IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Microcontroller Based Solar Power Measurement

[1]Arote Shweta Rajesh [2] Kharade Vivek Babaji, [3]Ughade Prathmesh Atish, [4]Shelake B.N. 5 Satpute A.S.

[1]Samarth Polytechnic Dept.of Electronic and Telecommunication Belhe, [2] Samarth Polytechnic Dept. of Electronic and Telecommunication Belhe, [3]Samarth Polytechnic Dept.of Electronic and Telecommunication Belhe,
[4]Samarth Polytechnic Dept. of Electronic and Telecommunication Belhe,
[5] Samarth Polytechnic Dept.of Electronic and Telecommunication Belhe.

Abstract

The rapid growth in renewable energy sources has highlighted the need for efficient and accurate monitoring of solar power generation. This paper presents the design and implementation of a microcontroller-based solar power measurement system aimed at improving the monitoring and management of solar energy systems. The proposed system utilizes a microcontroller (such as Arduino or Raspberry Pi) for real-time measurement and data logging of solar panel output, including voltage, current, and power. It incorporates sensors like voltage and current sensors, interfacing with the microcontroller to capture electrical parameters. The system is capable of calculating and displaying power generation data, ensuring optimal performance monitoring of the solar panels. Additionally, it can store data for historical analysis and send it to a cloud-based platform for remote monitoring. The results demonstrate the efficiency, cost-effectiveness, and scalability of this microcontroller-based solution for solar power measurement, making it a viable tool for both residential and industrial solar energy management.

Index Terms - **Component,formatting,style,styling,insert.**

1.INTRODUCTION

The growing demand for sustainable energy solutions has led to an increased adoption of solar power systems worldwide. As solar energy continues to be a key player in the renewable energy sector, the need for efficient monitoring and management of solar power generation has become paramount. Proper monitoring ensures that solar systems operate at their maximum efficiency, and timely detection of performance issues or faults can significantly extend the lifespan of solar panels.

2.NEED OF THE STUDY

1. **Cost-Effective Monitoring Solution:** Traditional solar power monitoring systems can be expensive, requiring specialized equipment and high installation costs. Microcontroller-based systems, by contrast, provide a low-cost solution, making it accessible to a broader range of users, from small residential systems to large industrial setups. These systems make solar energy more affordable to maintain by reducing the need for expensive commercial monitoring equipment.
2. **Real-Time Data Collection and Analysis:** Real-time monitoring of solar power generation is essential for optimizing energy production. A microcontroller-based system can continuously measure

electrical parameters such as voltage, current, and power output. This real-time data allows users to monitor the performance of their solar systems instantaneously, identify potential issues early, and take corrective actions before minor problems lead to significant inefficiencies or system failure.

3. **Efficiency Optimization:** Solar panels can be affected by various factors like dirt, shading, or degradation over time. A microcontroller-based monitoring system allows for the detection of deviations in expected power output. By analyzing real-time data, the system can identify underperforming panels or other system inefficiencies, allowing for timely interventions to optimize energy production and prevent energy loss.
4. **Integration with Other Smart Systems:** Microcontroller-based systems can be integrated with other smart devices and technologies such as cloud-based platforms, IoT applications, and smart grids. This integration allows for remote monitoring and control of solar power systems, providing users with the flexibility to track and manage their solar energy consumption from anywhere in the world. Furthermore, it allows system owners to make data-driven decisions based on historical performance.
5. **Data Logging and Predictive Maintenance:** One of the key advantages of a microcontroller-based measurement system is its ability to store historical data. This data can be used for long-term analysis, helping system owners predict performance trends and plan for necessary maintenance. Early detection of faults or inefficiencies can prevent costly repairs and ensure the longevity of solar equipment.
6. **Sustainability and Renewable Energy Monitoring:** As governments and organizations around the world emphasize the need for renewable energy solutions, reliable and efficient monitoring systems become increasingly vital. A microcontroller-based system provides an affordable and scalable solution to assess the performance of solar power systems, thereby contributing to the global goal of increasing renewable energy adoption and reducing dependence on fossil fuels.
7. **Educational and Research Opportunities:** The study of microcontroller-based solar power measurement offers significant educational value. It enables students, researchers, and enthusiasts to explore the intersection of renewable energy, electronics, and programming. This knowledge can lead to further innovations in solar energy monitoring, improving both technology and sustainability practices in the energy sector.

3.1 Data and Sources of Data

<u>1. Voltage Measurement Data</u>

- **Source of Data:**
  - **Voltage sensors** or voltage dividers are used to measure the voltage output from the solar panels.
  - Common voltage sensors include **ZMPT101B**, **INA219**, and **resistive voltage dividers** (which are simple circuits using resistors to scale down voltage).
- **Microcontroller Input:**
  - The voltage sensors provide analog data (voltage level), which is read by the **Analog-to-Digital Converter (ADC)** of the microcontroller.
- **Data Output:**
  - The voltage readings are converted to digital values and then processed to determine the real-time voltage produced by the solar panel.
- **Example Data:**
  - Solar panel voltage: 18.5V, 20.1V, etc., depending on the time of day and sunlight intensity.

2. Current Measurement Data

- **Source of Data:**
  - **Current sensors** like **ACS712**, **INA219**, or **Hall Effect sensors** are used to measure the amount of current generated by the solar panel.
  - These sensors output an analog voltage proportional to the current flowing through the circuit.
- **Microcontroller Input:**
  - The analog current readings are fed to the microcontroller's **ADC** to be converted into digital values.

- **Data Output:**
  - The microcontroller processes the current readings to calculate the solar power output ($P = V \times I$).
- **Example Data:**
  - Current measurements could be in the range of milliamps to amps (e.g., 0.5A, 1.2A, 5A), depending on the solar panel capacity and the time of day.

3. Power Measurement Data

- **Source of Data:**
  - Power is calculated by multiplying the measured **voltage** and **current**.
  - No additional external sensor is needed for direct power measurement; it's derived from voltage and current data.
- **Microcontroller Input:**
  - The **microcontroller** uses the voltage and current data to calculate power using the formula: $\mathbf{P = V \times I}$.
- **Data Output:**
  - The calculated power is displayed in watts (W) or kilowatts (kW).
- **Example Data:**
  - Power output could be values such as 20W, 50W, 200W, etc., depending on the **system and Solar Pannel Size.**

4. Irradiance (Solar Radiation) Measurement Data

- **Source of Data:**
  - **Light sensors** like **TSL2561**, **BH1750**, or **ML8511 UV sensor** measure the intensity of solar radiation (irradiance).
  - These sensors provide an analog or digital output representing the level of sunlight hitting the solar panels.
- **Microcontroller Input:**
  - The microcontroller processes the sensor output to determine the irradiance, often using the sensor's I2C or analog interface.
- **Data Output:**
  - The irradiance data is often displayed in **lux** or $\mathbf{W/m^2}$ (watts per square meter), representing the solar radiation intensity.
- **Example Data:**
  - Irradiance measurements could vary between 0 W/m^2 (nighttime) to 1000 W/m^2 (full sunlight).

5. Temperature Measurement Data (Panel and Ambient)

- **Source of Data:**
  - **Temperature sensors** like **LM35**, **DHT11**, **DS18B20**, or **TMP36** can be used to monitor both the temperature of the solar panels and the ambient temperature.
  - The temperature of the panels affects their efficiency, and the ambient temperature can provide insight into the overall environmental conditions.
- **Microcontroller Input:**
  - The analog temperature readings from the sensors are converted into digital data by the **ADC** of the microcontroller.
- **Data Output:**
  - The temperature is displayed in degrees Celsius (°C) or Fahrenheit (°F).
- **Example Data:**
  - Panel temperature might be 40°C under full sunlight, and ambient temperature might be 25°C.

6. Energy Measurement (Accumulated Data)

- **Source of Data:**
  - as 0.5 kWh, 1.2 kWh, etc., depending on the solar system's output. **Energy is typically accumulated over time** based on the calculated power output. The microcontroller uses a real-time clock (RTC) module to store time-based data and calculate the energy generated.
  - Energy is the integral of power over time (Energy = Power × Time).
- **Microcontroller Input:**
  - The microcontroller reads real-time data from the sensors, calculates the power, and integrates it over time.
- **Data Output:**
  - Energy is typically measured in kilowatt-hours (kWh).
- **Example Data:**
  - The energy generated might be recorded

7. Fault Detection Data

- **Source of Data:**
  - If the system detects abnormal voltage, current, or power readings, it can trigger an alert for a potential fault (e.g., a malfunctioning panel or wiring issue).
- **Microcontroller Input:**
  - Fault detection is done by comparing real-time data with expected or threshold values programmed into the microcontroller.
- **Data Output:**
  - The microcontroller will either output an error message on a display, trigger a buzzer, or send a notification (via SMS, email, or a cloud service).
- **Example Data:**
  - A sudden drop in voltage or current may indicate a fault, such as "Voltage below expected range: 10V," or "Current less than 0.5A."

8. Data Logging and Reporting

- **Source of Data:**
  - **SD cards** or **external storage modules** (e.g., EEPROM) are used for data logging. The data is stored periodically and can be accessed later for historical analysis.
- **Microcontroller Input:**
  - The microcontroller writes data to the SD card or storage at regular intervals, capturing real-time voltage, current, power, temperature, and irradiance readings.
- **Data Output:**
  - The logged data can be stored as CSV or text files, which can later be retrieved and analyzed.
- **Example Data:**
  - Data might be logged as timestamps with corresponding values: Timestamp: 12:30 PM, Voltage: 18.5V, Current: 1.2A, Power: 22W, Temp: 40°C.

9. Remote Monitoring and Cloud Data

- **Source of Data:**
  - **Communication modules** (Wi-Fi, GSM, Bluetooth, or Zigbee) can send real-time data to a cloud platform or a mobile application for remote monitoring.
- **Microcontroller Input:**
  - The microcontroller sends the collected sensor data to a remote server or cloud service, where it can be accessed via a user interface.
- **Data Output:**
  - The data can be viewed on a dashboard, with historical trends and real-time updates.
- **Example Data:**
  - A cloud-based platform might show a dashboard with graphs of power output, energy consumption, panel temperature, and irradiance over time.

Data Sources:

1. **Voltage Sensors** (e.g., ZMPT101B, INA219) for voltage data.
2. **Current Sensors** (e.g., ACS712, INA219) for current data.
3. **Light Sensors** (e.g., TSL2561, BH1750) for irradiance data.
4. **Temperature Sensors** (e.g., LM35, DHT11) for temperature data.
5. **Real-Time Clock (RTC)** for accurate time tracking for energy calculation.
6. **SD Cards/External Storage** for data logging.
7. **Communication Modules** (e.g., Wi-Fi, GSM) for remote monitoring and cloud data sharing.

3.2 literature Survey on Microcontroller-Based Solar Power Measurement

*1. **Introduction to Microcontroller-based Solar Power Measurement*** Accurate measurement and monitoring of solar power generation are essential for assessing the efficiency of solar power systems, identifying potential issues, and maximizing energy production. However, traditional methods of solar power monitoring are often expensive, complex, and not scalable for small to medium-scale applications. This has led to the exploration of more cost-effective and accessible alternatives, one of the most promising of which is using **microcontroller-based systems** for solar power measurement.

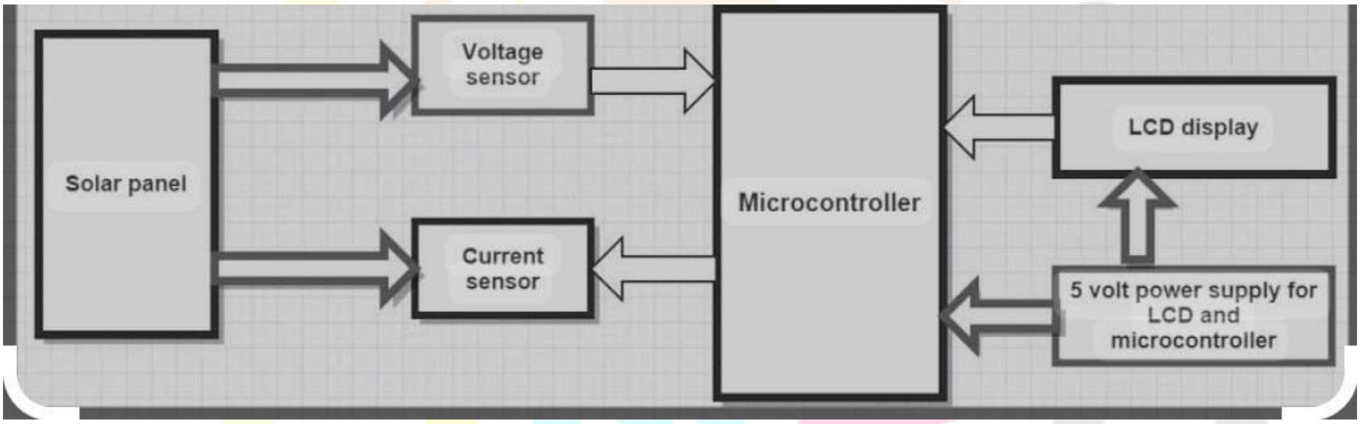
Microcontrollers, which are small and affordable embedded systems capable of processing data from external sensors, offer a practical solution for monitoring solar energy production. They allow for the real-time collection of data from various sensors that measure key parameters, including voltage, current, power output, and environmental factors such as temperature and irradiance. These systems can be easily programmed to calculate solar power, log historical data, display real-time readings, and even provide remote monitoring via cloud platforms or mobile applications.

2. Program

```
// include the library code:
#include <LiquidCrystal.h> //library for LCD

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

//Measuring Current Using ACS712
int sensitivity = 185; // use 100 for 20A Module and 66 for 30A Module
int adcvalue= 0;
int offsetvoltage = 2500;  //Sensor Callibration
double Voltage = 0; //voltage measuring
double ecurrent = 0;// Current measuring

void setup()
{
  lcd.begin(20, 4); // set up the LCD's number of columns and rows:
  lcd.setCursor(0,0);
  lcd.print("  THE BRIGHT LIGHT    ");
  lcd.setCursor(0,1);
lcd.print("  SOLAR MONITORING           ");
}

void loop()
```

```
{
  // Measure The Votage ************************************
  // read the input on analog pin A1:
  int sensorValue = analogRead(A1);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float vol = (sensorValue * 5.0) / 1023.0;

  float voltage = vol * 5;
  lcd.setCursor(0,2);
  lcd.print("  Voltage = ");
  lcd.print(voltage);
  lcd.print("V");

  // Measure The Current ************************************
  adcvalue = analogRead(A0);//reading the value from the analog pin A0
  Voltage = (adcvalue / 1024.0) * 5000; // Gets you mV
  ecurrent = ((Voltage - offsetvoltage) / sensitivity);
  lcd.setCursor(0,3);
  lcd.print("  Current = ");
  lcd.print(ecurrent);
  lcd.print("A"); //unit for the current to be measured
  delay(2000);
}
```

3.Block Digram



Future research may focus on improving the precision of Arduino-based systems, integrating artificial intelligence for better predictions and optimization, and making systems more resilient to environmental conditions for larger-scale applications.