路設備，已經成功地將船舶經過地方的每一時刻之各類氣象因子(風向、風速、氣溫、氣壓、相對溼度及日照輻射量)自動記錄起來，而且可經由電話網路傳送至氣象局；如果船舶在海上航行時，則即時氣象觀測結果，可經由國際海事衛星電話(IMARSAT)傳送。

商船參與海洋氣象觀測工作，如使用傳統的目儀記錄方式，則受限於船員人數精簡因素，往往不易執行。要鼓勵商船參與海洋氣象觀測工作，首先要讓商船上的氣象觀測及資料記錄與傳送完全自動化，而不增加船員的工作，如此才可能推動商船參與海洋氣象觀測。

本研究所完成的船舶海洋氣象觀測自動系統，已可提供商船此類系統設備之原型(Protocol)。茲將本研究成果「船舶海洋氣象觀測自動系統」所需之軟硬體設備條列如下，以提供商船裝置此類系統之參考。

硬體設備(這些儀器皆需要自記功能且能將資料數位化並且連接埠可將資料傳送至電腦)

(一) 氣象觀測儀器

風向、風速儀、溫度探針、濕度計、壓力計、日照探針

(二) 全球衛星定位系統(GPS)

(三) 個人電腦(CPU:INTEL PENTIUM-166, RAM:32MB, HD:4GB 級以上)及 33.6K 數據機

(四) 國際海事衛星電話(IMARSAT)

軟體：控制記錄的資料及傳送之程式。本計畫是以 BASIC 語言撰寫程式(如附錄)，其輸出結果如表一所示。

誌謝：

本計畫得以完成，主要歸功於「海研二號」研究船提供船上設備。曾安源、黃余達先生與蘇兆輝先生三位技士兼任本計畫之助理，盡心盡力地協助各項工作，才能順利達成計畫目標。最後感謝「海研二號」全體船上工作人員之合作。

附 錄

1. 研究船及海觀測儀器相關設備相片
2. 海象觀測系統原始程式



電羅經

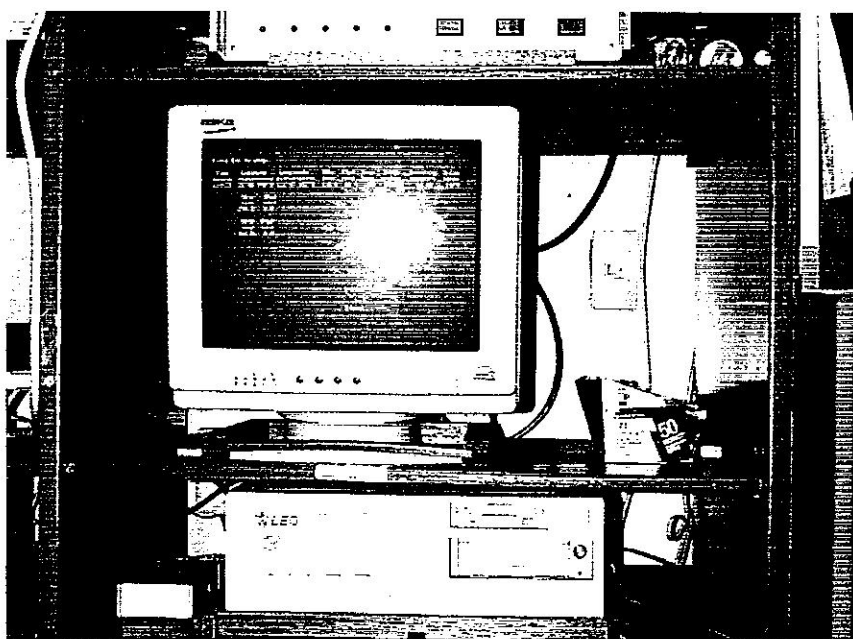


全球衛星定位系統 (GPS)





422



執行資料接收程式之情形

```
' Program:      Marine Star system for Receiving & Recording data
' Programmer:   David Sue
' Date:        98.05.01
```

```
ON ERROR GOTO ErrorHandler
```

```
'----- MAIN Loop -----
---
```

```
GOSUB IniInitialization
```

```
DO
```

```
  GOSUB CheckInkey
  GOSUB GetGPSData
  GOSUB getgyrodata
  GOSUB CheckGPSData
  GOSUB GetWeatherData
  GOSUB WindDirectionAndSpeed
  GOSUB AirHumidity
  GOSUB AirTemperature
  GOSUB SolarRadiation
  GOSUB BarometricPressure
  TimeTmp$ = LEFT$(TIME$, 2) + MID$(TIME$, 4, 2) + RIGHT$(TIME$, 2)
  GOSUB DisplayData
  GOSUB RecordData
```

```
LOOP
```

```
GOSUB ExitSystem
```

```
IniInitialization:
```

```
CLS
```

```
SCREEN 12
```

```
LINE (380, 310)-(450, 310), 3
LINE (380, 360)-(450, 360), 4
LINE (380, 405)-(450, 405), 2
LOCATE 20, 42: PRINT " Ship "
LOCATE 23, 42: PRINT "R-Wind"
LOCATE 26, 42: PRINT "T-Wind"
```

```
ViewXMin = 20
```

```
ViewYMin = 250
```

```
ViewLength = 200
```

424

```
VIEW (ViewXMin, ViewYMin)-(ViewXMin + ViewLength, ViewYMin +
ViewLength), , 6
WINDOW (-50, -50)-(50, 50)
```

```
LOCATE 2, 1: PRINT "Press Esc to stop."
LOCATE 4, 1: PRINT " Time      Latitude      Longitude      TDir  TSpd  Temp
Baro  Humidity Solar"
LOCATE 5, 1: PRINT "=====
=====
LOCATE 8, 9: PRINT " SDir  SSpd"
LOCATE 9, 9: PRINT " =====
LOCATE 12, 9: PRINT " RDir  RSpd"
LOCATE 13, 9: PRINT " =====
```

```
AvgInterval = 1
DisplayTimeChoice = 1      ' Choice of Seconds/Display
RecordTimeChoice = 2      ' Choice of Seconds/Record
                          ' SELECT CASE
RecordTimeChoice/DisplayTimeChoice
                          ' CASE 0:          Unlimited
                          ' CASE 1:          10 Seconds / Record
                          ' CASE 2:          1 Minutes / Record
                          ' CASE 3:          10 Minutes / Record
                          ' CASE 4:          1 Hours / Record
                          ' END SELECT
```

```
floppy$ = "T"
pi = 3.1415926#
AvgMemory = AvgInterval * 10
RawDataCount = 6
OldDate$ = " "
DIM WindSpeed(AvgMemory), WindDirect(AvgMemory)
DIM rawdata(RawDataCount - 1), filename$(5)
```

```
filename$(1) = "com1:4800,n,8,1,cs0,ds0"
OPEN filename$(1) FOR INPUT AS #1
filename$(2) = "com2:4800,n,8,1,cs0,ds0"
OPEN filename$(2) FOR INPUT AS #2
```

```
RETURN
```

```
ExitSystem:
```

```
CLOSE
```

STOP
END

GetGPSData:

```

GPSData$ = ""
gpsidflag = 0
DO WHILE gpsidflag < 3
  LINE INPUT #1, GPSData$
  GDHead$ = MID$(GPSData$, 2, 6)
  IF GDHead$ = "$GPGLL" THEN
    gpsidflag = 1
    GPSLat$ = MID$(GPSData$, 9, 2) + CHR$(248) + MID$(GPSData$, 11, 6)
+ "" + MID$(GPSData$, 18, 1)
    GPSTLon$ = MID$(GPSData$, 20, 3) + CHR$(248) + MID$(GPSData$, 23,
6) + "" + MID$(GPSData$, 30, 1)
  ELSEIF GDHead$ = "$GPRMC" THEN
    gpsidflag = 1
    GPSLat$ = MID$(GPSData$, 9, 2) + CHR$(248) + MID$(GPSData$, 11, 6)
+ "" + MID$(GPSData$, 18, 1)
    GPSTLon$ = MID$(GPSData$, 20, 3) + CHR$(248) + MID$(GPSData$, 23,
6) + "" + MID$(GPSData$, 30, 1)
  END IF
  IF GDHead$ = "$GPVTG" AND gpsidflag = 1 THEN
    gpsidflag = 2
    GPSTrack$ = MID$(GPSData$, 9, 3)
    GPSSpeed$ = MID$(GPSData$, 21, 5)
  END IF
  IF GDHead$ = "$GPZDA" AND gpsidflag = 2 THEN
    gpsidflag = 3
    GPSTime$ = MID$(GPSData$, 16, 10)
    GPSTime$ = MID$(GPSData$, 9, 6)
  END IF
LOOP

```

RETURN

getgyrodata:

```

gyrodata$ = ""
gyroflag = 0
DO WHILE gyroflag < 1
  LINE INPUT #2, gyrodata$
  IF MID$(gyrodata$, 2, 6) = "$AGHDT" THEN gyroflag = 1
  head$ = MID$(gyrodata$, 9, 5)

```

```

LOOP
RETURN

```

CheckGPSData:

```

    IF OldGPSLat$ <> GPSLat$ THEN OldGPSLat$ = GPSLat$ ' GPSLatitude
    IF OldGPSLon$ <> GPSLon$ THEN OldGPSLon$ = GPSLon$ ' GPSLongitude
    IF OldGPSTrack$ <> GPSTrack$ THEN OldGPSTrack$ = GPSTrack$ '
GPSGroundTrack
    IF OldGPSSpeed$ <> GPSSpeed$ THEN OldGPSSpeed$ = GPSSpeed$ '
GPSGroundSpeed

RETURN

```

GetWeatherData:

```

FOR idx = 0 TO RawDataCount - 1
    OUT 770, idx
    OUT 769, 0
    rawdata(idx) = INP(768) / 16 + INP(769) * 16 - 2048
    IF rawdata(idx) < 0 THEN rawdata(idx) = 0
NEXT idx

bpraw = rawdata(0)
wsraw = rawdata(1)
wdraw = rawdata(2)
srraw = rawdata(3)
atraw = rawdata(4)
ahRaw = rawdata(5)

RETURN

```

WindDirectionAndSpeed:

```

    shipspeed = VAL(GPSSpeed$)
    ShipDirect = VAL(GPSTrack$)
    ShipDirect = VAL(head$)
windshipspeed = INT(wsraw / 19.48)

windshipdirect = INT(wdraw / 5.686)
AntiWindShipDirect = windshipdirect + 180

```

```

GOSUB CalculateWindVector      ' Calculate Wind Vector

IF TrueWindVectorSum < AvgMemory THEN TrueWindVectorSum =
TrueWindVectorSum + 1
TrueWindVectorCounter = (TrueWindVectorCounter + 1) MOD AvgMemory
WindSpeed(TrueWindVectorCounter) = TrueWindSpeed
WindDirect(TrueWindVectorCounter) = TrueWindDirect
WindVectorX = 0: WindVectorY = 0
FOR n = 0 TO TrueWindVectorSum
    TVWRad = WindDirect(n) * (pi / 180)
    WindVectorX = WindVectorX + WindSpeed(n) * COS(TVWRad)
    WindVectorY = WindVectorY + WindSpeed(n) * SIN(TVWRad)
NEXT n
WindVectorX = WindVectorX / TrueWindVectorSum: IF WindVectorX = 0 THEN
WindVectorX = .000001
WindVectorY = WindVectorY / TrueWindVectorSum
AvgTrueWindSpeed = SQR(WindVectorY * WindVectorY + WindVectorX *
WindVectorX)
AvgTrueWindDirect = INT((ATN(WindVectorY / WindVectorX)) * (180 / pi)
+ .5)
IF WindVectorX < 0 THEN
    AvgTrueWindDirect = AvgTrueWindDirect + 180
ELSEIF WindVectorY < 0 THEN
    AvgTrueWindDirect = AvgTrueWindDirect + 360
END IF
IF AvgTrueWindSpeed <> oldAvgTrueWindSpeed THEN oldAvgTrueWindSpeed =
AvgTrueWindSpeed
IF AvgTrueWindDirect <> oldAvgTrueWindDirect THEN oldAvgTrueWindDirect
= AvgTrueWindDirect

RETURN

AirHumidity:

AirHumidity = INT(ahRaw / 32) + 32

RETURN

AirTemperature:

IF atraw < 164 THEN atraw = 164      'Range is -30 to +50 deg C
IF atraw > 819 THEN atraw = 819
AirTemp = INT((atraw - 164) / 8.188) - 30  ' CalculateAirTemperature
IF AirTemp <> oldAirTemp THEN oldAirTemp = AirTemp

```

RETURN

SolarRadiation:

' ----- Calculate Solar Radiation -----

Solar = INT(srrow / .8673) ' JOULES / meter / minute
 'Solar = INT(srrow / 1.44) 'if in WATTS 5V = 1422.5 W/M
 IF Solar <> OldSolar THEN OldSolar = Solar

RETURN

BarometricPressure:

' ----- Calculate Barometric Pressure -----

Baro = (bpraw / 8.6) + 926 '0V = 930 mb and 2V = 1030 mb
 IF Baro <> oldbaro THEN oldbaro = Baro

RETURN

CalculateWindVector:

AntiWindShipDirectRad = AntiWindShipDirect * (pi / 180)
 TrueWindSpeedY = shipspeed * SIN(AntiWindShipDirectRad)
 TrueWindSpeedX = windshipspeed + shipspeed * COS(AntiWindShipDirectRad)
 TrueWindSpeed = SQR(TrueWindSpeedX * TrueWindSpeedX + TrueWindSpeedY *
 TrueWindSpeedY)
 IF TrueWindSpeedX = 0 THEN TrueWindSpeedX = .001
 TrueWindCrossWindShipArc = ATN(TrueWindSpeedY / TrueWindSpeedX) * (180
 / pi)
 IF TrueWindSpeedX < 0 THEN
 IF TrueWindSpeedY >= 0 THEN
 TrueWindCrossWindShipArc = TrueWindCrossWindShipArc + 180
 ELSE
 TrueWindCrossWindShipArc = TrueWindCrossWindShipArc - 180
 END IF
 END IF
 TrueWindDirect = (ShipDirect + (windshipdirect -
 TrueWindCrossWindShipArc)) MOD 360
 RETURN

CheckInkey:

```

K$ = INKEY$
IF K$ = "" THEN
    RETURN
ELSEIF ASC(K$) = 27 THEN
    GOSUB ExitSystem
END IF

RETURN

```

RecordData:

```

GOSUB OutputRawData

IF RecordTimeChoice = 0 THEN
ELSEIF RecordTimeTmp$ <> MID$(TimeTmp$, 6 - RecordTimeChoice, 1) THEN
    RecordTimeTmp$ = MID$(TimeTmp$, 6 - RecordTimeChoice, 1)
ELSE
    RETURN
END IF

GOSUB OutputData

RETURN

```

DisplayData:

```

IF DisplayTimeChoice = 0 THEN
ELSEIF DisplayTimeTmp$ <> MID$(TimeTmp$, 6 - DisplayTimeChoice, 1) THEN
    DisplayTimeTmp$ = MID$(TimeTmp$, 6 - DisplayTimeChoice, 1)
ELSE
    RETURN
END IF
DisplayTime$ = TimeTmp$

GOSUB ShowData
GOSUB ShowWindVectorDiagram

RETURN

```

OutputData:

```

IF DATE$ <> OldDate$ THEN

```



```

PRINT #OFNo, " Solar : Solar Radiation (Joules/meter/minute)"
PRINT #OFNo, " Baro : Barometric Pressure (mb)"
PRINT #OFNo, " TDir : True Wind Direction (" + CHR$(248) + ")"
PRINT #OFNo, " TSpd : True Wind Speed (Knots)"
PRINT #OFNo, " SDir : Ship Direction (" + CHR$(248) + ")"
PRINT #OFNo, " SSpd : Ship Speed (Knots)"
PRINT #OFNo, " RDir : Wind - Ship Resultant Direction (" + CHR$(248)
+ ")"
PRINT #OFNo, " RSpd : Wind - Ship Resultant Speed (Knots)"

RETURN

```

OutputRawData:

```

IF RecordTimeTmp$ <> MID$(TIME$, 5 - ((RecordTimeChoice + 3) MOD 4), 1)
THEN
  IF RecordTimeTmp$ = MID$(TIME$, 2, 1) THEN
    RecordRawData$ = LEFT$(TIME$, 2)
  END IF
  RecordRawData$ = RecordRawData$ + MID$(TIME$, 4, 2)
  RecordRawData$ = RecordRawData$ + " " + GPSLat$ + " " + GPSLon$
  RecordRawData$ = RecordRawData$ + " " + GPSSpeed$ + " " + GPSTrack$
END IF
FOR rawidx = 1 TO 2
  RecordRawData$ = RecordRawData$ + LTRIM$(STR$(rawdata(rawidx))) + "
"
  FOR n = 0 TO TrueWindVectorSum
    IF rawidx = 1 THEN
      RecordRawData$ = RecordRawData$ + LTRIM$(STR$(WindSpeed(n))) +
" "
    ELSEIF rawidx = 2 THEN
      RecordRawData$ = RecordRawData$ + LTRIM$(STR$(WindDirect(n)))
+ " "
    END IF
  NEXT n
NEXT
RETURN

```

ShowData:

```

LOCATE 6, 1: PRINT USING "& & & ###.# ##.# ###  ####.# ###.##
####"; DisplayTime$, GPSLat$, GPSLon$, AvgTrueWindDirect, AvgTrueWindSpeed;

```

AirTemp; Baro; AirHumidity; Solar

LOCATE 10, 9: PRINT USING "###.# ##.#"; ShipDirect; shipspeed;

LOCATE 14, 9: PRINT USING "###.# ##.#"; windshipdirect; windshipspeed

RETURN

ShowWindVectorDiagram:

```

MaxLength = -1
ScaleDraw = 1
AngleShip = 90 - ShipDirect
AngleWind = 90 - (ShipDirect + windshipdirect)
xsnew = shipspeed * COS(AngleShip * pi / 180)
ysnew = shipspeed * SIN(AngleShip * pi / 180)
xwnew = windshipspeed * COS(AngleWind * pi / 180)
ywnew = windshipspeed * SIN(AngleWind * pi / 180)
IF shipspeed > MaxLength THEN MaxLength = shipspeed
IF windshipspeed > MaxLength THEN MaxLength = windshipspeed
IF ABS(MaxLength > .1) THEN ScaleDraw = 50 / MaxLength * .8 ELSE ScaleDraw
= 0
LINE (0, 0)-(xsold * ScaleDrawold, ysold * ScaleDrawold), 0
LINE (0, 0)-(xwold * ScaleDrawold, ywold * ScaleDrawold), 0
LINE (xwold * ScaleDrawold, ywold * ScaleDrawold)-(xsold * ScaleDrawold,
ysold * ScaleDrawold), 0
LINE (0, 0)-(xsnew * ScaleDraw, ysnew * ScaleDraw), 3
LINE (0, 0)-(xwnew * ScaleDraw, ywnew * ScaleDraw), 4
LINE (xwnew * ScaleDraw, ywnew * ScaleDraw)-(xsnew * ScaleDraw, ysnew *
ScaleDraw), 2
xwold = xwnew
ywold = ywnew
xsold = xsnew
ysold = ysnew
ScaleDrawold = ScaleDraw

```

RETURN

ErrorHandler:

```

RESUME NEXT
END

```

中華民國交通部中央氣象局

台北市100公園路64號

CENTRAL WEATHER BUREAU

MINISTRY OF TRANSPORTATION AND COMMUNICATIONS

64, KUNG YUAN ROAD, TAIPEI, TAIWAN, 100

REPUBLIC OF CHINA