

Welcome to Picominer® IoT Operational Data Platform

/*=====LICENSE NOTICE=====*/

/*

- * Software: IoT Enabled Business Solutions.
- * Product code: SDL-MODBUS-D21, PM-FIN-L21 (= NCAP-FIN-L21), PM-FIN-S21
- * Platform code: PM.ODP
- * Type of software: Application.
- * License: Refer to LICENSE file of the software package. GPL-v3.
- * Company: Vidcentum R&D Pvt Ltd, India (CIN: U72200TG2009PTC063723).
- * Copyright (c) 2020-2021.
- * Email: support@vidcentum.com
- * Website: <https://vidcentum.com>
- * Address: Plot No. 1-62/20, Kavuri Hills,
- * Hyderabad - 500081, Telangana, India.

*/

/*****

The software and services are confidential and proprietary of Vidcentum.
You shall not disclose such Confidential Information and shall use it only in
accordance with the terms of the license agreement you entered with Vidcentum.

Vidcentum acknowledges and honours the open source software modules and
respective licenses.

The software is distributed with source to its users / customers / stakeholders.

External Libraries: This software uses external libraries with licenses MIT,
GPL, Apache etc. The software is distributed with source to its users, customers
and stakeholders as part of its service contract.

*****/

/*****

You MAY redistribute it and/or modify it under the terms of the GNU General
Public License as published by the Free Software Foundation, either version 3
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<https://www.gnu.org/licenses/>>.

*****/

/*=====END LICENSE NOTICE=====*/

SUBSYSTEMS.

SDL-MODBUS. Current Release Version: 1.1.1 (28/Sep/2021)

Data collection is a critical step to sustain the production and the processes. *SDL-MODBUS* integrates factory, plant, and field operations. Smart Data Logger (Modbus) is a software program / framework to develop Modbus/TCP based applications.

Advantages

1. Reduces the application development time. It is easy to build and run in the lab or application environment.
2. Change the software to suit the target environment.
3. Full control over the data.
4. Extend it with new features or functionalities.
5. Acts as application test-bed.
6. Regular software updates.
7. Flexible licensing for better development and run-time management.
8. Get professional support (choose the plan that is suitable for you).

Features

1. Modbus/TCP support.
2. Simple configuration with YAML files.
3. Fine-grain control of reading Modbus/TCP registers, coils and input bits.
4. Support multiple Modbus devices simultaneously.
5. Manage metadata easily.
6. HDF5 datasets.
7. Upstream and downstream integrations.
8. Logging.
9. Deploy compiled sources (C/C++).
10. SI units.
11. Integration with Network Capabilities with MQTT, Websockets, REST etc.

System Requirements

Development Setup

1. Ubuntu 20.04+ Linux, x86_64
2. Compile on Raspberry Pi target.
3. CodeLite IDE, Code::Blocks IDE
4. C++17, CMake (Latest versions of CLang, GCC)

Deployment (Target System)

1. Ubuntu 20.04 Linux, x86_64 desktop / workstation or cloud VM

2. Raspberry Pi with at least 2GB RAM, 64 Bit (GUI is needed.)

Build process

Prerequisites

Install GCC / G++ Install latest gcc/g++, build-essential, binutils, autoconf, cmake etc from the linux distribution.

Install YAML/JSON utilities

Software: nodejs. Install from the linux distribution packages.

npm packages:

```
sudo npm install -g yamltojson
```

```
sudo npm install -g jsontoyaml
```

Install Boost Libraries

Software Module: Boost C++

You may compile the library and install locally. Follow the instructions of the package.

Source: <https://www.boost.org/>

Licence: <https://www.boost.org/users/license.html>

Linux Package:

You may install Boost libraries from the Linux package manager as well

```
sudo apt-get install libboost-dev
```

Or follow the build instructions - <https://www.boost.org/>

Install openssl-dev and curl

Software Module: cUrl

Source: <https://github.com/curl/curl>

Licence: Open Software

<https://github.com/curl/curl/blob/master/COPYING>

```
sudo apt install libcurl4-openssl-dev
```

Note: OpenSSL is used.

Prerequisites: openssl dev

Install OpenGL development libraries

```
sudo apt-get install build-essential libgl1-mesa-dev # Debian, Ubuntu
```

```
sudo apt-get install libglew-dev libsdl2-dev libsdl2-image-dev libglm-dev libfreetype6-dev
```

```
sudo apt-get install libglfw3-dev
```

```
sudo apt-get install libfreetype-dev libfreetype6 libfreetype6-dev
```

Install libmodbus

Software Module: libmodbus. Install from the linux distribution packages. You may compile the library and install locally. Follow the instructions of the package.

Source: <https://github.com/stephane/libmodbus/>

Licence: GNU Lesser General Public License v2.1.

<https://github.com/stephane/libmodbus/blob/master/COPYING.LESSER>

Linux Package:

You may install libmodbus from the Linux package manager as well

```
sudo apt install libmodbus-dev
```

Build libmodbus

Clone the libmodbus sources.

```
git clone --recursive --recurse-submodules https://github.com/stephane/libmodbus.git
```

```
// System-wide installation
```

```
export CFLAGS+=" -Os "
```

```
export CPPFLAGS+=" -g -Wall -fPIC "
```

```
export CXXFLAGS+=" -g -Wall -fPIC "
```

```
./autogen.sh
```

```
./configure --enable-static
```

```
make
```

```
make check
```

```
sudo make install
```

For static linking

libmodbus.a

For dynamic linking

modbus

Install HDF5

Software Module: HDF5. Install from the linux distribution packages.

You may compile the library and install locally. Follow the instructions of the package.

Source: <https://www.hdfgroup.org/solutions/hdf5/>

Licence: HDF5 (Hierarchical Data Format 5) Software Library and Utilities.

<https://github.com/HDFGroup/hdf5/blob/develop/COPYING>

HDF5 Version: 1.10 version. HDF5 Viewer is built with HDF5 1.10.

Linux Package:

You may install HDF5 from the Linux package manager as well

```
sudo apt install libhdf5-dev
```

Build HDF5

If you would like to build HDF5 on target, you may follow the instructions below.

```
export CFLAGS+=" -Os "  
export CPPFLAGS+=" -g -Wall -fPIC "  
export CXXFLAGS+=" -g -Wall -fPIC "
```

```
./configure --prefix=/usr/local \  
            --with-zlib \  
            --enable-build-mode=production \  
            --with-pic\  
            --enable-hl \  
            --enable-cxx \  
            --enable-build-all \  
            --enable-java
```

```
make  
make check  
sudo make install
```

C/C++ Libs: (preserve the order)

For static linking

libhdf5_hl_cpp.a	- HDF5 High Level C++ APIs
libhdf5_cpp.a	- HDF5 C++ Library
libhdf5_hl.a	- HDF5 High Level C APIs
libhdf5.a	- HDF5 C Library

For dynamic linking

hdf5_hl_cpp	- HDF5 High Level C++ APIs
hdf5_cpp	- HDF5 C++ Library
hdf5_hl	- HDF5 High Level C APIs
hdf5	- HDF5 C Library

Install OpenJDK 15 for HDFView

Released on 15 September 2020, the OpenJDK 15 is the latest feature release of JDK. But it is not the part of Ubuntu 20.04 base repository.

To install OpenJDK 15, download its Debian package from the Oracle official website <https://www.oracle.com/java/technologies/javase-jdk15-downloads.html>.

(OR)

JAVA JDK Archives: <http://jdk.java.net/archive/>

https://download.java.net/java/GA/jdk15.0.2/0d1cfde4252546c6931946de8db48ee2/7/GPL/openjdk-15.0.2-linux-x64_bin.tar.gz

Download the above `openjdk-15.0.2_linux-x64_bin.tar.gz`

```
extarct
```

```
..> jdk-15.0.2
```

copy to `/usr/lib/java/` or (or wherever you prefer for example, `/opt/java/`)

```
cd /usr/lib/java/  
    +/jdk-15.0.2
```

Use the above path (or wherever you prefer)

```
$ export PATH=$PATH:"/usr/lib/java/jdk-15.0.2/"
$ export JAVA_HOME="/usr/lib/java/jdk-15.0.2"
```

```
$ HDFView
```

Install libuv

Software Module: libuv. Install from the linux distribution packages.
You may compile the library and install locally. Follow the instructions of the package.
Source: <https://github.com/libuv/libuv>
Licence: Open Source.
<https://github.com/libuv/libuv/blob/v1.x/LICENSE>

Linux Package:

You may install libuv from the Linux package manager as well
`sudo apt install libuv1-dev`

Build libuv

Clone the libmodbus sources.
`wget https://github.com/libuv/libuv/archive/v1.41.0.tar.gz`

```
// System-wide installation
export CFLAGS+=" -Os "
export CPPFLAGS+=" -g -Wall -fPIC "
export CXXFLAGS+=" -g -Wall -fPIC "
./autogen.sh
./configure
```

```
make
sudo make install
```

For static linking
libuv.a

For dynamic linking
uv

Building Smart Data Logger Modbus (SDL-MODBUS)

cd to the source tree directory of the software.

```
$<source directory>
  mkdir build
  cd build
  cmake ..
  make
  cd ..
```

```
$<source directory>
```

```
cd run/bin
```

Execute the sdl-modbus
`./sdl-modbus`

You may use process management tools such as pm2 to run the app.
<https://www.npmjs.com/package/pm2>

Datasets

```
$<source directory>  
  run/  
    datasets
```

The sdl-modbus-d21 generates HDF5 datasets in the above directory.

Obtain a copy of HDF5 viewer

Email to support@vidcentum.com to get a copy of HDFView tool.

File: HDFView-3.1.2-Linux-x86_64.tar.gz

Install the viewer.

```
$ tar -xvf HDFView-3.1.2-Linux-x86_64.tar.gz  
$ sudo dpkg -i hdfview_3.1.2-1_amd64.deb
```

Configuration

```
$<source directory>  
  run/  
    config
```

PM-FIN-L21 (= NCAP-FIN-L21). Current Release Version: 1.0.3 (13/Oct/2021)

Network Capable Application Software (NCAP) for MODBUS - MQTT - Expert Systems Applications.

System Requirements

Development Setup

1. Ubuntu 20.04+ Linux, x86_64
2. Compile on Raspberry Pi target.
3. CodeLite IDE, Code::Blocks IDE
4. C++17, CMake (Latest versions of CLang, GCC)

Deployment (Target System)

1. Ubuntu 20.04 Linux, x86_64 desktop / workstation or cloud VM

2. Raspberry Pi with at least 2GB RAM, 64 Bit

Build process

Please see the prerequisites of SDL-MODBUS as well.

Build using clang.

```
$ cd pm-fin-l21 (ncap-fin-l21)
$ mkdir build
$ cd build
$ export CC=/usr/bin/clang
$ export CXX=/usr/bin/clang++
$ cmake ..
$ make -j4
```

PM-FIN-S21. Current Release Version: 1.0.2 (26/Oct/2021)

Web Application to work with IoT / Operational Data Platform

```
https://github.com/pyenv/pyenv
git clone https://github.com/pyenv/pyenv.git ~/.pyenv
https://github.com/pyenv/pyenv-virtualenv
```

Step 1: Create pyenv

Step 2: Install 3.8.12+ (3.8)

Step 3: Activate virtual environment of 3.8.12 (3.8)

Step 4: Install poetry

Step 5: cd to source directory/

Step 6: poetry install

Run 'poetry update' to update python dependencies.

Use the following command to run the application.

```
$ gunicorn -k geventwebsocket.gunicorn.workers.GeventWebSocketWorker -w 1 -b localhost:5000
```

Refer to <https://flask-socketio.readthedocs.io/en/latest/deployment.html>

To run development version (without OpenID Connect)

```
gunicorn -k geventwebsocket.gunicorn.workers.GeventWebSocketWorker -w 1 -b localhost:5000
```

PM-FIN-S21-SPA. Current Release Version: 1.0.2 (26/Oct/2021)

Web GUI SPA (Single Page App) to work with IoT / Operational Data Platform

```
$ cd /pm-fin-s21/web-gui/vue-spa-l21/
```


Build Setup

```
# install dependencies
$ npm install

# serve with hot reload at localhost:3000
$ npm run dev

# build for production and launch server
$ npm run build
$ npm run start

# generate static project
$ npm run generate
```

For detailed explanation on how things work, check out the documentation.

Special Directories

You can create the following extra directories, some of which have special behaviors. Only **pages** is required; you can delete them if you don't want to use their functionality.

assets

The assets directory contains your uncompiled assets such as Stylus or Sass files, images, or fonts.

More information about the usage of this directory in the documentation.

components

The components directory contains your Vue.js components. Components make up the different parts of your page and can be reused and imported into your pages, layouts and even other components.

More information about the usage of this directory in the documentation.

layouts

Layouts are a great help when you want to change the look and feel of your Nuxt app, whether you want to include a sidebar or have distinct layouts for mobile and desktop.

More information about the usage of this directory in the documentation.

pages

This directory contains your application views and routes. Nuxt will read all the ***.vue** files inside this directory and setup Vue Router automatically.

More information about the usage of this directory in the documentation.

plugins

The plugins directory contains JavaScript plugins that you want to run before instantiating the root Vue.js Application. This is the place to add Vue plugins and to inject functions or constants. Every time you need to use `Vue.use()`, you should create a file in `plugins/` and add its path to plugins in `nuxt.config.js`.

More information about the usage of this directory in the documentation.

static

This directory contains your static files. Each file inside this directory is mapped to `/`.

Example: `/static/robots.txt` is mapped as `/robots.txt`.

More information about the usage of this directory in the documentation.

store

This directory contains your Vuex store files. Creating a file in this directory automatically activates Vuex.

More information about the usage of this directory in the documentation.

General FAQ

1. What is the use of the "Smart Data Logger (Modbus)"? Modbus is a proven industry protocol to interface with sensors / transducers / PLCs etc., It is widely used in the automation systems. Industrial IoT (IIoT) brings all new applications connecting the assets (industrial, factory etc.,) to the digital platforms. SDL-(Modbus) is designed to connect the existing and/or new Modbus (TCP) based systems with the IIoT digital platforms.

2. I just want to learn Modbus protocol. Can "Smart Data Logger (Modbus)" be useful? Yes. Absolutely. Modbus standard is available from <https://www.modbus.org/>. Get familiar with the standard first. "Smart Data Logger (Modbus)" is a tool that surely helps you understand the protocol. It has inbuilt Modbus/TCP server and you really need not have a physical device to understand the protocol. Also, you can browse the data collected from the Modbus/TCP server simulations using HDF5 viewer. You can get a copy of the HDF5 viewer from <https://datacloud.vidcentum.com/s/tQyBrCZf5YxoH8j>.

3. Can I modify the software? Yes. You can. It is released under GNU GPL-3.

4. What is a support package subscription? You gain access to free training, development material, early releases, software updates. Also, you get discounts if you hire us for your projects.

5. Why should I subscribe to a support package? Let's make it win-win together. We are investing in developing useful software for industrial applications. It needs money and support. However, we do not compromise on software freedom. You get a copy of the software and you are free to modify as per your requirements. Subscribing to a support package get early access to the new features and updates. Furthermore, you can attend the training sessions for free or heavily discounted prices. Contact sales@vidcentum.com for further details.

6. Does support package subscription include development services? No. If you need more help in your project, you can hire us on a contract basis. If you have a support subscription, you may get an additional discount on contract value. Contact sales@vidcentum.com for further details.

7. Do you support academic or open source projects? Yes. Please do contact sales@vidcentum.com for further details.

8. How frequently do you release the updates (or patches)? We have plans to release monthly updates. If no update is available, it will be informed to all subscribers in advance.

9. Is there any refund of the support package? No. Sorry. We understand your concerns. Start small. Start with community support. Subscribe to a support package for a minimum period and see if it works for you. If you are convinced, go for 6 months to 12 months subscriptions.

10. Are you going to add other industrial protocols to the framework? Yes, progressively.

11. Will my support package subscription cover all protocols? No. You need additional subscriptions. Your support subscription covers updates, bug fixes, enhancements of one product line; for example, SDL-MODBUS-D21 (Modbus) support subscriptions cover only DL-MODBUS-D21 updates, bug fixes etc.,

12. Will you provide on-site support? No. Due to pandemic situations, we prefer remote support. Since it is software, it is easy to support remotely.

13. What is the mode of training? Online.

14. Can I use "Smart Data Logger (Modbus)" in my clients' projects? Yes. You may need additional support package licenses for your customers.

15. Can I conduct a training using "Smart Data Logger (Modbus)"? Yes. Please do contact sales@vidcentum.com for further details.