Solar Pro

Karthik Sukumar
Electrical and Computer Engineering
Technical University of Munich
Munich, Germany
karthik.sukumar@tum.de

Johannes Machleid
Electrical and Computer Engineering
Technical University of Munich
Munich, Germany
johannes.machleid@tum.de

***Abstract*—The source of future energy production has to be and is undoubtedly renewable and environmentally friendly. Solar power is one of the most readily available energy sources in almost all parts of the world. Although solar power is ubiquitous, certain physical and technological limitations allow for a maximum efficiency factor of 37% (and that's for commercially avaiable high end solar cells). This implies that only 37% of the sun's energy captured by the solar cell can be converted to useful electrical energy. This paper focuses on using Wireless Sensor Networks (WSN) to utilise the maximum possible energy of the solar cells without any further losses by aligning the solar panel orthogonal to the sunrays depending on the daytime.**

I. Introduction

The energy generated by the solar panels is highly dependent on the angle of incidence of the sunrays on the panel.

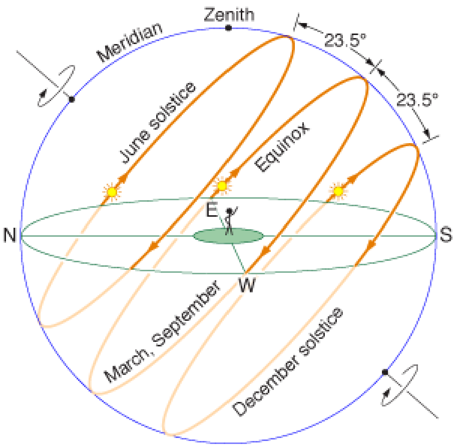


Fig. 1: Sun's path in winter and summer [**?**].

The sun's path in the horizon is dynamic and dependent on the time of the year. In the northern hemisphere, the sun is higher in the sky during summer and lower in the winter. In the summer the sun rises in the north east and sets in the north west. Whereas in winter, the sun rises in the south east and sets in the south west as it seen in Fig. 1. As shown in Fig. 2, consequent of the sun's path in the sky, the angle of incidence of the sun's rays on the panel changes not only during the day but as well as during the months of the year. In order to track the sun's rays, so that the solar panel is always perpendicular to it, a dynamic system that adjusts to the east-west changes in the sun's angle during the different times of the day as well the north-south tracking during the changes in the months is needed.

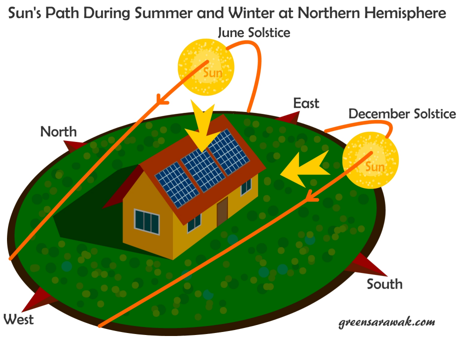


Fig. 2: Angle of incidence of the sun's rays on the panel [**?**].

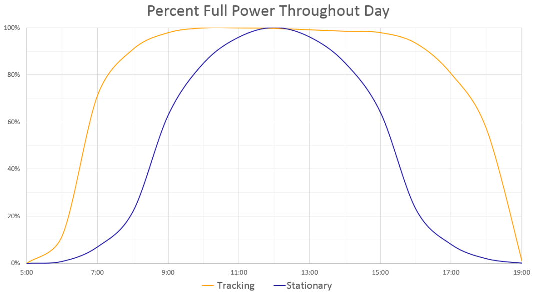


Fig. 3: Comparison of energy produced by a tracking vs non tracking system [**?**].

II. Application Set up

Fig. 3 is a comparison of the percentage of total power generated between a tracking and a non-tracking panel. It shows that there are definitely efficiency gains to be made from employing a tracking panel. Our proof of concept therefore showcases the potential to deploy a wireless sensor network to not only control the tracking of solar panels but log data at the same time. In order to achieve this, we employ three different types of sensors and eight wireless motes, of which one is set up as the base station connected to a desktop computer and the seven others deployed around the base station to function as sensor and panel control motes.

In this proof of concept as only four digital servo motors are available, not every mote in the field is able to align the solar panel according to the sun's rays. Never the less sensor values

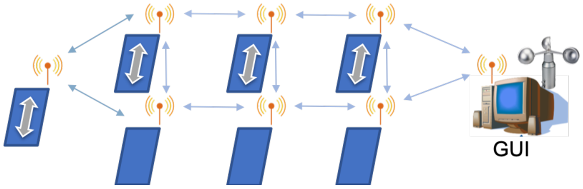


Fig. 4: Possible application setup with seven sensor motes and a base station ensuring multihop communication.

can still be measured and transmitted. A possible setup with a multihop communication towards the base station is depicted in Fig. 4.

A. *Sensors*

Described below are the different types of sensors used in our application along with their uses:

- Digital Servo Motor - HS-422 [1]
  A servo motor is a cheap rotary actuator which is connected to the mote via three wires: Ground (GND), Voltage (VDD) and Signal. The servo angle can be set via the signal wire by using a PWM-Signal. Most servos expect to see a pulse every 20ms [2]. The correspondence between pulsewidth and servo angle is depicted in formula (1). The servos will be used to change the angle of solar panels.

$$pw = 1.0ms + 1.0ms \cdot \left(\frac{\alpha}{\alpha_{max}} \right) \tag{1}$$

- Light Sensors
  The light sensors are used in this proof of concept to measure the luminosity and thus simulate the power output of the solar panel, since no real solar panels are available. As a result, the relative ADC 12-bit values of the sensors are of interest.

- Wind Speed Sensor
  The wind speed sensor, also called anemometer, is directly connected to the base station. It measures the wind speed at all times. In the case of excessive wind speeds, the panels will have to be stowed at a safe angle.
  The anemometer works with a simple reed contact and a magnet attached to the rotating axis [3]. Every time the magnet passes the reed contact, a digital HIGH is sent to the base station, which is processed by the base station using interrupts. The measured interval $t_{measure}$ together with the interrupt count n_{ticks} delivers the actual wind speed *ws* in km/h after formula (2).

$$ws = \frac{n_{ticks}}{t_{measure}} \cdot 2.4 \frac{km/h}{tick} \tag{2}$$

B. *Motes*

The Zolertia RE-mote platform [4] is a wireless module designed to be small, consume very little power, stay affordable and be easy to deploy in large quantities. In general, these type of devices are known as *motes*. These motes are the brain of the wireless sensor network and are running Contiki as an operating system. Eight motes are used in total out of which one mote functions as a base station and seven motes act as panel motes.

1) Base Station: The base station is connected to a desktop computer via a serial link and feeds the GUI with information. It runs a different code compared to the panel motes as it has to account for interfacing with the computer and collect data from the wind speed sensor at the same time.

2) Panel Motes: The panel motes run a more simplified subset of the base station code. They only react to unicast and broadcast messages and they respond based on the packet type.

III. Network description and Routing Technique

In this proof of concept we are demonstrating an WSN which provides information on the status of the solar panel along with the ability to control the angle of the panel in east-west axis. In a real world scenario it is intended that the mote will be powered by the solar panel and be able to draw minimal required power for its functioning.

A. *Network*

Considering our application, robustness and reliability were important factors for our design. Hence the decision was made to have a centralised base station polling the motes regularly. Having a master that queries, eliminates the need for system wide synchronisation (which are potentially required in a distributed system) and makes the implementation simpler. This also has the added advantage of avoiding collisions and random backoffs in unicast modes as at any point in time there is only ever one node using the channel. In this heterogenous network the routes are first discovered during the network discovery phase and the routing tables are exchanged as described in III-D.

B. *Base Station State Machine*

Fig. 5 shows the five states of operation and the transition conditions. The states are:

- IDLE
- NETWORKDISCOVERY
- PATHMODE
- UNICASTMODE
- EMERGENCY

1) IDLE: After bootup the device enters the IDLE state. In this state there are no broadcast or unicast messages that are sent out by the base station.

2) NETWORKDISCOVERY: This state is entered upon a user button press on the REmote or a trigger from the GUI. This event driven mode is used to obtain the network graph via a series of broadcast messages. A more detailed description can be found in III-D

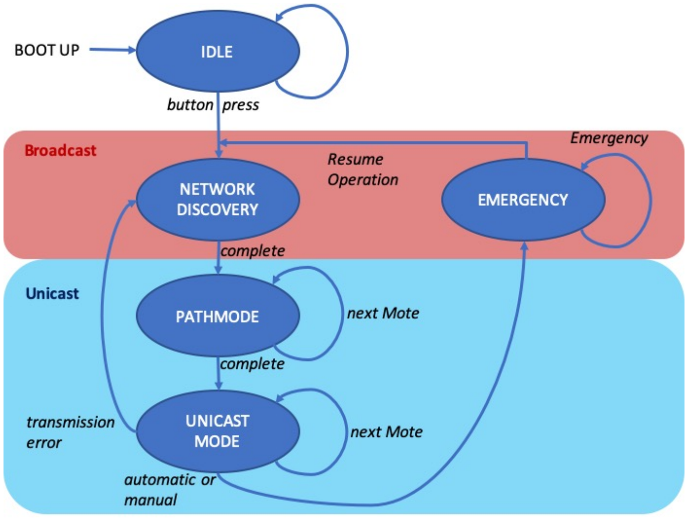


Fig. 5: Base station state machine.

3) PATHMODE: The base station transitions into PATHMODE state after the NETWORKDISCOVERY mode. This is a query-driven data reporting method, where the base station polls each node for their hop traces to the base station. This is required for the GUI to be able to display the topology.

4) UNICASTMODE: In this state, the nodes are polled in a round-robin manner and when the base station queries the information, it sends the servo angle data along with the query packet. The nodes respond by filling out the fields with their respective sensor values. Packet fields include battery voltage, temperature, light sensor value and servo angle. This state also follows a query driven data reporting method.

5) EMERGENCY: This state is only entered upon when the base station records an over threshold wind speed value. This event driven mode sends out a broadcast message to all nodes around it to stow their panels at a predefined safe angle.

C. Routing Cost Metric

A very important consideration is the cost for a route. There are many factors that can determine the calculation of a route cost. It could include battery state, transmission power, distance hop count and RSSI. In our application we have directly equated hop count as a distance cost metric.

D. Network Discovery

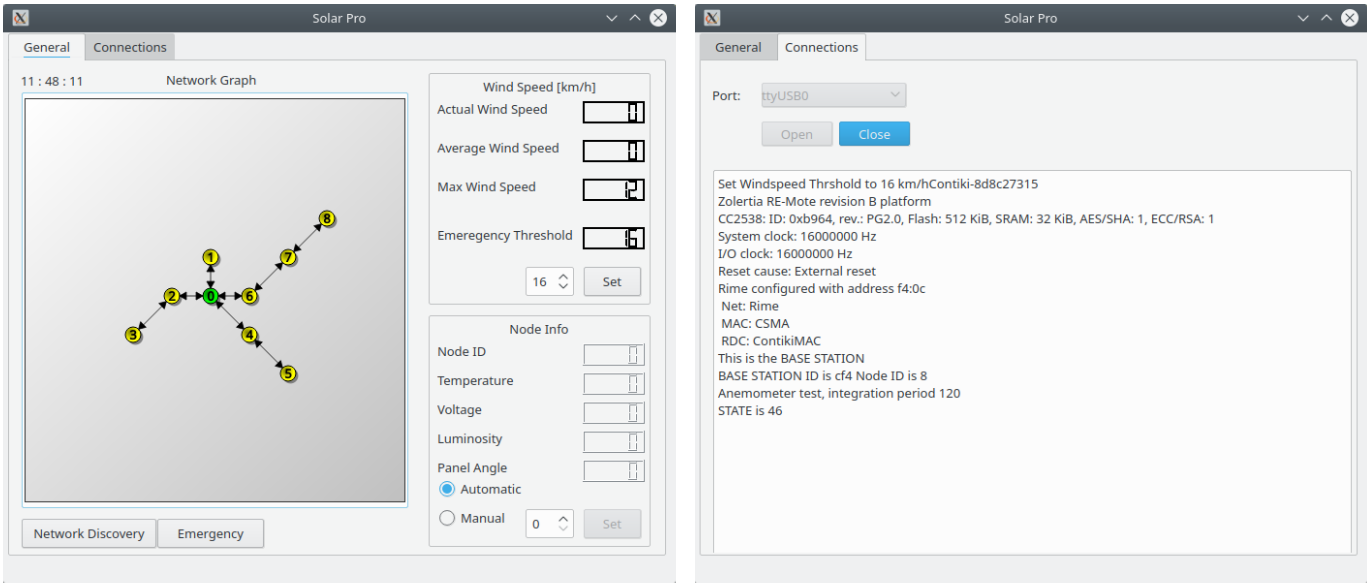
This is the most critical phase of the state machine. The network discovery here is started by a user button press for ease of demonstration of our proof of concept. Furthermore it can also be triggered from the GUI as explained in section IV. Any mote can initiate a network discovery which triggers a controlled flooding broadcast message. The node's complete routing table is sent as a payload to exchange with others. This method works in accordance to the distance vector routing and follows the Bellman-Ford shortest path tree algorithm. The routing itself is distributed where each node receives information from direct neighbours, performs calculation and distributes the result back to its neighbours, if there were changes to be made to the receiver's routing table. This means the network discovery is iterative and self terminating and also asynchronous.

E. Fault Tolerance

The wireless medium is prone to introducing errors, particularly because of noise, interference and attenuation. The dynamic nature of the wireless sensor network in the field could mean that nodes can be damaged, non-functional or removed from the network. To cope with these scenarios the application has to be fault tolerant. An acknowledgement ensures detecting transmission errors. If the sending node does not receive an acknowledge message after a certain number of tries, it initiates a network discovery.

IV. Graphical User Interface (GUI)

The graphical user interface (GUI) is a desktop application programmed with the aid of the QT Creator and focuses on displaying information about the motes operating in the field and allows the user to interact with a desired mote. The computer running the GUI is directly connected to the base station mote via a serial link.
The GUI basically consists of the two tabs ”General” and ”Connections” as shown in Fig. 6a and Fig. 6b.
The ”General” tab shows a network graph of the wireless sensor network and the system time. Communicating sensor motes are depicted with a yellow node and a node ID inside, whereas the base station is shown centered as green node.
The double sided arrows depict the wireless communication route and therefore the network topology. Only the shortest paths of the nodes towards the base station as sink in the network is depicted on the graph.
The two buttons ”Network Discovery” and ”Emergency” in the bottom left corner enable the user to trigger these two states III-B2 and III-B5 manually.
Two info boxes aligned to the right side of the window show the sensor information of the motes. The upper info box shows the values of the anemometer connected to the base station, such as the actual wind speed, the average windspeed, the maximum windspeed and the emergency threshold, at which the solar panels are aligned to a predefined safety angle. The GUI also allows to choose a user defined emergency threshold, which is set by pressing the ”Set” button next to the number field. The default value is 16 km/h.
The lower info box shows the sensor information of the selected mote. A mote is selected by clicking on a mote in the network graph. The particular information such as the Node ID, the temperature, voltage, luminosity and panel angle are presented. For maintenance reasons it is also possible to operate the selected mote by clicking on the ”Manual” button. It is then possible to enter an angle between 0 and 180 degrees to manually align the selected solar panel by clicking the ”Set” button next to the number field.
The second tab of the GUI allows the user to connect and disconnect the serial link to the base station via selecting the respective port and clicking the buttons ”Open” to connect or ”Close” to disconnect. The textbox aligned to the bottom of the tab shows possible debug information from the base station mote.



```
Set Windspeed Thrshold to 16 km/hContiki-8d8c27315
Zolertia RE-Mote revision B platform
CC2538: ID: 0xb964, rev.: PG2.0, Flash: 512 KiB, SRAM: 32 KiB, AES/SHA: 1, ECC/RSA: 1
System clock: 16000000 Hz
I/O clock: 16000000 Hz
Reset cause: External reset
Rime configured with address f4:0c
 Net: Rime
 MAC: CSMA
 RDC: ContikiMAC
This is the BASE STATION
BASE STATION ID is cf4 Node ID is 8
Anemometer test, integration period 120
STATE is 46
```

(a) GUI General Tab

(b) GUI Connections Tab

Fig. 6: Graphical user interface of the Solar Pro application.

V. Application Specific Features

The radio duty cycle (RDC) layer is not modified and runs the Contiki standard configuration ”ContikiMAC” according to [5]. The contiki MAC provides energy savings due to duty cycled wake up times. The carrier sense multiple access (CSMA) driver provides reliability by sensing busy channel medium and backing off by a random amount of time before transmitting the packet. This avoids collisions during the broadcasting modes although it might increase the latency.

VI. Future work and Optimisations

Our implementation does not focus on low latency so this could be a possible improvement. One way of improving latency can be achieved by using hash tables instead of array based routing tables. This reduces the complexity from O(n) to a constant time. Considering that for every UNICAST transmission routing table walk is necessary, this could save time and power. Further ideas to advance the application are a sun tracking algorithm which steers the solar panel according to sensor values. Another useful feature would be to integrate data visualization of the sensor values as a graph showing the history of e.g. the last 24 hours.

VII. Conclusion

The power efficiency of solar panels highly depends on the inclination angle of the sunrays. By aligning the solar panel orthogonally to the sunrays depending on the daytime, it is possible to increase its energy efficiency. In this application a wireless sensor network has been employed to control the solar panel angle via a digital servo and measure the amount of light hitting the panel with a light sensor. Furthermore the temperature of the controlling mote and its battery voltage is read out.

The application setup consists of seven sensor motes, of which four motes operate a digital servo. One mote functions as base station and delivers the sensor information to a desktop application with a GUI.

The network is set up by first initialising a network discovery routine. All motes broadcast and collect information about their neighbors to set up their very own distance vector table. After completion, the base station reaches out to each mote with a unicast request to collect the shortest path through their respective acknowledge messages.

With this information the base station is able to poll the sensor informations of each node by cycling through the network.

In case of lost packets or transmission errors, the network reorganises itself after a certain number of retransmissions.

The network graph and sensor information are visually presented in the graphical user interface, which also allows to operate the servo position manually, set a wind speed emergency threshold or simply trigger an emergency case manually. It is also possible to manually restart the network discovery.

References

[1] “Servo Motor HS-422,” https://www.multiplex-rc.de/produkte/112422-servo-hs-422, 2008, [Online; accessed 19-July-2008].

[2] Jameco.com, “How servo motors work,” accessed 2019-08-14. [Online]. Available: https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html

[3] A. Egger, “Arduino anemometer bauen,” 2018, accessed 2019-08-14. [Online]. Available: https://www.aeq-web.com/anemometer-mit-dem-arduino-bauen/

[4] “Zolertia Remote,” https://github.com/Zolertia/Resources/wiki/RE-Mote, 2008, [Online; accessed 19-July-2008].

[5] A. Dunkels, “The contikimac radio duty cycling protocol,” 2011.