

THE MAIDENHEAD QRA LOCATOR SYSTEM

Worked Example

The Maidenhead Locator System divides the world into an 18x18 matrix of rectangles, called fields, of 20° of longitude and 10° of latitude. These fields are then divided into a 10x10 matrix of “squares” of 2° of longitude and 1° of latitude. These “squares” are further divided into a 24x24 matrix of sub-squares of 5' of longitude and 2.5' of latitude.

To avoid negative numbers, latitude is measured from the South Pole, giving the equator a false northing of 90° and longitude is measured from the anti-meridian position giving the Prime Meridian a false easting of 180°.

Latitude and longitude angles are entered in the DD.MMSS format, with northern latitudes and eastern longitude positive, southern latitudes and western longitudes negative.

The fields are thus represented by the numbers 0 – 17, and coded by the letters A – R, with A = 0, B = 1, C = 2 etc.

The squares are represented by their number, from 0 – 9, and the sub-squares by a number from 0 – 23, represented by the letters A – X, with again, A = 0, B = 1, C = 2 etc.

The resultant 3-character alphanumeric codes of latitude and longitude are merged into a 6-character composite of the form: AB12CD, the first character letter being longitude, the second being latitude, the third longitude, the fourth latitude, the fifth longitude, and the sixth latitude.

For example, entering the data: latitude 34°38' 45" N, (34.3845) and longitude 32°12' 30" E, (32.1230) gives a latitude code of: M4P, and a longitude code of: K6C, which when merged gives the QRA locator code of: KM64CP.

Rather than trying to calculate the QRA locator code directly from the angular data, instead it is converted into a position number. In other words, the angular data is converted to its decimal format, and then quantised into 2.5' increments for latitude, or 5' increments for longitude.

There are 240 sub-squares and squares to a field designator, thus, by dividing the angular position number by 240, the integer part is the field number, between 0 – 17, and a simple look-up table gives its equivalent letter.

The remainder is multiplied by 240 and divided by 24, and the integer part of this quotient gives the square designator, between 0 – 9, and is used directly.

The remainder of this quotient is multiplied by 24, and the integer portion is now the sub-square designator, in the range 0 – 23, and a simple look-up table gives its equivalent letter.

To perform the inverse calculation, the 6-character composite QRA locator is split into its two 3-character latitude and longitude codes. Thus, for the example above, we would take the

composite KM64CP and recover K6C and M4P. In fact, these codes are recovered in their inverse order simply to make the calculation easier, thus C6K, and P4M respectively.

Taking the latitude code, the first letter, P, is converted into its position, 15. This is divided by 24 to give the decimal part of its angle, and has 0.5 added to it to place the resultant position in the centre of its rectangle.

The middle digit, in this case 4, is added to the number, and the result multiplied by 24.

The most significant character, in this case M, has its number calculated, 13, which is multiplied by 240 and added to the previous result, giving a final decimal angle.

If this is a latitude, then 90° is subtracted from it, and the result re-converted into its DD.MMSS format. If it was a longitude figure, then 180° would have been subtracted, and then it would have been re-converted.

By adding the 0.5 offset, the resultant angles are now located in the middle of the reference rectangle, and if these angles are then re-inserted into the locator program, exactly the same locator code is produced, making the whole process fully reciprocal.

The encoding process is:

Longitude $32^\circ 12' 30''$ E, in decimal form	: 32.2083... (recurring)
Add 180 to give its pseudo angle	: 212.2083...
Convert 5' to decimal	: 0.083...
Divide pseudo angle by 5' giving the position number	: 2,546.5
Divide by 240 to give the field	: 10.610416...
Field number is the integer portion, 10, which is the letter	: K
The fraction part is multiplied by 240 and divided by 24	: 6.10416...
The integer portion is 6 and is used directly	: 6
The fractional part is multiplied by 24	: 2.5
Sub-square is the integer portion, 2, which is the letter	: C

The decoding process is:

Recover the least significant character	: C
Convert to its position	: 2
Add 0.5	: 2.5
Divide by 24	: 0.10416...
Recover the middle digit	: 6
Add to the previous result	: 6.10416...
Multiply by 24 and divide by 240	: 0.610416...
Recover the most significant character	: K
Convert to its position	: 10
Add to the previous result	: 10.610416...
Multiply by 240	: 2,546.5
Multiply by decimal 5', 0.083...	: 212.2083...
Subtract 180°	: 32.2083...
Convert to dd.mmss	: 32° 12' 30''

Note that normally a recurring decimal digit is indicated with a dot above it, as this is not possible in this word processor, instead I have placed a ... after the digit.

For the latitude calculation and its inverse, the constants are 90° and $2.5'$.