Open Source SCADA Systems for Small Renewable Power Generation

M.Eng Student: Lawrence O. Aghenta
Thesis Supervisor: Prof. M. Tariq Iqbal

November 7, 2019
Overview:

1. Introduction
2. SCADA
3. Design and Dynamic Simulation of a Hybrid Power System
5. Open Source SCADA 2 Design: using Thinger.IO, ESP32 Thing & Wi-Fi
7. Conclusions
8. Key Contributions
9. Future Recommendations
10. Acknowledgements
11. Q/A & References

Open Source SCADA Systems for Small Renewable Power Generation
Introduction:

Motivation:

1. The distributed nature of Power Generation infrastructures

2. High cost of Commercial SCADA systems: Economically unjustifiable for smaller applications.

3. Interoperability issues, high power consumption, & the need for expensive standard communication systems in Commercial SCADA solutions.

4. The need for a cost-effective, reliable, secure, timely, flexible & sophisticated coordinated data monitoring and control system: Open Source SCADA
Introduction:

Objectives, Challenges and Work Around:

a. Objectives:
   - To design, model & dynamically simulate a Small Hybrid Power System.
   - To study some proven Commercial SCADA Systems & Open Source SCADA packages.
   - To study and test the available open source Internet of Things (IoT) platforms & use the best options.
   - To design some low-cost, low-power, reliable, secure IoT-based Open Source SCADA systems.
   - To test the designed Open Source SCADA systems with a Small Renewable Power Generation system with ESS.

b. Challenges:
   - Site for the Hybrid Power System Design.
   - Choice of SCADA components to meet low-power objective.
   - Security Strategies to meet security objective.
   - Complexity of the entire system

c. Work Around:
   - A House in Nigeria was chosen for the Hybrid Power System Design.
   - Tested Multiple components, and IoT Platforms.
   - The main data server was locally installed, self-managed and self-hosted on own private network.
   - Simple solutions proposed to overcome the complexity of the entire system.
SCADA: Description & Basic Elements

Supervisory Control

And

Data Acquisition

Field Instrumentation Devices (FIDs)
Sensors and Actuators connected to the process system being managed

Remote Terminal Units (RTUs)
Local collection points for gathering reports from FIDs (sensors)

Communications Network
Protocol needed to transfer data to and from different sites (Connects RTUs to MTU)

Master Terminal Unit (MTU)
Collecting information gathered by the remote stations to generate the necessary actions (HMI)
SCADA:

SCADA: Basic Functions

Data Acquisition

Networked Data Communication

Data Presentation

Monitoring/Control

SCADA: Architectures (Generations)

Monolithic
• 1st Generation

Distributed
• 2nd Generation

Networked
• 3rd Generation

Internet of Things
• 4th Generation

SCADA Applications

Electric power generation, transmission and distribution

Oil and Gas Production

Mass transit

Buildings, facilities and environments

Water and sewage

Open Source SCADA Systems for Small Renewable Power Generation
SCADA: Architectures (Generations)

1st Generation - Monolithic SCADA:

2nd Generation - Distributed SCADA:

3rd Generation - Networked SCADA:

4th Generation – Internet of Things (IoT)-based SCADA:

Open Source SCADA Systems for Small Renewable Power Generation
SCADA:

Desired Characteristics:

- Dynamism; flexibility.
- Retrofit; upgrading/updating
- Ease of Use/Ease of Installation
- Reliability and Availability
- Low Power Consumption
- Security: various attacks are possible; Security techniques must be decided and implemented.

Classes of SCADA:

- Proprietary (Commercial) SCADA; e.g. Siemens, ABB, GE, Wonderware, Schneider Electric, etc.
- Open Source SCADA
SCADA:

Some Studied Commercial SCADA Solutions:

1. SIMATIC WinCC: Siemens

2. ClearSCADA: Schneider Electric

3. MicroSCADA Pro: Allen Bradley

4. Ovation SCADA: Emerson

Specifications

<table>
<thead>
<tr>
<th>SCADA Communication Server Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Platform</strong></td>
</tr>
<tr>
<td><strong>Standard Communication Protocols</strong></td>
</tr>
<tr>
<td><strong>Communication Media</strong></td>
</tr>
<tr>
<td><strong>Redundancy Scheme</strong></td>
</tr>
<tr>
<td><strong>Number of Simultaneous Communication Channels</strong></td>
</tr>
</tbody>
</table>
**SCADA:**
Some Studied Open Source SCADA Solutions:

1. Rapid SCADA

2. Tango SCADA
Design and Simulation of a Hybrid Power System for a House in Nigeria:

Motivation:

- Extreme Electricity shortage and prolonged periods of power outages in Nigeria.

- Abundant Solar Irradiation in Nigeria (an average of 17.3 MJ/m²/day (4.81KWh/m²/day) in Benin City).

- Modelling and Simulation is the necessary first step in design, optimization and performance analysis.

- A Case Study of a Hybrid Power System with a Small Renewable Power Generation.
Design and Simulation of a Hybrid Power System for a House in Nigeria:

Project Steps:

- House Energy Needs determination: Thermal Modelling and Analyses using BEOpt software
- System Sizing, Optimum Design Technology and Cost Analyses using HOMER Pro software
- Dynamic/Mathematical model of each of the subsystems (components).
- Matlab/Simulink Simulation of the Proposed PV System Component of the Hybrid Power System
- Simulation results.
Open Source SCADA Systems for Small Renewable Power Generation
Main Floor and Building Foundation:

This case contains output associated with these inputs and therefore inputs are disabled. To modify inputs or create a new design, either close the existing output or create a new case.
Sample Modelling Outputs:

House Energy Needs:

Design and Simulation of a Hybrid Power System for a House in Nigeria:
House Energy Needs Determination: Thermal Modelling and Analysis (BEOpt):

Open Source SCADA Systems for Small Renewable Power Generation
The generated BEOpt Annual Hourly Load Data exported into HOMER Pro for HPS Sizing, and optimum system design.

HOMER (Hybrid Optimization of Multiple Energy Resources) software was developed by the National Renewable Energy Laboratory.

HOMER models micropower systems with single or multiple power sources: e.g. Photovoltaics, Wind turbines, etc.

HOMER, the micropower optimization model, helps to design off-grid and grid-connected systems.

HOMER analyses help to answer design questions e.g: Components size? Most cost effective technology? etc.

HOMER simulates various configurations to find the least cost combinations that meet electrical and thermal loads.
Design and Simulation of a Hybrid Power System for a House in Nigeria:

System Sizing, Optimum Technology and Cost: HOMER Pro:

HOMER Simulation Results:

Open Source SCADA Systems for Small Renewable Power Generation
Design and Simulation of a Hybrid Power System for a House in Nigeria:
System Sizing, Optimum Technology and Cost: HOMER Pro:

HOMER Simulation Results:

Open Source SCADA Systems for Small Renewable Power Generation
Design and Simulation of a Hybrid Power System for a House in Nigeria:
Dynamic/Mathematical model of each of the subsystems (components):

Block Diagram of PV System Components:
Design and Simulation of a Hybrid Power System for a House in Nigeria:

Dynamic/Mathematical model of each of the subsystems (components): Overview

- Jinko Solar JMK300M-72 PV Array: 4 strings of 12 panels PV modules. (12x4x300 = 14.4KW)

- The converter stabilizes and steps up (boosts) the DC Voltage from the PV Array to a fixed DC Voltage output.
- The converter provides the DC bus voltage (48V) for charging the Battery.
- The output voltage of the DC – DC Boost Converter is fed into the Inverter for conversion to AC voltage.
- The single phase voltage source inverter converts the fixed DC Voltage (48V) into a single-phase AC voltage (48V) with fixed frequency of 50Hz.
- The single-phase Step-up transformer steps up the 48V Inverter output to 220V AC at 50Hz for household loads
- Trojan SSIG 12 255 Lead Acid Batteries: 48V Bus Voltage, 1542Ah Capacity, 29.9 hrs Autonomy,
Incremental Conductance MPPT Algorithm and Inverter Voltage Control:

- $\frac{dI}{dV} = \text{incremental conductance.} \quad & \frac{I}{V} = \text{panel conductance}$
- Incremental and Panel conductance are determined.
- DC – DC Boost Converter Duty Cycle, D is adjusted until:
  - $\frac{dI}{dV} = -\frac{I}{V}$ or $dP/dV=0 = \text{MPPT}$
Design and Simulation of a Hybrid Power System for a House in Nigeria:

Matlab/Simulink Simulation of the PV System Components of the HPS

Open Source SCADA Systems for Small Renewable Power Generation
Design and Simulation of a Hybrid Power System for a House in Nigeria:

Simulation Results:

- **V\_PV**
- **I\_PV**
- **Ir (W/m^2)**
- **Temp (deg. C)**
- **P\_mean (kW)**
- **V\_mean (V)**
- **Duty Cycle**
Design and Simulation of a Hybrid Power System for a House in Nigeria:

Simulation Results:

Fig.: Battery I, SOC & V

Fig.: DC – DC Boost Converter Output Voltage

Fig.: Inverter Output Voltage

Fig.: Transformer Output Voltage: Load Voltage

Open Source SCADA Systems for Small Renewable Power Generation
Components of SCADA 1 System:

- A Small Renewable Power Generation System (Solar PV System with ESS).
- Sensors (Current, Voltage.): FIDs
- Arduino Uno R3 Microcontroller: RTU
- Raspberry Pi2 Single-Board Computer: RTU
- Node-RED Programming Tool: SCADA Communication/Data Transfer
- EMONCMS Local Server IoT Platform: MTU
- Computers and Mobile Devices: HMIs/GUIs
Open Source SCADA 1 Design: IoT-based, Emoncms, Arduino Uno, Raspberry Pi, Node-RED & Ethernet

System Configuration & Flow Chart:
Field Instrumentation Devices: Sensors

The ACS 712 Hall Effect Current Sensor:

- Hall Effect-Based Linear Current Sensor, 5.0 V single supply operation, 66 to 185 mV/A output sensitivity, Low cost
- Output voltage proportional to AC or DC currents, Nearly zero magnetic hysteresis
- Used to measure the DC Current from the solar PV System (the 30A Model was used for this application)

The MH Electronic Voltage Sensor Module:

- Two modules used.
- Detects the supply voltage from 0.025 to 25V
- Analog input, Low Cost
- Uses the concept of Voltage Divider to measure the supply voltage
A microcontroller board based on the ATmega328P; Version R3 used

- 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

- Programmed to measure DC Current & Voltage from the PV Panel.
Open Source SCADA 1 Design: IoT-based, Emoncms, Arduino Uno, Raspberry Pi, Node-RED & Ethernet

Raspberry Pi Single-Board Computer: RTU

- A single-board computer device with BCM2836 quad core ARMv7 processor
- Hosted the Node-RED Programming Tool for parsing data to Emoncms IoT Platform
- Connected to MUN Network via an Ethernet cable.
- Arduino Uno board connected to the RPi via a USB cable.
Node-RED Programming Tool:

- An open source programming tool for wiring together hardware devices, APIs & online services (IoT)
- Flows written to acquire Arduino Sensor Data via Serial, and Post to Emoncms IoT Server.
A powerful open-source web-app for processing, logging & visualising energy data, temperature, etc.

- Allows Users to create customized reports in the form of charts, data logs & alarms (local, email, web, and mobile apps) for remote data monitoring and control.

- Part of the OpenEnergyMonitor.org project.

- Both Local and Web-Server options available.

- Free Local Server Software option for various Operating Systems.

- Locally installed on Jetson TK1 Dev. Kit Linux Machine; Data Security technique.

- Server Self-managed and Self-hosted on Private Network (MUN Network); Data Security technique.
Open Source SCADA 1 Design: IoT-based, Emoncms, Arduino Uno, Raspberry Pi, Node-RED & Ethernet

Hardware Implementation & Experimental Setup on a Small Renewable Power Generation System:

- Hardware Prototype designed & set up in MUN ECE Laboratory for PV System Monitoring & Supervisory Control:
### Inputs

<table>
<thead>
<tr>
<th>Node</th>
<th>Key</th>
<th>Name</th>
<th>Process list</th>
<th>Updated</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>PV Current</td>
<td>Log to feed</td>
<td>5s</td>
<td>0.6</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>PV Voltage</td>
<td>Log to feed</td>
<td>5s</td>
<td>15.48</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>PV Power</td>
<td>Log to feed</td>
<td>5s</td>
<td>9.28</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>Battery Voltage</td>
<td>Log to feed</td>
<td>5s</td>
<td>14.4</td>
</tr>
</tbody>
</table>

### Feeds

<table>
<thead>
<tr>
<th>Id</th>
<th>Tag</th>
<th>Name</th>
<th>Process list</th>
<th>Public</th>
<th>Datatype</th>
<th>Engine</th>
<th>Size</th>
<th>Updated</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>…</td>
<td>PV Power_W</td>
<td></td>
<td>🗽</td>
<td>REALTIME</td>
<td>PHPFINA</td>
<td>r/a</td>
<td>8s</td>
<td>9.28</td>
</tr>
<tr>
<td>16</td>
<td>…</td>
<td>PV Current_A</td>
<td></td>
<td>🗽</td>
<td>REALTIME</td>
<td>PHPFINA</td>
<td>r/a</td>
<td>8s</td>
<td>0.6</td>
</tr>
<tr>
<td>19</td>
<td>…</td>
<td>PV Voltage_V</td>
<td></td>
<td>🗽</td>
<td>REALTIME</td>
<td>PHPFINA</td>
<td>r/a</td>
<td>8s</td>
<td>15.5</td>
</tr>
<tr>
<td>25</td>
<td>…</td>
<td>Battery Voltage_V</td>
<td></td>
<td>🗽</td>
<td>REALTIME</td>
<td>PHPFINA</td>
<td>r/a</td>
<td>8s</td>
<td>14.4</td>
</tr>
</tbody>
</table>
Open Source SCADA Systems for Small Renewable Power Generation

Open Source SCADA 1 Design: IoT-based, Emoncms, Arduino Uno, Raspberry Pi, Node-RED & Ethernet

Human Machine Interface: Operator Dashboards and Events: Emoncms IoT Platform:
Open Source SCADA 1 Design: IoT-based, Emoncms, Arduino Uno, Raspberry Pi, Node-RED & Ethernet

Human Machine Interface: Operator Dashboards and Events: Emoncms IoT Mobile App Platform:

- Screenshots of PV Current, Voltage & Power from EMONCMS Mobile App:

![Screenshots of PV Current, Voltage & Power from EMONCMS Mobile App](image_url)
Conclusions: Some Key Features of SCADA 1:

- IoT-based, Secure, Reliable, Low-cost, Low-Power, & Open Source
- Reporting, Remote Data Monitoring & Supervisory Control, Data Acquisition, Historic Data Storage & Trending, etc.

<table>
<thead>
<tr>
<th>S/N</th>
<th>COMPONENT</th>
<th>QTY</th>
<th>PRICE (CAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Emoncms Server (Jetson TK1 Dev. Kit)</td>
<td>1</td>
<td>250.00</td>
</tr>
<tr>
<td>2</td>
<td>Emoncms Software</td>
<td>1</td>
<td>00.00</td>
</tr>
<tr>
<td>3</td>
<td>Node-RED Software</td>
<td>1</td>
<td>00.00</td>
</tr>
<tr>
<td>4</td>
<td>Raspberry Pi 2 B</td>
<td>1</td>
<td>45.95</td>
</tr>
<tr>
<td>5</td>
<td>Arduino Uno</td>
<td>1</td>
<td>29.00</td>
</tr>
<tr>
<td>6</td>
<td>Current Sensor</td>
<td>1</td>
<td>5.25</td>
</tr>
<tr>
<td>7</td>
<td>Voltage Sensor</td>
<td>2</td>
<td>11.98</td>
</tr>
<tr>
<td>8</td>
<td>8GB SD Card</td>
<td>1</td>
<td>12.66</td>
</tr>
<tr>
<td>9</td>
<td>Miscellaneous (Wires, Boxes, etc.)</td>
<td>1</td>
<td>50.00</td>
</tr>
<tr>
<td></td>
<td><strong>Grand Total:</strong></td>
<td></td>
<td><strong>$ 404.84 CAD</strong></td>
</tr>
</tbody>
</table>

Table A.2: Power consumption of hardware components.

<table>
<thead>
<tr>
<th>S/N</th>
<th>HARDWARE</th>
<th>POWER (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raspberry Pi 2 + Arduino Uno + Sensors</td>
<td>2.4</td>
</tr>
<tr>
<td>2</td>
<td>Emoncms Server</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td><strong>Total Power Consumption:</strong></td>
<td><strong>5.1 W</strong></td>
</tr>
</tbody>
</table>
Open Source SCADA 2 Design: IoT-based, Thinger.IO, Raspberry Pi, ESP32 Thing & Wi-Fi

Components of SCADA 2 System:

- A Small Renewable Power Generation System (Solar PV System with ESS).
- Sensors (Current, Voltage.): FIDs
- ESP32 Thing Micro-controller: RTU
- Raspberry Pi2 machine (Thinger.IO IoT Server): MTU
- Wi-Fi Router: SCADA Communication/Data Transfer
- Computers and Mobile Devices: HMIs/GUIs
System Configurations; Two Configurations considered

Configuration A: MUN Network (Internet Required)

Configuration B: Private Industrial Network created (Internet Optional)
Field Instrumentation Devices: Same Sensors as in SCADA 1

ESP32 Thing Micro-controller: RTU

- Low-Cost ($20), Low-Power (0.5W) Micro-controller for IoT Projects
- Supports Wi-Fi, 30 Input/Output Pins
- 5 V Power Supply, 2.2 V to 3.6 V (3.3 V) ADC Signal Voltage
- Programmed with Arduino IDE to measure & acquire Sensor PV Data & parse data over Wi-Fi to Thinger.IO IoT Server

Open Source SCADA 2 Design: IoT-based, Thinger.IO, Raspberry Pi, ESP32 Thing & Wi-Fi

SparkFun ESP32 Thing (DEV-13907)
Open Source SCADA 2 Design: IoT-based, Thinger.IO, Raspberry Pi, ESP32 Thing & Wi-Fi

Raspberry Pi Single-Board Computer: MTU

- Hosted the Thinger.IO IoT Server
- Connected to MUN Network via an Ethernet cable in Configuration A.
- Connected to Wi-Fi Router LAN Port via an Ethernet cable in Configuration B.

SCADA Communication Channel: Wi-Fi Router for Private Wi-Fi Network

- D-Link Router (DI-524 Airplus G) used to create a private Wi-Fi Network
Thinger.IO IoT Server: Overview

- A Powerful Open Source platform for the Internet of Things; supported by GitHub
- Supports REST API for controlling & reading smart devices; e.g: sensors
- Uses HTTP methods like OPTIONS, GET, PUT, POST, and DELETE.
- Specific response of each resource gotten in JSON/XML format.
- Allows integration of Arduino-compatible hardware; e.g: ESP32
- Both Web-Server and Self-Hosted options available
- Connected Devices identified with Unique Identifier and Credentials
- Cloud Console shows Google Maps of Connected Devices
- Allows an operator to define 4 resources (Input, Output, Input/Output, Callback resources)
- Supports Real-time visualization dashboards & charts for remote monitoring
- Supervisory Controls possible using Endpoints (emails, HTTP Requests, etc)
- Free Android & iOS Apps for Server integration available

Open Source SCADA Systems for Small Renewable Power Generation
Thinger.IO IoT Server: Configurations

- Main Data Server installed on a Local Machine (Rpi), Self-hosted, Self-managed on private networks
Prototype Design & Experimental Setup:

- Hardware implemented & System Set up on a Small Renewable Power Generation System (Solar PV System) at MUN ECE Lab:
Open Source SCADA 2 Design: IoT-based, Thinger.IO, Raspberry Pi, ESP32 Thing & Wi-Fi

Implementation Methodology & Flow Chart:

**Algorithm 1: Data Logging Algorithm:**

Initialization;
1. Read Sensor Values on Analog Pins 32, 34 and 35, and Calculate Values for Pins 32 x 34;
2. Display the above Values on Arduino IDE Serial Monitor;
3. Connect to Local Wi-Fi Network with Wi-Fi Name and Password;
4. Connect to Thinger.IO Local Server IP Address;
5. Identify the specified Thinger.IO Account Name, Device ID and Credentials;
6. Post Sensor Data to the specified Thinger.IO Device;

while Thinger.IO Server Acknowledges Data Receipt do
  7. Display Sensor Data on Thinger.IO Cloud Console, and;
  8. Display “Ok” on Arduino IDE Serial Monitor;
  if No Data Receipt Acknowledgement from Thinger.IO Server then
    9. Display Debug/Error Message on Arduino IDE Serial Monitor;
  else
    10. Go to Step 1;
end
Open Source SCADA 2 Design: IoT-based, Thinger.IO, Raspberry Pi, ESP32 Thing & Wi-Fi

Data Logging & Historic Storage:
Open Source SCADA 2 Design: IoT-based, Thinger.IO, Raspberry Pi, ESP32 Thing & Wi-Fi

Human Machine Interface (HMI): PV Data Monitoring & Supervisory Control:

- PV Voltage (V)
- PV Current (A)
- PV Voltage 2 (V)
- PV Current 2 (A)
- Battery Voltage (V)
- PV Power (W)
- PV Power 2 (W)
- Battery Voltage 2 (V)

Google Map of Device Location

Open Source SCADA Systems for Small Renewable Power Generation
Conclusions: Some Key Features of SCADA 2:

- IoT-based, Secure, Reliable, Low-cost, Low-Power, & Open Source
- Reporting, Remote Data Monitoring & Supervisory Control, Data Acquisition, Historic Data Storage & Trending, etc.

Table 4.1: Bill of Materials.

<table>
<thead>
<tr>
<th>S/N</th>
<th>COMPONENT</th>
<th>QTY</th>
<th>PRICE (CAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Thinger.IO RPi ISO Image</td>
<td>1</td>
<td>15.62</td>
</tr>
<tr>
<td>2</td>
<td>Raspberry Pi 2 B</td>
<td>1</td>
<td>45.95</td>
</tr>
<tr>
<td>3</td>
<td>ESP32 Thing</td>
<td>1</td>
<td>31.90</td>
</tr>
<tr>
<td>4</td>
<td>Current Sensor</td>
<td>1</td>
<td>5.25</td>
</tr>
<tr>
<td>5</td>
<td>Voltage Sensor</td>
<td>2</td>
<td>11.98</td>
</tr>
<tr>
<td>6</td>
<td>D-Link D1-524 Wireless Router</td>
<td>1</td>
<td>98.51</td>
</tr>
<tr>
<td>7</td>
<td>8GB SD Card</td>
<td>1</td>
<td>12.66</td>
</tr>
<tr>
<td>8</td>
<td>Miscellaneous (Breadboard, Resistors, Wires, Boxes, etc.)</td>
<td>1</td>
<td>70.00</td>
</tr>
<tr>
<td></td>
<td><strong>Grand Total:</strong></td>
<td></td>
<td><strong>$ 291.87 CAD</strong></td>
</tr>
</tbody>
</table>

Table 4.2: Power consumption of hardware components.

<table>
<thead>
<tr>
<th>S/N</th>
<th>HARDWARE</th>
<th>POWER (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raspberry Pi 2 B</td>
<td>1.7</td>
</tr>
<tr>
<td>2</td>
<td>ESP32 Thing (alone)</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>Breadboard (with Sensors, ESP32, Resistors, etc. connected)</td>
<td>3.3</td>
</tr>
<tr>
<td>4</td>
<td>D-Link D1-524 Wireless Router</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td><strong>Total Power Consumption (less ESP32 alone):</strong></td>
<td><strong>9.4 W</strong></td>
</tr>
</tbody>
</table>
Components of SCADA 3 System:

- A Small Renewable Power Generation System (Solar PV System with ESS).
- Sensors (Current, Voltage.): FIDs
- ESP32 with OLED Micro-controller: RTU
- Wi-Fi Router: SCADA Communication Network
- MQTT Protocol: Data Transfer
- Raspberry Pi2 machine (ThingsBoard IoT Server): MTU
- Computers and Mobile Devices: HMIs/GUIs
Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

System Configurations; Two Configurations considered

Configuration A: MUN Network (Internet Required)

Configuration B: Private Industrial Network created (Internet Optional)
Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

Field Instrumentation Devices: Same Sensors as in SCADA 1

ESP32 with OLED: RTU

- TTGO ESP32 LoRa32 micro-controller board with 0.96 inch OLED display
- 18 ADC pins and over 30 GPIO pins (I/O Pins), Supports Wi-Fi
- Low-Cost ($20), Low-Power (0.9W) Micro-controller for IoT Projects
- 5 V Power Supply, 1.8 V to 3.7 V ((3.3 V) ADC Signal Voltage
- Programmed as an MQTT Client using Arduino IDE & MQTT Client Library (PubSubClient).
**Open Source SCADA 3 Design:** IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

**Raspberry Pi Single-Board Computer: MTU**

- ThingsBoard IoT Server & PostgreSQL Database installed on the Raspberry Pi
- Connected to MUN Network via an Ethernet cable in Configuration A.
- Connected to Wi-Fi Router LAN Port via an Ethernet cable in Configuration B.

**SCADA Communication Channel:** Wi-Fi Router for Private Wi-Fi Network:

- D-Link Router (DI-524 Airplus G) used to create a private Wi-Fi Network
Internet of Things (IoT) & MQTT Protocol: Overview

- A reliable communication is necessary in an IoT-based application like SCADA
- IoT Technologies use the four layers of the general TCP/IP model.
- Important choice for IoT-based SCADA: communication protocols, message encoding format, & the web/IoT platform
- MQTT: a lightweight M2M data communication protocol; supports applications with limited resources, e.g., low bandwidth
- MQTT Protocol implemented over TCP/IP Wi-FI for sensor data transfer.

Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

Open Source SCADA Systems for Small Renewable Power Generation
ThingsBoard IoT Server: Overview

- An Open Source IoT Platform for data collection, processing, visualization & device management
- 100% Support for standard IoT protocols, e.g.; MQTT, CoAP, etc.
- Live Demo (Web), Cloud Server (e.g.; AWS) & On-premise Server options available.
- Supports various hardware integration, e.g.; ESP32, Arduino, ESP8266, etc.
- Supports customizable real-time dashboards for data visualization, alarms, device management, etc.
- Server nodes act as MQTT Broker that supports QoS levels 0 & 1
- Uses Database to store Entities (e.g.; devices, dashboards) & Telemetry Data (e.g.; Sensor Readings)
- Supports 3 different Database options; SQL (e.g.; Postgres), NoSQL (e.g.; Cassandra), & Hybrid
- Stores received values as Telemetries: Time-series key-value pairs.
- Community & Professional Editions available: Community Edition used (Free & supported by GitHub)
- On-premise Server option installed on a Raspberry Pi alongside PostgreSQL Database: Data Security strategy.
Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

ThingsBoard IoT Server: Architecture

System Rule Engine:

Alarm Rule Engine:
Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

ThingsBoard IoT Server: Local Server Configuration

Rpi-installed ThingsBoard Server Platform:

Connected ESP32 OLED Device:
Open Source SCADA Systems for Small Renewable Power Generation

Implementation Methodology:

Algorithm 2: Data acquisition and logging algorithm:

Initialization:
1. Analog sensors measure and collect PV system data;
2. ESP32 reads sensor values on analog Pins 32, 34 and 35, and calculates values for Pins 32x34;
3. ESP32 displays the above values on Arduino IDE Serial Monitor and ESP32 OLED Screen;
4. ESP32 connects to local TCP/IP Wi-Fi Network with Wi-Fi Name and Password;
5. ESP32 MQTT Client identifies the local ThingsBoard IoT Server (MQTT Broker) via the Server IP Address;
6. ESP32 MQTT Client publishes sensor data to MQTT Broker over the TCP/IP Wi-Fi connectivity;
7. ThingsBoard Server displays data as Telemetry Messages on the specified Device using the Device Name and Access Token;
8. ThingsBoard Server Node logs the Telemetry Messages to Dashboards for data visualization;

while ThingsBoard Server acknowledges data receipt do
9. Display sensor data on ThingsBoard Server Node, Dashboards and ESP32 OLED Screen, and;
10. Display "DONE" on Arduino IDE Serial Monitor;
if No data receipt acknowledgement from ThingsBoard Server Node then
11. Display "FAILED......retrying in 5 seconds" on Arduino IDE Serial Monitor;
else
12. Go to step 1;
end
end

Real-time PV Data Posting:
Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

Prototype Design & Experimental Setup:

- Hardware implemented & System Set up on a Small Renewable Power Generation System (Solar PV System) at MUN ECE Lab:
Open Source SCADA Systems for Small Renewable Power Generation

Human Machine Interface (HMI): PV Data Monitoring & Supervisory Control:

Dashboards:

Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

Alarm:
Open Source SCADA 3 Design: IoT-based, ThingsBoard, Raspberry Pi, ESP32 with OLED, Wi-Fi & MQTT Protocol

Conclusions: Some Key Features of SCADA 3:

- IoT-based, Secure, Reliable, Low-cost, Low-Power, & Open Source
- Reporting, Remote Data Monitoring & Supervisory Control, Data Acquisition, Historic Data Storage & Trending, etc.
- Local Operator Data Monitoring Interface

Table 5.1: Bill of Materials.

<table>
<thead>
<tr>
<th>S/N</th>
<th>COMPONENT</th>
<th>QTY</th>
<th>PRICE (CAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ThingsBoard IoT Server</td>
<td>1</td>
<td>00.00</td>
</tr>
<tr>
<td>2</td>
<td>PostgreSQL</td>
<td>1</td>
<td>00.00</td>
</tr>
<tr>
<td>3</td>
<td>Raspberry Pi 2 B</td>
<td>1</td>
<td>45.95</td>
</tr>
<tr>
<td>4</td>
<td>TTGO LoRa32 ESP32 OLED</td>
<td>1</td>
<td>17.49</td>
</tr>
<tr>
<td>5</td>
<td>16GB Memory Card</td>
<td>1</td>
<td>19.99</td>
</tr>
<tr>
<td>6</td>
<td>Voltage Sensor</td>
<td>2</td>
<td>11.98</td>
</tr>
<tr>
<td>7</td>
<td>Current Sensor</td>
<td>1</td>
<td>5.25</td>
</tr>
<tr>
<td>8</td>
<td>D-Link D1-524 Wireless Router</td>
<td>1</td>
<td>98.51</td>
</tr>
<tr>
<td>9</td>
<td>Miscellaneous (Boxes, Breadboard, Resistors, Wires, etc.)</td>
<td>1</td>
<td>80.00</td>
</tr>
<tr>
<td></td>
<td><strong>Grand Total:</strong></td>
<td></td>
<td><strong>$279.17 CAD</strong></td>
</tr>
</tbody>
</table>

Table 5.2: Power Consumption of Hardware Components.

<table>
<thead>
<tr>
<th>S/N</th>
<th>HARDWARE</th>
<th>POWER(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raspberry Pi 2 B</td>
<td>1.8</td>
</tr>
<tr>
<td>2</td>
<td>ESP32 OLED (alone)</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>D-Link D1-524 Wireless Router</td>
<td>4.2</td>
</tr>
<tr>
<td>4</td>
<td>Breadboard (with ESP32 OLED, Sensors, Resistors, etc. connected)</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td><strong>Overall System Power Consumption (less ESP32 OLED alone):</strong></td>
<td><strong>9.3 W</strong></td>
</tr>
</tbody>
</table>

Open Source SCADA Systems for Small Renewable Power Generation
Conclusions:

- Designed and simulated a Hybrid Power System with a Small Renewable Power Generation System & ESS.
- Designed & implemented 3 different Low-cost, Low-Power, Secure, Reliable, IoT-based Open Source SCADA Systems.
- The designed Open Source SCADA systems used for a Small Renewable Power Generation System.

Table 6.1: Comparison Between The Three IoT-Based Open Source SCADA Systems.

<table>
<thead>
<tr>
<th>Key Properties</th>
<th>Emoncms-based SCADA (SCADA 1)</th>
<th>Tinker.IO-based SCADA (SCADA 2)</th>
<th>ThingsBoard-based SCADA (SCADA 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT Server (MTU) RTU</td>
<td>Emoncms Arduino Uno/Raspberry Pi Node-RED/Ethernet</td>
<td>Tinker.IO ESP32 Thing</td>
<td>ThingsBoard ESP32 with OLED</td>
</tr>
<tr>
<td>Communication Channel</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FIDs</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security Measure</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reliable from Testing? Alarm &amp; Notifications</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline Option</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Local Operator Interface</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cost</td>
<td>$404.84 CAD 5.1 W</td>
<td>$291.87 CAD 9.4 W</td>
<td>$279.17 CAD 9.3 W</td>
</tr>
<tr>
<td>Power Consumption</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Key Contributions:

- A house in Nigeria chosen for the HPS Design; House Energy Assessments; Optimum Renewable Generation System Sizing.
- Implementation of SCADA System security strategies on each system.
- Selection & Testing of each SCADA Element for overall system design goals, e.g. Low-power, low-cost, reliability, etc.

List of Publications:

**Refereed Journal Articles:**


**Refereed Conference Publication:**


**Regional Conference Publications:**


**Poster Presentation:**

Future Recommendations:

- Detailed Reliability Calculations & Analysis of each of the Open Source SCADA Systems for Reliability Comparison.

- Inclusion of Data Encryption on each SCADA Communication Channel for greater security.

- Emoncms installed on a more recent Linux machine, while using ESP32 Thing as the RTU, and MQTT protocol for data transfer from the RTU to the MTU to improve on SCADA 1 & reduce cost.

- Configuring the Raspberry Pis in SCADA 2 & 3 as a Wireless Access Point to provide the needed Wi-Fi connections, thereby eliminating the Wi-Fi Router in each case to further reduce power consumption.

- Developing new Open Source SCADA Systems using CoAP API and HTTP API of the ThingsBoard platform for data transfer so as to compare their performance with the MQTT API used in SCADA 3 option.
Acknowledgements:

Thank you!

- My Supervisor: Professor M. Tariq Iqbal.

- Funding Support: School of Graduate Studies, Faculty of Engineering & Applied Science MUN, NSERC & NESTNet.

- My Colleagues, Friends & Family.
Acknowledgements:

Thank you!

Contributions & Questions???
Some References:


