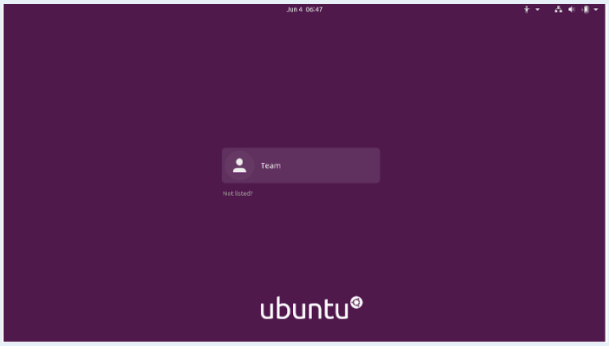
CHIPKIN

Case Study

BACnet Stack on Ubuntu to provide a Self-Configuring BACnet Server

Overview - BACnet Stack, Ubuntu and Self-Configuring BACnet Server



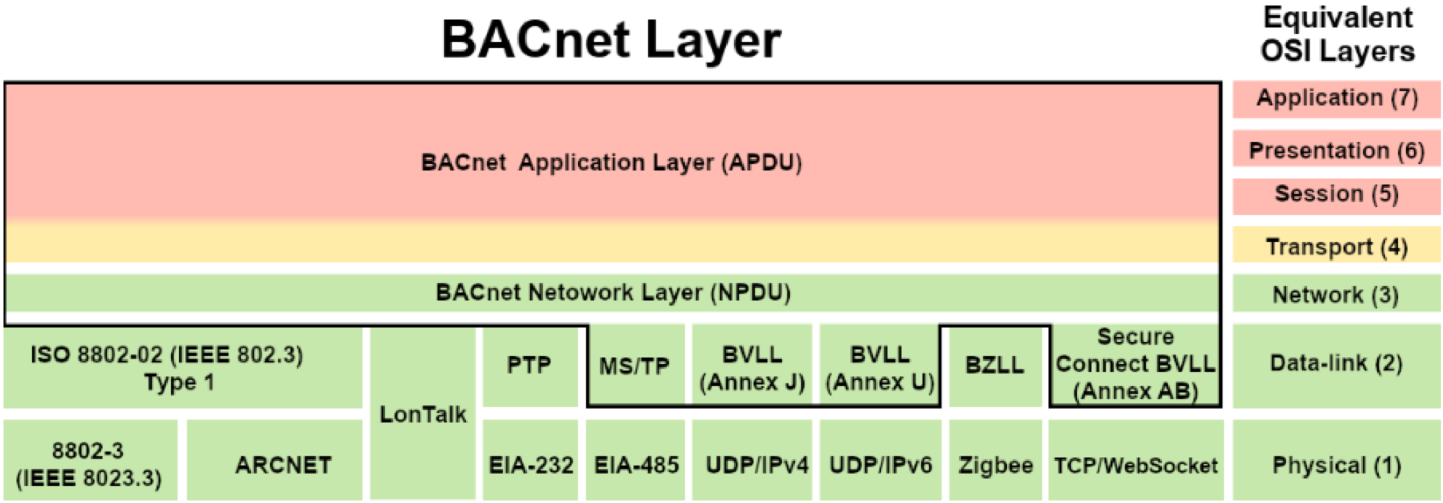
This case study delves into the implementation of a Self-Configuring BACnet Server using the CAS BACnet Stack on the Ubuntu operating system. BACnet, a widely adopted communication protocol in building automation systems, is harnessed to create a dynamic and adaptive server configuration.

The primary goal is to showcase the seamless integration of a BACnet Stack on Ubuntu, transforming it into a Self-Configuring BACnet Server. This approach aims to enhance the flexibility and efficiency of building automation by automating the server configuration process.

About Chipkin Automation Systems

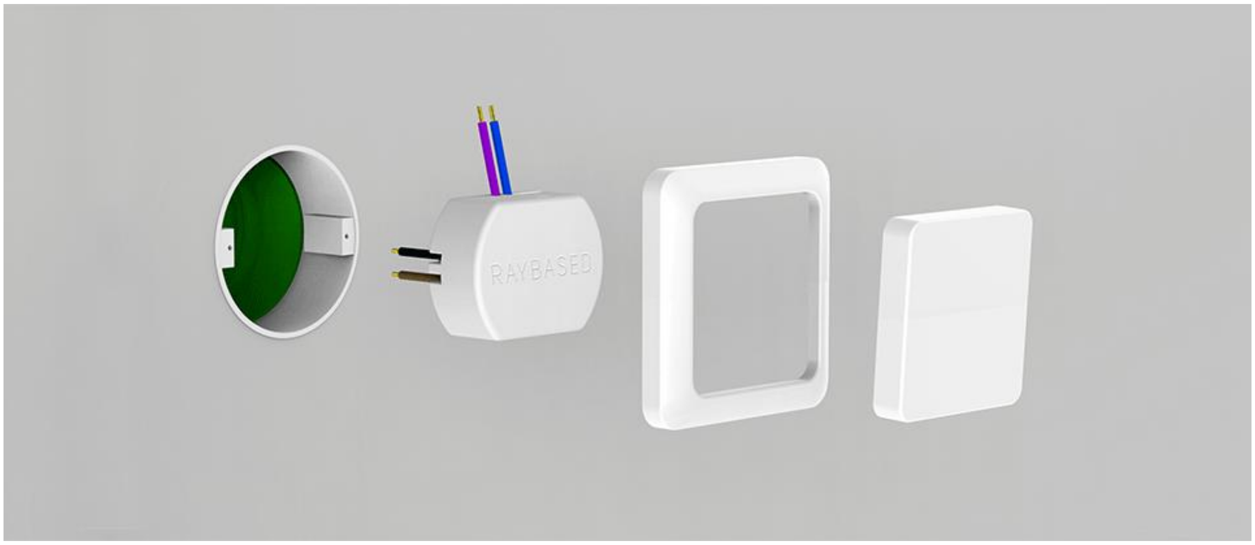
Chipkin excels in protocol solutions. Supporting 140+ protocols and providing custom drivers for over a dozen customers yearly, Chipkin has garnered a reputation for unparalleled Customer support.

Chipkin's BACnet stack guarantees 100% copyright infringement indemnity and supports the BACnet Layers listed on the image below. As part of the BACnet Stack, Chipkin offers direct access to stack developers for troubleshooting, training, and implementation. Chipkin's stack is a royalty-free software library that allows you to add a native BACnet interface to your embedded devices or applications. With support for protocol revision 19, Chipkin's BACnet Stack is a market leader and one of the most up-to-date stacks available.



About Raybased

Raybased began operations on a limited scale in 2011. Since then, Raybased has developed an open wireless platform for advanced building automation. The system makes it possible to design applications that control and optimize building functions, such as heating, ventilation, lighting, and security. Raybased thus combines the Internet of Things with building automation. Raybased primarily targets reconstruction of existing commercial properties. These often have a great need of improved energy efficiency but struggle to motivate the high total cost that professional systems entail.



Project Overview: Integrating Raybased's REST JSON API with BACnet

Raybased is actively developing a REST JSON API designed to retrieve real-time sensor data, manage setpoints, and control relays. Our project involves the seamless integration of this valuable data into the BACnet protocol using the robust CAS BACnet stack. A key requirement is the development of a Linux application specifically tailored for Ubuntu 14.04.

In the implementation, we harnessed the power of several essential libraries:

- **CAS BACnet API:** This library serves as the linchpin, providing crucial functionality for seamless integration with the BACnet protocol.

❒ **Curl:** Instrumental in the process, Curl takes charge of constructing, sending, and decoding web requests, ensuring efficient communication between systems.
❒ **Rapidjson:** A pivotal component, Rapidjson specializes in parsing complex JSON documents, facilitating the smooth exchange of information.

Raybased has thoughtfully exposed three primary REST API endpoints, shaping the interaction for the BACnet application:

1. **Metadata:** The metadata endpoint facilitates the initial setup of the BACnet interface by describing settings, device properties, objects, and object properties in JSON format.
2. **Read:** The read endpoint provides current values of BACnet-mapped data points, incorporating reliability indicators for accurate reflection of sensor outages, with data stored in the memory database and accessible through BACnet.
3. **Write:** The write functionality enables the BACnet application to send commands to the REST API for altering setpoints or activating/deactivating relays.

This comprehensive integration not only aligns Raybased's innovative REST JSON API with BACnet but also sets the stage for enhanced functionality and interoperability within the broader building automation ecosystem. The utilization of robust libraries ensures the reliability and efficiency of data exchange, marking a significant stride towards a seamlessly connected and intelligently controlled environments. The Project was a great success which led to the client hiring Chipkin to implement the same for Modbus.