English

# Asema IoT Central

## Hardware integration manual

ASEMA

# Table of Contents

# 1. Introduction

Remotely monitoring hardware sensors and controlling actuators is the heart of IoT. Unsurprisingly, integrating and running various types of hardware is also at the core of Asema IoT. This manual illustrates various ways in which this can be achieved.

Hardware integration is somewhat of an art because there are a multitude of ways hardware units can talk with software and ways connections can be made. Much depends on the capabilities of the hardware in use. If there is plenty of processing power, money to pay for the bandwidth, deployment does not need to be made in masses, and real-time synchronization is not an issue, traditional networking solutions from cable Ethernet to wireless 4G and WiFi work very neatly. But when millions of units need to be connected cheaply over long distance, when bandwidth is limited, and when the hardware devices run on very low level protocols, something else is needed. Understanding and applying the seemingly endless permutations of wired and wireless technologies, protocols, and standards requires quite a bit of knowledge and also a pile of creativity.

There is no definite way to go about in integrating a certain piece of hardware. This is why this document is not a step-by-step guide to integration but rather an illustration of the possibilities and components available. It is up to the reader to pick those pieces that fit their project the best.

As the means and ways of integrating hardware differ, so do the tools and steps needed to achieve integration. In Asema IoT, those steps are performed by different parts of the system which have their own user manuals. In this document you will find a short description of the principles of operation of each type of integration. The actual instructions on how to proceed are in the corresponding manuals. This document gives you the pointers to those manuals which you should read to get the implementation details.

# 2. Connecting standard, supported hardware

## 2.1. Integration without parsers

There are some devices where everything from the connection protocol all the way up to the payload in messages is so well defined that no further configuration is needed. These devices are usually low power wireless devices with a limited bandwidth radio. Currently, such devices supported by Asema IoT include NFC tags and Eddystone and iBeacon tags.

When you use such devices, you just need to scan them in. The device is recognized and linked to its ID, and an object representing this hardware is created. After activation, when the same device is detected, it triggers a property change in the corresponding object. This object can be used in IoT applications.

For instructions on how to scan such devices, please refer to the hardware section of the **Asema IoT Central User Manual**.

## 2.2. Integration using data parsers

Some devices have a standardized communication protocol but an unknown payload. Such devices include serial port devices, Ethernet and WiFi-connected devices, as well as Bluetooth Classic and Bluetooth Smart devices. These devices can be connected easily, either by entering the port information (such as the connected serial port) or by scanning them (in the case of Bluetooth devices).

However, this network level connectivity is not enough for applications as the payloads of the messages can be anything. To make the payload formats understandable for applications you need a parser. Asema IoT has internally built parser classes which you can configure with a schema. The schema tells the parser how to actually parse the payload and which parts of the incoming data match with which object properties of the system.

Parsers and serializers are the topic of another manual, the **Asema IoT Central Object data parsing and serializing manual**. So take a look at that manual to learn more about this integration option.

# 3. Gateways and converters

A gateway is a device that sits between your system and some sensor or actuator. Gateways are commonly used to centrally connect short range radio devices into one hub that then offers a long range, broadband Internet connection. This lowers the cost of communication as most devices can run on some very inexpensive network technology. But while the gateways do the connectivity, they typically also handle data conversion.

In the gateway option you do not connect the hardware directly to a large host but rather have a smaller but still network capable embedded computer in between. You can build such a gateway either from other components or by using the Asema IoT Edge software package. If you use Asema IoT Edge, this software can automatically sync all its objects (and therefore the measurements and controls from sensors and actuators) with an Asema IoT Central without further integration work. The communication protocol also handles transforming the messages into an understandable format. A schema or a driver is used on the gateway to translate proprietary messages from the sensors and actuators. The traffic from Asema IoT Edge requires no further translation.

If you have a proprietary gateway, then the easiest way to integrate it with Asema IoT is to use the JSON API. This lets you manipulate the objects that represent the sensors over HTTP calls and also receive notifications and commands that make it possible to do two-way communication and control solutions.

Instructions on how to use the JSON API can be found in the **Asema IoT Central JSON API 1.0 manual**.

# 4. Custom drivers

If you don't have standard tags, parsers are insufficient, and gateways are too slow or unavailable (or you need a driver to run on the gateway), then custom hardware drivers are the way forward. With custom drivers you essentially program a driver with C and C++ and get to operate your hardware in any way that is needed and enabled by purely custom software code. Because drivers are fully custom code, you pretty much have the freedom to interpret data the way you want.

Do note that because Asema IoT runs in user space, the drivers are also user space drivers, not kernel drivers. If you actually need a kernel driver, then you need to program one and load it into the kernel. Expose some device interface from this driver to user space. The "driver" that is loaded into Asema IoT is simply a connector that reads this interface you've exposed and transforms data into objects.

Programming drivers into Asema IoT takes place as software plugins. To learn more about plugins and to see an example hardware driver, read the manual **Asema IoT Central Plugin Interface 1.0**.

# 5. Hardware deployment process

Once you have made hardware integration ready and some devices are running smoothly at the office, lab or pilot, the next challenge typically is repeating the same in large scale on the field. This deployment process is often the trickiest part; error-prone and notoriously expensive if things don't go smoothly. Asema IoT offers various methods to make the process smoother.

First, debugging and testing tools. The driver interface used to connect the hardware contains several debugging methods such as `selfDiagnoseDevice()`, `locateDevice()`, `pingDevice()` and `restartDevice()` which are meant as simple tests of connectivity, starting everything again, and for receiving heath data from the device. If your device supports two-way communication, it is highly recommended to implement such methods in the driver.

The driver interface also supports methods for requesting information on device configuration, clock, software version, etc. An incorrect software version or wrong clock time are very typical errors that result in erroneous operation or data. So, again, it is recommended to implement these in sufficient detail. These methods include `requestDeviceConnectionState()`, `requestDeviceStatus()`, `requestDeviceConnectionQuality()`, `requestDeviceFirmwareVersion()`, and `requestDeviceClock()`.

Second, the deployment process. In an ideal situation (in terms of ensuring all is OK), devices are connected, configured and tested already before they leave a factory or a warehouse. All that is left is turning the power on during installation. However, in many cases this is not possible. Devices are delivered unconfigured to distributors, installers, and similar parties where they may sit on shelf for an undetermined period of time. Because of this, usually pre-configuration cannot be made prior to installation.

This is where the deployment tools of Asema IoT Central become handy. The deployment process uses the Asema IoT Dashboard mobile application and offers field engineers a mobile interface for configuration. For example, you can print QR codes that identify each device and attach them to the devices on delivery. The code automatically associates the device with the correct configuration once scanned. The engineers simply need to scan the code with their hand-held devices. Alternatively, you can use the deployment tool to offer the field engineers forms to fill in and to manually type in the correct configuration. Once entered, this initiates an automatic deployment process that configures the object to the system.

For more details about the deployment tools, read the corresponding chapters in the **Asema IoT Central User Manual**.