

# Intel Edison

- [Configuring Intel Edison board](#)
- [Installing third-party components for C++ SDK](#)
- [Creating applications based on C++ SDK](#)
- [Example](#)
- [Known Issues](#)

This guide explains how to build applications for Intel Edison based on the Kaa C++ endpoint SDK (further, the C++ SDK).

**NOTE:** The instructions below are for [the official Intel Edison firmware](#).

## Configuring Intel Edison board

If this is the first time you use the Intel Edison technology, you have to start with configuring your board. For this purpose, refer to [the official Edison getting started guide](#).

After finishing with the guide, you will be able to access your board through both a serial terminal and an ssh client.

## Installing third-party components for C++ SDK

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

- **Mandatory:** [Boost](#) (1.54 min.v.), [Avro](#) (1.7.5 - 1.7.7 v.), and [Botan](#) (only v.1.10) 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 Edison board**, and **with the root permissions**.

To install these libraries, proceed as follows:

1. Connect to the board either through the serial terminal or an ssh client. Refer to [the official Edison getting started guide](#) for any help required.
2. Install the "install" utility.

```
opkg install
http://repo.opkg.net/edison/repo/core2-32/coreutils_8.22-r0_core2-32.ipk
```

3. Install Boost (1.54 or higher).

```
wget
http://sourceforge.net/projects/boost/files/boost/1.58.0/boost_1_58_0.tar.gz
tar -zxf boost_1_58_0.tar.gz
cd boost_1_58_0/
./bootstrap.sh
./b2 install
```

4. Install Avro (1.7.5-1.7.7).

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

5. Install Botan (1.11).

```
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
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 -zxf sqlite-autoconf-3081002.tar.gz
cd sqlite-autoconf-3081002/
./configure
make install
```

## Creating applications based on C++ SDK

To create an application based on the C++ SDK, you need to build static/shared Kaa libraries from the generated SDK at first 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 Intel Edison 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 Kaa demo applications from the Kaa Sandbox and run it on the Edison 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 assigned to the Edison WiFi interface:

```
Edison
ifconfig wlan0 | grep 'inet addr:' | cut -d: -f2 | awk '{ print $1}'
```

2. Copy the demo application source package to Edison from your 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 Edison board:

## Edison

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

If the build is successful, you will see the following output on the Edison 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
```

## Known Issues

On some Edison boards, the build crashes due to the lack of system resources. If you experience this issue, reduce the number of workers used by the *make* utility by replacing 'make -j4' with 'make' in the *build.sh* script and re-running it:

## Edison

```
sed -i 's/make -j4/make/g' build.sh
./build.sh deploy
```