

UDOO

- [Configuring UDOO board](#)
- [Installing third-party component for C SDK](#)
- [Installing third-party components for C++ SDK](#)
- [Creating applications based on C++ SDK](#)
 - [Example](#)

This guide explains how to build applications for UDOO based on the Kaa C/C++ endpoint SDKs.

Configuring UDOO board

If this is the first time you use the UDOO technology, you have to start with configuring your board. For this purpose, refer to [UDOO official documentation](#).

Installing third-party component for C SDK

The further instructions are expected to be executed directly on the UDOO board and with the Linux installation and root permissions.

1. Connect to the board via an micro OTG client.
2. Install common prerequisites.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install cmake libssl-dev
```

Installing third-party components for C++ SDK

The following third-party components must be installed on the UDOO board before building the C++ SDK.

- **Mandatory:** Boost (min v1.54), Avro (v1.7.5 - v1.7.7), and Botan (only v1.11) libraries.
- **Optional:** SQLite library. It should be installed only if you are going to use a persistent log storage for the Kaa data collection feature.

NOTE: The further instructions are expected to be executed in the order given, directly on the UDOO board, and with Linux installation and root permissions.

To install these libraries, proceed as follows:

1. Connect to the board via a micro OTG client.
2. Install common prerequisites.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install cmake libssl-dev python
```

3. Install Boost (v1.54 or higher).

```
sudo apt-get install libbz2-dev libbz2-1.0 zlib1g zlib1g-dev
wget http://sourceforge.net/projects/boost/files/boost/1.54.0/boost_1_54_0.tar.gz
tar -zxvf boost_1_54_0.tar.gz
cd boost_1_54_0/
./bootstrap.sh
sudo ./b2 install
```

4. Install Avro (v1.7.5 - 1.7.7).

```
wget http://apache.ip-connect.vn.ua/avro/avro-1.7.7/cpp/avro-cpp-1.7.7.tar.gz
tar -zxvf avro-cpp-1.7.7.tar.gz
cd avro-cpp-1.7.7/
cmake -G "Unix Makefiles"
sudo make install
```

5. Install Botan (v1.11).

```
wget https://github.com/randombit/botan/archive/1.11.28.tar.gz
tar -zxf 1.11.28.tar.gz
cd botan-1.11.28/
./configure.py --cpu=arm
sudo make install
ln -s /usr/local/include/botan-1.11/botan /usr/local/include/botan
```

6. Install SQLite (latest).

```
wget https://www.sqlite.org/2015/sqlite-autoconf-3081002.tar.gz
tar -zxvf sqlite-autoconf-3081002.tar.gz
cd sqlite-autoconf-3081002/
./configure
sudo make install
```

Creating applications based on C++ SDK

To create an application based on the C++ SDK, at first you need to [build static/shared Kaa libraries](#) from the generated SDK and then link your application to those libraries. You can either do it manually or create some build script to automate the building process (see the *Example* section).

The UDOO platform allows building the source code directly on the board. The only thing you need to do prior to that is export your code onto the board. For this purpose you can use, for example, the [scp](#) utility.

Example

To quickly start with [the Kaa IoT platform](#), you can download one of the Kaa demo applications from [the Kaa Sandbox](#) and run it on the UDOO board.

For this example, you need to download the Notification demo from the Kaa Sandbox to your host machine. After that, export the downloaded archive to the board and run the demo, as follows:

1. Find the IP address of the UDOO board (you can use the WIFI predefault integrated one).

UDOO board

```
ifconfig
```

2. Copy the source application to the UDOO board from the host machine.

Host machine

```
scp /path/to/downloaded/demo/notification_demo.tar.gz root@<put ip address here>:notification_demo.tar.gz
```

3. Build and run the Notification demo from your UDOO board.

UDOO board

```
tar -zxf notification_demo.tar.gz
cd CppNotificationDemo/
./build.sh deploy
```

If the build is successful, you will see the following output on the UDOO terminal:

```
Notification demo started
--= Press Enter to exit =--
Topic list is empty
Topic list was updated
Topic: id '603', name 'Sample mandatory topic', type 'MANDATORY_SUBSCRIPTION'
Topic: id '604', name 'Sample optional topic', type 'OPTIONAL_SUBSCRIPTION'
Subscribing to optional topic '604'
Notification for topic id '603' received

Notification body: 'This is sample notification for mandatory topic.
Client automatically receive notifications for manadatory topics. More
details here: https://docs.kaaproject.org/display/KAA/Using+notifications'
Notification for topic id '604' received

Notification body: 'This is sample notification for optional topic.
Client should subscribe to optional topics in order to receive
notifications. This application subscribes to optional topics
automatically.'
Notification demo stopped
```