ABSTRACT
In the recent years, energy extraction from renewable sources has become increasingly important. Among different renewable sources, solar energy is becoming attractive for its environmental friendliness and universal availability. However, to maximize power output from a solar panel, many kinds of efficiency improvement techniques are needed. In this paper, we implemented two such techniques: maximum sun-light tracking and optimum battery charge monitoring for a solar panel. We implemented these techniques using ZigBee based wireless sensor network. The ZigBee network has become an affordable wireless solution for many monitoring and control problems. In this paper, we presented a comprehensive description of our solution approach.

KEY WORDS
Solar Power; Solar Tracker; Battery Charge Monitoring; Wireless Sensor Network; ZigBee; Arduino;

1. Introduction
Solar energy has become an important renewable energy source globally. In our own country Bangladesh, solar energy shows huge potential for both rural and urban areas [1, 2]. To achieve maximum power from a solar system, many kinds of efficiency improvement techniques are needed [3]. We have implemented two of these techniques in a solar power system in our research: a maximum sun-light tracking system and an optimum battery charge monitoring system. In our system, the solar panel charges a lead acid battery, that we monitored for optimum battery charging cycle using ZigBee wireless sensor network. In the recent years ZigBee based wireless network has become popular choice due to their energy efficiency and easy configurability [4, 5]. Thus, ZigBee-based monitoring gives cost reduction, long term maintainability and energy efficiency for the system. Besides, our system has implemented a solar tracking module that changes the orientation of the solar panel to draw maximum energy from the sun-light. The system achieve these through implementing an array of sensors and a tracking system that provide enough information about the health and status of the solar power generation unit. On top of these techniques, we also built a user friendly web-interface for monitoring, so that any user can monitor different status information from a computer or a mobile phone through the Internet.

The work presented in this paper has been divided into several sections. Section two explains our approach for maximizing energy extraction from the solar panel. In section three, we give a brief description of the components used in our system. In section four we give a brief overview of ZigBee network. In section five, we explain implementation of different system modules and their integration. Section six explains results from our project and section seven discusses several problems that we faced during implementation phases. Section eight ends the paper with conclusions that summarizes the important aspects of the project with some future research direction.

We implemented a ZigBee-based wireless sensor network to monitor sun-light and charging status of a solar photovoltaic system. Furthermore, within the solar system, a delicate balance between all the sub systems like voltage, temperature and light has been established in a careful manner. One of the important features of the projects is that the user can monitor system status information using the Internet. The ZigBee wireless network operates at low power, it is inexpensive and ensures a secure networking. Following subsections: Sensor Network, Solar Tracker, ZigBee Network Interface and Web based User Interface explain our methodology.
2.1 Sensor Network

The sensor network part involves reading instantaneous voltage and other values from an analog to digital converter (ADC) on an Arduino Mega [6, 7] microcontroller board. The ADC requires the input signal to fall within the range of 0V and 5V. Therefore, signal from battery line voltage has been modified for reading. In order to reduce the line voltage to a readable value, a voltage divider circuit has been implemented. For temperature measurement a LM35 temperature sensor is used. Similarly, photocells are used for measuring light intensity. All data captured through these sensors are passed through the ZigBee network to the web interface.

2.2 Solar Tracker

This part of the project has been built using a concept where two signals from different light intensity sensors are compared. The Arduino Mega microcontroller board takes care of the control functionality using other support circuits, such as ULN2004 for controlling stepper motor movement. The microcontroller sends data to the bidirectional unipolar stepper motor via relay to ensure that solar panel is perpendicular towards the Sun. Arduino microcontroller also controls the rotation of the stepper motor either to rotate clockwise or anticlockwise. The solar panel that attached to the stepper motor acts according to the direction of the motor. Status of the movement is reported through the ZigBee network.

2.3 ZigBee Network Interface

To transfer data to the Internet a communication link between Arduino mega microcontroller and a computer is created through the ZigBee wireless mesh networking. The computer is connected to the ZigBee network through XStick wireless modem, which acts as a coordinator between two devices. This provides a channel to receive and transmit energy and control data wirelessly for processing and Internet based display.

2.4 User Interface

The user interface allows the user to see and understand the complete detail of the solar panel monitoring system. User gets access to these data through a particular website using a computer or mobile device. Furthermore, user can also monitor the graphs for different parameters, which may give him/her a more clear view of the system.

3. Devices used to Build the System

We used several components to build our system. A brief explanation of each of these components is given here.

3.1 Arduino Microcontroller and Support Components

Arduino is an open source electronic prototyping platform based on a flexible and easy to use hardware and software [6, 7]. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). In our project we used Arduino to sense the environment by receiving input from a variety of sensors connected through ZigBee module for data transmission.

With Arduino, we have also used a voltage divider module, which is a simple circuit consisting of two resistors that is useful of changing a higher input voltage into a lower output voltage. It does this by dividing the input voltage by a ratio determined by the values of two resistors. Besides, voltage regulator ICs are used to give precise regulated voltage. It also includes built in current limiter and thermal shutdown protection modules, which make it virtually immune to damage from output overloads.

3.2 Temperature and Light Intensity Sensors

System uses LM35 precision integrated circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. Photocells are sensors that allowed us to detect variation in light intensity. They are small, inexpensive, low power and easy to use. Photocells change their resistive values in ohms depending on how much light is shining on them.

3.3 Actuator Stepper Motor

We used a stepper motor, which is a brush-less DC motor that divides a full rotation into a number of equal steps. The motor’s position can then be commanded to move and hold at one of these steps without any feedback sensor (an open loop controller), as long as the motor is carefully sized to the application. As the microcontroller can’t drive the stepper motor directly, a driving circuit is needed. For this ULN2004A IC was used. The ULN2004A is a high voltage, high current Darlington transistor arrays, each consists of seven NPN Darlington pairs that feature high voltage outputs with common cathode clamp diodes for switching inductive loads.

3.4 XBee Wireless Modem and Adapter

Wireless sensor network in our project was built using XBee and XStick radios. The XBee radio [8] provides an embedded solution for wireless end point connectivity to devices, based on the ZigBee wireless network standard. It operates on the 2.4GHz frequency, and provides 250Kbps bandwidth to an RF node and has a range of 3.2Km.
Figure 1: (a) XBee Wireless Modem and Adapter and (b) XStick Wireless Modem

The XStick [9] is a USB adapter that provides wireless connectivity to a low power mesh or multipoint network. Housing Digi’s XBee module, this adapter provides instant wireless connectivity from a laptop or PC, allowing network configuration, diagnostics or device monitoring to be conducted locally. The XStick is USB powered and does not require batteries or a power adapter.

3.5 Solar Panel and Battery

The solar panel we used in our project was for prototyping a small electricity generation unit. It was a small 5W solar panel attached with a 12V battery. There are several types of battery available in the market with different current ratings. In this project, we have used a 12V 7.5AH Sealed Lead Acid battery which is typically used in UPS. The charging capacity of the panel depends on the variation of available sun-light.

4. ZigBee Network Overview

ZigBee wireless networking standard fits into a section that is not full filled by other wireless technologies. When most wireless technologies are aiming to go faster, ZigBee aims for low data rates and using a tiny stack that fits on 8 bit microcontroller. While other wireless technologies look to provide the last mile to the Internet or deliver streaming high-definition media, ZigBee looks to control a light or send temperature. ZigBee wireless networking standard provides improved networking capabilities over the IEEE802.15.4, which only provides endpoint-to-endpoint connection. Due to these several benefits, ZigBee based network has shown wide adoption in industrial applications [4, 5].

Figure 2: ZigBee Mesh Topology

ZigBee networking devices can be connected in several different layouts or topologies to give the wireless network its structure. These topologies indicate how the radios are logically connected to each other. There are three major ZigBee topologies: Star, Tree and Mesh. In our project we have used mesh topology (Fig-2). The mesh topology employs router nodes in addition to the coordinator radio. These router nodes can pass messages along to other routers and end devices. The coordinator radio acts to manage the network and also route messages. Various end devices may be connected to any router or to the coordinator. ZigBee wireless network implemented by the XBee [8] modules. The coordinator is responsible for starting the network, setting up the Personal Area Network (PAN) ID and establishing encryption if required. The routers and end devices can then join the PAN. In this topology, using a router is optional. However routers help to expand the network range.

4.1 ZigBee Network Setup

To set up XStick and XBee radios, we download the X-CTU tools from the Digi’s website to configures required control parameters. In order to correctly set up a ZigBee network, following parameters must be set:

- ID – PAN ID must be the same
- DL – Destination Address Low:
  - Coordinator set to 0xFFFF (Broadcast Mode)

Figure 3: (a) Wireless Sensor Network Unit and (b) Constructed Solar Tracker Prototype
Router set to Coordinator’s SL – Serial Number Low
• BD – Baud Rate must be the same

Once the connection is established, the yellow LED on the XStick and the green LED on the XBee should blink moderately.

4.2 Data Capture over ZigBee Network

A 12V battery has been used in the project. However, as Arduino cannot read data directly from any source more than 5V, we setup an appropriate voltage divider circuit for our system. Arduino has also a current rating limitation, where the current should not be greater than 60mA. Considering these factors, the voltage divider circuit has been implemented with two resistors R1 and R2 with appropriate values, so that output voltage is less than 5V and current is less than 60mA. We have chosen R1= 20KΩ and R2= 5KΩ. The Arduino board is getting analog input from the voltage divider using the analog Pin 0 and then it is communicating with XBee in order to send the data over the ZigBee network. We have selected following values:

\[
I_o = \frac{V_{in}}{R_o + R_1} = 0.48mA
\]

Voltage Drop Across R1

\[
V_{drop}^{R_1} = V_{in} \left( \frac{R_1}{R_1 + R_2} \right) = 12 \left( \frac{20}{20+5} \right) = 9.6V
\]

Output Voltage

\[
V_{out} = V_{in} \left( \frac{R}{R_1 + R_2} \right) = 12 \left( \frac{5}{20+5} \right) = 2.4V
\]

Voltage Ratio

\[
\frac{V_{out}}{V_{in}} = \frac{R}{R_1 + R_2} = \frac{5}{20+5} = 0.2
\]

In the purpose to measure temperature, LM35 sensor is used. Output pin of LM35 is connected with analog pin 1 of Arduino. Light intensity sensor photocell is connected with a 10KΩ resistor in series and that node is wired up with analog Pin 2 of Arduino. Since the photocell sensors are relatively inexpensive, they are not very precise. Accordingly, we used a 10KΩ resistor in series to improve stability and accuracy.

4.3 Building the Solar Tracker

First, we connected all devices ULN 2004A, two Photocells with two 10K Resistors and LED in a breadboard as shown in Figure 3, then wire the unipolar stepper motor with the pin number 13, 14, 15, 16 of ULN 2004A. Pin number 1, 2, 3, 4 of UNL 2004A is serially connected with 8, 9, 10, 11 PWM pins of Arduino. The left photocell is connected with pin 1 and right photocell is connected with pin 2 of Arduino’s analog input section. Rest of the connection is done according to the Figure 3. Four voltage regulator ICs are used between four connections of ULN 2004A and unipolar stepper motor to protect the ULN 2004A uneven excessive voltage drawn from the stepper motor.

4.4 Integrating XBee and Arduino

The next part of the project is to integrate the XBee module to the Arduino 1280 microcontroller to enable the Arduino to send serial data wirelessly. It is done through programing Arduino using a cross platform integrated development environment (IDE). The IDE has been written in Java and includes a code editor which features syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. The Arduino IDE comes with a C/C++ library called "Wiring". The following code segment shows how to interface XBee with the Arduino 1280:

```java
void setup() {
  Serial.begin(9600);
  Serial.begin(9600);
}

void loop() {
  while(Serial.available()) {
    Serial.print((char) Serial.read());
  }
  while( Serial.available()) {
    Serial1.print((char)Serial.read());
  }
  delay(10000);
}
```

Arduino uses port 18 and 19 as the TX/RX pins of its hardware Serial 1 port. The RX Pin of the XBee module has to be connected to the TX (Pin 18) of the Arduino, and the TX Pin to the RX (Pin 19). The power pins 5V and GND should be connected to the appropriate power pins on the Arduino. The XStick’s USB module should be connected to the computer, with the X-CTU Terminal running. The above code makes the Arduino poll for an input from either the Arduino Serial Monitor or the X-CTU Terminal. Typing into the X-CTU Terminal should show the message on the Arduino Serial Monitor, and vice versa (Figure 4). Once the ZigBee network was established, the next move was to make integration between the network and the computer or local server. In order to do the task, following steps have been followed.

To show the results online, it was necessary to take the data out from the ZigBee network and send it to an online server. Java programming has been used as a gateway to take the data out of the network. In Java, some properties of Java library have been used.

```java
import com.rapplogic.xbee.api.XBee;
import com.rapplogic.xbee.api.XBeeException;
import com.rapplogic.xbee.api.XBeeResponse;
```
When the integration between XStick and Java was done, the next focus was to exact value from the Arduino byte stream. To perform this task, a pattern in which Arduino was uploading data into ZigBee network was scanned. In order to receive the data in the same manner, it was essential to filter the data because there were some garbage values coming with the original data. A colon ("\:"\) has been used as a delimiter, noticing the fact that in the garbage value has no colon ("\:"\). Using this simple trick, we were able to filter actual data from the garbage value.

### 4.5 Building Web Interface

This is the last step of the project where data has been uploaded from the local server to an online server for public access. The Java.net library was used for the local server. From the local server, a post request has been sent on the online server in order to store data into a MySQL database. In this process, PHP based web programs (CodeIgniter framework) has been used.

```java
import java.net.HttpURLConnection;
import java.net.URL;
```

After all these steps, it is possible to monitor three parameter voltage, temperature and light intensity from the webpage as show in the Figure 4.

### 5 Results

The following graphs (Figure 5, 6) shows output from our system. The data for these graphs was collected from a daylong test of the ZigBee Wireless Sensor Network system with a functional solar panel. The first graph (Figure 5) shows changes in voltage and temperature with time. Our 12V 7.5AH Sealed Lead Acid battery has 11.22 Volt remaining at 6:00 AM on January 07, 2013. As the day progresses and light intensity increases battery starts charging and goes its peak point approximately 12.30 PM Volt around noon. Both Voltage and Temperature values are coming from our remotely located sensor network wirelessly through the ZigBee wireless network and uploaded into a website near real time. Our optimum charge monitoring circuits are suppose to track this variation (not shown).

The second graph (Fig-6) illustrates the light intensity over a period of 12 hours in a particular day. Vertical axis represents the light intensity in Lux where the horizontal axis represents the time in hours. In the beginning of the day, 6:00 AM on January 07, 2013, the light intensity almost touches the 45 Lux mark which is considered as dim. As the time progresses, a gradual increase in the light intensity is observed touching at the peak at 721 Lux at 2:30 PM where a session from 9:30 AM to 11:00 AM experiences a bright state and afterwords approximately 11:30 AM to 3:30 PM session experiences a very bright sunlight. After a dramatic fall is observed by the photocell sensor from the 2:30 PM to the end of the day at 6:00 PM touching the finishing line at 33 Lux which is considered as dim. Light intensity values are coming from our remotely located sensor network wirelessly through the ZigBee wireless network and uploading on a website near real time. Looking at the data any time from anywhere, a user can easily get an idea about the amount of sunlight
falling on the solar panel. Optical sensor triggers DC stepper motor based on this light intensity variations to change orientation of the solar panel.

![Figure 6: Light intensity around solar PV recorded in January 07, 2013](image)

6 Discussions

In our project, our main aim was to build a cost effective solution for solar energy tracking with automated actuator mechanism which can guide optimum charging sequence in a solar panel. To achieve these objectives, we have successfully built a ZigBee-based robust wireless network with an array of sensors, built control circuits and also built a user friendly web interface to monitor all those sensor values near real time. We have also tested our prototype system and have received very good results. The whole process was full of trials and errors with stiff learning curve. We had to deal with hardware limitations and had to learn three different programming languages C, Java and PHP, and we also developed our knowledge on HTML and CSS coding.

The first complexity we faced was on voltage divider circuit, where we had to choose resistor R1 and R2 in such a ratio that output becomes more accurate. In our circuit R1:R2 is 4:1 and output accuracy was 98%.

Next problem we faced during the modem configuration, as our coordinator radio XStick in COORDINATOR API mode and router radio XBee in ROUTER AT mode. We have easily configure our XStick in COORDINATOR API, but for configuring XBee in ROUTER AT mode, we had to write it twice, first in ROUTER AT mode then set the DH - Destination Address High and DL - Destination Address Low with the serial number we took from the Coordinator radio XStick. As our radios were in two different modes, response was broken up into bytes and represented hex-encoded ASCII string. In order to present the data in traditional way we wrote filtering code in Java and successfully filtered and represent the sensor values on web pages.

Another problem we faced was excessive heating of motor driver ULN 2004A in solar tracker circuit. We have used voltage regulator ICs in circuit to give precise regulated voltage coming towards motor driver and give it a thermal protection.

7 Conclusion

Our ZigBee based wireless network solution for maximum sun-light tracking and optimum battery charging offers significant benefits for solar panel. Customers may monitor energy creation in real time, from anywhere. They will also be able to identify any problem in the solar panels. Such ability to monitor power generation is an important advantage for preventive maintenance for the manufacturer and the users. Wireless solution reduces infrastructure costs; remote troubleshooting reduces maintenance costs and system downtime and also maximizes the efficiency of energy generation. In the near future we want to make our system more accurate and less expensive.

References