

## **EPHEMERIDES OF THE SUN AND DETERMINATION OF POSITION**

A Program (Version 2) for the HP-28S Scientific Calculator

The program outlined below will run on an HP-28S calculator and will provide quick and accurate ephemeris of the sun without the bulk or cost of the yearly almanac. The program can be used to calculate the declination (Dec) and Greenwich hour angle (GHA) of the sun for any time between March 1, 1900 and February 28, 2100. If the sun's observed altitude and your assumed position are also entered the program will provide the calculated altitude (Hc), azimuth angle (Azn) and intercept (Ic) thus allowing you to plot a line of position (St Hilaire's method). Although it has not been tested exhaustively, several test cases have been checked with good results against the U.S. Naval Observatory's Interactive Computer Ephemeris program (ICE Version 0.51).

The main routine and the sub-routines have all been placed in one user directory. I have sorted the directory such that the main routine (SUN) is first on the menu for easy access. This routine provides the user interface and prompts for the necessary information required to complete the calculations. The majority of these calculations are also performed within this routine but calculations and/or logic statements that are required more than once have been saved as separate sub-routines so that they may be called as needed. The program stores the results of the various calculations in the same user directory as the main program routine and sub-routines. This allows the user to manually access the results of the intermediate calculations as the program itself only displays the final calculations listed above.

To run the program simply input the data requested using the format indicated on the display (ie. h,m,s). The "continue" key on the HP key pad is then used to continue the program. As a note of caution, I did not create any subroutines to check the validity of the data entered. As such, be careful to enter valid data in the correct format.

The formulas used for this program were sourced from Henning Umland's website at "[home.t-online.de/home/h.umland/](http://home.t-online.de/home/h.umland/)" and, as I have not explained them here in any detail you may want to peruse this site for a better understanding of what is happening. This is an excellent site and is great for those with a bit of mathematics behind them who want to understand the why of everything (rather than the how covered by most books on the subject). The Interactive Computer Ephemeris program mentioned above is also available from his website.

## MAIN ROUTINE: SUN

The main routine starts with << and ends with >>; the text in the right hand column is information for the programmer and is not to be included in the code.

```
<< CLLCD
  "Celestial Navigation" 1 DISP .5 WAIT
  "Ephemerides of" 3 DISP
  "the Sun" 4 DISP 2 WAIT
  CLLCD
  "Enter hours, minutes" 1 DISP
  "and seconds (GMT) of" 2 DSIP
  "the sun sight (h,m,s)" 3 DISP
  HALT
  → H M S 'H + M/60 + S/3600'
  'GMT' STO CLLCD
  "Enter year, month" 1 DISP
  "and day of the sight (y,m,d)" 2 DISP
  HALT
  'GMT' EVAL 4 ROLL 4 ROLL 4 ROLL
  → GMT Y M D '367*y - IP(1.75 * IP(Y +
  IP((M+9)/12))) + IP(275 * M/9) + D +
  GMT/24 - 730531.5'
  'T' STO CLLCD
  'T' EVAL 'GMT' EVAL
  → T GMT '0.9856473*T + 15*GMT +
  100.46023'
  'RANG' EVAL
  'GHAA' STO
  'T' EVAL
  → T '0.98560038*T - 2.472
  'RANG' EVAL
  'G' STO
  'T' EVAL
  → T '0.9856473*T - 79.54'
  'LM' STO
  'G' EVAL 'LM' EVAL
  → G LM 'LM + 1.915*SIN(G) +
  0.02*SIN(2*G)'
  'RANG' EVAL
  'LT' STO
```

Clears display

Time must be entered in GMT

Halts program for data entry

**Greenwich mean time decimal conversion**  
Stores the result as 'GMT'

Puts GMT back on stack and positions data  
**Time variable calculation** (see notes)

Stores the result at 'T'

Puts 'T' and 'GMT' on the stack  
**Greenwich hour angle (aries) calculation**

Calls '**RANG**' routine to ensures angle is < 360°  
Stores result as 'GHAA'

**Mean anomaly of the sun calculation** (see notes)  
Calls '**RANG**' routine to ensures angle is < 360°  
Stores the result as "G"

**Mean longitude of the sun calculation** (see notes)  
Stores the result as 'LM'

**Mean longitude of the sun calculation** (see notes)  
Calls '**RANG**' routine to ensures angle is < 360°  
Stores the result as "LT"

```
'T' EVAL
→ T '23.439 - 0.0000004*T'
'GE' STO
```

**Obliquity of the ecliptic calculation** (see notes)  
Store as 'GE'

```
'LT' EVAL 'GE' EVAL
→ LT GE 'ASIN(SIN(LT) * SIN(GE))'
'DECL' STO
```

**Declination of the sun calculation** (see notes)  
Store as 'DECL'

```
'LT' EVAL 'GE' EVAL 'DECL' EVAL
→ LT GE DECL '2*ATAN(COS(GE) *
SIN(LT)/(COS(DECL) + COS(LT)))'
'RA' STO
```

**Right ascension of the sun calculation** (see notes)  
Store as 'RA'

```
'GHAA' EVAL 'RA' EVAL
→ GHAA RA 'GHAA - RA'
'RANG' EVAL
'GHAS' STO
```

**Greenwich hour angle of the sun** (see notes)  
Calls '**RANG**' routine to ensures angle is < 360°  
Store as 'GHAS'

```
'DECL' EVAL GHAS EVAL
→ DECL GHAS
<< CLLCD
  "The declination and" 1 DISP
  "GHA of the sight are" 2 DISP
  " DECL =" DECL →STR + 3 DISP
  " GHAS =" GHAS →STR + 4 DISP
  DO UNTIL KEY END
  DROP
  CLMF
>>
```

Picks declination and GHA (sun) from the stack to display them

Displays declination on third row  
Displays GHA (sun) on fourth row  
Displays until any key is pressed  
Drops pressed key from the stack

```
CLLCD
"Enter your apparent" 1 DISP
"Latitude and longitude" 2 DISP
" Lat (d, m.mm)" 3 DISP
" Long (d, m.mm)" 4 DISP
HALT
```

Prompts for entry of apparent latitude and longitude

Halts program for data entry

```
'RLLD' EVAL 'LONA' STO
'RLLD' EVAL 'LATA' STO
```

Calls **RLLD** routine to convert d:m.m to d.ddd  
Calls **RLLD** routine to convert d:m.m to d.ddd

```
'GHAS' EVAL 'LONA' EVAL
→ GHAS LONA 'GHAS+LONA'
'RANG' EVAL
'LHA' STO CLLCD
```

**Local hour angle calculation** (see notes)  
Calls '**RANG**' routine to ensures angle is < 360°

```

‘DECL’ EVAL ‘LHA’ EVAL ‘LATA’ EVAL
→ DECL LHA LATA ‘ASIN(SIN(LATA) *
SIN(DECL) + COS(LATA) * COS(DECL) *
COS(LHA))’
‘HC’ STO

```

**Calculated altitude of the sun from apparent latitude and longitude calculation** (see notes)

```

‘DECL’ EVAL ‘LATA’ EVAL ‘HC’ EVAL
→ DECL LATA HC ‘ACOS((SIN(DECL)-
SIN(HC) * SIN(LATA)) / (COS(HC) *
COS(LATA)))’
‘AZ’ STO

```

**Azimuth angle calculation** (see notes)

```

‘AZ’ EVAL ‘LHA’ EVAL
→ LHA ‘SIN(LHA)’

```

Puts AZ (needed for below) and LHA on the stack  
Takes the sin of the LHA, AZ still on stack

```

→ LHAS
<< IF LHAS 0 <
  THEN
  ELSE 360 SWAP –
  END

```

Takes the sin of the LHA off the stack as LHAS  
**Azimuth angle relative to true north logic and calculation** (see notes)

```

>>
‘AZN’ STO

```

Stores the result as AZN

```

CLLCD
“Enter the sun’s” 1 DISP
“observed altitude” 2 DISP
“Ho (d, m.mm)” 3 DISP
HALT

```

Halts program for data entry

```

‘RLLD’ EVAL
‘HO’ STO

```

Calls **RLLD routine** to convert d:m.m to d.ddd  
Stores the result as HO

```

‘HO’ EVAL ‘HC’ EVAL
→ HO HC ‘40031.6*(HO – HC)/360’
‘IC’ STO

```

**Intercept calculation** (see notes)

```

‘AZN’ EVAL ‘HC’ EVAL ‘IC’ EVAL
→ AZN HC IC
<< CLLCD
“Final calculations” 1 DISP
“ HC   =” HC →STR + 2 DISP
“ AZn  =” AZN →STR + 3 DISP
“ IC(km)=” IC →STR + 4 DISP
DO UNTIL KEY END
DROP
CLMF

```

Converts HC to a string and adds it to “HC =” so that it displays on one line

```

>>
HOME CEL

```

```

>>

```

## **SUBROUTINES:**

The following are the subroutines called on by the main routine SUN (except for AZn which is a routine within SUN). Be sure to save them as I have indicated (ie. RANG, RLLD) or else the main routine won't be able to find them when needed (or, if you do chose to change a routine name you must also change it in the main routine as well). The routines start with << and end with >>; the text above and below is only provided as information for the programmer.

### **RANG SUBROUTINE: ANGLE CONVERT**

```
<<
  WHILE DUP 360 >= REPEAT 360 - END
  WHILE DUP 0 < REPEAT 360 + END
>>
```

Several of the subroutines return results with units of degrees. This subroutine ensures that the final result is between 0 and 360 degrees by continually subtracting or adding 360 until the result is between 0 and 360 degrees.

### **RLLD SUBROUTINE: LATITUDE/LONGITUDE CONVERSION**

```
<<
  → M 'M/60' SWAP
  → D
  << IF D 0 >=
    THEN D +
    ELSE D SWAP -
    END
  >>
>>
```

This routine is used to covert lat/long from d:m.mm format to d.dddddd. The logic part of the routine is used to take care of negative entries like -79,30. It ensures that in this case the minutes are 'added' to the negative degrees such that the result is -79.5 degrees and not -78.5.

### **AZn ROUTINE: CALCULATES THE AZIMUTH ANGLE RELATIVE TO TRUE NORTH**

```
'AZ' EVAL 'LHA' EVAL
→ LHA 'SIN(LHA)'
→ LHAS
<< IF LHAS 0 <
  THEN
  ELSE 360 SWAP -
  END
>>
```

AZ is the angle between the meridian line through AP and the great circle through AP and GP. If it is before local noon (ie. LHA > 180) then AZn = AZ. If it is after local noon (ie. LHA < 180) the ACOS formula for AZ returns a complimentary angle to the one needed - it must be subtracted from 360 degrees to give the bearing from true north (AZn). This relationship can be readily seen by drawing a quick hand sketch of the north pole, AP and GP.

Note: Both 'AZ' and 'LHA' must be on the stack for this routine to work thus the first line where they are both evaluated from memory

The sin of LHA is then taken and stored in the local memory as LHAS  
LHAS is then compared to 0 and if:

LHA > 180 it returns AZ (still on the stack)

LHA < 180 it returns 360 - AZ

## **CALCULATIONS:**

The following are the calculations contained in the main routine SUN. Some notes are included.

### **T: TIME VARIABLE CALCULATION**

The time variable represents the number of days before or after 01/01/2000 12:00:00 GMT. The formula is valid from March 1, 1900 through Feb. 28, 2100.

→ GMT Y M D '367\*y - IP(1.75 \* IP(Y + IP((M+9)/12))) + IP(275 \* M/9) + D + GMT/24 - 730531.5'

### **GMT: GREENWICH MEAN TIME CALCULATION**

Converts the hours, minutes and seconds entered into decimal format.

→ H M S 'H + M/60 + S/3600'

### **GHAA: GREENWICH HOUR ANGLE (ARIES) CALCULATION**

Calculates the greenwich hour angle of Aries.

Final result must be between 0 and 360 degrees (add/sub 360 as necessary)

→ T GMT '0.9856473\*T + 15\*GMT + 100.46023'

### **G: MEAN ANOMALY OF THE SUN CALCULATION**

Final result must be between 0 and 360 degrees (add/sub 360 as necessary)

→ T '0.98560038\*T - 2.472                      Units = degrees

### **LM: MEAN LONGITUDE OF THE SUN CALCULATION**

→ T '0.9856473\*T - 79.54'                      Units = degrees

### **LT: TRUE LONGITUDE OF THE SUN CALCULATION**

Final result must be between 0 and 360 (add/sub 360 as necessary), units = degrees

→ G LM 'LM + 1.915\*SIN(G) + 0.02\*SIN(2\*G)'

**GE: OBLIQUITY OF THE ECLIPTIC CALCULATION**

→  $T \cdot 23.439 - 0.0000004 \cdot T$  Units = degrees

**DECL: DECLINATION OF THE SUN CALCULATION**

→  $LT \cdot GE \cdot \text{ASIN}(\text{SIN}(LT) \cdot \text{SIN}(GE))$  Units = degrees

**RA: RIGHT ASCENSION OF THE SUN CALCULATION**

→  $LT \cdot GE \cdot \text{DECL} \cdot \text{'2} \cdot \text{ATAN}(\text{COS}(GE) \cdot \text{SIN}(LT) / (\text{COS}(\text{DECL}) + \text{COS}(LT)))$  Units = degrees

**GHAS: GREENWICH HOUR ANGLE OF THE SUN CALCULATION**

Final result must be between 0 and 360 (add/sub 360 as necessary), units = degrees

→  $\text{GHAA} \cdot \text{RA} \cdot \text{'GHAA} - \text{RA}$

**LHA: LOCAL HOUR ANGLE CALCULATION**

LONA = apparent longitude, LATA = apparent latitude

→  $\text{GHAS} \cdot \text{LONA} \cdot \text{'GHAS} + \text{LONA}$  Units = degrees

**HC: CALCULATED ALTITUDE OF THE SUN FROM APPARENT LAT/LONG**

→  $\text{DECL} \cdot \text{LHA} \cdot \text{LATA} \cdot \text{'ASIN}(\text{SIN}(\text{LATA}) \cdot \text{SIN}(\text{DECL}) + \text{COS}(\text{LATA}) \cdot \text{COS}(\text{DECL}) \cdot \text{COS}(\text{LHA}))$

**AZ: AZIMUTH ANGLE CALCULATION**

→  $\text{DECL} \cdot \text{LATA} \cdot \text{HC} \cdot \text{'ACOS}((\text{SIN}(\text{DECL}) - \text{SIN}(\text{HC}) \cdot \text{SIN}(\text{LATA})) / (\text{COS}(\text{HC}) \cdot \text{COS}(\text{LATA})))$

**IC: CALCULATES THE INTERCEPT**

→  $\text{HO} \cdot \text{HC} \cdot \text{'4003} \cdot 1.6 \cdot (\text{HO} - \text{HC}) / 360$  Units = kilometers