

An open-source albedo-wand for the measurement of sea ice albedo

Christian KATLEIN^{1,2*}

¹ Alfred-Wegener-Institut, Helmholtz Zentrum für Polar- und Meeresforschung, 27570 Bremerhaven, Germany;

² Takuvik Joint International Laboratory, Université Laval and CNRS, Québec, QC G1V 0A6, Canada

Received 11 February 2019; accepted 24 April 2019; published online 18 June 2019

Abstract Surface albedo is defined as the ratio of incident and reflected solar irradiance and describes the ability of a surface to reflect, rather than absorb incident solar shortwave radiation. It is thus a crucial parameter in the climate system, particularly in the polar oceans. Sea ice albedo is a main driver for light transmission into the polar oceans and thus has a high impact on ocean warming, ice melting and marine primary production. During spring and summer, sea ice albedo can exhibit a significant spatial variability caused by meltwater accumulations on the ice. While complex and expensive solutions for albedo measurements are already available, we want to present a simple open-source design that allows for affordable mapping of spatially varying surface albedo on sea ice and beyond. Our solution is based on off-the-shelf components, such as an Arduino microcontroller integrating affordable light sensors, a GPS unit, data recording on memory card and data display into a simple field strengthened unit. We provide example data from two Arctic field deployments showing the capabilities and limitations of this system.

Keywords sea ice, albedo, spatial variability, mapping, GPS, open-source

Citation: Katlein C. An open-source albedo-wand for the measurement of sea ice albedo. *Adv Polar Sci*, 2019, 30(2): 106-117, doi: 10.13679/j.advps.2019.0002

1 Introduction

Surface albedo is a key parameter in our earth's climate system (Curry et al., 1995). Albedo describes the reflective properties of a surface and is defined by the ratio of reflected to incident solar shortwave radiation (Perovich, 1996). In the polar regions, surface albedo decreases with the retreat of snow and ice (Perovich et al., 2002; Perovich and Polashenski, 2012). The exposed dark ocean or land surface then leads to further warming. This positive feedback loop is known as the ice albedo feedback.

In spring and summer the albedo of sea ice also varies on the scale of several meters, when melt water ponds form on the surface (Fetterer and Untersteiner, 1998; Flanner and

Zender, 2006; Flocco et al., 2012). This variability impacts the optical properties of the ice pack and is thus a critical factor in understanding the energy partitioning across the atmosphere-ice-ocean interface (Perovich, 1990; Perovich et al., 1998; Light et al., 2015).

Due to its importance sea ice albedo has been already measured for many decades. Early measurements were single point measurements, followed by an increasing number of albedo measurements along transect lines (Perovich, 1990, 2005; Perovich and Richter-Menge, 2006). Since then, sea ice albedo has also been measured on manned and unmanned drifting stations (Nicolaus et al., 2010a; Wang et al., 2014) or most recently integrated into complex multi-sensor albedo mapping sled systems (Hudson et al., 2012). Modern sensor setups include high quality and even hyperspectral optical sensors that are able to endure the harsh environmental conditions in the Arctic (Nicolaus et al., 2010b). However such sensor setups are

* Corresponding author, E-mail: Christian.Katlein@awi.de

usually very expensive and complicated to operate, prohibiting wide use of such equipment.

Albedo measurements can vary in respect to the covered spectral range. While only spectral albedo can be attributed precisely to surface properties, broadband integrated surface albedo in planetary meteorology most frequently covers the entire range of solar shortwave radiation between 280 nm and 4000 nm. The measurement of surface albedo in different spectral bands, such as the visible or photosynthetically active part of the spectrum can however be better suited for investigations of the under-ice light field, or in primary production studies, where the near infrared portion of the solar spectrum is of negligible importance.

Simple techniques have been used for albedo measurements using cameras or even smartphones photographing the surface in question, as well as a white paper (Barnard et al., 2014) or other reflection standard. These simpler approaches are accessible to a wide range of projects and could be used in citizen science data collection. The techniques are however not suited to acquire a high number of spatially distributed measurements to provide large datasets on the albedo of spatially varying surfaces.

For the measurement of snow thickness, such a professional tool allowing the easy acquisition of thousands of spatially distributed georeferenced datapoints is available commercially: The MagnaProbe (SnowHydro, Fairbanks, Alaska, USA) combines a magnetostrictive snow height measurement stick with a GPS receiver, user interface and datalogger (Sturm and Holmgren, 2018).

The objective of this paper is to present an open-source low-cost solution for easy recording of high numbers of spatially distributed and georeferenced measurements of surface albedo in the spectral range of 360–1120 nm. We describe the device design based on open-source components and describe its operation. Exemplary data from two polar research expeditions demonstrate the field capabilities and provides an overview about capabilities and limitations of the device.

2 Methods

2.1 Design and construction

The albedo wand is shown in Figure 1a. It consists of two SP-215-SS silicon cell pyranometers (Apogee Instruments Inc., Logan, UT, USA) mounted at one end of a telescopic rod and a small transparent watertight case holding all electronics. The three parts for attaching the sensors and the watertight case (Figure 2) were manufactured using a 3D printer (Renkforce RF100, Conrad Electronics SE Hirschau, Germany). The generated Computer Aided Design (CAD) files are included in the electronic supplementary material.

The electronic components are off-the-shelf and can be

simply assembled. Only basic soldering skills are required, as most electric connections can be achieved by simply plugging jumper cables onto the device header pins. Good tutorials, providing code examples and documentation on the integration of the individual components are provided on the respective manufacturer websites (e.g. <https://www.sparkfun.com/>, <https://www.adafruit.com/> or <https://www.arduino.cc/>). A full wiring diagram of the components is presented in Figure 3 and a detailed parts list is provided in Table 1. The albedo wand is based on an Arduino UNO microcontroller. It is powered using four AA batteries. To facilitate operation, the power supply is connected to a simple on/off switch. Data and device status is displayed on a two line liquid crystal display (LCD). All acquired data are saved on an SD card module also connected to the Arduino. A GPS sensor provides geographic position information via a software serial interface on the Arduino. Both pyranometers generate an analogue 0–5 V output signal that is read directly using the analogue/digital converters onboard the Arduino. Measurements are initiated by a pushbutton connected to an interrupt port on the Arduino. A resistor pulls down the signal line to ground after each button contact.

The code for configuration of the Arduino is given in the supplementary material. It can easily be modified for individual requirement.



Figure 1 a, Photo of the completely assembled albedo wand; b, Albedo wand used to measure the albedo of a melt pond during the *Amundsen* 2018 expedition in Baffin Bay, July 2018; c, Albedo wand used in harsh Arctic autumn climate close to the geographic North Pole during the *Oden* expedition, September 2018.

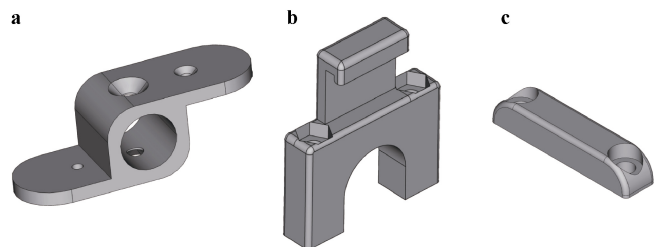


Figure 2 Three dimensional CAD drawings of the parts produced using 3d printing: a, Double sensor mount with attachment for the telescopic rod; b, Clamp for attachment of the waterproof case to the telescopic rod; c, Counter bar to fix part b around the telescopic rod.

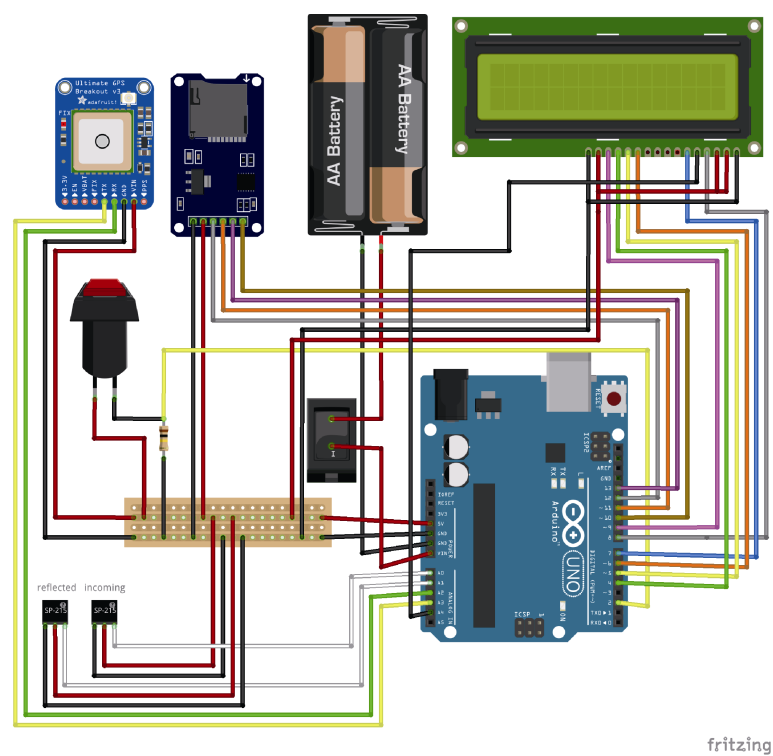


Figure 3 Sketch of the wire connections between the different components of the Albedo-Wand. Components from top left to bottom right are GPS receiver, SD-Card module, battery pack, display, button, switch, pulldown resistor, breadboard, the two sensors and the Arduino UNO microcontroller. Sketch was generated using the Fritzing Software (<http://fritzing.org>).

Table 1 List of all parts necessary for building the system. Webpages provide examples of possible part distributors			
Name	#	Model	Website
Irradiance sensor	2	Apogee SP-215_SS	https://www.apogeeinstruments.com/sp-215-ss-amplified-0-5-volt-pyranometer/
Microcontroller	1	Arduino UNO	https://www.sparkfun.com/products/11021
GPS	1	Adafruit Ultimate GPS	https://www.adafruit.com/product/746
Sd-card	1	Micro SD card FAT16	https://www.adafruit.com/product/2693
SD module	1	Adafruit SD card breakout	https://www.adafruit.com/product/254
LED display	1	Standard 16x2 LCD	https://www.adafruit.com/product/181
Resistor	1	-	-
Jumper cable	40	male/male & male/female	https://www.sparkfun.com/products/12794
Breadboard	1	micro breadboard	https://www.adafruit.com/product/2463
Switch	1	on/off switch	https://www.sparkfun.com/products/8837
Button	1	-	https://www.adafruit.com/product/481
Battery holder	1	4×AA	https://www.adafruit.com/product/3905
Batteries	4	AA	-
Screw & nut	4	M6 × 45	-
Telescopic rod	1	Bürkle TeleScoop	https://www.buerkle.de/en/telescopic-rod#optional-accessory
Waterproof case	1	Peli 1020 transparent	https://www.pelishop.com/peli-1020-micro-case.html
Cable bulkheads	2	Lapp Skintop STR-M12	https://de.rs-online.com/web/p/kabelverschraubungen/4442919/
Bubble level	1	-	-

2.2 Sensors and calibration

The two SP-215-SS silicon cell pyranometers (Apogee Instruments Inc., Logan, UT, USA) have a measurement range of $0\text{--}2000\text{ W}\cdot\text{m}^{-2}$ of estimated broadband irradiance based on a measurement of the sensor in the spectral range of $360\text{--}1120\text{ nm}$ which is mapped to a voltage output of $0\text{--}5\text{ V}$. The manufacturer specifies a linearity of better than 1% and a directional cosine response better than 5% (Further information can be found in the manufacturer's manual <https://www.apogeeinstruments.com/content/SP-212-215-manual.pdf>). The analogue/digital converters on the Arduino provide a resolution of 10 bit (1024 counts) on the range of $0\text{--}5\text{ V}$. This results a resolution of $1.95\text{ W}\cdot\text{m}^{-2}$ for both the incoming and reflected sensors. While these sensors are calibrated in a way that provides a reading of full solar shortwave flux ($280\text{--}4000\text{ nm}$) in a typical cloud free scenario, care must be taken because under cloudy skies and above ice and snow the measured fluxes do not necessarily represent true broadband shortwave radiation fluxes. These uncertainties can affect our final albedo values. According to the manufacturer specification the calibration uncertainty is 5%, which is a maximum of $100\text{ W}\cdot\text{m}^{-2}$. This is manifested mainly in an unknown calibration offset, so that the value measured in the dark does not equal $0\text{ W}\cdot\text{m}^{-2}$. This offset was found to vary between different power cycles of the instrument. To overcome this problem, we developed an easy in-field calibration routine: After start-up of the instrument, the optical heads are covered in completely opaque material while twenty measurements of the dark value were recorded. The average of these dark values was subtracted from the data during post-processing. An easy way for complete covering of the instruments is to put it firmly under the warm field clothes worn by the investigator. This procedure was capable to reduce the absolute calibration uncertainty of both sensors to approximately $5\text{ W}\cdot\text{m}^{-2}$. This accounts for a typical albedo uncertainty of 2% for a downwelling shortwave irradiance of $500\text{ W}\cdot\text{m}^{-2}$. However, at low incident fluxes, the associated measurement uncertainty increases significantly to 5% at $200\text{ W}\cdot\text{m}^{-2}$ and 10% at $100\text{ W}\cdot\text{m}^{-2}$. The device is thus suitable for albedo determination in typical polar summer investigations where incident fluxes exceed $200\text{ W}\cdot\text{m}^{-2}$, but it is of limited use for low incident light scenarios such as in late autumn or winter.

2.3 Field tests

Field test data was acquired during nine surveys on four different ice floes, covering a wide range of incident light and sea ice surface conditions: On 18 and 21 July 2018, two different ice floes were surveyed in Baffin Bay as part of the Sentinel North International PhD School on board the Canadian coast guard icebreaker *CCGS Amundsen*. Multiple smaller surveys were carried out on each floe by different

student groups being introduced to the device. On 21 and 29 August 2018, one floe in the vicinity of the geographic North Pole was sampled repeatedly. This long term station enabled extensive surveys covering the entire circumference of the ice floe.

To account for drift motion of all ice floes, GPS data were corrected using either logs from the ship's GPS receiver (18 July, stationary ship), or from GPS loggers separately deployed at a fixed location on the ice floe. Aerial images of the sampled ice floes were taken using helicopters and remotely controlled unmanned aerial vehicles. Several oblique photographs from multiple viewing angles were then processed into orthographic photos using the "PhotoScan Standard" photogrammetry software (Agisoft LLC, St. Petersburg, Russia).

3 Results

Results for all surveys are summarized in Table 2. Figure 4 provides spatial maps of all four survey days, while Figure 5 shows the respective histograms of albedo distribution. Figure 6 provides aerial images of the investigated ice floe surfaces.

On both 18 and 21 July, albedo values exhibit a bimodal distribution due to the typical surface dominated by extensive melt pond cover. Modal bare ice albedo on 18 July is 0.7, significantly higher than the value of 0.6 measured on 21 July. This is likely caused by the thinner ice and subsequently wetter surface on the latter station. The ice floe sampled on 21 July shows distinct linear pond patterns (Figure 6b) and was formed from landfast sea ice which had just broken out from its initial position a few weeks before sampling, as confirmed by radar satellite imagery (not shown). The associated modal albedo value of ponded sea ice on this floe is 0.2. The floe surveyed on 18 July shows a larger variety of surface features (Figure 6a) which apart from higher pond albedos also manifests in a wider distribution of pond albedo values.

Measurements conducted on 21 August also comprise a large range of albedo values, but as melt ponds are already covered by ice and some snow cover (Figure 6d), the bimodal distribution is not present. Due to the low incoming fluxes, under $50\text{ W}\cdot\text{m}^{-2}$, and the associated albedo uncertainty in this measurement setup, the distribution includes a large number of albedo values larger than one. This is reduced somewhat in the full floe survey due to higher incoming fluxes. Increased fresh snow cover (Figure 6e) however drastically reduces surface variations and the albedo histogram is characterized by a narrow single mode of 0.95. As this value is for a limited spectral range of $380\text{--}1100\text{ nm}$ it is higher than the literature value of 0.85 (Perovich and Polashenski, 2012) given for the full solar shortwave range ($200\text{--}2000\text{ nm}$) due to the significantly higher spectral albedo of snow in the visible compared to the infrared part of the spectrum.

Table 2 Summary of the conducted albedo surveys, ice properties, mean incident and reflected radiation as well as calibrated mean albedos with standard deviations. Also included are the measured dark offsets including their standard deviation and the derived maximum error related to calibration uncertainties

Expedition	<i>Amundsen</i>	<i>Amundsen</i>	<i>Amundsen</i>	<i>Amundsen</i>	<i>Amundsen</i>	<i>Amundsen</i>	<i>Oden</i>	<i>Oden</i>	<i>Oden</i>
Survey name	Floe 3-1	Floe 3-2	Floe 3-3	Floe 4-1	Floe 4-2	Floe 4-3	Survey area	Floe-part 1	Floe-part 2
Start date	18 July 2018	18 July 2018	18 July 2018	21 July 2018	21 July 2018	21 July 2018	21 August 2018	29 August 2018	29 August 2018
Start time /(UTC)	16:43	17:27	18:29	15:56	18:42	20:19	13:52	09:30	12:35
End time /(UTC)	17:02	17:29	19:01	16:06	18:47	21:10	14:28	11:40	13:30
Latitude/(°N)	67.8115			67.5326			89.5894	89.5705	
Longitude/(°E)	-62.3122			-63.5628			20.3213	50.7932	
Thickness of bare sea ice	131			101			200	200	
Thickness of ponded sea ice	70			41			140	140	
Snow thickness	0			0			2	5	
Number of measurements	221	99	132	552	23	571	325	2630	1283
Mean incident radiation/(W·m ⁻²)	750	720	717	863	716	639	44	82	87
Σ incident radiation/(W·m ⁻²)	43	61	86	62	54	67	15	13	18
Mean reflected radiation/(W·m ⁻²)	504	396	414	434	318	297	40	73	85
σ reflected radiation/(W·m ⁻²)	49	136	93	120	128	106	13	11	16
Mean albedo	0.67	0.55	0.60	0.50	0.44	0.47	0.97	0.91	0.98
σ albedo	0.08	0.19	0.30	0.14	0.18	0.17	0.35	0.12	0.07
Calibration offset incident sensor/(W·m ⁻²)	59	57	22	59	59	59	71	50	50
σ calibration offset incident sensor/(W·m ⁻²)	17	9	6	8	8	8	8	23	23
Calibration offset reflected sensor/(W·m ⁻²)	6	8	-4	-1	-1	-1	11	22	22
σ calibration offset reflected sensor/(W·m ⁻²)	3	4	2	1	1	1	8	13	13
Max. albedo uncertainty calibration	0.02	0.01	0.01	0.01	0.01	0.00	0.36	0.55	0.55

Apart from the inherent albedo uncertainty associated with low incident fluxes, the albedo wand successfully provided a large number of reliable georeferenced albedo measurements. The simple battery packs lasted for more than two hours of surveying in below 0°C conditions. Operations were reliable and could easily be performed in parallel with transect measurements of snow-thickness and electromagnetic measurements of ice thickness.

4 Discussion

The price of the albedo wand is mostly driven by the connected sensors. Here we chose off-the-shelf sensors of a medium price in the order of 200 €. This comes with the disadvantage of a limited spectral range. Due to the sensor's calibration, the absolute numbers given for total shortwave (280–4000 nm) fluxes are thus biased high by up to 15%. These biases due to the spectral composition of the incoming light field need to be kept in mind when interpreting albedo data. However the bias is acceptable in the low-cost solution presented here. Using spectral albedo

data for melt ponded and bare sea ice from Light et al. (2015), as well as reference solar spectra from Grenfell and Perovich (2008) for overcast sky and the spectral response of the detector, we estimate that these biases are typically about 3% for ponded and 10% of the respective albedo value for bare ice. This drawback could be easily overcome by using more expensive traditional pyranometers covering the entire spectral range from 280–4000 nm. The albedo wand can also be equipped with either cheaper custom built sensors or be upgraded to state of the art scientific sensors that would significantly reduce the calibration uncertainty. The exact wiring and calibration procedures will need to be adjusted, but the Arduino can accommodate most existing sensor interfaces. Even small and lightweight hyperspectral irradiance sensors, such as the STS-VIS-RAD (Ocean Optics, Largo, FL, USA) could be linked to two software serial ports on the Arduino.

The open design makes it very easy to adapt the device to a specific mission. Implementing continuous recording requires only a few lines of extra code and could allow the easy recording of albedo transect measurements e.g. when

the sensors are mounted to a snowmobile. With an adapted power supply, the system could in turn also be used for stationary time series albedo measurements.

Currently sensor levelling is controlled by a simple bubble level attached to the sensor holder. To further improve the calibration stability, we suggest to perform dark measurements not only before, but also after a set of albedo measurements.

A further useful addition would be the incorporation of an attitude sensor recording pitch and roll values. This

would allow the rejection of not well levelled data points during post processing and a real time control of the achieved levelling accuracy achieved with the bubble level. At lower latitudes, a heading sensor would also allow a more precise post processing of the surveyed positions: The heading value enables a correction of the offset between the GPS in the casing and the sensor holder. In the current setup, four pins on the Arduino remain free which can be used for the integration of such auxiliary sensors.

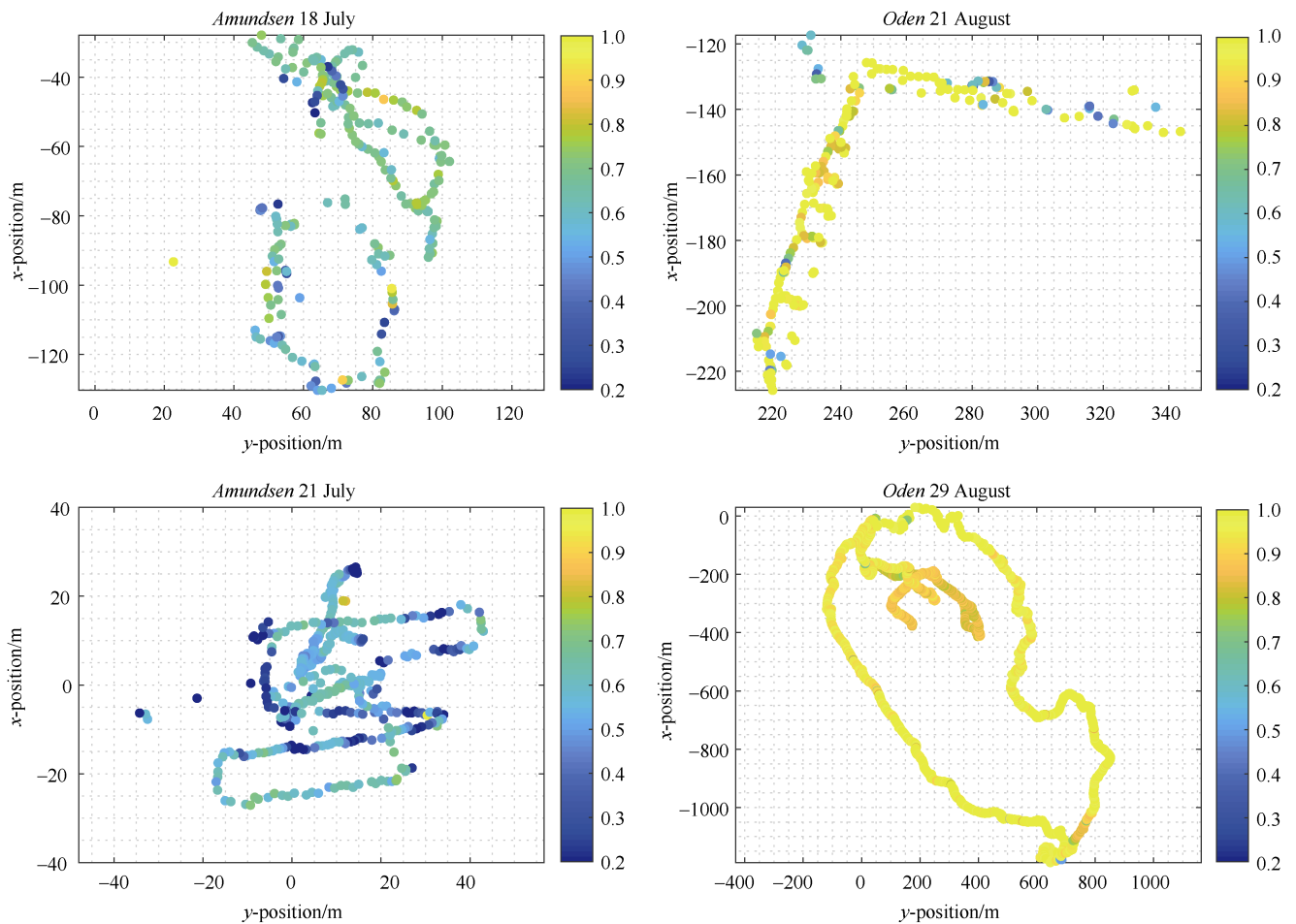


Figure 4 Maps of spatially distributed albedo measurements acquired during the four presented surveys. Note that the positions from 18 July, 2018 are not necessarily accurate, as the vessel, which was used as GPS base station for drift correction, might have moved in between the three different surveys. Also note the different spatial scales of the maps, where measurements taken on 29 August, 2018 have the largest spatial extent.

5 Summary and conclusion

Here we presented the complete open source design for a device able to record spatially distributed measurements of sea ice surface albedo (360–1120 nm). The open design provides researchers across the world with easily accessible and affordable technology fit for use in the rough environmental conditions of the Earth's polar regions. The design can be easily adapted to different sensing scenarios, where the spatial distribution of a measured parameter is to be recorded using a single measurement device.

We have shown that this device can provide science grade data, as long as the incident solar fluxes are significant. Data quality and comparability to total solar shortwave fluxes (280–4000 nm) could be further improved by the use of more expensive sensors, but we assume that the achieved accuracy of our setup is sufficient for most applications.

Due to its easy operation and low logistical footprint, the device can easily be incorporated into existing measurement workflows and ease the acquisition of previously manually logged data and thus increases its scientific potential.

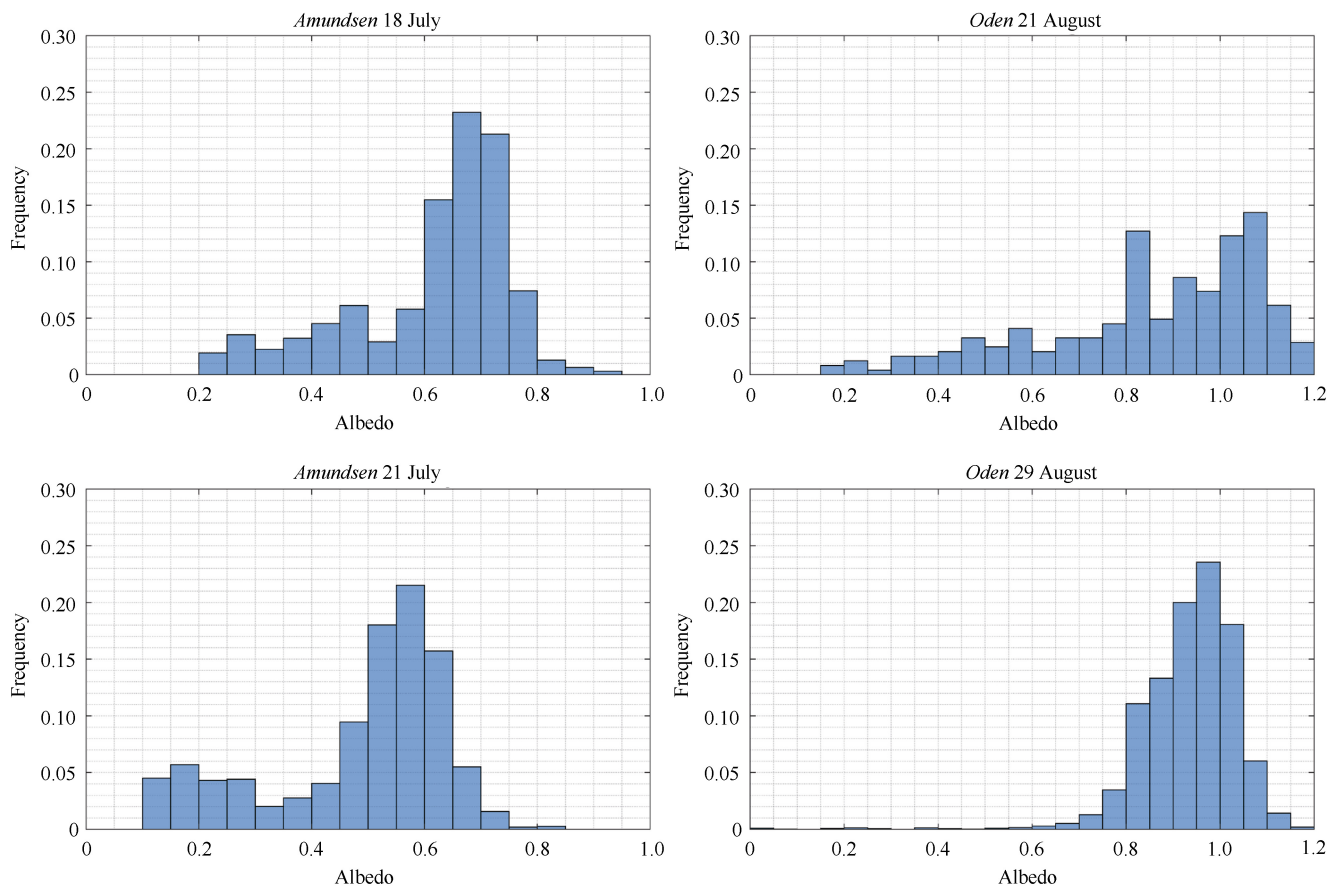


Figure 5 Histograms of albedo measurements acquired during the four presented surveys. Note that albedo values during the two *Oden* surveys include values >1 due to calibration uncertainties resulting from the low incident solar radiation.

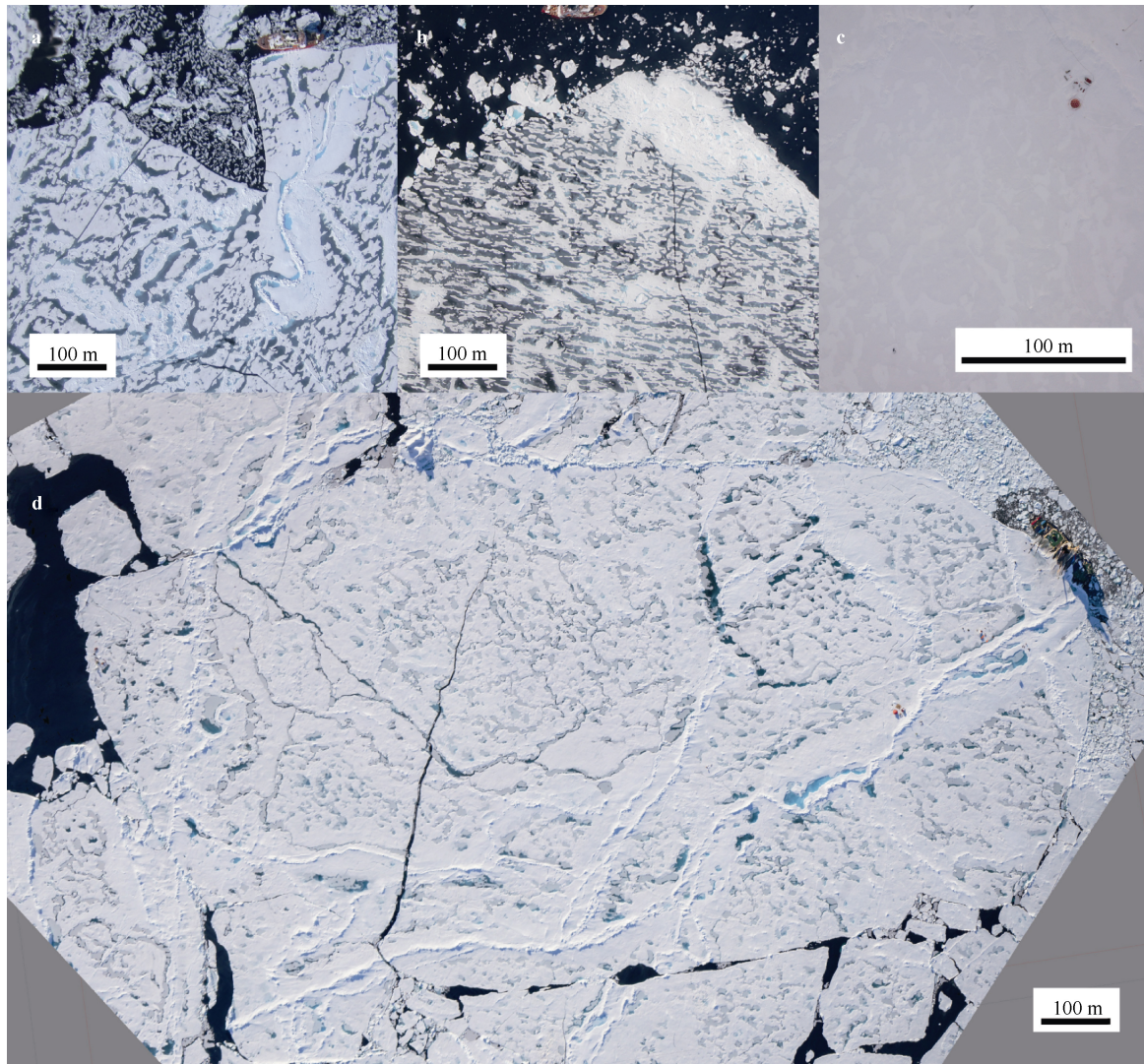


Figure 6 Processed orthographic photos of the investigated survey areas taken during helicopter flights: **a**, Summer drift ice in Baffin Bay on 18 July; **b**, Broken off land-fast sea ice near Qikiqtarjuaq, Baffin Island on 21 July 2018; **c**, Survey area investigated on 21 August 2018 during the drift of icebreaker *Oden* as photographed by drone on 13 September 2018; **d**, Overview over the entire ice floe of the *Oden* drift as surveyed on 29 August 2018, photographed on 16 August 2018 five days before the first albedo survey.

Acknowledgements I want to thank the Alfred-Wegener-Institut Helmholtz-Zentrum für Polar und Meeresforschung, and the Helmholtz infrastructure initiative “Frontiers in Arctic marine Monitoring” (FRAM) for funding this project. The writing of this manuscript was funded by a Sentinel North Postdoctoral Research Fellowship at Université Laval, Canada. Also, I want to thank the captain and crew, as well as all participants and mentors of the Sentinel North international PhD school (IPS) on board *CCGS Amundsen* — in particular Simon Lambert Girard and Marcel Babin — for their help during the cruise in Baffin Bay. Similarly I like to thank the captain, crew and SPRS support staff from the Swedish icebreaker *IB Oden* for their friendly welcome. In particular, Mario Hoppmann, Philipp Anhaus and Lars Lehnert are acknowledged for

their help during the measurements. A special thanks goes to the helicopter pilots on both ships providing the platform for aerial pictures. The design heavily relies on example code and the libraries “SPI”, “SD”, “Adafruit_GPS”, “SoftwareSerial” and “LiquidCrystal” generously provided by the Arduino community. I also want to thank Anja Nicolaus for her help in the procurement of most necessary components. I thank the two anonymous reviewers and Editor Ian Allison, who helped improving this manuscript with valuable comments.

References

- Barnard H R, Findley M C, Csavina J. 2014. PARduino: a simple and inexpensive device for logging photosynthetically active radiation. *Tree Physiology*, 34(6): 640-645, doi:10.1093/treephys/tpu044.
- Curry J A, Schramm J L, Ebert E E. 1995. Sea ice-albedo climate feedback mechanism. *J Clim*, 8(2): 240-247, doi:10.1175/1520-0442(1995)008<0240:SIACFM>2.0.CO;2.
- Fetterer F, Untersteiner N. 1998. Observations of melt ponds on Arctic sea ice. *J Geophys Res-Oceans*, 103(C11): 24821-24835, doi:10.1029/98jc02034.
- Flanner M G, Zender C S. 2006. Linking snowpack microphysics and albedo evolution. *J Geophys Res*, 111(D12): D12208, doi:10.1029/2005jd006834.
- Flocco D, Schroeder D, Feltham D L, et al. 2012. Impact of melt ponds on Arctic sea ice simulations from 1990 to 2007. *J Geophys Res-Oceans*, 117(C9): C09032, doi:10.1029/2012JC008195.
- Grenfell T C, Perovich D K. 2008. Incident spectral irradiance in the Arctic Basin during the summer and fall. *J Geophys Res-Atmos*, 113(D12), doi:10.1029/2007jd009418.
- Hudson S R, Granskog M A, Karlsen T I, et al. 2012. An integrated platform for observing the radiation budget of sea ice at different spatial scales. *Cold Reg Sci Tech*, 82: 14-20, doi: 10.1016/j.coldregions.2012.05.002.
- Light B, Perovich D K, Webster M A, et al. 2015. Optical properties of melting first-year Arctic sea ice. *J Geophys Res-Oceans*, 120(11): 7657-7675, doi:10.1002/2015JC011163.
- Nicolaus M, Gerland S, Hudson S R, et al. 2010a. Seasonality of spectral albedo and transmittance as observed in the Arctic Transpolar Drift in 2007. *J Geophys Res-Oceans*, 115: C11011, doi:10.1029/2009jc006074.
- Nicolaus M, Hudson S R, Gerland S, et al. 2010b. A modern concept for autonomous and continuous measurements of spectral albedo and transmittance of sea ice. *Cold Reg Sci Tech*, 62(1): 14-28, doi:10.1016/j.coldregions.2010.03.001.
- Perovich D K. 1990. Theoretical estimates of light reflection and transmission by spatially complex and temporally varying sea ice covers. *J Geophys Res-Oceans*, 95(C6): 9557-9567, doi:10.1029/JC095iC06p09557.
- Perovich D K. 1996. The optical properties of sea ice. Monograph 96-1: 31. US Army Corps of Engineers: Cold Regions Research & Engineering Laboratory.
- Perovich D K. 2005. On the aggregate-scale partitioning of solar radiation in Arctic sea ice during the Surface Heat Budget of the Arctic Ocean (SHEBA) field experiment. *J Geophys Res-Oceans*, 110(C3): C03002, doi:10.1029/2004jc002512.
- Perovich D K, Grenfell T C, Light B, et al. 2002. Seasonal evolution of the albedo of multiyear Arctic sea ice. *J Geophys Res-Oceans*, 107(C10): SHE 20-1-SHE 20-13, doi:10.1029/2000jc000438.
- Perovich D K, Polashenski C. 2012. Albedo evolution of seasonal Arctic sea ice. *Geophys Res Lett*, 39(8): L08501, doi:10.1029/2012gl051432.
- Perovich D K, Richter-Menge J A. 2006. From points to Poles: extrapolating point measurements of sea-ice mass balance. *Ann Glaciol*, 44(1): 188-192, doi:10.3189/172756406781811204.
- Perovich D K, Roesler C S, Pegau W S. 1998. Variability in Arctic sea ice optical properties. *J Geophys Res-Oceans*, 103(C1): 1193-1208, doi:10.1029/97jc01614.
- Sturm M, Holmgren J. 2018. An automatic snow depth probe for field validation campaigns. *Water Resour Res*, 54(11): 9695-9701, doi:10.1029/2018WR023559.
- Wang C, Granskog M A, Gerland S, et al. 2014. Autonomous observations of solar energy partitioning in first-year sea ice in the Arctic Basin. *J Geophys Res-Oceans*, 119(3): 2066-2080, doi:10.1002/2013JC009459.

Appendix-1

Supplementary material

1 CAD files for 3D-printed parts

The following files include the designs for all three 3D-printed parts shown in Figure 2. The files were generated using FreeCAD 0.16 (<https://www.freecadweb.org/>). “.FCStd”-files can be opened in FreeCAD to modify the design, while “.stl” files can be directly read into most slicing software for 3D-printing:

- albedo_wand_bar.stl
- albedo_wand_bar.FCStd
- albedo_wand_sensor.stl
- albedo_wand_sensor.FCStd
- albedo_wand_clamp.stl
- albedo_wand_clamp.FCStd

2 Code for the Arduino Micro-Controller

/* ALBEDO WAND
This is the code for the Albedo Wand

Wiring:

```
* analog sensors on A0, A1,
* GPS on A2 A3
* SD card attached to SPI bus as follows:
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 10
* Display attached as follows:
**RS - pin 8
**EN - pin 7
**DB4- pin 6
**DB5- pin 5
**DB6- pin 4
**DB7- pin 9
```

Version 1.0 running

(c) Christian Katlein (Alfred-Wegener-Institut), ckatlein@awi.de
based on various example codes and the included libraries from the
Arduino Community*/

```
#include <SPI.h> // for SD card
#include <SD.h> // for SD card
#include <Adafruit_GPS.h> //for GPS
#include <SoftwareSerial.h> //for GPS
#include <LiquidCrystal.h> //for LCD Display
```

```
// Deboune definitions for Button
const int buttonPin = 2; // the number of the pushbutton pin
int buttonState = LOW; // the current reading from the
input pin
int lastButtonState = LOW; // the previous reading from the input pin
unsigned long lastDebounceTime = 0; // the last time the output pin was
toggled
```

```
unsigned long debounceDelay = 500; // the debounce time; increase if
the output flickers
```

```
////////// INITIALIZE LCD //////////// LCD
//////////
```

```
// initialize the LCD interface pins
// with the arduino pin number it is connected to
const int rs = 8, en = 7, d4 = 6, d5 = 5, d6 = 4, d7 = 9;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
////////// INITIALIZE GPS //////////// GPS ////////////
SoftwareSerial GPSSerial(A3, A2); // RX, TX software serial ports
Adafruit_GPS GPS(&GPSSerial); // connect to GPS
// Set to 'true' if you want to debug and listen to the raw GPS sentences
#define GPSECHO false
```

```
////////// INITIALIZE SD //////////// SD ////////////
const int chipSelect = 10; // The Chip Select pin for SD card here it is 10
```

```
void setup() {
```

```
pinMode(buttonPin, INPUT);
```

```
////////// SETUP SERIAL ////////////
```

```
Serial.begin(9600); // Open hardware serial communications for
development and debugging
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}
```

```
////////// SETUP DISPLAY ////////////
```

```
pinMode(A4, OUTPUT); //Pull down A4 to ground level to pull pin 5 of
LCD to ground
digitalWrite(A4, LOW);
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a welcome message to the LCD.
lcd.print("AWI Albedo-wand:");
lcd.setCursor(0, 1);
lcd.print("Version: 1.0");
delay(3000);
```

```
////////// SETUP GPS ////////////
```

```
Serial.println("Adafruit GPS library basic test!");
// Open software serial for GPS serial communications and wait for
port to open:
// 9600 NMEA is the default baud rate for Adafruit MTK GPS's- some
use 4800
GPS.begin(9600);
//** uncomment this line to turn on RMC (recommended minimum) and
GGA (fix data) including altitude
GPS.sendCommand(PMTK_SET_NMEA_OUTPUT_RMCGGA);
// Set the update rate
GPS.sendCommand(PMTK_SET_NMEA_UPDATE_1HZ); // 1 Hz
update rate
// For the parsing code to work nicely and have time to sort thru the data,
and
// print it out we don't suggest using anything higher than 1 Hz
```

```
// Request updates on antenna status, comment out to keep quiet
GPS.sendCommand(PGCMD_ANTENNA);
```

```
// Ask for firmware version
GPSSerial.println(PMTK_Q_RELEASE); /*
delay(1000);
```

```
////////// SETUP SD CARD ////////////
```

```
Serial.println("Initializing SD card...");
```

```

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
  Serial.println("Card failed, or not present");
  lcd.clear();
  lcd.print("Card failure");
  // don't do anything more:
  return;
}
Serial.println("card initialized.");
lcd.clear();
lcd.print("SD-Card OK");
delay(3000);

//Write File Header
File dataFile = SD.open("albedo.txt", FILE_WRITE);

dataFile.println("YYYY,MM,DD,HH,MM,SS,FFF,LAT,LON,IN,REF,AL
BEDO");
dataFile.close();
lcd.clear();
lcd.print("READY");
}
////////// LOOP LOOP LOOP
//////////
void loop() {

  attachInterrupt(digitalPinToInterrupt(2), single_sample, RISING); //
  Button Function

  //Read and parse GPS
  char c = GPS.read();
  // if (c) Serial.print(c);
  if (GPS.newNMEAreceived()) {
    // Serial.println(GPS.lastNMEA()); // this also sets the
newNMEAreceived() flag to false
    if (!GPS.parse(GPS.lastNMEA())) // this also sets the
newNMEAreceived() flag to false
      return; // we can fail to parse a sentence in which case we
should just wait for another
  }

}

////////// BUTTON //////////////////////////////////// BUTTON
//////////

void single_sample() {

  //Read GPS //////////////////////////////////// GPS ////////////////////////////////////
  /*
  Serial.print("\nTime: ");
  Serial.print(GPS.hour, DEC); Serial.print('.');
  Serial.print(GPS.minute, DEC); Serial.print('.');
  Serial.print(GPS.seconds, DEC); Serial.print('.');
  Serial.println(GPS.milliseconds);
  Serial.print("Date: ");
  Serial.print(GPS.day, DEC); Serial.print('/');
  Serial.print(GPS.month, DEC); Serial.print("/20");
  Serial.println(GPS.year, DEC);
  Serial.print("Fix: "); Serial.print((int)GPS.fix);
  Serial.print(" quality: "); Serial.println((int)GPS.fixquality);
  if (GPS.fix) {
    Serial.print("Location: ");
    Serial.print(GPS.latitude, 4); Serial.print(GPS.lat);
    Serial.print(", ");

```

```

    Serial.print(GPS.longitude, 4); Serial.println(GPS.lon);
    Serial.print("Speed (knots): "); Serial.println(GPS.speed);
    Serial.print("Angle: "); Serial.println(GPS.angle);
    Serial.print("Altitude: "); Serial.println(GPS.altitude);
    Serial.print("Satellites: "); Serial.println((int)GPS.satellites);
  } */

```

```

// make a string for assembling the data to log:
String dataString = "";
dataString += "20";
dataString += String(GPS.year, DEC);
dataString += ',';
dataString += String(GPS.month, DEC);
dataString += ',';
dataString += String(GPS.day, DEC);
dataString += ',';
dataString += String(GPS.hour, DEC);
dataString += ',';
dataString += String(GPS.minute, DEC);
dataString += ',';
dataString += String(GPS.seconds, DEC);
if (GPS.fix) {
  dataString += ',';
  dataString += String(convertDegMinToDecDeg(GPS.latitude),6);
  dataString += ',';
  dataString += String(convertDegMinToDecDeg(GPS.longitude),6);
  dataString += ',';
}
else {
  dataString += ',';
  dataString += ',';
  dataString += ',';
}
}

```

```

int sensor1 = analogRead(A0); //First sensor
sensor1=(sensor1*4.9*0.25)-20; // W/m^2 = counts * 4.9 mV/count
*0.25
dataString += String(sensor1);
dataString += ",";

```

```

int sensor2 = analogRead(A1); //Second sensor
sensor2=(sensor2*4.9*0.25)-20; // W/m^2 = counts * 4.9 mV/count
*0.25
dataString += String(sensor2);
dataString += ",";

```

```

float albedo;
albedo = (float)sensor2 / (float)sensor1; // Albedo
dataString += String(albedo);

```

```

////////// Display data on LCD ////////////////////////////////////

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("I:");
lcd.print(sensor1);

```

```

lcd.setCursor(7, 0);
lcd.print("R:");
lcd.print(sensor2);

```

```

lcd.setCursor(13, 0);
lcd.print("W/m");

```

```

lcd.setCursor(0, 1);
lcd.print("A:");
lcd.print(String(albedo, 2));
lcd.setCursor(7, 1);
lcd.print("GPS:");
lcd.print(GPS.fix);

```

```

////////// WRITE DATA TO FILE ////////////////////////////////////
// open the file. note that only one file can be open at a time,

```

```

// so you have to close this one before opening another.
File dataFile = SD.open("albedo.txt", FILE_WRITE);
  Serial.println(dataString);

// if the file is available, write to it:
if (dataFile) {
  dataFile.println(dataString);
  dataFile.close();
  // print to the serial port too:
  Serial.println(dataString);
}
// if the file isn't open, pop up an error:
else {
  Serial.println("error opening albedo.txt");
  lcd.setCursor(13, 1);
  lcd.print("ERR");
}

delay(500);

}

////////////////////////////////////// CONVERT GPS COORDINATES 2 DECIMAL
DEGREES
//required for fmod()
#include <math.h>

// converts lat/long from Adafruit
// degree-minute format to decimal-degrees
double convertDegMinToDecDeg (float degMin) {
  double min = 0.0;
  double decDeg = 0.0;

  //get the minutes, fmod() requires double
  min = fmod((double)degMin, 100.0);

  //rebuild coordinates in decimal degrees
  degMin = (int) ( degMin / 100 );
  decDeg = degMin + ( min / 60 );

  return decDeg;
}

```