Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Northern Watch

Arctic Site Meteorology

J.L. Forand
DRDC Valcartier

Defence R&D Canada – Valcartier
Technical Report
DRDC Valcartier TR 2010-003
September 2010

Canada

Northern Watch

Arctic Site Meteorology

J. Luc Forand
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Report
DRDC Valcartier TR 2010-003
September 2010

Principal Author

J.Luc Forand

Defence Scientist

Approved by

A. Jouan

Head Spectral and Geospatial Section

Approved for release by

C. Carrier

Chair, Document Review Panel

Northern Watch (15aa)

Abstract

The principal objective of the Northern Watch Project is to install and test the performance of sensors at the Arctic site near Gascoyne Inlet. To accomplish this task the meteorological conditions to be expected at the site need to be determined. As no data has yet been obtained at the site, historical data from the nearby Environment Canada station at Resolute Bay was used, along with a solar insolation model.

This document shows the types of atmospheric conditions that are to be expected at the Arctic site based on the historical meteorological data sets and the insolation model. It shows that, if the Gascoyne Inlet installation is to operate year-round, equipment and personnel must withstand quite rigorous conditions. Equipment must be able to support winds gusting to 40 m/s and temperatures between -45 and 13^{o}C. It also shows that the sky is clear 17 % of the time, cloudy 43 % of the time, foggy 18 % of the time, and that there is likely to be some kind of precipitation 30 % of the time. Finally, using historical solar irradiance data from Barrow, Alaska, and an insolation model, it shows that for equipment requiring 100 A-h at 12 V, a 1 square meter 100 % efficient solar panel could not provide more than 190 days of power and that alternate power would need to be available for at least 175 days or about half the year.

Résumé

L'objectif principal du projet « Northern Watch » est d'installer et de tester la performance des capteurs sur un site arctique proche de Gascoyne Inlet. Pour accomplir cette tâche, nous avons besoin de bien déterminer les conditions météorologiques du site. Comme aucune donnée n'a encore été prise sur ce site, des données historiques de la station de Resolute Bay ont été obtenues d'Environnement Canada et nous avons employé un modèle d'ensoleillement.

Ce document démontre les types de conditions atmosphériques que nous devrions prévoir au site arctique à partir de ces données historiques et du modèle d'ensoleillement. Il démontre que l'installation d'équipement ou de personnel pendant un an est très exigeante. L'équipement doit être capable de subir des rafales jusqu'à 40 m/s et des températures entre -45 et 13 ^{o}C. Il démontre aussi que le ciel est clair 17 % du temps, nuageux 43 % du temps, brumeux 18 % du temps, et qu'il y a une bonne chance de précipitations 30 % du temps. Finalement, en employant les données d'ensoleillement historique de Barrow Alaska, et un modèle de l'ensoleillement, on démontre que pour l'équipement qui a besoin de 100 A-h à 12 V, un panneau solaire d'un mètre carré d'un efficacité de 100 % ne peut fournir suffisamment de courant que pour 190 jours et qu'un autre source de courant sera nécessaire pendant au moins 175 jours ou à peu près la moitié de l'année.

This page intentionally left blank.

Executive summary

Northern Watch: Arctic Site Meteorology

J. Luc Forand; DRDC Valcartier TR 2010-003; Defence R&D Canada – Valcartier

Background: The principal objective of the Northern Watch Project is to install and test the performance of sensors at the Arctic site near Gascoyne Inlet. To accomplish this task the meteorological conditions to be expected at the site need to be determined. As no data has yet been obtained at the site, historical data from the nearby Environment Canada station at Resolute Bay and a solar insolation model were used.

Results: This document shows the types of atmospheric conditions that are to be expected at the Arctic site near Gascoyne Inlet based on an analysis of this information. As expected, it shows that for year-round operation, the operational requirements for all shelter and equipment to be installed or used at the site, and the conditions under which personnel would have to work, are quite rigorous.

Sommaire

Northern Watch: Arctic Site Meteorology

J. Luc Forand; DRDC Valcartier TR 2010-003; R et D pour la défense Canada – Valcartier

Contexte: L'objectif principal du projet « Northern Watch » est d'installer et de tester la performance des capteurs sur un site arctique pas loin de Gascoyne Inlet. Pour accomplir cette tâche nous avons besoin de prévoir les conditions météorologiques sur le site. Comme aucune donnée n'a encore été prise sur ce site, les données historiques d'Environnement Canada à la station de Resolute Bay et un modèle d'ensoleillement ont été utilisées.

Résultats: Ce document décrit les conditions atmosphériques prévues sur le site arctique et se base sur l'analyse de ces informations. Comme prévu, on démontre que pour une opération d'un an, les besoins opérationnels d'une habitation et l'équipement installé sur le site, ainsi que les conditions dans lesquelles le personnel aura besoin de travailler, sont très exigeantes.

Table of contents

List of figures

List of tables

Acknowledgements

The author would like to thank Environment Canada and the National Oceanographic and Atmospheric Administration for making so much of their archived material easily available through the internet. In addition, the author would like to thank NOAA and particularly Jean Meeus and Greg Pelletier for providing their astronomical codes for calculating the position of the sun in the sky for any time of the year.

This page intentionally left blank.

1 Introduction

One of the objectives of the Northern Watch Project is to characterize the conditions which equipment, buildings and personnel placed in an Arctic environment would have to withstand. In particular, the project will test the performance of a suite of sensors at the Arctic site near Gascoyne Inlet on the south shore of Devon Island. The site of the entire current infrastructure is commonly referred to as the Base Camp (Ref. 1). As no known data has ever been obtained for the site, historical data from Environment Canada's (EC) nearby station at Resolute Bay was chosen as being representative. Resolute Bay is a community on the south shore of Cornwallis Island located about 110 km due west of the Base Camp. Figure 1, taken from Ref. 2, shows the meteorological station at Resolute Bay and the Base Camp on the shores of Barrow Strait.

Figure 1: The house in the top-left corner is the meteorological station at Resolute Bay and the house near the top-right is the Base Camp.

Chapter 2 presents the two data sets obtained from Environment Canada (EC), the monthly statistics for the entire year, and the hourly statistics for periods of operation during the summer. Chapter 3 presents solar radiation estimates and related issues, and Chapter 4 provides some conclusions.

This work was done between July 2007 and December 2009 as part of the Northern Watch Technological Demonstration Program.

2 Environment Canada Datasets for Resolute Bay

Two types of datasets were obtained for Resolute Bay from the meteorological archives belonging to Environment Canada (EC) and can easily be accessed through their website [Ref. 3]. The first dataset provides monthly statistics and the second dataset provides hourly statistics.

2.1 Monthly Statistics

The monthly statistics obtained from EC cover the period from January 1963 to December 2008 and includes the following information: Year, Month, Mean Maximum Temperature, Mean Temperature, Mean Minimum Temperature, Extreme Maximum Temperature, Extreme Minimum Temperature, Total Rain, Total Snow, Total Precipitation, Snow on Ground Last Day, Direction of Maximum Gust, and Speed of Maximum Gust. Table 1 shows the minimum and maximum values for most of these parameters along with their 5% percentile and 95% percentile values.

Table 1: Minimum, maximum, .05 percentile and 0.95 percentile monthly values for Resolute Bay from 1963 to 2008

	Min.	Max.	0.05 Percentile	Median	0.95 Percentile
Mean Temp. (C)	-40.9	7.4	-34.2	-18.4	3.9
Mean Min. Temp. (C)	-44.1	3.7	-37.6	-21.8	1.3
Mean Max. Temp. (C)	-37.6	11.0	-31.0	-14.7	6.6
Extreme Min. Temp. (C)	-52.2	0.1	-45.0	-31.4	-1.7
Extreme Max. Temp. (C)	-27.4	18.5	-22.2	-4.2	13.4
Total Rain (mm)	0.0	65.2	0.0	0.0	25.6
Total Snow (cm)	0.0	54.4	0.3	6.2	25.7
Total Precipitation (mm)	0.0	78.9	0.9	7.6	39.4
Speed of Max. Gust (m/s)	10.8	43.9	15.6	21.4	30.3

Table 1 shows that over the 45 year period between 1963 and 2008 that 90% of the time the temperature is between -45 and 13.4 degrees Celsius with an absolute minimum temperature of -52.2 ^{o}C and an absolute maximum temperature of 18.5 ^{o}C. Likewise, maximum wind gusts are expected to be less than 30.3 m/s (109 km/h) 95% of the time with an absolute maximum of 43.9 m/s (158 km/h). Finally, total rain and snow is expected to be less than 26 mm/month and 26 cm/month, respectively; however the median amounts are much less at 0.0 mm/month of rain and 6.2 cm/month of snow.

While Table 1 does provide information about the extremes of temperature and wind and the amounts of rain and snow that any equipment would have to survive, it doesn't provide a good idea of how they will vary during the year. Figures 2 to 4 graphically show how some of these parameters have varied over the ten year period between 1999 and 2009. Figure 2 shows that summer conditions (extreme maximum temperatures above 0 ^{o}C) exist for about 5 months of the year, and that winter conditions (extreme minimum temperatures below -30 ^{o}C) exist for about 6 months. Figure 3 shows that wind gusts are rarely greater then 30 m/s and that these gusts tend to come from a northerly direction. Finally, Figure 4 shows that rain falls essentially only during a 3 month period in the summer, and that while it can snow at any time during the year, most of the snow tends to arrive during a three month period in the fall.

2.2 Hourly Statistics

The hourly statistics obtained from EC cover the period from 1 July to 30 September for the years 2002 to 2009. These time periods were chosen to cover the period when personnel are expected to occupy the site. This data includes the following information: Year, Month, Date, Time, Air Temperature, Dew-Point Temperature, Relative Humidity, Wind Speed and Direction, Visibility, Atmospheric Pressure, and Weather Observations. The Weather Observations describe the meteorological conditions in text format (clear, cloudy, rain, fog, etc.). These data were then put together to provide certain statistics over this 8-year period. These statistics include the hourly mean, minimum, maximum, median, 5% percentile and 95% percentile values for the air temperature, water vapour pressure, relative humidity, wind speed, visibility, and atmospheric pressure. It also determines statistics about the meteorological conditions from the Weather Observations. The twenty-three conditions can be organized into 6 different types. These types are Cloudy, Clear, Rain, Snow, and Fog. While the cloudy and clear types are independent, fog and precipitation are often given together. For example, fog and drizzle often occur at the same time. The Cloudy type includes the Cloudy and Mostly Cloudy conditions, and the Clear type includes the Clear and Mostly Clear conditions. The Fog type includes the Fog and Freezing Fog conditions, and the Rain type includes the Rain, Rain Showers, Drizzle, and Freezing Drizzle conditions, and the Snow type includes the Snow, Snow Grains, and Snow Shower conditions. As the other 10 conditions generally occur much less than 1% of the time, they have not been included in any of the types. Table 2 shows the frequency of these types and conditions.

Table 2: Frequency of meteorological types and conditions for Resolute Bay between 1 July and 30 September between 2002 and 2009

Cloudy	43.0%	**Clear**	17.0%	**Fog**	18.2%
Cloudy	21.3%	Clear	3.1%	Fog	14.9%
Mostly Cloudy	21.7%	Mostly Clear	13.9%	Freezing Fog	3.3%
Rain	15.2%	**Snow**	16.0%	**Other**	1.2%
Rain	5.2%	Snow	12.9%	Blowing Snow	0.9%
Rain Showers	0.8%	Snow Grains	1.6%	Moderate Snow	0.2%
Drizzle	7.5%	Snow Showers	1.5%	Ice Pellets	0.1%
Freezing Drizzle	1.7%				

Table 2 essentially states that it is clear 17% of the time, cloudy 43% of the time, foggy 18% of the time, and that there is likely to be some kind of precipitation 30% of the time. As the total is 110%, this also means that 10% of the time more than 1 type of precipitation is present or it is foggy.

Table 3 shows the minimum and maximum values for most of these parameters along with their median, 5% percentile and 95% percentile values.

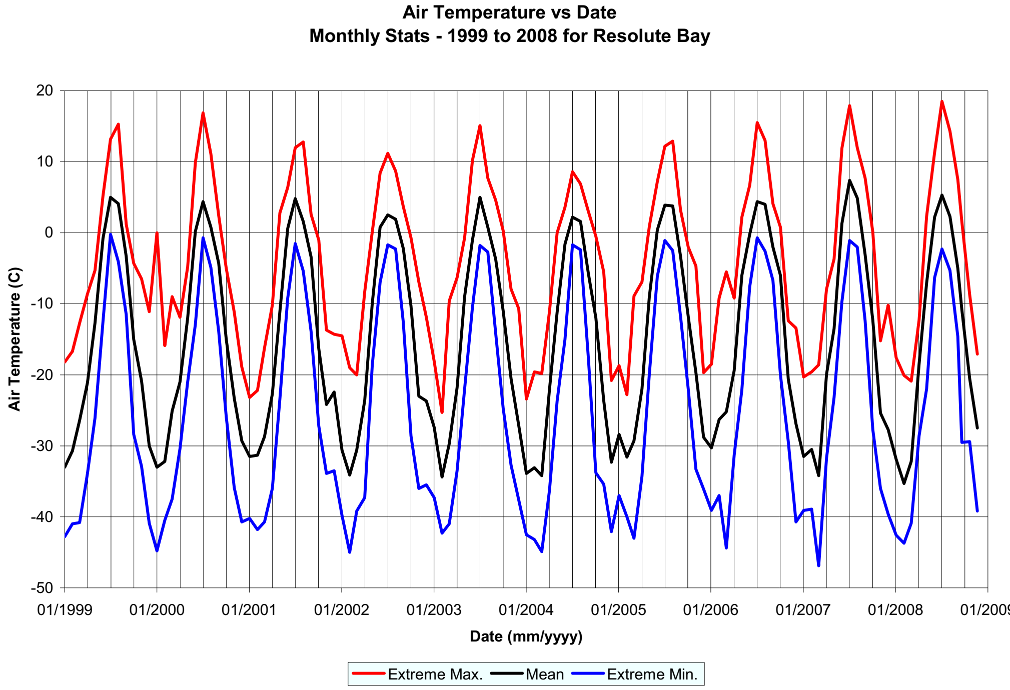

Figure 2: Monthly mean, extreme maximum and extreme minimum air temperatures for Resolute Bay

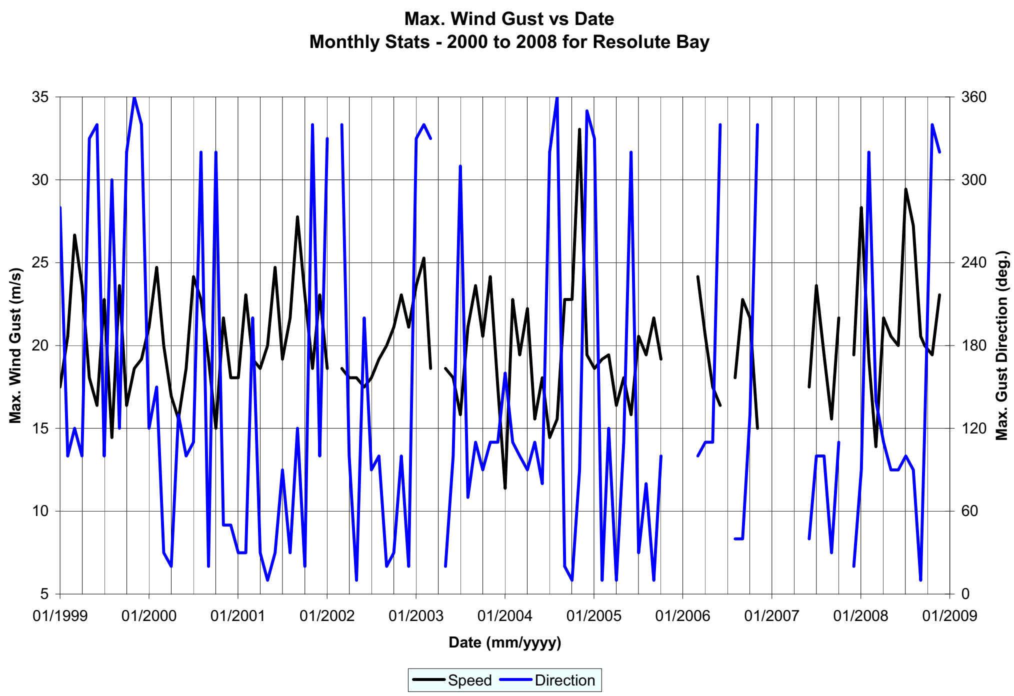

Figure 3: Maximum wind gust speed and direction for Resolute Bay

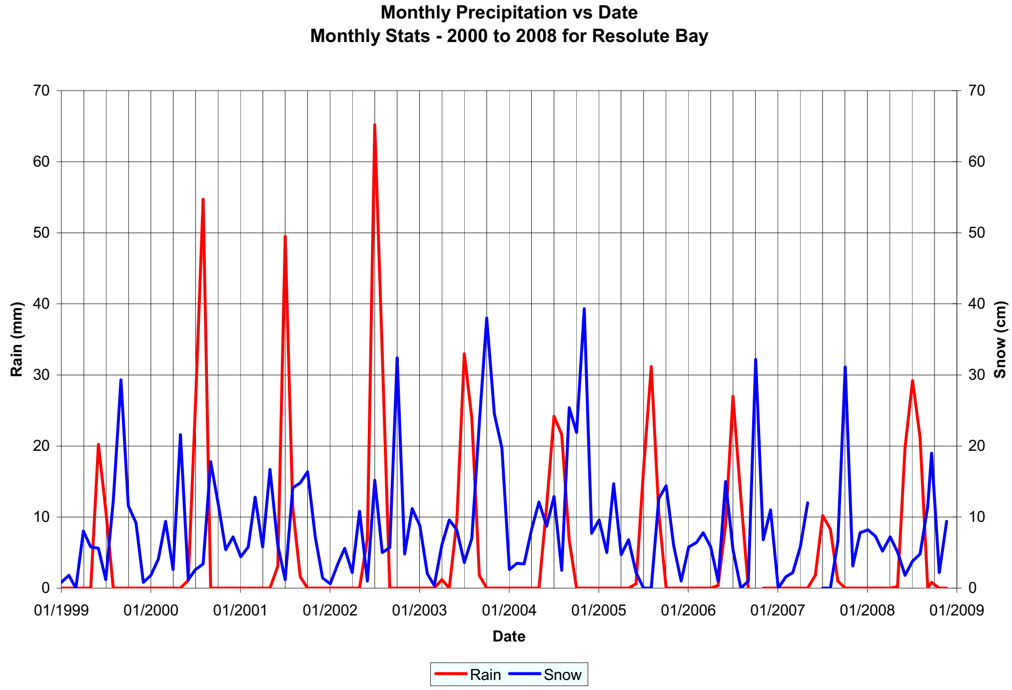

Figure 4: Monthly snow and rain accumulated precipitation for Resolute Bay

Table 3 states that 90% of the time during this time of the year, the air temperature is between -11.6 and 13.2 ^{o}C with a mean temperature between -6.5 and 6.0 ^{o}C. Likewise, the atmospheric pressure is between 98 and 102 kPa with a mean pressure between 100 and 101 kPa, and the water vapour pressure is between 0.21 and 1.04 kPa with a mean vapour pressure between 0.34 and 0.75 kPa, such that it is always less than 1% of the atmospheric pressure. The corresponding relative humidity is between 53 and 100% with a mean relative humidity between 79 and 93%, and the visibility is between 0.2 and 24 km with a mean visibility between 12 and 22 km. Finally, the wind speed is between 0 and 16.4 m/s with a mean wind speed between 3.5 and 7.7 m/s. Comparing these wind speeds with those of Table 1 would seem to indicate that wind speeds are stronger during other times of the year than during the summer.

While Table 3 does provide information about the range that their values can have, it doesn't provide a good idea of how they vary during the period and it isn't able to provide any information about the wind direction.

Figure 5 shows the mean, minimum and maximum temperatures observed during this time period. It shows that the temperature can vary between -16 and +16 ^{o}C with temperatures varying between 0 and 12 ^{o}C until about 10 August when the temperatures start dropping about 1 degree/day. As will be shown in Chapter 3, this corresponds with the end of 24 hours of sunlight and the beginning of the arctic night.

Table 3: Minimum, maximum, 0 .05 percentile and 0.95 percentile hourly values for Resolute Bay from 2002 to 2009 between 1 July and 30 September

	Min.	Max.	0.05 Percentile	Median	0.95 Percentile
Mean Air Temp. (C)	-8.5	9.7	-6.5	2.6	5.9
Min. Air Temp. (C)	-17.1	3.9	-11.6	-1.0	1.5
Max. Air Temp. (C)	-4.5	17.9	-2.4	6.6	13.2
Mean Pressure (kPa)	99.6	100.8	99.8	100.2	100.7
Min. Pressure (kPa)	97.6	100.2	98.3	99.2	100.0
Max. Pressure (kPa)	100.3	102.3	100.6	101.1	101.8
Mean Vap. Press. (kPa)	0.30	0.81	0.34	0.64	0.75
Min. Vap. Press. (kPa)	0.12	0.73	0.21	0.50	0.61
Max. Vap. Press. (kPa)	0.40	1.43	0.47	0.79	1.04
Mean Rel. Humidity (%)	65.2	96.6	78.9	88.5	93.4
Min. Rel. Humidity (%)	26.0	90.0	53.0	74.0	85.0
Max. Rel. Humidity (%)	84.0	100.0	95.0	100.0	100.0
Mean Visibility (km)	7.8	24.1	12.1	17.6	21.5
Min. Visibility (km)	0.0	24.1	0.2	0.8	12.9
Max. Visibility (km)	24.1	24.1	24.1	24.1	24.1
Mean Wind Speed (m/s)	1.7	10.4	3.5	5.5	7.7
Min. Wind Speed (m/s)	0.0	6.1	0.0	1.7	3.6
Max. Wind Speed (m/s)	3.1	25.8	7.2	10.8	16.4

Figure 6 shows mean, minimum and maximum atmospheric pressure observed during this time period. It shows that the atmospheric pressure varies between 98 and 102 kPa with a mean value of approximately 100.5 kPa.

Figure 7 shows the mean, minimum and maximum water vapour pressure observed during this time period. It shows that the vapour pressure varies between 0.1 and 1.3 kPa with pressures varying between 0.5 and 1.0 kPa until about 10 August when, as seen for the air temperature, the pressures (mean, minimum and maximum) start falling by about 0.008 kPa/day. This shows, as expected, that as the temperature starts falling the amount of water vapour also begins to fall.

Figure 8 shows the mean, minimum and maximum visibility observed during this time period. It shows that while the visibility can be very poor, it can also be very good at any time during the period with the average visibility varying between 15 and 20 km.

Figure 9 shows the mean, minimum and maximum wind speed observed during this time period. It shows that wind speed can exceed 16 m/s, but that the average wind speeds are between 4 and 8 m/s throughout the entire period.

Figure 10 shows the probability (percentage) that the wind comes from various directions during this time period. It shows that the wind mostly comes from the North-North-West and the South-East but very rarely from the South-West.

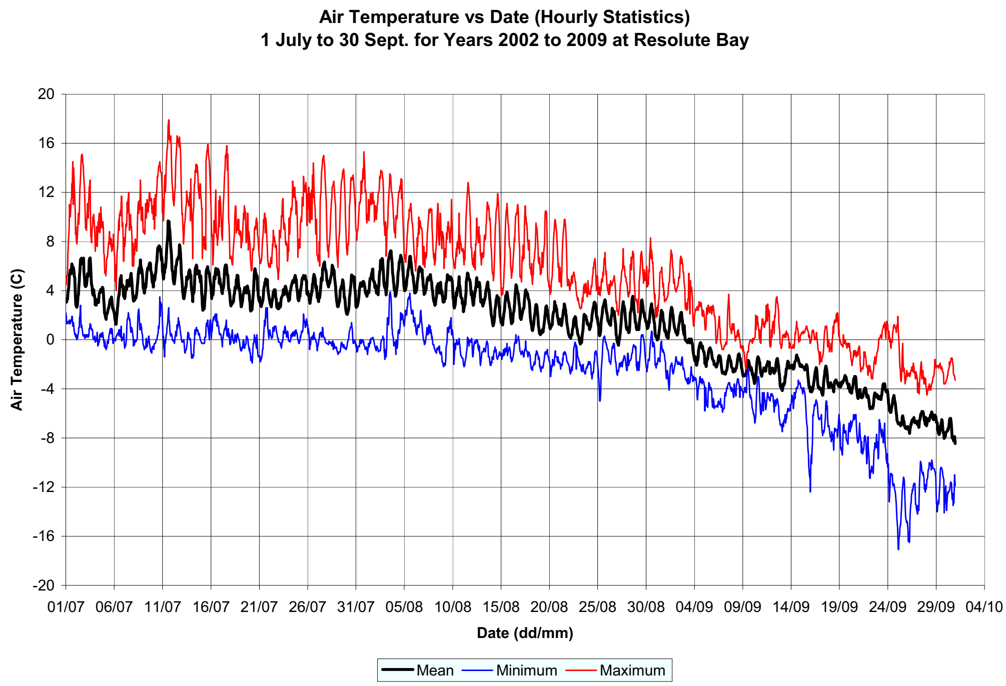

Figure 5: Mean, minimum and maximum air temperatures for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

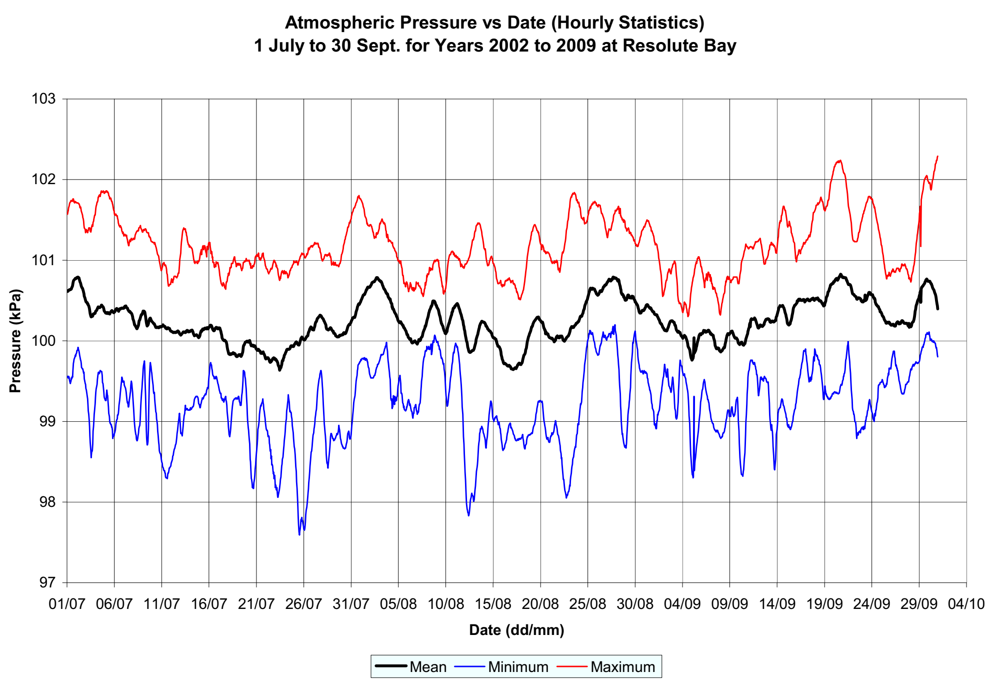

Figure 6: Mean, minimum and maximum atmospheric pressure for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

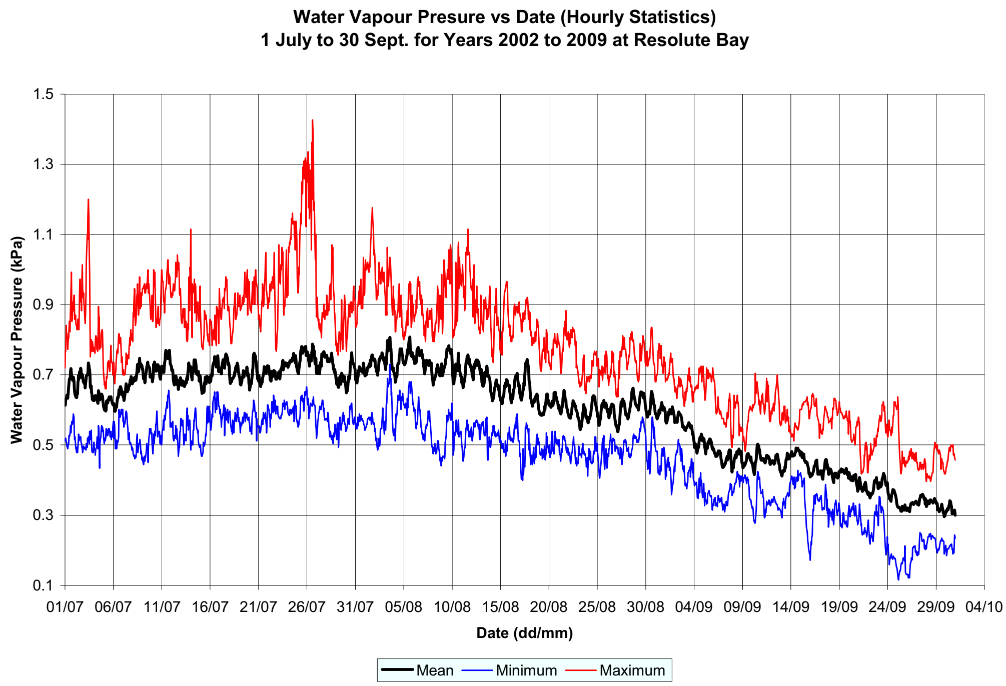

Figure 7: Mean, minimum and maximum water vapour pressure for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

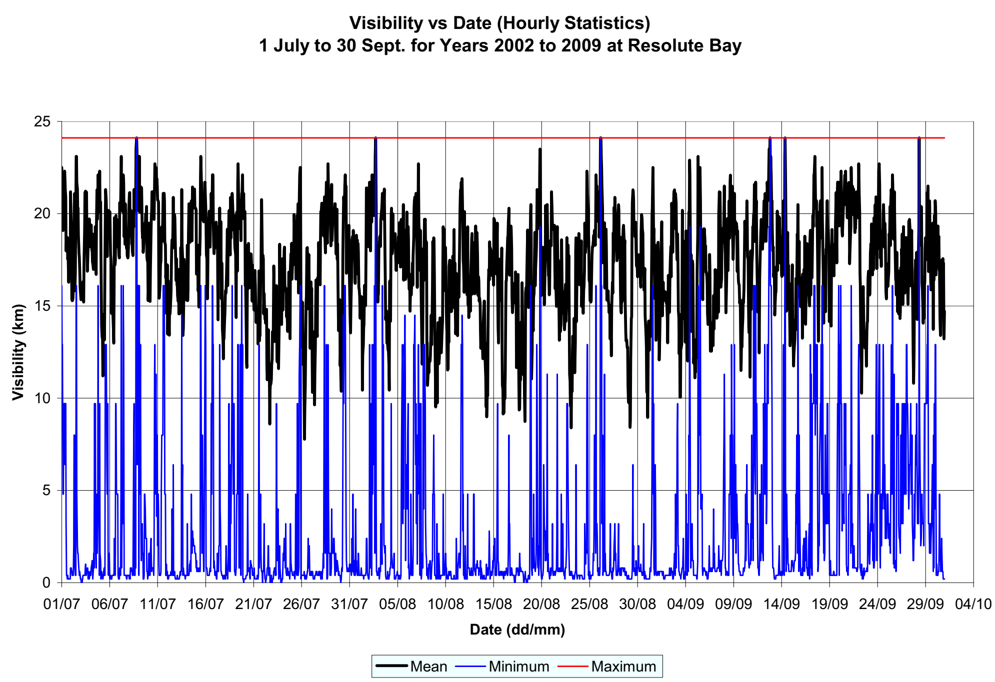

Figure 8: Mean, minimum and maximum visibility for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

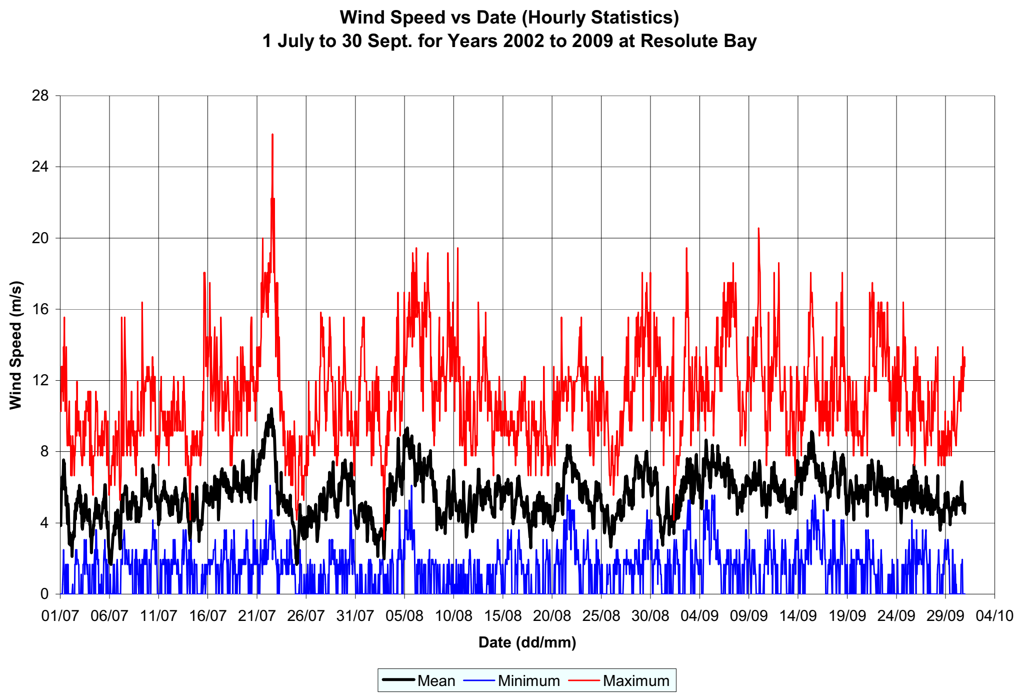

Figure 9: Mean, minimum and maximum wind speed for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

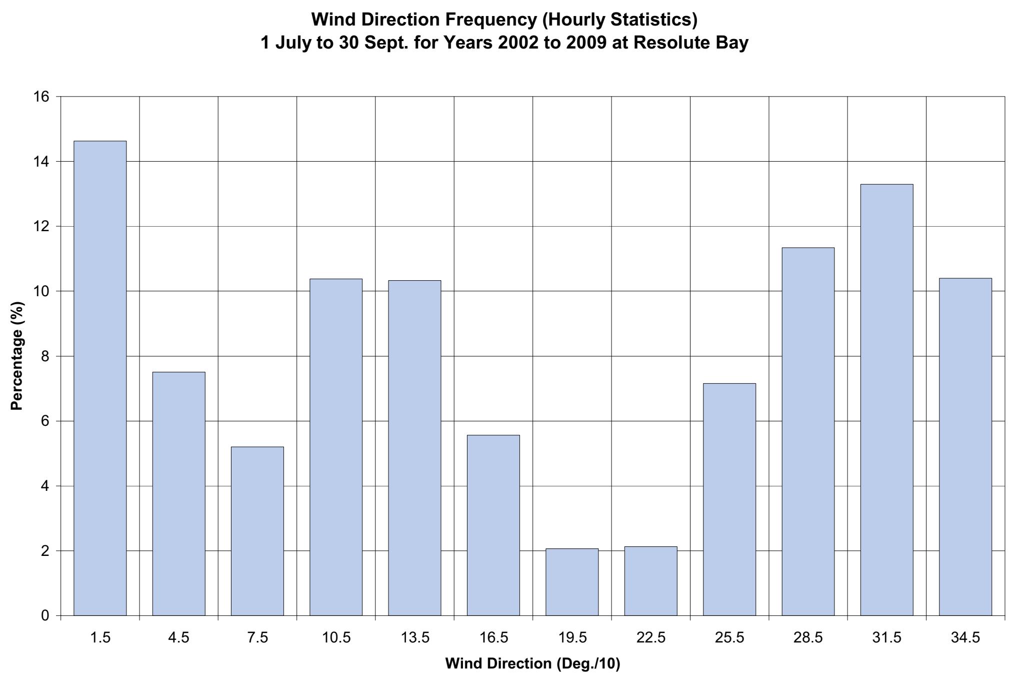

Figure 10: Percentage that the wind comes from a certain direction for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

Figure 11 shows the probability of having Clear type weather during this time period. It shows that conditions seem to be a bit clearer early in July than later in the summer, but this may not be significant as only 8 years have been used in the analysis. The stair step nature of figures 11 to 15 is due to the fact that only 8 years were used, such that each step equals 0.125 (1/8) probability.

Figure 12 shows the probability of having Cloud type weather during this time period. It shows that the chances of having cloudy conditions are the same during the entire period.

Figure 13 shows the probability of having Fog type weather during this time period. It shows that conditions are most foggy during late July and August, less foggy during early July, and the least foggy in September.

Figure 14 shows the probability of having Rain type weather during this time period. It shows that the occurrence of Rain is fairly stable until about Aug. 20th after which it starts dropping as the mean temperature drops.

Figure 15 shows the probability of having Snow type weather during this time period. It shows that the occurrence of Snow is about 10% until about Aug. 15th when it starts to increase quite rapidly and it snows 60% of the time by the end of September. This ties in very well with the remark in Chapter 2.1 when it was noticed that most snow arrives in the fall.

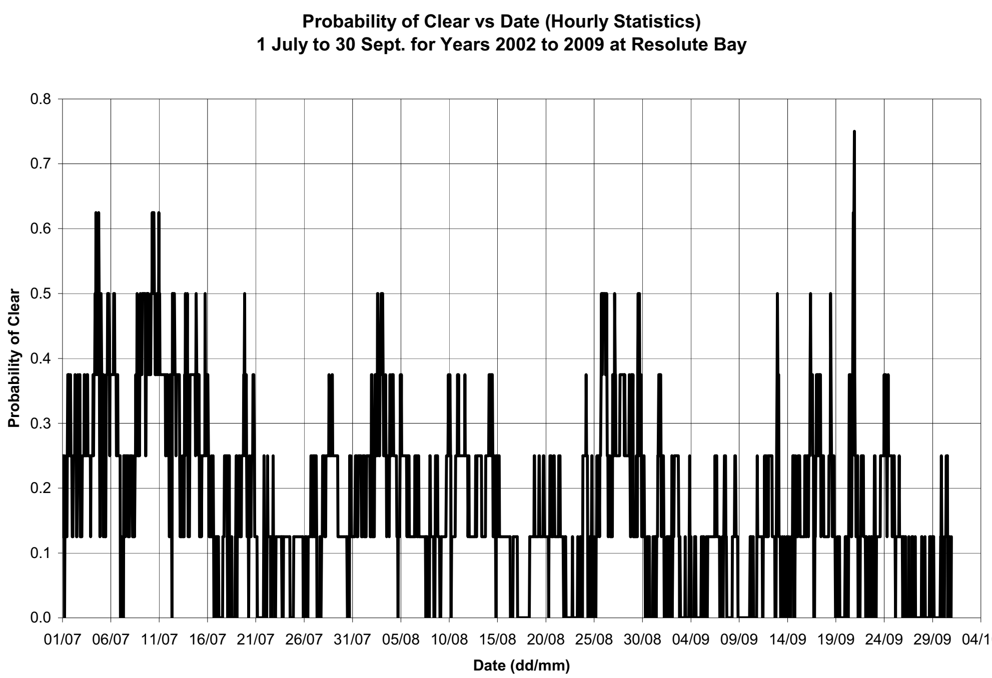

Figure 11: Probability that the weather is Clear for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

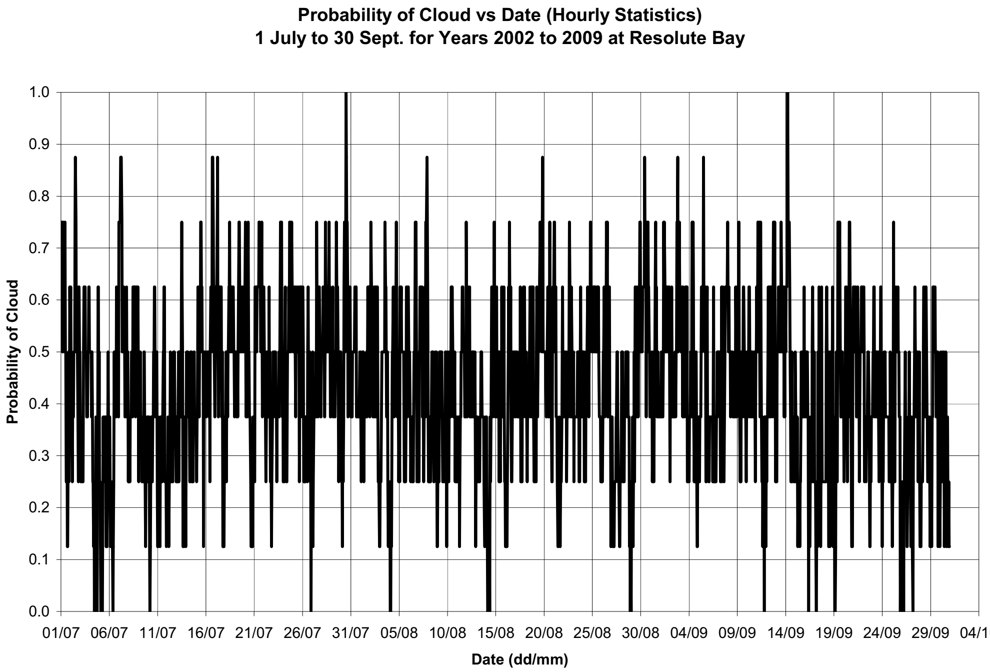

Figure 12: Probability that the weather is Cloudy for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

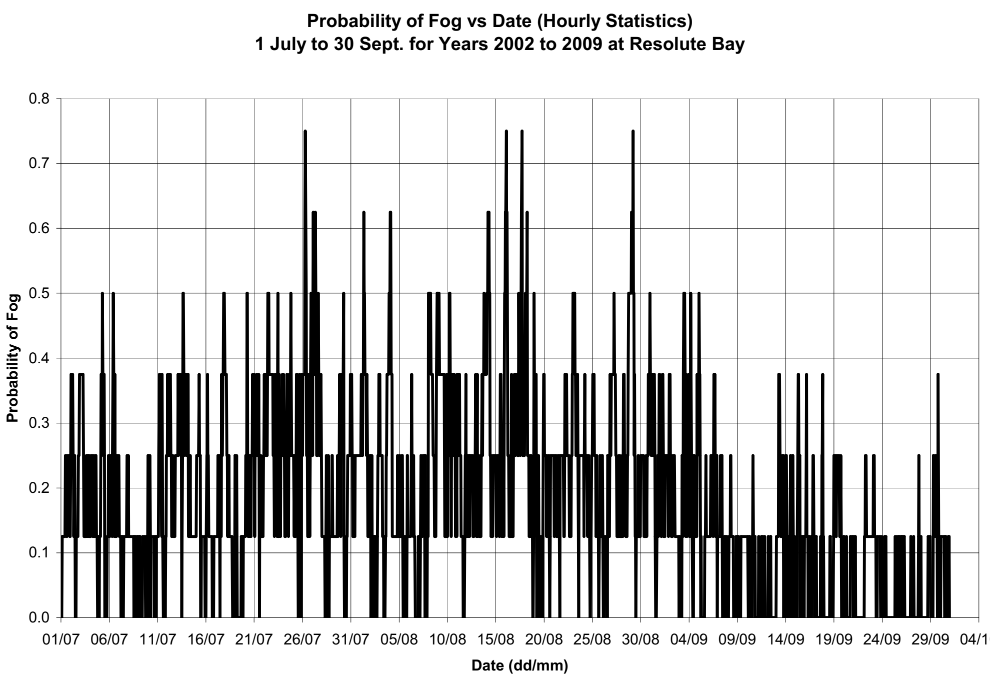

Figure 13: Probability that the weather is Foggy for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

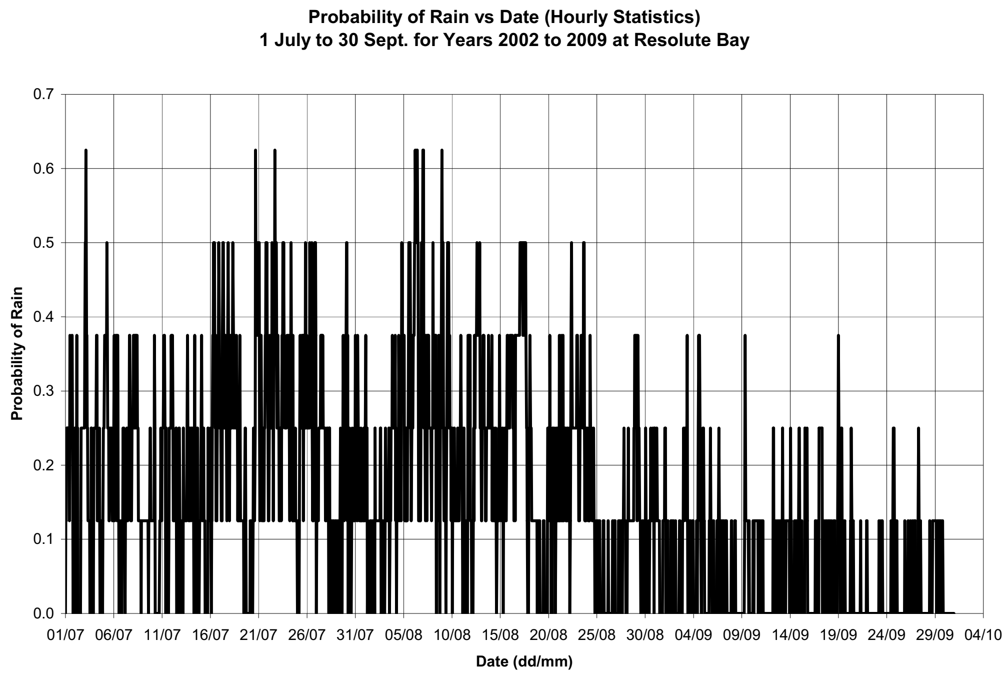

Figure 14: Probability that the weather is Rainy for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

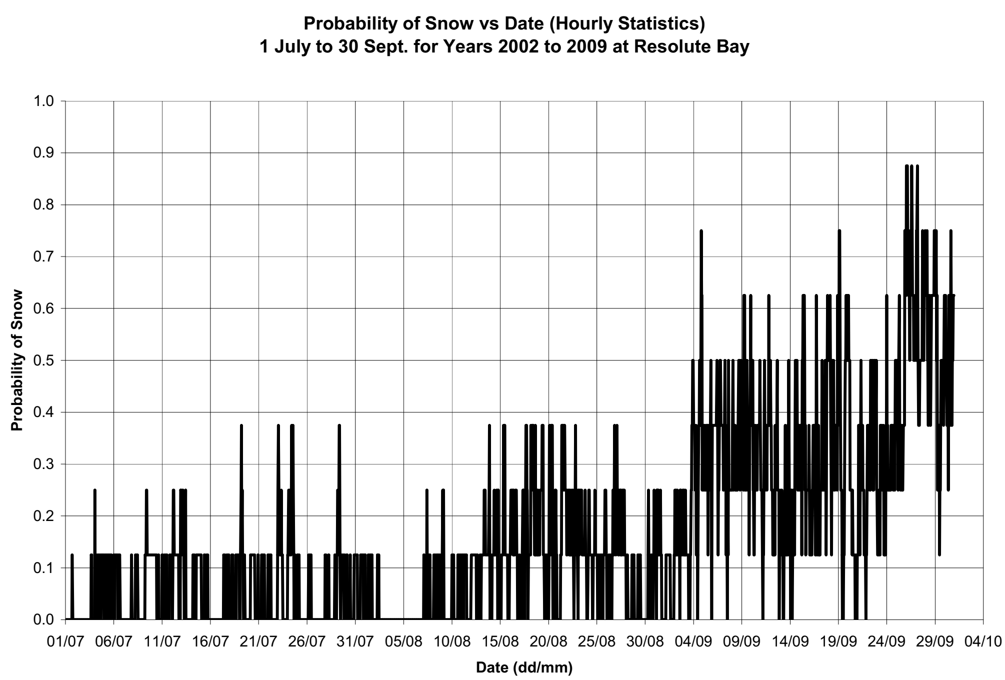

Figure 15: Probability that the weather is Snowy for each hour between 1 July and 30 Sept. for the years 2002 to 2009 at Resolute Bay

3 Solar Radiation Estimates

The amount of solar radiation present at the Arctic site is of importance as it provides information as to the length of the day and nigh, the maximum elevation of the sun, and the amount of energy that could be available for solar panels. As no data was obtained for Resolute Bay from Environment Canada's online site [Ref. 3], it was decided to determine these parameters, and others, using a model of the sun's position in the sky as a function of Julian day and to validate the model using data obtained for Barrow, Alaska [Ref. 11].

Appendix A shows the functions that were written in Microsoft Visual Basic Editor for use in Microsoft Excel. The first two functions were written by the author while the calculation of local times of sunrise, solar noon, and sunset are based on the calculation procedure by NOAA in the Javascript found at Ref. 4 and Ref. 5. The calculators are based on equations from Astronomical Algorithms by Jean Meeus. NOAA also included atmospheric refraction effects. The sunrise and sunset results are reported by NOAA to be accurate to within +/- 1 minute for locations between +/- 72° latitude, and within ten minutes outside of those latitudes. Thus, as Gascoyne Inlet is near 74^{o} latitude, an accuracy within 10 minutes is expected. This translation from NOAA's Javascript to Excel Visual Basic was tested for selected locations and found to provide results within +/- 1 minute of the original Javascript code. The translation was done by Greg Pelletier [Ref. 6] and the functions have been modified so as to output the results required by the author for this work. The functions have also been simplified and written in standard Visual Basic 6.3 with all variables types defined. All modifications were compared against Greg's results and no discrepancies were observed. The following principal functions are available for use in an Excel worksheet:

dawndusk(mode, lat, lon, year, month, day, timezone, dlstime, solardepression)
- returns local time of dawn or dusk for all latitudes given a solar depression

sunriseset(mode, lat, lon, year, month, day, timezone, dlstime)
- returns local time of sunrise or sunset for all latitudes assuming a solar depression of 0.833 deg.

solarnoon(lat, lon, year, month, day, timezone, dlstime)
- returns local time of solar noon for all latitudes

solarposition(mode, lat, lon, year, month, day, hour, minute, second, timezone, dlstime)
- returns the solar azimuth, zenith or elevation depending on the 'mode' parameter as a function of local time

where the sign convention for the inputs to all functions are:
- positive latitude decimal degrees for northern hemisphere
- negative longitude degrees for western hemisphere
- negative time zone hours for western hemisphere.

3.1 Estimates for Gascoyne Inlet

This section provides figures for the number of hours of sunshine, and the number of hours between civil dawn and dusk throughout the year, the solar elevation and solar irradiance during a day for various times during the year for Gascoyne Inlet. The final figure provides the daily energy in Amp-hrs that a 1 m^2 100% efficient solar panel could provide to a 12V system throughout the year.

Figure 16 shows the number of hours of sunshine per day throughout the year between sunrise and sunset (red curve) and between civil dawn and dusk (blue curve). Civil dawn and dusk occurs when the sun is at an elevation of -6 degrees. The graph shows that the period of 24 hour sunshine lasts from Julian Day 120 to 225 and that some light should be visible from Julian Day 105 to 240. Likewise, the period of 24 hour darkness lasts from Julian Day 311 to 35 but that some light should be visible starting at Julian Day 18 and ending at Julian Day 338. Thus, there is probably only about 45 days of absolute darkness.

Figure 17 shows the elevation of the sun during the day for various times during the year. Only half the year is shown as the function is quite symmetrical about June 21st, the summer solstice. The graph shows that the maximum elevation of the sun is about 39 degrees and that at the summer solstice the elevation of the sun varies between 9 and 39 degrees.

Figure 18 shows the estimated maximum solar irradiance (insolation) at Gascoyne Inlet during the day for nine times during the year. Again only half the year is shown as the function is quite symmetrical about June 21st. The curves for Jan. 01 and Feb. 09 are difficult to see as they overlap one another and because the solar irradiance during these times is zero. The values of solar irradiance or insolation (I) have been determined, for clear conditions, using Equation 1;

$$I = (S_0 / d^2) e^{-cL} \cos Z \,, \tag{1}$$

where S_0 is the Solar Constant, d is the distance of the earth from the sun in astronomical units (a.u.), c is the extinction coefficient, L is the optical path through the atmosphere, and Z is the sun's zenith angle. The optical path, L, is given by

$$L = R_e \sin(A - Z) / sinA \,, \tag{2}$$

where

$$A = \sin^{-1}\left[R_e / (R_e + O_d) \sin Z \right], \tag{3}$$

O_d is the Optical Thickness of the atmosphere, and R_e is the Earth's Radius. A value of of 1366 W/m^2 is taken for the solar constant [Ref. 7, 8]. A value of 6360 km is chosen for the Earth's Radius as Resolute Bay is closer to the pole than the equator {6360 ~ 6357 + (90-75/90)*21} [Ref. 9]. A value of 10 km is used for the optical thickness as this corresponds closely to the height to the tropopause in the arctic [Ref. 10]. A value 0.014 km^{-1} used for the extinction coefficient was determined through a fit of Eq. 1 to insolation data obtained from the meteorological station at Barrow Alaska (Barrow W Post-W Rogers (BRW), of which more details are provided in the next section. If instead we follow the development as given by Ref. 13 and others for Equation 1 and use the optical depth (τ) and optical air mass (m), then Equation 1 becomes

$$I = (S_0 / d^2) e^{-\tau m} \cos Z \,, \tag{4}$$

where the optical air mass is given by *L/10*, and the optical depth is given by *10c* (= 0.14), a value that agrees quite well with those given by Freund [Ref. 13].

Finally, Figure 19 shows the maximum estimated daily current that could be provided by a 100% efficient solar panel to a 12V system throughout the year. It also shows a 6 degree polynomial fit to determine this value for any day during the year.

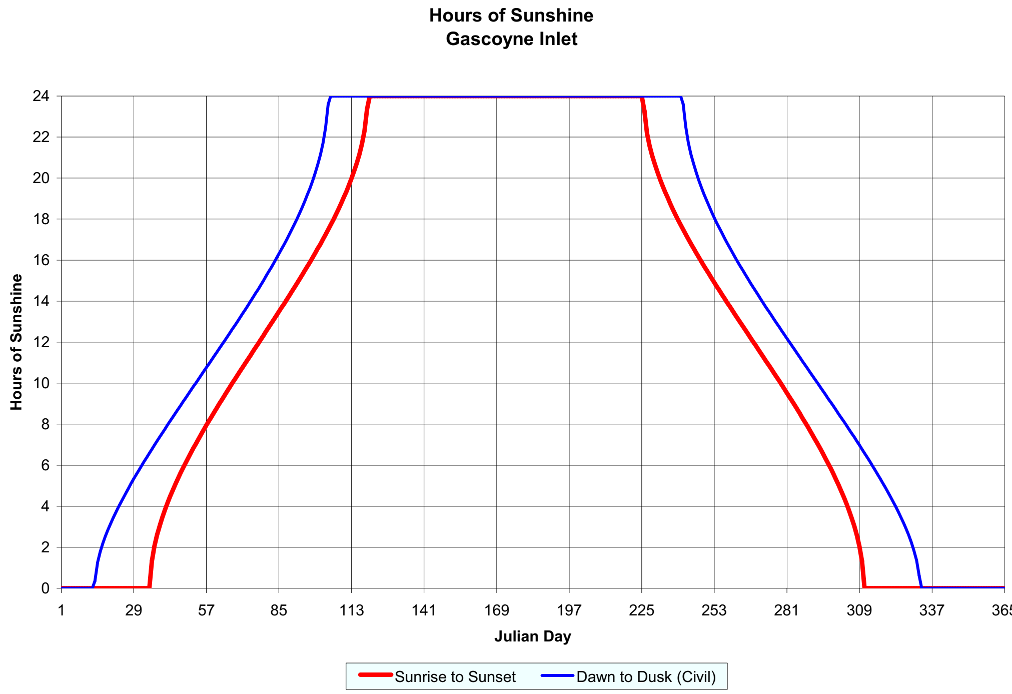

Figure 16: Number of hours of sunshine for Gascoyne Inlet versus Julian Day

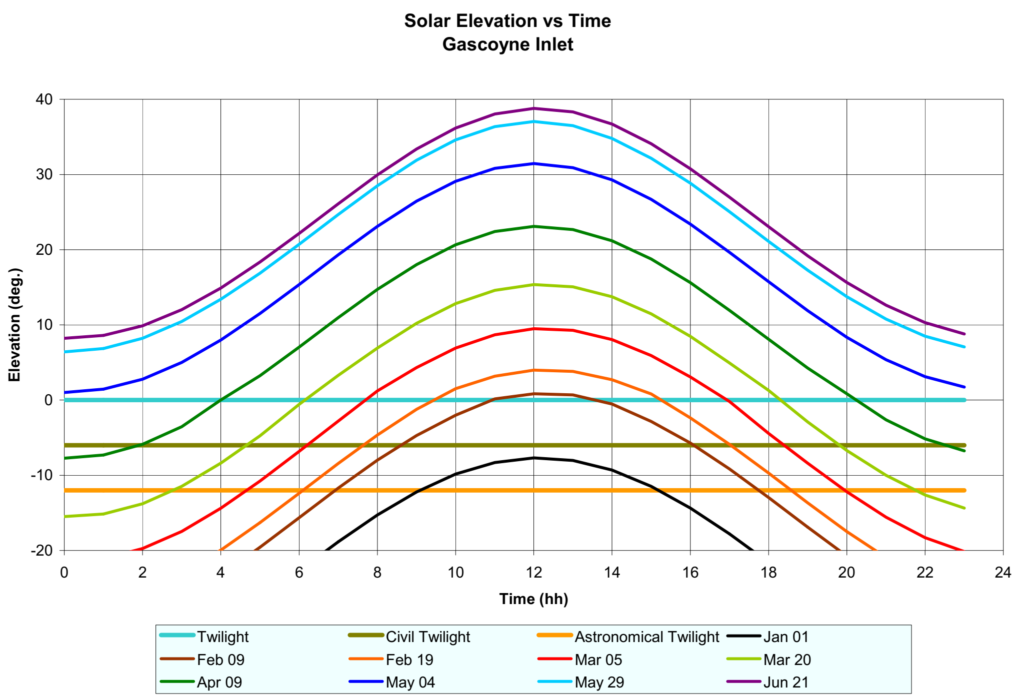

Figure 17: Solar elevation versus time of day for various days during the year. The three vertical lines provide the solar elevations used for twilight, civil twilight and astronomical twilight

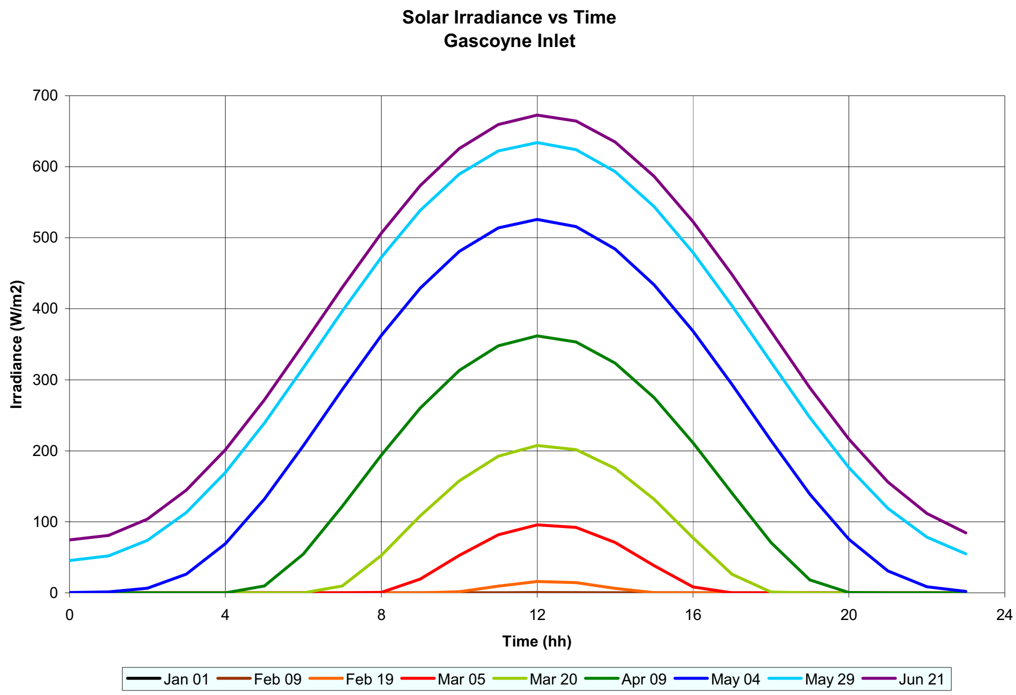

Figure 18: Solar irradiance versus time of day for various days during the year.

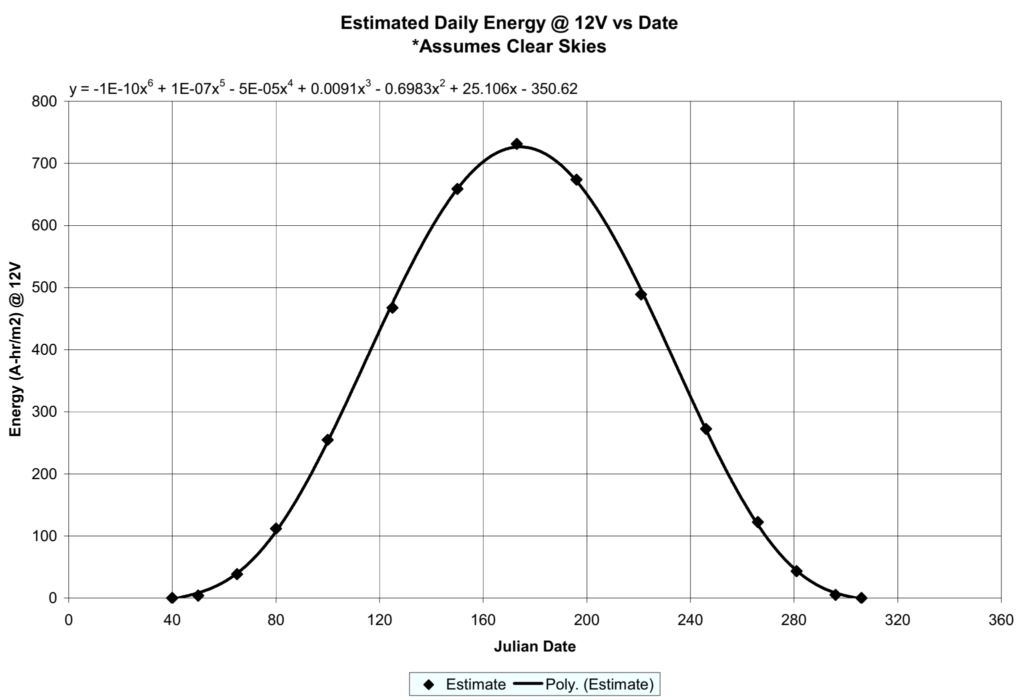

Figure 19: Estimated total daily energy available from a 100% efficient solar panel to a 12V system

3.2 Barrow (BRW) Insolation Data & Model Validation

The Barrow meteorological station is located at 71.2834 degrees North and 156.7815 degrees West, so it is less than 4 degrees further south than the Arctic site at Gascoyne Inlet. Consequently, the maximum solar irradiance or insolation at the two sites should be very similar. The insolation data was obtained from an ftp page [Ref. 12] on NOAA's Earth Sciences Research Laboratory internet site [Ref. 11] and analyzed in an Excel spreadsheet. Figure 20 shows the average and maximum daily power (A-h/m^2) that could be provided by a 1 m^2 100% efficient solar panel to a 12V system throughout the year at Barrow, Alaska. The red curve shows the maximum daily measured incident solar power over the 10 year period from 1995 to 2004; whereas, the blue curve shows the average daily measured incident solar power over the same 10 year period. The shape and variance of both these curves also show that conditions are generally less cloudy in the winter and spring than in the summer and fall. The average curve also shows that, at least for Barrow, users of solar energy should only expect to receive about 70% of the maximum expected power. Finally, the black curve and its data points show the maximum power as determined by the model and are accompanied by a 6 order polynomial fit and that the fit of the estimate to the measured maximum power is excellent.

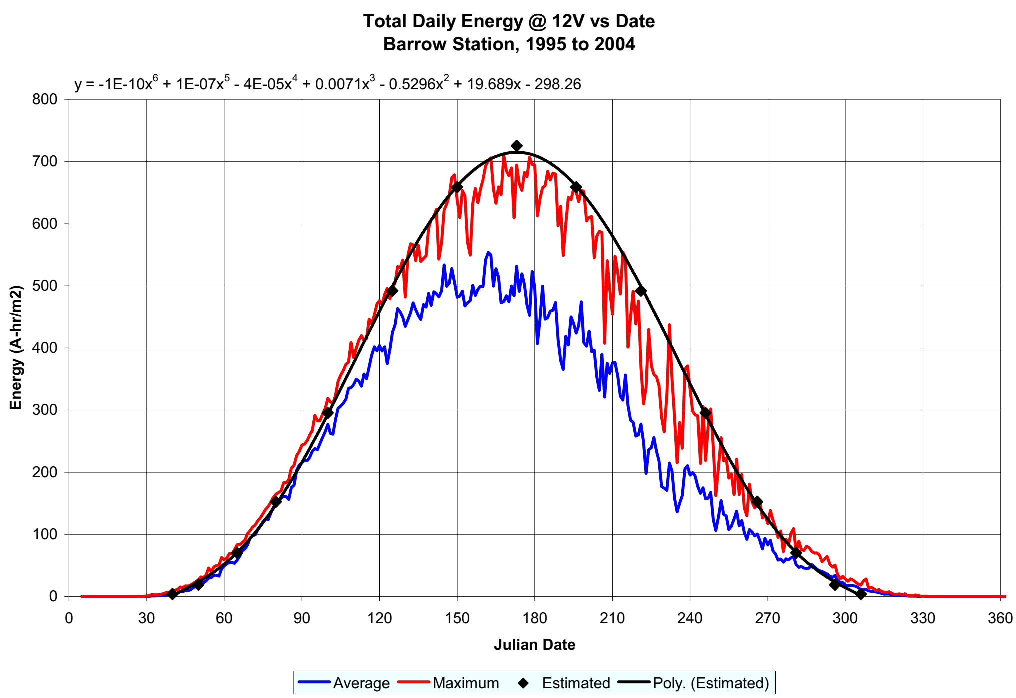

Figure 20: Total daily energy available from a 100% efficient solar panel to a 12V system as measured at Barrow, Alaska, and as estimated from the model

4 Conclusions

The results of this study show that equipment installed for year-round operation at the Gascoyne Inlet site must withstand extreme environmental conditions. Equipment must be able to tolerate winds gusting to 40 m/s with 95% of the gusts being less than 30 m/s, and air temperatures 95% of the time ranging from -45^o to 13^{o}C, with an extreme minimum of -52^{o}C. The study also shows that the sky is clear 17% of the time, cloudy 43% of the time, and foggy 18% of the time. There is likely to be some kind of precipitation 30% of the time, but more precipitation and more clouds are expected in the summer and fall than in the winter and spring, with rain being predominant in the summer and snow in the fall. A total of about 50 to 100 cm of snow and 50 mm of rain are expected each year. Using data from Barrow, Alaska, a sun position model and a solar irradiation model, it was shown that 24-hour continuous daylight should last about 105 days and 24-hour continuous darkness should last about 45 days. Thus, any power system that depends on solar panels must be able to provide at least 45 days of alternate power. This is an absolute minimum, as just outside this 45-day period of darkness, the sun provides little power. For example, if the equipment requires 100 A-h for operation at 12V, then Figure 19 shows that a 1 m^2 100% efficient solar panel could not provide more than 190 days of power. Therefore, alternate power would need to be available for at least 175 days or about half the year.

This study assembled the meteorological data required to make decisions regarding the conditions under which equipment and personnel must operate and survive, and provides information regarding the wind and solar energy potential of the site.

5 References

1. Forand, J.L., "Northern Watch: Trial Site Description", DRDC TN 2008-163, June 2008, UNCLASSIFIED
2. Forand, J.L., "Northern Watch: Barrow Strait Meteorological Data in 2008", DRDC TR 2009-145, November 2009, UNCLASSIFIED
3. http://www.climate.weatheroffice.ec.gc.ca/prods_servs/index_e.html
4. http://www.srrb.noaa.gov/highlights/sunrise/sunrise.html
5. http://www.srrb.noaa.gov/highlights/sunrise/azel.html
6. Greg Pelletier, Department of Ecology, State of Washington, Olympia, WA, Email: gpel461@ ecy.wa.gov
7. http://en.wikipedia.org/wiki/Solar_variation#cite_note-4
8. Willson, R. C., and A. V. Mordvinov, "Secular total solar irradiance trend during solar cycles 21–23", Geophys. Res. Lett., 30(5), 1199, 2003.
9. "CRC Standard Mathematical Tables", 23rd Edition, CRC Press, p.5, 1973.
10. Willett, H.C., "Descriptive Meteorology", Academic Press, New York, NY, p. 68, 1944.
11. http://www.esrl.noaa.gov/gmd/dv/ftpdata.html
12. ftp://ftp.cmdl.noaa.gov/radiation/brw
13. Freund, J., "Aerosol Optical Depth in the Canadian Arctic", Atmosphere-Ocean 21(2), p. 158-167, 1983.

6 Appendix A

```
Function Solar_Irradiance(JD As Double, Solar_Zenith As Double, ExtCoef As Double) As Double
'This function calculates the solar irradiance (insolatio) using the solar zenith and an extinction value.
'This function can be compared to pyranometer data where it is parallel to the earth's surface.

Dim Solar_Constant As Double    'Solar Constant
Dim Extinction, L, A, SINA, B, JC, RS, S  As Double
Dim Optical_Air As Double        'Optical thickness of atmosphere
Dim Earth_Radius As Double      'Radius of Earth

Solar_Constant = 1366          'W/m2
Optical_Air = 10               'km (~height of troposhere)
Earth_Radius = 6360            'km

'Determine solar constant
JC = calcTimeJulianCent(JD)      'Determine Julian century date
JD = JC
RS = calcSunRadVector(JD)       'Determine distance to the sun (AU)
S = Solar_Constant / RS ^ 2       'Correct solar constant for changing distance to the sun

'Determine length through the atmosphere
B = Solar_Zenith * Application.Pi() / 180
SINA = (Earth_Radius / (Earth_Radius + Optical_Air)) * Sin(B)
A = Application.Asin(SINA)
L = Optical_Air
If (SINA > 0) Then
   L = Earth_Radius * Sin(B - A) / SINA
End If

'Determine Solar Irradiance (can be compared to pyranometer data)
Extinction = Exp(-ExtCoef * L)
Solar_Irradiance = S * Cos(Solar_Zenith * Application.Pi() / 180) * Extinction
If Solar_Irradiance < 0 Then Solar_Irradiance = 0
End Function

Function Solar_Intensity(JD As Double, Solar_Zenith As Double, ExtCoef As Double) As Double
'This function calculates the solar intensity using the solar zenith and an extinction value.
'The solar intensity is the energy received for a surface perpendicular to the direction to the sun.

Dim Solar_Constant As Double    'Solar Constant
Dim Extinction, L, A, SINA, B, JC, RS, S  As Double
Dim Optical_Air As Double       'Optical thickness of of atmosphere
Dim Earth_Radius As Double      'Radius of Earth

Solar_Constant = 1366          'W/m2
```

```
Optical_Air = 10                 'km (~height of troposhere)
Earth_Radius = 6360           'km

'Determine solar constant
JC = calcTimeJulianCent(JD)      'Determine Julian century date
JD = JC
RS = calcSunRadVector(JD)       'Determine distance to the sun (AU)
S = Solar_Constant / RS ^ 2       'Correct solar constant for changing distance to the sun

'Determine length through the atmosphere
B = Solar_Zenith * Application.Pi() / 180
SINA = (Earth_Radius / (Earth_Radius + Optical_Air)) * Sin(B)
A = Application.Asin(SINA)
L = Optical_Air
If (SINA > 0) Then
   L = Earth_Radius * Sin(B - A) / SINA
End If

'Determine Solar Irradiance (surface perpendicular to the direction to the sun)
Extinction = Exp(-ExtCoef * L)
Solar_Intensity = Solar_Constant * Extinction
If Solar_Intensity < 0 Then Solar_Intensity = 0
If Solar_Zenith > 90 Then Solar_Intensity = 0
End Function

' Calculation of local times of sunrise, solar noon, and sunset
' based on the calculation procedure by NOAA in the javascript in
' http://www.srrb.noaa.gov/highlights/sunrise/sunrise.html and
' http://www.srrb.noaa.gov/highlights/sunrise/azel.html
'
' The calculations in the NOAA Sunrise/Sunset and Solar Position
' Calculators are based on equations from Astronomical Algorithms,
' by Jean Meeus. NOAA also included atmospheric refraction effects.
' The sunrise and sunset results were reported by NOAA
' to be accurate to within +/- 1 minute for locations between +/- 72°
' latitude, and within ten minutes outside of those latitudes.
'
' This translation was tested for selected locations
' and found to provide results within +/- 1 minute of the
' original Javascript code.
'
' Translated from NOAA's Javascript to Excel VBA by:
'
' Greg Pelletier
' Department of Ecology
' P.O.Box 47600
' Olympia, WA 98504-7600
' email: gpel461@ ecy.wa.gov'
'-----------------------------------
' Luc Forand
```

' DRDC Valcartier
' email: luc.forand@drdc-rddc.gc.ca
'
' The functions provided by Greg Pelletier have been modified so as to output
' the results required by Luc Forand. The functions have also been simplified and
' written in standard Visual Basic 6.3 with all variables types defined.
' All modifications were compared against Greg's results and no discrepancies were observed.
' The following principal functions are availabe for use from an Excel worksheets:
'
' - dawndusk(mode, lat, lon, year, month, day, timezone, dlstime, solardepression)
'   - returns local time of dawn or dusk for all latitudes given a solar depression
' - sunriseset(mode, lat, lon, year, month, day, timezone, dlstime)
'   - returns local time of sunrise or sunset for all latitudes assuming a solar depression of 0.833 deg.
' - solarnoon(lat, lon, year, month, day, timezone, dlstime)
'   - returns local time of solar noon for all latitudes
' - solarposition(mode, lat, lon, year, month, day, hour, minute, second, timezone, dlstime)
'   - returns the solar azimuth, zenith or elevation depending on the 'mode' parameter as a function of local time
'
' The sign convention for the inputs to all functions are:
'
' - positive latitude decimal degrees for northern hemisphere
' - negative longitude degrees for western hemisphere
' - negative time zone hours for western hemisphere
'
'--

Function radToDeg(angleRad As Double) As Double
' Convert radian angle to degrees

radToDeg = 180# * angleRad / Application.Pi()
End Function

Function degToRad(angleDeg As Double) As Double
' Convert degree angle to radians

degToRad = Application.Pi() * angleDeg / 180#
End Function

Function calcJD(year As Integer, month As Integer, day As Integer, hours As Integer, minutes As Integer, seconds As Integer) As Double
'***
'* Name: calcJD
'* Type: Function
'* Purpose: Julian day from calendar day
'* Arguments:
'* year : 4 digit year
'* month: January = 1
'* day : 1 - 31
'* Return value:

```
'*   The Julian day corresponding to the date
'* Note:
'*   Number is returned for start of day.  Fractional days have been added.
'********************************************************************************
Dim A As Double, B As Double
Dim Yr, Mo As Integer

Yr = year
Mo = month
If (Mo <= 2) Then
   Yr = Yr - 1
   Mo = Mo + 12
End If

A = Application.Floor(Yr / 100, 1)
B = 2 - A + Application.Floor(A / 4, 1)

calcJD = Application.Floor(365.25 * (Yr + 4716), 1) + Application.Floor(30.6001 * (Mo + 1), 1)
+ day + B - 1524.5
calcJD = calcJD + (hours + minutes / 60 + seconds / 3600) / 24
End Function

Function calcTimeJulianCent(JD As Double) As Double
'*************************************************************************
'* Name:    calcTimeJulianCent
'* Type:    Function
'* Purpose: convert Julian Day to centuries since J2000.0.
'* Arguments:
'*   jd : the Julian Day to convert
'* Return value:
'*   the T value corresponding to the Julian Day
'*************************************************************************
calcTimeJulianCent = (JD - 2451545#) / 36525#
End Function

Function calcJDFromJulianCent(t As Double) As Double
'*************************************************************************
'* Name:    calcJDFromJulianCent
'* Type:    Function
'* Purpose: convert centuries since J2000.0 to Julian Day.
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   the Julian Day corresponding to the t value
'*************************************************************************
calcJDFromJulianCent = t * 36525# + 2451545#
End Function

Function calcGeomMeanLongSun(t As Double) As Double
'*************************************************************************
```

```
'* Name:    calGeomMeanLongSun
'* Type:    Function
'* Purpose: calculate the Geometric Mean Longitude of the Sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   the Geometric Mean Longitude of the Sun in degrees
'**********************************************************************
Dim l0 As Double

l0 = 280.46646 + t * (36000.76983 + 0.0003032 * t)
Do
   If (l0 <= 360) And (l0 >= 0) Then Exit Do
   If l0 > 360 Then l0 = l0 - 360
   If l0 < 0 Then l0 = l0 + 360
Loop
calcGeomMeanLongSun = l0
End Function

Function calcGeomMeanAnomalySun(t As Double) As Double
'**********************************************************************
'* Name:    calGeomAnomalySun
'* Type:    Function
'* Purpose: calculate the Geometric Mean Anomaly of the Sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   the Geometric Mean Anomaly of the Sun in degrees
'**********************************************************************
calcGeomMeanAnomalySun = 357.52911 + t * (35999.05029 - 0.0001537 * t)
End Function

Function calcEccentricityEarthOrbit(t As Double) As Double
'**********************************************************************
'* Name:    calcEccentricityEarthOrbit
'* Type:    Function
'* Purpose: calculate the eccentricity of earth's orbit
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   the unitless eccentricity
'**********************************************************************
calcEccentricityEarthOrbit = 0.016708634 - t * (0.000042037 + 0.0000001267 * t)
End Function

Function calcSunEqOfCenter(t As Double) As Double
'**********************************************************************
'* Name:    calcSunEqOfCenter
'* Type:    Function
'* Purpose: calculate the equation of center for the sun
```

```
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   in degrees
'***********************************************************************
Dim m As Double, mrad As Double, sinm As Double, sin2m As Double, sin3m As Double
Dim c As Double

m = calcGeomMeanAnomalySun(t)
mrad = degToRad(m)
sinm = Sin(mrad)
sin2m = Sin(mrad + mrad)
sin3m = Sin(mrad + mrad + mrad)

c = sinm * (1.914602 - t * (0.004817 + 0.000014 * t)) + sin2m * (0.019993 - 0.000101 * t) +
sin3m * 0.000289
calcSunEqOfCenter = c
End Function

Function calcSunTrueLong(t As Double) As Double
'***********************************************************************
'* Name:    calcSunTrueLong
'* Type:    Function
'* Purpose: calculate the true longitude of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's true longitude in degrees
'***********************************************************************
calcSunTrueLong = calcGeomMeanLongSun(t) + calcSunEqOfCenter(t)
End Function

Function calcSunTrueAnomaly(t As Double) As Double
'***********************************************************************
'* Name:    calcSunTrueAnomaly (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the true anamoly of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's true anamoly in degrees
'***********************************************************************
calcSunTrueAnomaly = calcGeomMeanAnomalySun(t) + calcSunEqOfCenter(t)
End Function

Function calcSunRadVector(t As Double) As Double
'***********************************************************************
'* Name:    calcSunRadVector (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the distance to the sun in AU
```

```
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun radius vector in AUs
'***********************************************************************
Dim v As Double, e As Double

v = calcSunTrueAnomaly(t)
e = calcEccentricityEarthOrbit(t)

calcSunRadVector = (1.000001018 * (1 - e * e)) / (1 + e * Cos(degToRad(v)))
End Function

Function calcSunApparentLong(t As Double) As Double
'***********************************************************************
'* Name:    calcSunApparentLong (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the apparent longitude of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's apparent longitude in degrees
'***********************************************************************
Dim omega As Double

omega = 125.04 - 1934.136 * t
calcSunApparentLong = calcSunTrueLong(t) - 0.00569 - 0.00478 * Sin(degToRad(omega))
End Function

Function calcMeanObliquityOfEcliptic(t As Double) As Double
'***********************************************************************
'* Name:    calcMeanObliquityOfEcliptic
'* Type:    Function
'* Purpose: calculate the mean obliquity of the ecliptic
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   mean obliquity in degrees
'***********************************************************************
Dim seconds As Double

seconds = 21.448 - t * (46.815 + t * (0.00059 - t * (0.001813)))
calcMeanObliquityOfEcliptic = 23# + (26# + (seconds / 60#)) / 60#
End Function

Function calcObliquityCorrection(t As Double) As Double
'***********************************************************************
'* Name:    calcObliquityCorrection
'* Type:    Function
'* Purpose: calculate the corrected obliquity of the ecliptic
```

```
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   corrected obliquity in degrees
'***********************************************************************
Dim omega As Double

omega = 125.04 - 1934.136 * t
calcObliquityCorrection = calcMeanObliquityOfEcliptic(t) + 0.00256 * Cos(degToRad(omega))
 End Function

Function calcSunRtAscension(t As Double) As Double
'***********************************************************************
'* Name:    calcSunRtAscension (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the right ascension of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's right ascension in degrees
'***********************************************************************
Dim e As Double, lambda As Double, tananum As Double, tanadenom As Double

e = calcObliquityCorrection(t)
lambda = calcSunApparentLong(t)

tananum = (Cos(degToRad(e)) * Sin(degToRad(lambda)))
tanadenom = (Cos(degToRad(lambda)))

calcSunRtAscension = radToDeg(Application.Atan2(tanadenom, tananum))
End Function

Function calcSunDeclination(t As Double) As Double
'***********************************************************************
'* Name:    calcSunDeclination
'* Type:    Function
'* Purpose: calculate the declination of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's declination in degrees
'***********************************************************************
Dim e As Double, lambda As Double, sint As Double

e = calcObliquityCorrection(t)
lambda = calcSunApparentLong(t)

sint = Sin(degToRad(e)) * Sin(degToRad(lambda))
calcSunDeclination = radToDeg(Application.Asin(sint))
End Function
```

Function calcEquationOfTime(t As Double) As Double

```
'***********************************************************************
'* Name:    calcEquationOfTime
'* Type:    Function
'* Purpose: calculate the difference between true solar time and mean
'*     solar time
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   equation of time in minutes of time
'***********************************************************************
Dim epsilon As Double, l0 As Double, e As Double, m As Double
Dim y As Double, sin2l0 As Double, sinm As Double
Dim cos2l0 As Double, sin4l0 As Double, sin2m As Double, Etime As Double

epsilon = calcObliquityCorrection(t)
l0 = calcGeomMeanLongSun(t)
e = calcEccentricityEarthOrbit(t)
m = calcGeomMeanAnomalySun(t)
y = (Tan(degToRad(epsilon) / 2#)) ^ 2

sin2l0 = Sin(2# * degToRad(l0))
sinm = Sin(degToRad(m))
cos2l0 = Cos(2# * degToRad(l0))
sin4l0 = Sin(4# * degToRad(l0))
sin2m = Sin(2# * degToRad(m))

Etime = y * sin2l0 - 2# * e * sinm + 4# * e * y * sinm * cos2l0 - 0.5 * y * y * sin4l0 - 1.25 * e * e
* sin2m

calcEquationOfTime = radToDeg(Etime) * 4#
```

End Function

Function calcHourAngleDawn(Lat As Double, solarDec As Double, solardepression As Double) As Double

```
'***********************************************************************
'* Name:    calcHourAngleDawn
'* Type:    Function
'* Purpose: calculate the hour angle of the sun at dawn for the
'*        latitude
'*        for user selected solar depression below horizon
'* Arguments:
'*   lat : latitude of observer in degrees
'*   solarDec : declination angle of sun in degrees
'*   solardepression: angle of the sun below the horizion in degrees
'* Return value:
'*   hour angle of dawn in radians
'***********************************************************************
Dim latRad As Double, sdRad As Double, HAarg As Double
```

```
latRad = degToRad(Lat)
sdRad = degToRad(solarDec)
HAarg = (Cos(degToRad(90 + solardepression)) / (Cos(latRad) * Cos(sdRad)) - Tan(latRad) * Tan(sdRad))
If HAarg < -1 Then HAarg = -1
If HAarg > 1 Then HAarg = 1
calcHourAngleDawn = Application.Acos(HAarg)
End Function

Function calcHourAngleSunrise(Lat As Double, solarDec As Double) As Double
'***********************************************************************
'* Name:    calcHourAngleSunrise
'* Type:    Function
'* Purpose: calculate the hour angle of the sun at sunrise for the
'*         latitude
'* Arguments:
'*   lat : latitude of observer in degrees
'* solarDec : declination angle of sun in degrees
'* Return value:
'*   hour angle of sunrise in radians
'*
'* Note: For sunrise and sunset calculations, we assume 0.833° of atmospheric refraction
'* For details about refraction see http://www.srrb.noaa.gov/highlights/sunrise/calcdetails.html
'*
'***********************************************************************
Dim latRad As Double, sdRad As Double, HAarg As Double

latRad = degToRad(Lat)
sdRad = degToRad(solarDec)
HAarg = (Cos(degToRad(90.833)) / (Cos(latRad) * Cos(sdRad)) - Tan(latRad) * Tan(sdRad))
If HAarg < -1 Then HAarg = -1
If HAarg > 1 Then HAarg = 1
calcHourAngleSunrise = Application.Acos(HAarg)
End Function

Function calcHourAngleSunset(Lat As Double, solarDec As Double) As Double
'***********************************************************************
'* Name:    calcHourAngleSunset
'* Type:    Function
'* Purpose: calculate the hour angle of the sun at sunset for the
'*         latitude
'* Arguments:
'*   lat : latitude of observer in degrees
'* solarDec : declination angle of sun in degrees
'* Return value:
'*   hour angle of sunset in radians
'*
'* Note: For sunrise and sunset calculations, we assume 0.833° of atmospheric refraction
'* For details about refraction see http://www.srrb.noaa.gov/highlights/sunrise/calcdetails.html
```

```
'*
'**********************************************************************
Dim latRad As Double, sdRad As Double, HAarg As Double

latRad = degToRad(Lat)
sdRad = degToRad(solarDec)
HAarg = (Cos(degToRad(90.833)) / (Cos(latRad) * Cos(sdRad)) - Tan(latRad) * Tan(sdRad))
If HAarg < -1 Then HAarg = -1
If HAarg > 1 Then HAarg = 1
calcHourAngleSunset = -Application.Acos(HAarg)
End Function

Function calcHourAngleDusk(Lat As Double, solarDec As Double, solardepression As Double) As Double
'**********************************************************************
'* Name:    calcHourAngleDusk
'* Type:    Function
'* Purpose: calculate the hour angle of the sun at dusk for the
'*         latitude
'*         for user selected solar depression below horizon
'* Arguments:
'*   lat : latitude of observer in degrees
'*   solarDec : declination angle of sun in degrees
'*   solardepression: angle of sun below horizon in degrees
'* Return value:
'*   hour angle of dusk in radians
'**********************************************************************
Dim latRad As Double, sdRad As Double, HAarg As Double

latRad = degToRad(Lat)
sdRad = degToRad(solarDec)
HAarg = (Cos(degToRad(90 + solardepression)) / (Cos(latRad) * Cos(sdRad)) - Tan(latRad) * Tan(sdRad))
If HAarg < -1 Then HAarg = -1
If HAarg > 1 Then HAarg = 1
calcHourAngleDusk = -Application.Acos(HAarg)
End Function

Function calcDawnUTC(JD As Double, latitude As Double, longitude As Double, solardepression As Double) As Double
'**********************************************************************
'* Name:    calcDawnUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of dawn
'*         for the given day at the given location on earth
'*         for user selected solar depression below horizon
'* Arguments:
'*   JD  : julian day
'*   latitude : latitude of observer in degrees
'*   longitude : longitude of observer in degrees
```

```
'*   solardepression: angle of sun below the horizon in degrees
'* Return value:
'*   time in minutes from zero Z
'*******************************************************************
Dim t As Double, eqtime As Double, solarDec As Double, hourangle As Double
Dim delta As Double, timeDiff As Double, timeUTC As Double
Dim newt As Double

t = calcTimeJulianCent(JD)

'*** First pass to approximate sunrise in minutes of time
eqtime = calcEquationOfTime(t)
solarDec = calcSunDeclination(t)
hourangle = calcHourAngleSunrise(latitude, solarDec)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

' *** Second pass includes fractional jday in gamma calc
newt = calcTimeJulianCent(calcJDFromJulianCent(t) + timeUTC / 1440#)
eqtime = calcEquationOfTime(newt)
solarDec = calcSunDeclination(newt)
hourangle = calcHourAngleDawn(latitude, solarDec, solardepression)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

calcDawnUTC = timeUTC
End Function

Function calcSunriseUTC(JD As Double, latitude As Double, longitude As Double) As Double
'*******************************************************************
'* Name:    calcSunriseUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of sunrise
'*          for the given day at the given location on earth
'* Arguments:
'*   JD  : julian day
'*   latitude : latitude of observer in degrees
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time in minutes from zero Z
'*******************************************************************
Dim t As Double, eqtime As Double, solarDec As Double, hourangle As Double
Dim delta As Double, timeDiff As Double, timeUTC As Double
Dim newt As Double

t = calcTimeJulianCent(JD)
```

```
'*** First pass to approximate sunrise in minutes of time
eqtime = calcEquationOfTime(t)
solarDec = calcSunDeclination(t)
hourangle = calcHourAngleSunrise(latitude, solarDec)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

'*** Second pass includes fractional jday in gamma calc
newt = calcTimeJulianCent(calcJDFromJulianCent(t) + timeUTC / 1440#)
eqtime = calcEquationOfTime(newt)
solarDec = calcSunDeclination(newt)
hourangle = calcHourAngleSunrise(latitude, solarDec)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

calcSunriseUTC = timeUTC
End Function

Function calcSolNoonUTC(t As Double, longitude As Double) As Double
'***********************************************************************
'* Name:    calcSolNoonUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of solar
'*     noon for the given day at the given location on earth
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time in minutes from zero Z
'***********************************************************************
Dim newt As Double, eqtime As Double, solarNoonDec As Double, solNoonUTC As Double

newt = calcTimeJulianCent(calcJDFromJulianCent(t) + 0.5 - longitude / 360#)
eqtime = calcEquationOfTime(newt)
solarNoonDec = calcSunDeclination(newt)
solNoonUTC = 720 - (longitude * 4) - eqtime

calcSolNoonUTC = solNoonUTC
End Function

Function calcSunsetUTC(JD As Double, latitude As Double, longitude As Double) As
Double
'***********************************************************************/
'* Name:    calcSunsetUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of sunset
'*         for the given day at the given location on earth
'* Arguments:
```

```
'*   JD  : julian day
'*   latitude : latitude of observer in degrees
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time in minutes from zero Z
'**********************************************************************
Dim t As Double, eqtime As Double, solarDec As Double, hourangle As Double
Dim delta As Double, timeDiff As Double, timeUTC As Double
Dim newt As Double

t = calcTimeJulianCent(JD)

'First calculates sunrise and approx length of day in minutes
eqtime = calcEquationOfTime(t)
solarDec = calcSunDeclination(t)
hourangle = calcHourAngleSunset(latitude, solarDec)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

'first pass used to include fractional day in gamma calc
newt = calcTimeJulianCent(calcJDFromJulianCent(t) + timeUTC / 1440#)
eqtime = calcEquationOfTime(newt)
solarDec = calcSunDeclination(newt)
hourangle = calcHourAngleSunset(latitude, solarDec)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

calcSunsetUTC = timeUTC
End Function
Function calcDuskUTC(JD As Double, latitude As Double, longitude As Double,
solardepression As Double) As Double
'**********************************************************************
'* Name:    calcDuskUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of dusk
'*         for the given day at the given location on earth
'*         for user selected solar depression below horizon
'* Arguments:
'*   JD  : julian day
'*   latitude : latitude of observer in degrees
'*   longitude : longitude of observer in degrees
'*   solardepression: angle of sun below horizon
'* Return value:
'*   time in minutes from zero Z
'**********************************************************************
Dim t As Double, eqtime As Double, solarDec As Double, hourangle As Double
Dim delta As Double, timeDiff As Double, timeUTC As Double
Dim newt As Double
```

```
t = calcTimeJulianCent(JD)

'First calculates sunrise and approx length of day in minutes
eqtime = calcEquationOfTime(t)
solarDec = calcSunDeclination(t)
hourangle = calcHourAngleSunset(latitude, solarDec)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

'first pass used to include fractional day in gamma calc
newt = calcTimeJulianCent(calcJDFromJulianCent(t) + timeUTC / 1440#)
eqtime = calcEquationOfTime(newt)
solarDec = calcSunDeclination(newt)
hourangle = calcHourAngleDusk(latitude, solarDec, solardepression)
delta = -longitude - radToDeg(hourangle)
timeDiff = 4 * delta
timeUTC = 720 + timeDiff - eqtime

calcDuskUTC = timeUTC
End Function

Function dawndusk(mode As Integer, Lat As Double, lon As Double, year As Integer, month As Integer, day As Integer, timezone As Integer, dlstime As Double, solardepression As Double) As Double
'***********************************************************************
'* Name:    dawndusk
'* Type:    Main Function called by spreadsheet
'* Purpose: calculate time of dawn or dusk for the entered date
'*      and location.
'* For latitudes greater than 72 degrees N and S, calculations are
'* accurate to within 10 minutes. For latitudes less than +/- 72°
'* accuracy is approximately one minute.
'* Arguments:
'   mode = 0 (dawn) or 1 (dusk)
'   latitude = latitude (decimal degrees)
'   longitude = longitude (decimal degrees)
'   year = year
'   month = month
'   day = day
'   timezone = time zone hours relative to GMT/UTC (hours)
'   dlstime = daylight savings time (0 = none, 1 = 1 hour, etc.) (hours)
'   solardepression = angle of sun below horizon in degrees
'* Return value:
'*   dawn or dusk time in local time (days)
'***********************************************************************
Dim longitude As Double, latitude As Double, JD As Double
Dim TimeGMT As Double, TimeLST As Double
```

```
' Latitude & longitude check
longitude = lon
latitude = Lat
If (latitude > 90) Then latitude = 90
If (latitude < -90) Then latitude = -90

JD = calcJD(year, month, day, 0, 0, 0)

If mode = 0 Then
   'Calculate sunrise for this date
   TimeGMT = calcDawnUTC(JD, latitude, longitude, solardepression)
Else
   'Calculate sunset for this date
   TimeGMT = calcDuskUTC(JD, latitude, longitude, solardepression)
End If

'Adjust for time zone and daylight savings time in minutes
TimeLST = TimeGMT + (60 * timezone) + (dlstime * 60)

'Convert to days
dawndusk = TimeLST / 1440
End Function

Function sunriseset(mode As Integer, Lat As Double, lon As Double, year As Integer, month As Integer, day As Integer, timezone As Integer, dlstime As Double) As Double
'***********************************************************************
'* Name:    sunriseset
'* Type:    Main Function called by spreadsheet
'* Purpose: calculate time of sunrise or sunset for the entered date
'*          and location.
'* For latitudes greater than 72 degrees N and S, calculations are
'* accurate to within 10 minutes. For latitudes less than +/- 72°
'* accuracy is approximately one minute.
'* Arguments:
'  mode = 0 (sunrise) or 1 (sunset)
'  latitude = latitude (decimal degrees)
'  longitude = longitude (decimal degrees)
'  year = year
'  month = month
'  day = day
'  timezone = time zone hours relative to GMT/UTC (hours)
'  dlstime = daylight savings time (0 = none, 1 = 1 hour, etc) (hours)
'* Return value:
'*   sunrise or sunset time in local time (days)
'***********************************************************************
Dim longitude As Double, latitude As Double, JD As Double
Dim TimeGMT As Double, TimeLST As Double

'Latitude & longitude check
longitude = lon
```

```
latitude = Lat
If (latitude > 90) Then latitude = 90
If (latitude < -90) Then latitude = -90

JD = calcJD(year, month, day, 0, 0, 0)

If mode = 0 Then
    'Calculate sunrise for this date
    TimeGMT = calcSunriseUTC(JD, latitude, longitude)
Else
    'Calculate sunset for this date
    TimeGMT = calcSunsetUTC(JD, latitude, longitude)
End If

'Adjust for time zone and daylight savings time in minutes
TimeLST = TimeGMT + (60 * timezone) + (dlstime * 60)

'Convert to days
sunriseset = TimeLST / 1440
```

End Function

Function solarnoon(Lat As Double, lon As Double, year As Integer, month As Integer, day As Integer, timezone As Integer, dlstime As Double) As Double

```
'**********************************************************************
'* Name:    solarnoon
'* Type:    Main Function called by spreadsheet
'* Purpose: calculate the Universal Coordinated Time (UTC) of solar
'*     noon for the given day at the given location on earth
'* Arguments:
'    year
'    month
'    day
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time of solar noon in local time days
'**********************************************************************
Dim longitude As Double, latitude As Double, JD As Double
Dim t As Double, newt As Double, eqtime As Double
Dim solarNoonDec As Double, solNoonUTC As Double

'Latitude & longitude check
longitude = lon
latitude = Lat
If (latitude > 90) Then latitude = 90
If (latitude < -90) Then latitude = -90

JD = calcJD(year, month, day, 0, 0, 0)
t = calcTimeJulianCent(JD)

newt = calcTimeJulianCent(calcJDFromJulianCent(t) + 0.5 + longitude / 360#)
```

```
eqtime = calcEquationOfTime(newt)
solarNoonDec = calcSunDeclination(newt)
solNoonUTC = 720 + (longitude * 4) - eqtime

'adjust for time zone and daylight savings time in minutes
solarnoon = solNoonUTC + (60 * timezone) + (dlstime * 60)

'convert to days
solarnoon = solarnoon / 1440
End Function

Function solarposition(mode As Integer, Lat As Double, lon As Double, year As Integer, month As Integer, day As Integer, hours As Integer, minutes As Integer, seconds As Integer, timezone As Integer, dlstime As Double) As Double
'***********************************************************************
'* Name:    solarposition
'* Type:    Main Function
'* Purpose: calculate solar azimuth (deg from north) for the entered
'*          date, time and location.
'*
'* Arguments:
'*   mode,latitude, longitude, year, month, day, hour, minute, second,
'*   timezone, daylightsavingstime
'* Return value:
'*   If mode = 0:  solar azimuth in degrees from north
'*   If mode = 1:  solar zenith in degrees from vertical
'*   If mode = 2:  solar elevation in degrees from horizon
'*
'***********************************************************************
Dim longitude As Double, latitude As Double
Dim zone As Double
Dim hh As Double, mm As Double, ss As Double, timenow As Double
Dim JD As Double, t As Double, R As Double
Dim alpha As Double, theta As Double, Etime As Double, eqtime As Double
Dim solarDec As Double, earthRadVec As Double, solarTimeFix As Double
Dim trueSolarTime As Double, hourangle As Double, harad As Double
Dim csz As Double, zenith As Double, azDenom As Double, azRad As Double
Dim azimuth As Double, exoatmElevation As Double
Dim step1 As Double, step2 As Double, step3 As Double
Dim refractionCorrection As Double, te As Double, solarzen As Double

'Latitude, longitude & timezone check
longitude = lon
latitude = Lat
If (latitude > 90) Then latitude = 90
If (latitude < -90) Then latitude = -90
zone = timezone
hh = hours - dlstime
mm = minutes
```

```
ss = seconds

'timenow is GMT time for calculation in hours since 0Z
timenow = hh + mm / 60 + ss / 3600 - zone

JD = calcJD(year, month, day, 0, 0, 0) + (timenow - 12) / 24#
t = calcTimeJulianCent(JD)
R = calcSunRadVector(t)
alpha = calcSunRtAscension(t)
theta = calcSunDeclination(t)
Etime = calcEquationOfTime(t)

eqtime = Etime
solarDec = theta     'in degrees
earthRadVec = R

solarTimeFix = eqtime + 4# * longitude - 60# * zone
trueSolarTime = hh * 60# + mm + ss / 60# + solarTimeFix   'in minutes

Do While (trueSolarTime > 1440)
   trueSolarTime = trueSolarTime - 1440
Loop

hourangle = trueSolarTime / 4# - 180#
'Thanks to Louis Schwarzmayr for the next line:
If (hourangle < -180) Then hourangle = hourangle + 360#
harad = degToRad(hourangle)

csz = Sin(degToRad(latitude)) * Sin(degToRad(solarDec)) + Cos(degToRad(latitude)) *
Cos(degToRad(solarDec)) * Cos(harad)
If (csz > 1#) Then
   csz = 1#
ElseIf (csz < -1#) Then
   csz = -1#
End If

zenith = radToDeg(Application.Acos(csz))
azDenom = (Cos(degToRad(latitude)) * Sin(degToRad(zenith)))
If (Abs(azDenom) > 0.001) Then
   azRad = ((Sin(degToRad(latitude)) * Cos(degToRad(zenith))) - Sin(degToRad(solarDec))) /
azDenom
   If (Abs(azRad) > 1#) Then
      If (azRad < 0) Then
         azRad = -1#
      Else
         azRad = 1#
      End If
   End If

   azimuth = 180# - radToDeg(Application.Acos(azRad))
```

```
    If (hourangle > 0#) Then
        azimuth = -azimuth
    End If
Else
    If (latitude > 0#) Then
        azimuth = 180#
    Else
        azimuth = 0#
    End If
End If

If (azimuth < 0#) Then
    azimuth = azimuth + 360#
End If
exoatmElevation = 90# - zenith

'Refraction correction
If (exoatmElevation > 85#) Then
    refractionCorrection = 0#
Else
    te = Tan(degToRad(exoatmElevation))
    If (exoatmElevation > 5#) Then
        refractionCorrection = 58.1 / te - 0.07 / (te * te * te) + 0.000086 / (te * te * te * te * te)
    ElseIf (exoatmElevation > -0.575) Then
        step1 = (-12.79 + exoatmElevation * 0.711)
        step2 = (103.4 + exoatmElevation * (step1))
        step3 = (-518.2 + exoatmElevation * (step2))
        refractionCorrection = 1735# + exoatmElevation * (step3)
    Else
        refractionCorrection = -20.774 / te
    End If
    refractionCorrection = refractionCorrection / 3600#
End If

'Include refraction correction
solarzen = zenith - refractionCorrection

'Output desired parameter
If (mode = 0) Then     'solar azimuth
    solarposition = azimuth
ElseIf (mode = 1) Then 'solar zenith
    solarposition = solarzen
Else                   'solar elevation
    solarposition = 90# - solarzen
End If
End Function
```

List of abbreviations

A-h	Amp-Hour
BRW	Barrow Meteorological Station
DRDC	Defence Research & Development Canada
DRDKIM	Director Research and Development Knowledge and Information Management
EC	Environment Canada
NOAA	National Oceanographic and Atmospheric Administration
CORA	Centre of Operational Research and Analysis
R&D	Research & Development

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) DRDC Valcartier	2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.) UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Northern Watch: Arctic Site Meteorology

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Forand, J. Luc

5. DATE OF PUBLICATION (Month and year of publication of document.) September 2010	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 54	6b. NO. OF REFS (Total cited in document.) 13

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.) Northern Watch (15aa)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC Valcartier TR 2010-003	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The principal objective of the Northern Watch Project is to install and test the performance of sensors at the Arctic site near Gascoyne Inlet. To accomplish this task the meteorological conditions to be expected at the site need to be determined. As no data has yet been obtained at the site, historical data from the nearby Environment Canada station at Resolute Bay was used, along with a solar insolation model.

This document shows the types of atmospheric conditions that are to be expected at the Arctic site based on the historical meteorological data sets and the insolation model. It shows that, if the Gascoyne Inlet installation is to operate year-round, equipment and personnel must withstand quite rigorous conditions. Equipment must be able to support winds gusting to 40 m/s and temperatures between -45 and 13 ^{o}C. It also shows that the sky is clear 17 % of the time, cloudy 43 % of the time, foggy 18 % of the time, and that there is likely to be some kind of precipitation 30 % of the time. Finally, using historical solar irradiance data from Barrow, Alaska, and an insolation model, it shows that for equipment requiring 100 A-h at 12 V, a 1 square meter 100 % efficient solar panel could not provide more than 190 days of power and that alternate power would need to be available for at least 175 days or about half the year.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Northern Watch, Gascoyne Inlet, meteorology, solar irradiance

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca