Stand Alone PV System Supervisory Control and Data Acquisition using Arduino

SCADA system with Arduino Micro-Controller using Modbus

[1]Abhijith A R, [2]Kavya B,
[1]Project Engineer, [2]Student,
[1]Denar Automation and Controls LLP, Pune, India
[2]College of Engineering Attingal, Thiruvananthapuram, India
[1]abhijithar.eng@gmail.com, [2]kavyabiju333@gmail.com

***Abstract*—The SCADA systems are widely used to control and monitor the real time trends, production analysis etc. in industrial environments. SCADA functions will all supervisory control in the industries. It uses PLC, actuators, field devices and a reliable network protocol for gathering data and provide it's supervisory function. Due to the higher cost and requirement of many devices including PLC, small scale systems are not able to use SCADA. Here we introduce a low cost and easily configurable SCADA system which can control and monitor a Stand Alone system using Arduino Uno Micro controller. This technology will enable the use of SCADA's in low end setups also. Thus it unlocks a wide range of new applications in the control system field.**

***Index Terms*- SCADA, Arduino, PLC, Control System.**

I. Introduction

The SCADA systems are highly capable systems which can monitor and control the whole activities in a control system. It can monitor real time trends, data's, sensor values, etc. and can control the whole system using PLC signals and actuators connected in the field. As the SCADA system is having components which requires higher technical expertise for maintenance and commissioning, it is only used in large industries. Add on to that these systems are costly. So, small scale industries, schools, hospitals etc. are not able to afford such systems. The never ending possibilities of SCADA systems are being unexplored due to lack of technical expertise.

Here we have designed a SCADA model which will suit for small scale purposes which relay on an Arduino microcontroller development board for the data transfer. The Arduino acts as the heart of the system and the SCADA acts as the brain of the system. The SCADA is connected with the Arduino using the MODBUS RTU protocol. The system is integrated with a PV stand-alone system and it controls all the connected loads and measure the real time voltage and current values using sensors integrated with Arduino.

II. System Design

The system comprises of a stand-alone PV system, which can be monitored and controlled by SCADA by using Arduino Uno as the brain of the system. The developed SCADA is having supervisory control powers, in which it can switch any connected device in the system. Also, it is able to monitor the real time voltage, current readings from the solar panel and the battery storage unit. The voltage and current sensors are implemented to the PV system for the real time monitoring. The Arduino reads the sensor data and convert it into suitable format and transfer to SCADA. Using the data from the Arduino, the SCADA system plots the real time trends. The system efficiency is calculated by the Arduino and transferred to SCADA. For the supervisory control, the command will be prompted from the SCADA and that signal forces the Digital pins of Arduino. The SCADA is able to show the switching status also.

Using an Arduino Uno development board, the whole system can be brought under the control of a SCADA system. It makes the system more reliable and could be controlled from a control station. As there are tremendous possibilities for SCADA environments, it can also be controlled using all the connected devices which have the Administrator access to SCADA. In this paper we have utilized the possibilities of Modbus protocol communication for the data transfer and control transfer between Arduino and SCADA.

Here we are using the free version of Reliance SCADA Designer 4. This SCADA is able to communicate using Modbus RTU protocol. The sensor data from the field is read by the Arduino and store it into a register bank address in Modbus protocol. Which means Arduino converts the data to Modbus data. This SCADA communicate using a Modbus ID which is set in SCADA and Arduino program. If the ID matches, the SCADA will be able to read the data in Modbus register bank. The topology of the system is given in the *Fig.1, Architecture of the system.*

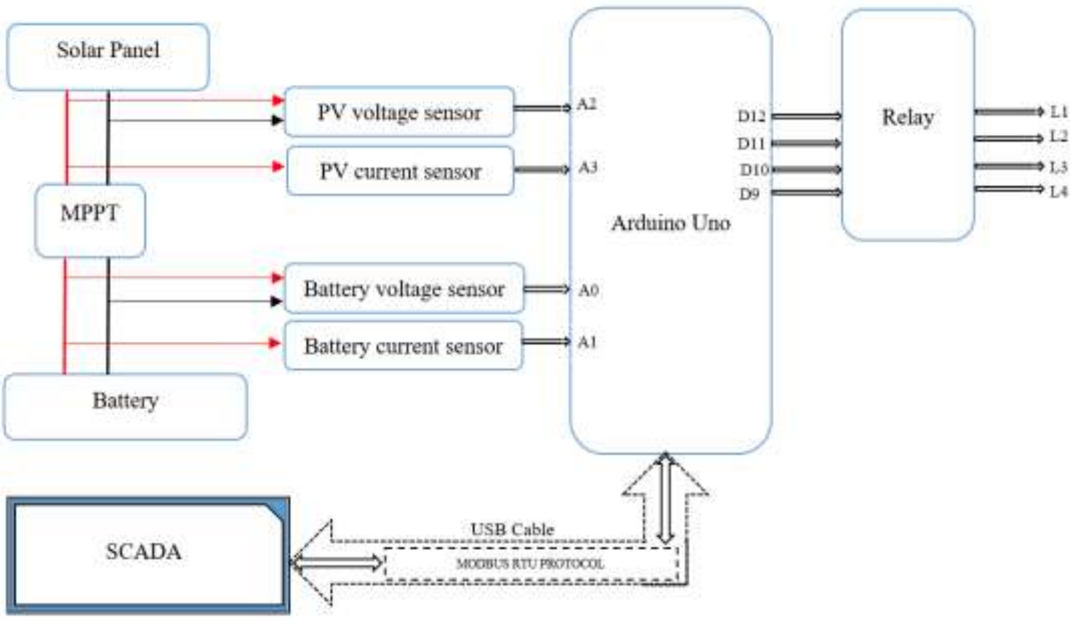

Fig.1, Architecture of the system

III. Hardware Setup

III.A. Arduino UNO Microcontroller

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. Here Arduino acts as the CPU of the system. Arduino reads sensor data from the field and transfer it to SCADA. It reads command from SCADA and actuate the relay unit.

Specifications

Microcontroller	ATmega328
Operating voltage	5V
Input voltage	7–12 V
Input voltage (limits)	6–20V
Digital I/O pins	14 (6PWMoutputs)
Analog input pins	6
DC current per I/O pin	40mA
DC current for 3.3V pin	50mA
Flash memory	32KB(ATmega328)
SRAM	2 KB(ATmega328)
EEPROM	1 KB(ATmega328)
Clock speed	16MHz

Table.1, Arduino Specifications

III.B. Current Sensor

Here ACS712 current sensor is used. It is a hall effect based linear current sensor IC designed by Allegro Microsystems.

It is having a current measurement range of ±20A. Here we use 2 current sensors. One to monitor the PV current and other to monitor the battery current.

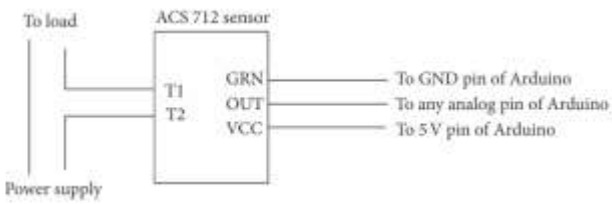

Fig.2, Current Sensor

III.C. Voltage Sensor

The voltage sensor here used is a 25V sensor with two resistors of 30KΩ and 7.3KΩ. Here we use two sensors, one for PV voltage monitoring and another one for Battery voltage monitoring.

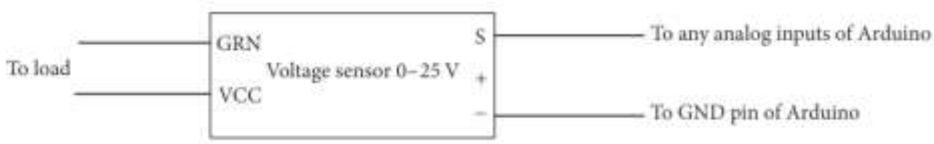

Fig.3, Voltage Sensor

III.D. Relay

Here ELEGOO 4 Channel relay module is used. This relay module is 5V active low with maximum output contact AC 250V – 10A and DC 30V – 10A. It can independently switch 4 relays which is able to control 4 field devices in our system.

IV. SOFTWARE SETUP

IV.A. Reliance Designer 4

Here the free version of Reliance Designer 4 SCADA is used. It is able to communicate with the Arduino using Modbus RTU protocol. The developed SCADA UI is having control option to control four field devices switching and monitoring options. The real-time efficiency of the system is displayed at the top.
The fig.4 shows the developed UI.

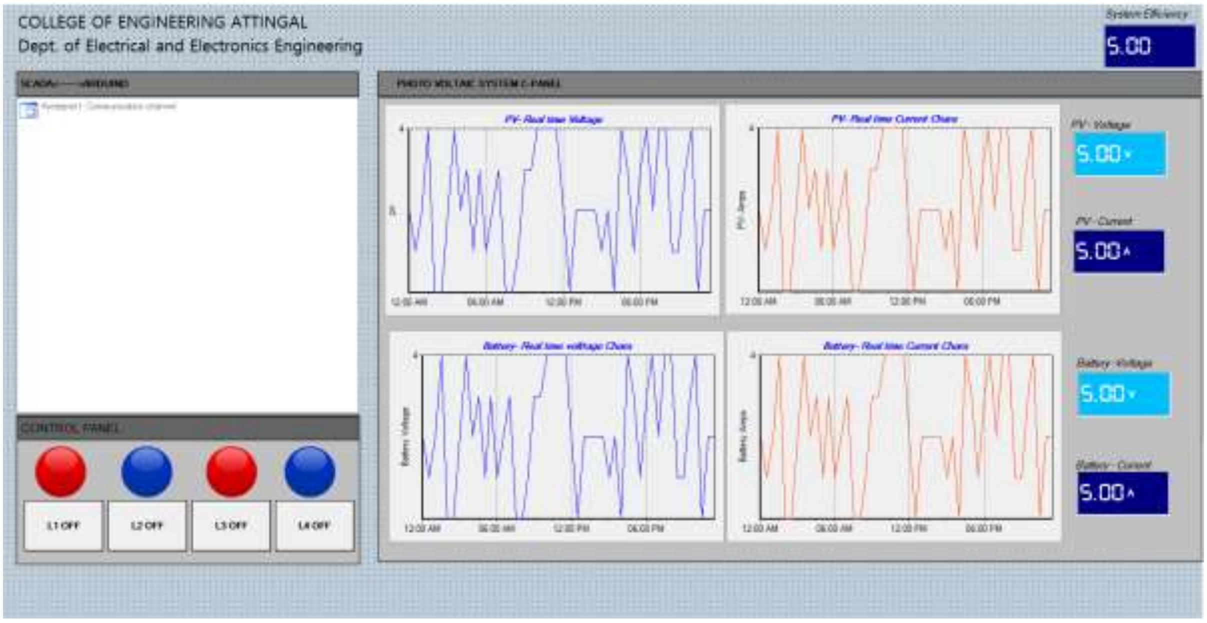

Fig.4, Developed SCADA UI

IV.B. Arduino IDE

The Arduino IDE software is used to write the program to the Arduino development board. Here we have installed Modbus library in the Arduino IDE. The Modbus addresses will be accessible after the library installation. The table shows the addressing modes in Arduino with Modbus and Reliance SCADA

Memory Area	Modbus Addressing	Reliance Addressing	
		Register Type	Address
Discrete Outputs	From 00001	Outputs(Coils)	From 0
Discrete Inputs	From 10001	Inputs	From 0
Input Registers	From 30001	Input Registers	From 0
Holding Registers	From 40001	Holding Registers	From 0

Table.2, Modbus addressing

Arduino Program:

```
#include <modbus.h>
#include <modbusDevice.h>
#include <modbusRegBank.h>
#include <modbusSlave.h>

modbusDevice regBank;
modbusSlave slave;
//PV MON
 float out1 = 0.0; float out1_dec = 0.0; float out2 = 0.0;float out2_dec = 0.0; float out3 = 0.0; float out4=0.0;float AIO;float AI1;float AI2;int AI3;
 float vpv = 0.0; float vbatt = 0.0; float eff = 0.0; float p1 =0.0; float p2 =0.0; int mVperAmp=100;int ACSoffset= 2500; double Voltage2 = 0; double Voltage3 = 0;
 float Ampspv = 0; float Ampsbatt = 0;
float R1 =30000.0; float R2 = 7500.0;
float adc_voltage = 0.0;float adc_voltage1 = 0.0;

float ref_voltage = 5.0;
```

```
int adc_value = 0;int adc_value1 = 0;
//PV MON
//const int analogPin = A3; // Analog pin connected to the ACS712 output
const float sensitivity = 0.185; // Sensitivity of ACS712-30A in V/A
const float Vcc = 5.0; // Supply voltage in volts
const float VoutQ = 2.5; // Quiescent output voltage in volts

void setup(){
  Serial.begin(9600);
  regBank.setId(10); ///Set Slave ID IN SCADA
// CONTROL IDs- 9,10,11,12

  regBank.add(9);
  regBank.add(10);
  regBank.add(11);
  regBank.add(12);
// FOR PV
 regBank.add(30001);
 regBank.add(30002);
 regBank.add(30003);
 regBank.add(30004);
 regBank.add(30005);

  slave._device = &regBank;
  slave.setBaud(9600);
//PIN MODE : CONTROLS
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
////PIN MODE : PV MON
 pinMode(1,INPUT);
 pinMode(2,INPUT);
 pinMode(3,INPUT);
}
void loop(){
  while(1){
   //THIS FOR CONTROL PURPOSE; START

    int DO9 = regBank.get(9);
    if (DO9 <= 0)digitalWrite(9,LOW);
    if (DO9 >= 1)digitalWrite(9,HIGH);
    int D10 = regBank.get(10);
    if (D10 <= 0)digitalWrite(10,LOW);
    if (D10 >= 1)digitalWrite(10,HIGH);
    int D11 = regBank.get(11);
    if (D11 <= 0)digitalWrite(11,LOW);
    if (D11 >= 1)digitalWrite(11,HIGH);
    int D12 = regBank.get(12);
    if (D12 <= 0)digitalWrite(12,LOW);
    if (D12 >= 1)digitalWrite(12,HIGH);

  int value = analogRead(A1); // Read the analog value
  float voltage = value * Vcc / 1024.0; // Convert to voltage
  float Ampsbatt = (voltage - VoutQ) / sensitivity; // Calculate current

  int value1 = analogRead(A3); // Read the analog value
  float voltage1 = value1 * Vcc / 1024.0; // Convert to voltage
  float Ampspv = (voltage1 - VoutQ) / sensitivity; // Calculate current

  adc_value = analogRead(A0);
  adc_voltage = (adc_value * ref_voltage) / 1024.0;
  out2 = adc_voltage * (R1 + R2) / R2;

  adc_value1 = analogRead(A2);
  adc_voltage1 = (adc_value1 * ref_voltage) / 1024.0;
  out1 = adc_voltage1 * (R1 + R2) / R2;
```

```
    p1= out1*Ampspv;
    p2 = out2 * Ampsbatt;
    eff = p2/p1;

    Ampspv=abs(Ampspv);
    Ampsbatt=abs(Ampsbatt);

    eff=eff*100;
   regBank.set(30001,word (out1) );
    regBank.set(30002, word (Ampspv));
    regBank.set(30003,word(out2));
    regBank.set(30004, word (Ampsbatt));
    regBank.set(30005, word (eff));
     delay(60);
     slave.run();
    }
  }
```

V. CONCLUSION

The proposed system shows that, using Modbus protocol we can communicate SCADA with different environments. That is, using this communication protocol and simple design, we can implement SCADA systems with a lesser cost. This cost effectiveness will make the system more popular among small scale setups. The never ending possibilities of SCADA is now open to small system by integrating the field with an Arduino microcontroller.

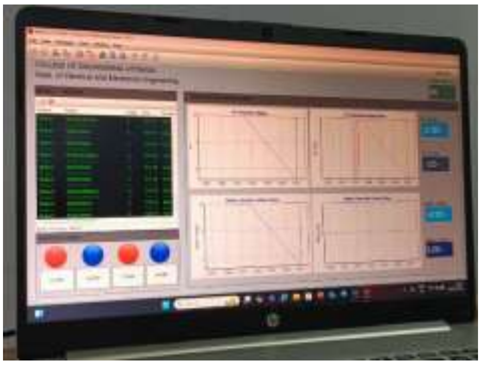

Fig.5, SCADA connected with field device

VI. ACKNOWLEDGMENT

We like to thank Department of Electrical and Electronics Engineering, College of Engineering Attingal for supporting this research by providing all hardware support.

REFERENCES

[1] IRENA International Renewable Energy Agency, Renewable Energy Technologies: Cost Analysis Series,2012.

[2] C.DumitruandA.Gligor,"SCADA Based Software for Renewable Energy Management System," Procedia Economics and Finance, vol. 3, pp. 262–267, 2012.

[3] K.Kim,G.Parthasarathy,O.Uluyol,W.Foslien,S.Sheng,and P. Fleming, "Use of scada data for failure detection in wind turbines," in Proceedings of the ASME 2011 5th International Conference on Energy Sustainability, ES 2011, pp. 2071–2079, USA, August 2011.

[4] L. Wang and K.-H. Liu, "Implementation of a web-based real time monitoring and control system for a hybrid wind-PV battery renewable energy system".

[5] Figueiredo, Jo˜ao M. G., and M. G. S´a da Costa, "Intelligent sun tracking system for efficiency maximization of International," in Proceedings of the International Conference On Renewable Energy and Power Quality,vol.2008,Santader.

[6] Figueiredo, J. M. J. Sa da Costa, and R. R. Ramalho, "Intelligent sun-tracking system for efficiency maximization of photovoltaic energy production," in Proceedings of the International Confer ence On Renewable Energy and Power Quality,vol.1,Santader, 2008.

[7] http://www.sma.de/en/products/monitoring-control/sunny-view .html.

[8] http://www.ebay.com.au/itm/SMA-Sunny-View-Colour-Touch screen-Solar-Monitor-Screen-/251743364128.

[9]M. E. Treter, L. P. Pietta, P. Xavier, andL.Michels, "Data acquisition and cloud storage system applied photovoltaic sys tems," in Proceedings ofthe13thIEEEBrazilianPowerElectronics Conference and 1st Southern Power Electronics Conference, COBEP/SPEC 2016, Brazil, December 2015.