ALGORITHMS USED BY ELECTRONIC LOGBOOKS FOR THE COMPUTATION OF DEW POINT TEMPERATURE

Name of e-logbook	TurboWin
Agency	KNMI
Country	Netherlands
Contact point (name, e-mail)	martin.stam@knmi.nl
Web site for the software	http://www.knmi.nl/turbowin/
Version number of e-logbook	All versions
Date of version of e-logbook	-
Version number of algorithm	-
Date of version of algorithm	-
Name of algorithm	`Bereken_Dauwpunt_en_RV`

Variables used as input for the computation of Dew Point Temperature		
Name	**Units**	**Precision required**
num_Tdry_total	degr C	0.1 degr
num_Twet_total	degr C	0.1 degr
ijs_wetbulb	boolean	
exposure	string	

Variables returned by the algorithm		
Name	**Units**	**Resulted precision**
num_rv	%	% rounded
num_Tdew	degr C	0.1 degr

Pseudocode of the algorithm[1]

```
#define c1                          6.112       // (up to and including version 1.
#define c2                          17.62       // (up to and including version 1.
#define c3                          243.12      // (up to and including version 1.
#define c4                          0.000646    // psychrom. const. in 1/K
#define c5                          5419        // K
#define c1_ijs                      6.112       // (up to and including version 1.
#define c2_ijs                      22.46       // (up to and including version 1.
#define c3_ijs                      272.62      // (up to and including version 1.
#define FP                          1.004719    // (up to and including version 1.
#define SP                          1013.25
#define t0                          273.15
#define eps                         1
#define A_SLING_ICE                 0.000575
#define A_SCREEN_ICE                0.000711    // (0.720 / 1013.25)
#define A_SCREEN_WATER              0.000789    // (0.799 / 1013.25)



Vochtigheid::Vochtigheid()
{
}


float Vochtigheid::SVP(float temp)
```

[1] : Possibly based on C++ alike syntax whenever possible; otherwise using original source language that was used

```
{
	float ew = c1 * exp(c2 * temp / (c3 + temp));
	return ew;
}


float Vochtigheid::SVP_ijs(float temp)
{
	float ei = c1_ijs * exp(c2_ijs * temp / (c3_ijs + temp));
	return ei;
}


float Vochtigheid::Bereken_Dauwpunt(float num_Rh, float num_Tdry_total)
{
  float num_Tdew;

  if (num_Rh == 100)                                  // 100 % r.v.
     num_Tdew = num_Tdry_total;
  else
	num_Tdew = c3 / (-1 + c2 / (log((num_Rh / 100) * SVP(num_Tdry_total) / c1)));

	return num_Tdew;
}


float Vochtigheid::fi_intern(float x, float e)
{
	float y = SVP(x);
	float fi = (y - e) / (c5 * y / pow(x + t0, 2) + c4 * SP);
	return fi;
}


float Vochtigheid::Bereken_NatteBol(int num_Rh, float num_Tdry_total)
{
	float f;
	float x = num_Tdry_total;                         // setting of start values
	float e = SVP(num_Tdry_total) * num_Rh / 100;

	do
	{
		f = fi_intern(x , e);
		x = x - f;
		e = e + c4 * SP * f;
	}while (fabs(f) >= eps);

	float num_Twet_total = x;
	return num_Twet_total;
}


void Vochtigheid::Bereken_Dauwpunt_en_RV(float num_Tdry_total, float num_Twet_total, float
ijs_wetbulb, string exposure)
{
  float A_water;
  float A_ice;


  /* after discussion with ‘dewpoint calculation specialists’ of the UK Met Office:
  /* “UK marine screens do not work with yes or no iced wet bulb for dew point calculation
  /* If Twet < 0 degr C -> for marine screens always use the formula above ice. */
  /* Regardless the observer reported yes or no iced wet bulb!” */
  /* (see also tables in Marine Observers handbook) */
```

```
  if ((exposure == SCREEN) && (num_Twet_total < 0.0))
     ijs_wetbulb = TRUE;                    // to be sure correct formula for marine scre


  /* for sling psychrometers: if Twet < 0 C it is possible to use the formulas above ice c
  /* The yes or no iced wet bulb reporting by the observer makes the difference */
  /* (see also Handleiding voor het verrichten van meteorologische waarnemingen op zee */


  /* WATER */
  if (ijs_wetbulb == FALSE)
        {
     // determine psychrometer coefficient (A)
     if (exposure == SCREEN)               // num_Twet_total here always >= 0 !!
        A_water = A_SCREEN_WATER;
     else
        A_water = 0.000653 * (1 + 0.000944 * num_Twet_total);


              num_rv = (SVP(num_Twet_total) - A_water * SP * (num_Tdry_total - num_Twet_t


              // relative humidity: between 1 - 100 %
              if (num_rv >= 0.01 && num_rv <= 1.0)
              {
                     float term = log(num_rv) + (c2 * num_Tdry_total / (c3 + num_Tdry_to
                     num_Tdew = c3 * term / (c2 - term);
              }
              else // dus (num_rv < 0.01 of num_rv > 1.0)
              {
                     num_Tdew = MAXINT;                              // invalid
                     num_rv = MAXINT;                                // invalid
              }
        } //if (ijs_wetbulb == FALSE)

  /* ICE */
  else if (ijs_wetbulb == TRUE)
        {
     // determine psychrometer coefficient (A)
     if (exposure == SCREEN)
        A_ice = A_SCREEN_ICE;
     else
        A_ice = A_SLING_ICE;


              num_rv = (SVP_ijs(num_Twet_total) - A_ice * SP * (num_Tdry_total - num_Twet

              if (num_rv >= 0.01 && num_rv <= 1.0)
              {
                     float term = log(num_rv) + (c2 * num_Tdry_total / (c3 + num_Tdry_to
                     num_Tdew = c3 * term / (c2 - term);
              }
              else
              {
                     num_Tdew = MAXINT;                              // invalid
                     num_rv = MAXINT;                                // invalid
              }
        } // else
}
```