On using IoT Protocols for Automation and Monitoring Solar Tracker Devices

Renan Augusto Starke[1], Allan Ricardo Starke[2], Julio Nelson Scussel[2] and Sergio Colle[2]

[1] DAELN, Departament of Eletronics, Federal Institute of Santa Catarina (IFSC), Florianópolis (Brazil)

[2] LEPTEN - Laboratory of Energy Conversion Engineering and Energy Technology, Department of Mechanical Engineering, Federal University of SantaCatarina (UFSC), Florianópolis (Brazil)

Abstract

It is known that measurements of long-term solar radiation are climatically, economically and anthropologic important. Moreover, reliable and low uncertainty irradiation measurements are fundamental for assessment of solar system and for improve the bankability of systems. This work describes a compact control system for a solar tracker, which was successfully tested to an Eppley SMT tracker. The control unit uses a single board computer BeagleBone Black, managed by MQTT IoT network protocol and azimuth and zenith angles are estimated by NREL's SPA algorithm. Since the goal is to reduce the number of problems related to the measurements, mainly to the miss alignment, we proposed a supervisory system, composed by web monitoring, alarms, cameras, databases, time-lapses videos, among other possible services. The supervisory system is simple to implement and uses Grafana, that allows to obtain, visualize data and quality control, alert and understand any metrics independent of the type of storage.

Keywords: Solar tracker control unit, IoT, supervisory system, single board computer.

1. Introduction

Solar irradiation is directly responsible for our climate, atmospheric circulation and ocean currents. Small variations in the energy balance at the Earth's surface, if persistent, can lead to significant climatic, economic and anthropologic effects (Chylek et al., 2007; Knox & Untersteiner, 2011 apud. Driemel et al., 2018). Irradiation data have such importance in atmospheric studies – validation of satellite models, validation of atmospheric radiative transfer models, weather research and forecasting models and monitoring long-term variation – that in 1988 the WMO proposed the creation of the Baseline Surface Radiation Network (BSRN) (Driemel, Augustine, Behrens, & Etal., 2018) which provides measurements of solar irradiation with high temporal resolution.

On the other hand, the search for means for supply the energy demand of the modern society while reducing GHG (green-house gases) emissions and other environmental impacts has increased interest in renewable energies (RE). Among the renewable technologies, the solar energy stands out, not only for its high operating potential, but for the wide range of technologies that can contribute to supplying the demand for various energy services. According to (Polo et al., 2015), the proper design and consequent success of solar energy systems requires adequate knowledge of the components of solar irradiation. Solar concentration systems (CSP) require accurate and reliable measurements of direct solar irradiation (DNI). In the case of flat (PV or thermal) collectors, precise data are required for the total slope irradiation (GTI), which can be calculated through the three components, global irradiation (GHI), diffuse irradiation (DHI) and DNI. For example, (Hirsch, 2017) estates that in performance evaluation of CSP plants, direct normal irradiance data (DNI) present the third largest contribution in the uncertainties of the results.

Moreover, to assess the financial risk of the project, the performance evaluation of the CSP system should consider the long-term inter-annual variability of the solar resource, therefore, solar irradiance data series with 10, 20 or 30 year must be available (Ramírez et al., 2017). Although irradiance data derived from satellite models achieved acceptable uncertainties, and can be used for performance evaluation, ground measurements of solar irradiance data are still fundamental. These measurements are fundamental to validate and identify uncertainties of satellite models, to carry out measurement campaigns at potential locations for the installation of solar systems, and provide irradiance data with high temporal resolution (minutes), which are fundamental to assess the dynamic behavior of the solar systems and increase the bankability of these projects (Ramírez et al.,

doi:10.18086/swc.2019.44.07 Available at http://proceedings.ises.org

2017).

Measurements of the three terrestrial solar irradiance components are performed through sensors that require accurate and reliable alignment with the sun. Sensors for DNI and DHI required an accurate and reliable alignment with the sun during all day. This because DNI uses a pyrheliometer normally aligned with the sun's rays and DHI uses a thermopile pyranometer with a shading ball that slowly traces the sun. GHI is measured by a thermopile pyranometer, without any kind of alignment. The pyrheliometer and DHI pyranometers are aligned with the sun using a solar tracker device. Such equipment is composed of electro-mechanical systems, electronics and algorithms (Reda & Andreas, 2004) that estimate the solar position. The SOLYS 2 Sun Tracker (Kipp & Zonen, 2019), Eppley SMT (Eppley Lab, 2019) and the work of (Roth, Georgiev, & Boudinov, 2004) are very good examples of solar tracker devices. In addition to trackers, a solar monitoring station requires a storage infrastructure for the collected data. Measuring irradiance data for such long periods, maintaining the standard recommended by BSRN requires a tremendous effort and resources, mainlining because of the cost of the equipment and rigorous inspections, cleaning and calibration schedules. To minimize the possibility of losing or discarding data, it is desirable an automation system that provides and interprets status information so that possible problems could be quickly detected and solved.

Solar station automation can be implemented using Internet of Thinks (IoT) network protocols, providing an infrastructure that ensures a better data reliability. Regarding the research activities of solar energy and IoT protocols, it is possible cite the following works (Chieochan & Ph, 2017; Phung, Villefromoy, & Ha, 2017; Williams & Qouneh, 2017). Where, (Chieochan & Ph, 2017) demonstrated a system of current and voltage intelligent sensors to monitor the charge of voltaic cells. (Williams & Qouneh, 2017) uses IoT protocols to control the solar positioning of voltaic cells in order to find the maximum output power. (Phung et al., 2017) demonstrated the use of IoT protocols to control and manage the energy flow collected by solar panels and a microgrid. Control optimization includes not only sensor data but also meteorological data and a fault tolerance system containing multiple controllers to maximize the capture of solar radiation.

We present two main contributions in this work. First, we describe a control system for a solar tracker implemented with a single embedded computer board. This system can handle real-time requirements to control stepper motors and functional requirements to process solar positioning algorithms and IoT-related network interfaces. The control system is successfully applied to an Eppley SMT tracker (Eppley Lab, 2019) without any electro-mechanical modifications, just the control board. The second contribution is our solar station supervisory system, composed by web monitoring, alarms, cameras, databases, time-lapses videos, among other possible services. The goal is to reduce the number of problems related to the measurements, mainly to the miss alignment, which can be hard to detect. We created all solutions using open source software and systems. Using IoT protocols on solar energy research is not new (Chieochan & Ph, 2017; Phung et al., 2017; Williams & Qouneh, 2017), but ours, as far as we know, it is an interesting approach for solar resource assessment and energy meteorology.

2. Overview of the tracker control board

In order to create a reliable and compact control system, we use a single board computer BeagleBone Black (Beagleboard, 2019). This board uses an AM3358 System-On-Chip (SoC) which embedded an ARM Cortex A8 general-purpose processor and a Programmable Real-Time Unit and Industrial Communication Subsystem (PRU-ICSS). The general-purpose processor runs Debian GNU/Linux 9.x operating system and a high-level application responsible for solar positioning calculations, IoT protocols, logs and a simple text-mode user interface. The PRU contains two 32-bit 200MHz RISC cores and they have their own local memory allocation, can access general purpose pins, trigger interrupts and share memory with the Linux host device. Those cores are a valuable resource since they provide support for interfacing applications that have hard real-time constraints. The software we implemented for the PRU is responsible for low-level protection and generation of digital pulses sent to the power drivers attached to the stepper motors. Control pulses and protection signals must be held by a real-time subsystem to avoid timing interferences from the operating system and/or applications with lower priority.

The BeagleBone Black card allows the connection of expansion cards, also known as capes, which can be used to add application sensors and to implement electronics for signal conditioning of input and output data. Particularly for this project, we build a cape to embed a GPS module and some electronics for voltage conditioning for the stepper motors drivers. Fig. 1 presents an overview of the proposed tracker control unit and

Fig. 2 shows the physical implementation of the protype, where the BeagleBone is shown on the left and the original Eppley power unit and power drivers is shown on the right.

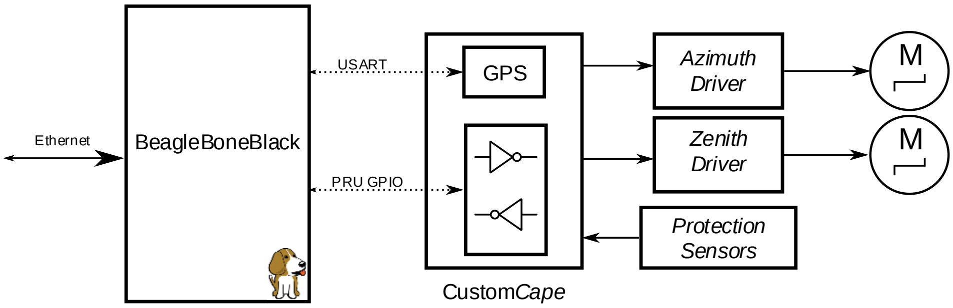


Fig. 1: Overview of the tracker control system

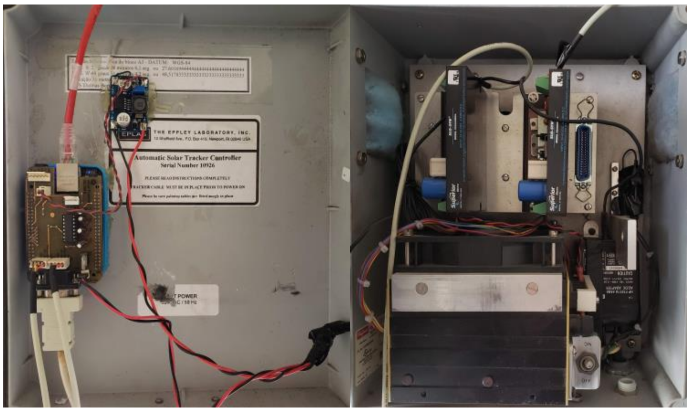


Fig. 2: Tracker control unit: Beaglebone with our custom cape (left) and stepper motor drivers (right).

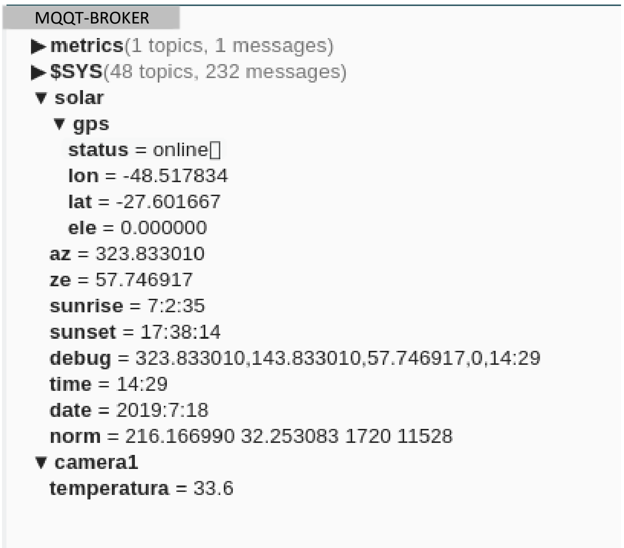
3. Overview of the tracker control software

The control software runs on the Debian GNU / Linux 9.x operating system. Only a few modifications on the configuration files were required, such as the activation of the PRU module (core activation and mapping of input and output pins - GPIO) and the serial transceiver (USART). Regarding the *kernel*, the 4.9.x-bone11 stable version was adopted. It is important highlight the *kernel* version, since there are some differences on the use of the PRU in different versions.

We use MQTT (MQTT.org, 2019) as IoT network protocol. It uses a publish/subscribe communication model where all messages are handled by a broker server. The broker is responsible for receiving, queuing, and sending messages from clients who publish data on a particular topic to clients subscribed to receive that information. In this sense, the tracker software periodically (each second) sends a message with the value of the azimuth angle in the topic "solar/az". Any application, as long as it achieves a network connection with the broker using MQTT protocol, can subscribe to "solar/az" and receive all values of the azimuth angle. This architecture is applied for all state variables of the solar tracker, as shown in Fig. 3. Moreover, the tracker also receives commands, such as setting and calibrating start positions, fine adjustments of movement, start and stop and other commands.

The developed tracker control system can be characterized as multithreaded and multiprocessing but with non-homogeneous processors, since part of the system is executed by Cortex A8 while another by PRU. Fig. 4 shows an overview of the tracker software, depicted the main modules that were developed. The application was implemented in C ++, C, and assembly with the use of some external components. The main class is called *SolarTracker*, which instantiates the GPS, MqttComm and PRU modules. The GPS class is responsible for receiving and interpreting data received from the GPS sensor that sends them in text format in the marine standard NMEA 0183 (Garmin, 2019). In addition to informing the geographical position of the tracker, the GPS provides the date and time for the system. The MqqtComm class is responsible for sending the state data and receiving the MQTT protocol commands. The low-level communication functions are implemented by the

Mosquitto external library (Mosquitto, 2019).



```
MQQT-BROKER
▶metrics(1 topics, 1 messages)
▶$SYS(48 topics, 232 messages)
▼solar
  ▼gps
    status = online
    lon = -48.517834
    lat = -27.601667
    ele = 0.000000
  az = 323.833010
  ze = 57.746917
  sunrise = 7:2:35
  sunset = 17:38:14
  debug = 323.833010,143.833010,57.746917,0,14:29
  time = 14:29
  date = 2019:7:18
  norm = 216.166990 32.253083 1720 11528
▼camera1
  temperatura = 33.6
```

Fig. 3: Status information of the tracker control system (broker topics)

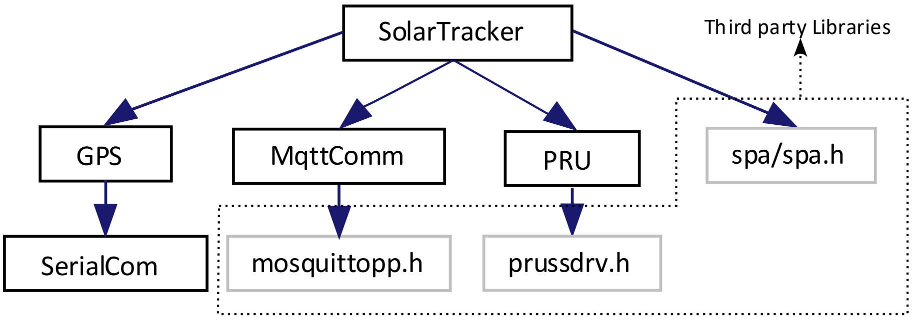


Fig. 4: Overview of the tracker software modules

The solar position calculation is performed by the external high precision Solar Position Algorithm – SPA library (Reda & Andreas, 2004), which can calculate the solar position (zenith and azimuth angles) in the period from the year -2000 to 6000, with uncertainties of ±0.0003 (degrees). A new solar position is calculated by SPA at a frequency of 1 Hz (synchronized by the GPS). After SPA calculations, zenith and azimuth angles are converted to stepper motor pulses: Eppley has a resolution of 0.01875 degrees/pulse. The necessary number of pulses to move to the new solar position is sent to the PRU core by the external library *prussdrv* (Molloy, 2014), which is responsible for low-level communication between the software running in the Linux operating system and the PRU. The MQTT communication system is implemented in a separate thread and does not interfere with the calculation of the solar position, in the same way that the generation of the movement pulses is independently performed by the PRU.

In order to validate our prototype, we used a camera recording the calibration shadow of the Eppley NIP pyrheliometer every 30 seconds. Depicted in Fig. 5 is shown two images generated by the calibration camera on February 8, 2019. About 1200 images are recorded daily and a time-lapse video is generated for visual inspection of the tracker. The alignment of the solar tracker can be confirmed by viewing the dot of light from the direct solar beam, which passes through the small hole in the front flange of the instrument and intercepts a target mounted on the rear flange. When the dot of light is centered with outer diameter of the black dot of the target there is an alignment error of 0.3 degree. In the case that the dot of light is centered with outer diameter of the white circle of the target there is an alignment error of 1.0 degree.

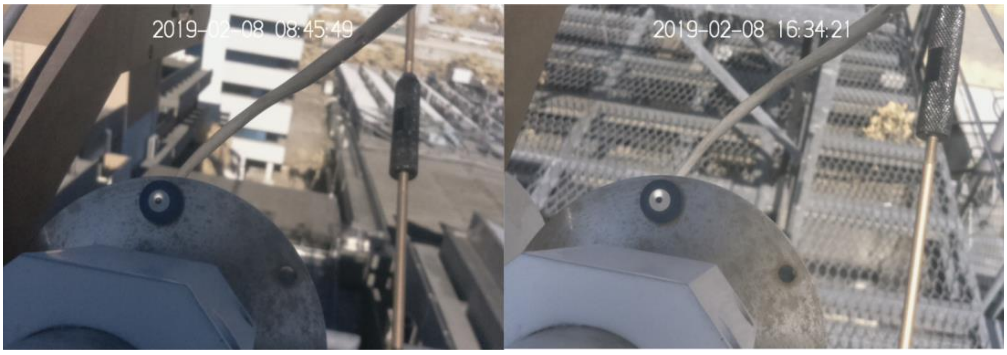


Fig. 5: Tracker calibration test on 8th February 2019.

4. Overview of the solar station monitoring system

Our solar station is composed by an Eppley SMT tracker, a CM21 and CM11 Kipp & Zonen pyranometers, for global and diffuse irradiance, respectively, both using the CV2 ventilation unit. An Eppley NIP pyrheliometer was used for the DNI measurements, and for the data acquisition system, we adopt a CR10X, integrated with the NL100 network link and PS100 power unit, the last three from Campbell Scientific. Fig. 6 shows the tracker and the acquisition system.

Fig. 6: Instruments and data acquisition system of the solar station.

Fig. 7 shows the design of our monitoring system. A solar station can contain several solar trackers and smart sensors connected through a TCP/IP network. These devices send and receive information using MQTT protocol and any device can connect to the broker (considering correct authentication and encryption) and obtain values of the various sensors as long as the MQTT topics are known. It is interesting to note that it is not necessary to know network addresses of the various devices since the broker server is responsible for redirecting all data.

In order to visualize real-time data from a specific tracker on a smartphone, we can use a simple MQTT dash application (Routix software, 2019), as shown in Fig. 8. The configuration screen is used to correct misalignments and issue operation mode commands to the tracker like start, stop, enter manual mode or turn it off.

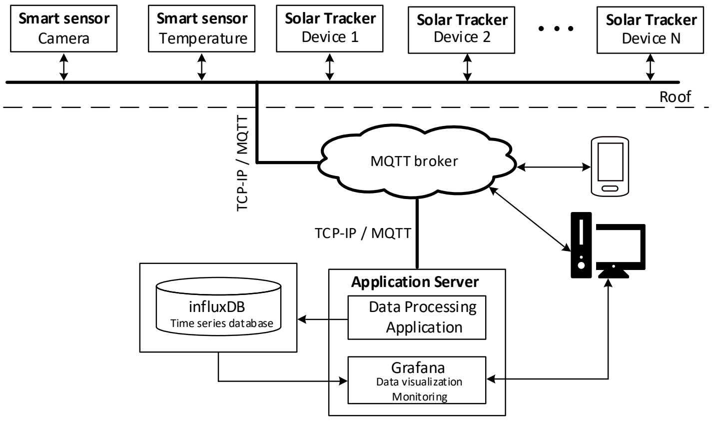
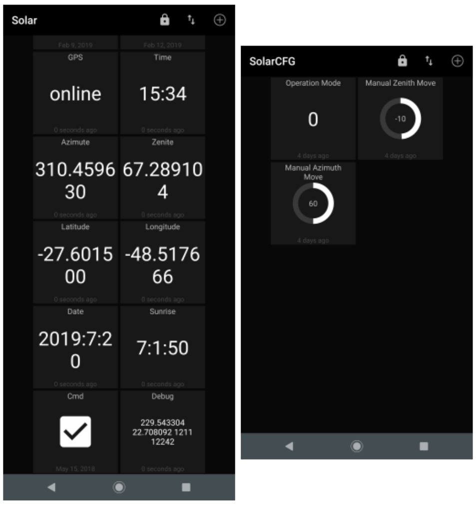


Fig. 7: Overview of the solar station monitoring system.


Solar

Feb 9, 2019 | Feb 12, 2019

GPS: online (0 seconds ago)
Time: 15:34 (0 seconds ago)
Azimute: 310.459630 (0 seconds ago)
Zenite: 67.289104 (0 seconds ago)
Latitude: -27.601500 (0 seconds ago)
Longitude: -48.517666 (0 seconds ago)
Date: 2019:7:20 (0 seconds ago)
Sunrise: 7:1:50 (0 seconds ago)
Cmd: ☑ (May 15, 2018)
Debug: 229.543304 22.708092 1211 12242 (0 seconds ago)

SolarCFG

Operation Mode: 0 (4 days ago)
Manual Zenith Move: -10 (4 days ago)
Manual Azimuth Move: 60 (4 days ago)

Fig. 8: MQTT dash: real-time information screen (left) and configuration screen (right)

The solar station data is stored and monitored by an *application server*. This application subscribes to all sensor and tracker topics and stores relevant information in an InfluxDB time-series database (InfluxData, 2019). We use Grafana (Labs, 2019) that allows to obtain, visualize, alert and understand any metrics independent of the type of storage in order to create our supervisory application. Fig. 9 shows the solar tracker metrics of our prototype on our WEB dash created in Grafana. It is possible to monitor the azimuth and zenith angles, camera board temperatures and the alignment of the solar tracker, which is based on the images generated by the calibration camera, installed on the flange of the pyrheliometer. With the aid of this image and the MQTT dash configuration screen, it is possible to remotely align the solar tracker.

Since we have a functional time-series database, we can also store irradiation data to create a quality control system that identifies errors, labels, ignores and informs invalid data. A sample of 6th June, 2019 of irradiation data is shown in Fig. 10.

The objective of the supervisory application, depicted on Fig. 10, is provide a real time graphical representation of the quality of the measured data. This facilitates the identification of problems on the stations, such as misaligned, dirty optics or any catastrophic failure. The Grafana tool allow us to implement alerts in various notification channels like email and instant messaging (Telegram, i.e.), this feature can help station operators identify problems as soon as possible, and prevent missing data for large periods, especially in the cases of unattended stations.

In order to implement and improve our quality control, we will follow the BSRN recommendations published by the World Meteorological Organization (König-Langlo et al., 2013), which consist of “flagging” the measurements as “Physical Possible Limits”, “Extreme Rare Limits” and “Comparisons”. In addition, we are also implementing the SERI QC (Maxwell, Wilcox, & Rymes, 1993), which consist of “flagging” each measurement based on dimensionless parameters – clearness index (K_t), effective diffuse horizontal transmittance (K_d), and direct beam transmittance (K_n). This method allows us to plot the measured data as a K_n versus K_t scatter plot (Fig. 10), which facilitates the visualizations of data that do not couple well in K-space.

As shown in Fig. 10, on the last sub-figure, the K-space identify the wrong measured data, especially when compared with the theoretical limit's lines. It is also possible to verify it the measured data is in the correct region of the K-space, corresponding to the atmospheric condition of measurement. If the measured data falls outside of the acceptance region of the K-space a set of alarm are triggered to warn the station staff. We also plot the diffuse fraction (d) versus clearness index (K_t), (Fig. 10), which also facilitates the visualization of data that are not physically possible, or rare events such as cloud enhancement. With our implementations is also possible to apply the work of (Moreno-Tejera, Ramírez-Santigosa, & Silva-Pérez, 2015), which provide a visualization tool of the quality "flags" and facilitates identify common errors such as time shifts, changes in recording frequency, misalignment of pyrheliometers, or nighttime and daytime offsets. As a further development the cameras updating the tracker status alignment (1200 images recorded daily), will be used to identify the alignment precisions of the tracker. In that context, it is possible to perform a statistical analysis of the imagens or use computer vision and machine learning to alert and correct possible tracker misalignments.

Fig. 9: Grafana dash created to visualize tracker information

Fig. 10: Grafana dash created to visualize irradiation data

5. Conclusions

It is known that measurements of long-term solar radiation are climatically, economically and anthropologic important. Thus, in this article we present a control system that uses a Beaglebone single board computer to control a two-axis solar tracker using NREL's solar position algorithm. The validation of the system was first performed through a camera recording the calibration shade of the pyrheliometer, in future works, we will compare irradiation measurements with a calibrated tracker.

Since we developed all the tracker software, except the solar positioning algorithm, we created an MQTT topic system that allows adjustments and exports all real-time status information. With these and smart sensors measurements, we have implemented a solar station monitoring system. All data are stored in time series database and analyzed by the Grafana tool that can generate alarms in case of station problem, improving the reliability of the irradiation data.

For future work, we will implement an automatic initial positioning system in case of power failure, fine-tuning with a closed loop control for better alignment, add new sensors in the solar station, add gateways in order to gather information of commercial trackers and create a quality control following the BSRN recommendations.

It is important to note that all software systems implemented in this work are open source technologies and can be obtained online. The solar tracker software is available at *https://github.com/xtarke/solarTracker*. All the images of the earlier calibration tests are available at https://github.com/xtarke/solarTracker/tree/master/calibimages and time-lapse videos are available at:

- https://www.youtube.com/watch?v=RLjJRj-Cs38
- https://www.youtube.com/watch?v=b4RHYuFXH9A
- https://www.youtube.com/watch?v=lTlY52YFF1I

6. References

Beagleboard. (2019). BeagleBone Black. Retrieved February 22, 2019, from https://beagleboard.org/black

Chieochan, O., & Ph, D. (2017). Internet of things (IOT) for smart solar energy : A case study of the smart farm at Maejo University. In *Proceedings of 2017 International Conference on Control, Automation and Information Sciences (ICCAIS)* (pp. 262–267). Chiang Mai, Thailand: IEEE.

Chylek, P., Lohmann, U., Dubey, M., Mishchenko, M., Kahn, R., & Ohmura, A. (2007). Limits on climate sensitivity derived from recent satellite and surface observations. *Journal of Geophysical Research Atmospheres*, *112*(24), 1–8. https://doi.org/10.1029/2007JD008740

Driemel, A., Augustine, J., Behrens, K., & Etal. (2018). Baseline Surface Radiation Network (BSRN): structure and data description (1992–2017). *Earth System Science Data*, *10*(3), 1491–1501. https://doi.org/10.5194/essd-10-1491-2018

Eppley Lab. (2019). Eppley SMT Tracker. Retrieved February 22, 2019, from http://www.eppleylab.com/instrument-list/automatic-solar-tracker/

Garmin. (2019). GPS 17x NMEA. Retrieved February 22, 2019, from https://buy.garmin.com/en-SG/digital/p/11626

Hirsch, T. (2017). *SolarPACES Guideline for Bankable STE Yield Assessment*. *IEA-SolarPACES*. Retrieved from http://www.solarpaces.org/wp-content/uploads/ SolarPACES_Guideline_for_Bankable_STE_Yield_Assessment_-_Version_2017.pdf

InfluxData, I. (2019). Influx database. Retrieved from https://www.influxdata.com

Kipp & Zonen. (2019). SOLYS2-Sun-Tracker. Retrieved February 22, 2019, from https://www.kippzonen.com/Product/20/SOLYS2-Sun-Tracker

Knox, R., & Untersteiner, N. (2011). Thinning of arctic sea ice. *Physics Today*, 1166–1169. https://doi.org/10.1007/978-90-481-2642-2_576

König-Langlo, G., Sieger, R., Schmithüsen, H., Bücker, A., Richte, F., & Dutton, E. G. (2013). *Baseline*

Surface Radiation Network (BSRN) Update of the Technical Plan for BSRN Data Management. *Worls Meteorological Organization*. Geneva, Switzerland. Retrieved from http://www.wmo.int/pages/prog/gcos/Publications/gcos-174.pdf

Labs, G. (2019). Grafana. Retrieved February 22, 2019, from https://grafana.com/

Maxwell, E., Wilcox, S., & Rymes, M. (1993). *User's Manual for SERI QC Software. Assessing the Quality of Solar Radiation Data*. Golden, CO.

Molloy, D. (2014). *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. John Wiley & Sons. Retrieved from https://books.google.com.br/books?id=F7zAnQAACAAJ

Moreno-Tejera, S., Ramírez-Santigosa, L., & Silva-Pérez, M. A. (2015). A proposed methodology for quick assessment of timestamp and quality control results of solar radiation data. *Renewable Energy*, *78*, 531–537. https://doi.org/10.1016/j.renene.2015.01.031

Mosquitto. (2019, July). Eclipse Mosquitto: An open source MQTT broker. Retrieved from https://mosquitto.org/

MQTT.org. (2019). MQTT Standard and Project Page. Retrieved from https://mqtt.org

Phung, M. D., Villefromoy, M. D. La, & Ha, Q. (2017). Management of Solar Energy in Microgrids Using IoT-Based Dependable Control. In *Proceedings of 2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*. Sydney, NSW, Australia: IEEE.

Polo, J., Wilbert, S., Ruiz-Arias, J. A., Meyer, R., Gueymard, C., Šúri, M., … Geuder, N. (2015). Integration of ground measurements to model- derived data. *A Report of IEA SHC Task 46 Solar Resource Assessment and Forecasting, Solar Heating and Cooling Programme, International Energy Agency*, (October), 1–34.

Ramírez, L., Pagh, K., Vignola, F., Blanco, M., Blanc, P., Meyer, R., & Wilbert. (2017). Road Map for Creation of Advanced Meteorological Data Sets for CSP Performance Simulations. *SolarPACES. International Energy Agency*, 14. https://doi.org/10.1021/ja210528v

Reda, I., & Andreas, A. (2004). Solar position algorithm for solar radiation applications. *Solar Energy*, *76*(5), 577–589. https://doi.org/10.1016/j.solener.2003.12.003

Roth, P., Georgiev, A., & Boudinov, H. (2004). Design and construction of a system for sun-tracking. *Renewable Energy*, *29*(3), 393–402. https://doi.org/10.1016/S0960-1481(03)00196-4

Routix software. (2019). MQTT Dash (IoT, Smart Home). Retrieved from https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=pt_BR

Williams, K., & Qouneh, A. (2017). Internet of Things : Solar Array Tracker. In *Proceedings of 2017 20th International Conference on Electrical Machines and Systems (ICEMS)* (pp. 1057–1060). Chiang Mai, Thailand: IEEE. https://doi.org/10.1109/ICCAIS.2017.8217588