
Subject: sun rise/set calculator multiple zenits update

From: KM (km777@centrum.sk)

To: jllammi@yahoo.com;

Date: Sunday, September 7, 2014 8:39 PM

Hello

could you help me please update this your source code for using multiple zenits

```
#define znt_offical 90 // a 50min
#define znt_civil 96 // degrees
#define znt_nautical 102 // degrees
#define znt_astronomical 108 // degrees
```

```
// example
// suncalc(2014,8,SUNRISE,48.142,17.099,znt_official){
//
```

where to assing zenits value ?

```
double suncalc(int year,char month,day,suncalctype, double lat1,lot1, char
zenith);
```

```
// example
// suncalc(2014,8,SUNRISE,48.142,17.099,znt_official){
//
```

```
double longitude = lot1;
double latitude = lat1;
```

```
// 1. first calculate the day of the year
```

```
double N1,N2,N3,N;
double M,L,H,T;
double Lquadrant,Rquadrant,RA;
double sinDec,cosDec,cosH;
```

```
double UT,localT,localOffset,lngHour;
```

```
N1 = floor(275 * month / 9);
N2 = floor((month + 9) / 12);
```

```
N3 = (1 + floor((year - 4 * floor(year / 4) + 2) / 3));
N = N1 - (N2 * N3) + day - 30;
printf("N1 = %f, N2 = %f, N3 = %f, N = %f\n", N1, N2, N3, N);

// 2. convert the longitude to hour value and calculate an approximate time

lngHour = longitude / 15;
// ak pocitam vychod inak zapad slnka
if (suncalctype == SUNRISE ) t = N + ((6 - lngHour) / 24)
else t = N + ((18 - lngHour) / 24);

// 3. calculate the Sun's mean anomaly

M = (0.9856 * t) - 3.289;

// 4. calculate the Sun's true longitude

L = M + (1.916 * sin(M)) + (0.020 * sin(2 * M)) + 282.634;
// NOTE: L potentially needs to be adjusted into the range [0,360) by
adding/subtracting 360

// 5a. calculate the Sun's right ascension

RA = atan(0.91764 * tan(L));
NOTE: RA potentially needs to be adjusted into the range [0,360) by
adding/subtracting 360

// 5b. right ascension value needs to be in the same quadrant as L

Lquadrant = (floor( L/90)) * 90;
RAquadrant = (floor(RA/90)) * 90;
RA = RA + (Lquadrant - RAquadrant);

//5c. right ascension value needs to be converted into hours

RA = RA / 15;

//6. calculate the Sun's declination

sinDec = 0.39782 * sin(L);
cosDec = cos(asin(sinDec));

//7a. calculate the Sun's local hour angle
```

```
cosH = (cos(zenith) - (sinDec * sin(latitude))) / (cosDec * cos(latitude));

if (cosH > 1) printf("the sun never rises on this location \\\n");
if (cosH < -1) printf("the sun never sets on this location \\\n");

// 7b. finish calculating H and convert into hours

// if if rising time is desired:
if (suncalctype == SUNRISE ) H = 360 - acos(cosH)
else H = acos(cosH);
// if setting time is desired:

H = H / 15;

// 8. calculate local mean time of rising/setting

T = H + RA - (0.06571 * t) - 6.622;

// 9. adjust back to UTC

UT = T - lngHour;
// NOTE: UT potentially needs to be adjusted into the range [0,24) by
adding/subtracting 24

// 10. convert UT value to local time zone of latitude/longitude

localT = UT + localOffset;

}
```