IoT Embedded Systems

The rapid progress of technology has blurred the lines between sectors. This is particularly evident in IoT (Internet of Things) embedded systems where the terms IoT and embedded system have become mishmashed to mean the same thing. But an embedded system and IoT are two different concepts, despite some similarities.

Embedded systems have been around since 1965 when a company called Autonetics developed a tiny computer that was used as part of a missile guidance system.

Embedded systems are standalone devices that have usually been designed to do one specific thing. An IoT embedded system is an embedded system that also has connectivity to the internet and can therefore communicate with other IoT embedded systems.

We are surrounded by IoT embedded systems such as:

- Set-top boxes
- Point-of-Sale (POS) terminals
- A wide variety of medical devices
- Any "smart" devices such as smart refrigerators, smart bicycles, fitness trackers, etc.
- Parking meters

Here, we will look in detail at embedded systems for IoT devices, what an embedded system for IoT is, what an embedded system for IoT has to offer, and how you can put an embedded system IoT into practice.

The difference between IoT and embedded system devices

Let's look at the difference between these two terms:

What is IoT?

IoT stands for Internet of Things and refers to the interconnected nature of devices that we use in our daily lives. These days "smart" devices can connect to the internet and to other devices to facilitate everyday actions. For example, a smart fridge can detect the items in it as well as expiry dates and provide information to the owner about it. This is a typical use-case for IoT in everyday use.

What is an embedded system?

An embedded system is a tiny computer that was built for a custom purpose. Such a computer usually has a microprocessor or microcontroller, which is an integrated circuit that typically contains processor, memory, and I/O (input/output) peripherals on one chip.

The defining factor of an embedded system is that it can carry out some form of digital processing. This is what distinguishes it from plain hardware devices that contain only some circuitry and possibly a battery.

An embedded system often has sensors that let it monitor environmental factors.

What is the difference between the two?

An embedded system does not necessarily have internet connectivity, but an IoT embedded system always does. The difference between an embedded system and IoT is that the embedded system can be a subset of IoT. Embedded system before IoT were not able to connect to the internet, therefore IoT embedded systems generally encompass a broader range of devices and use cases.

What is embedded system in IoT?

An IoT embedded system is an embedded system that has internet connectivity. Another word for what is IoT embedded system is a "smart" device. A touch screen and a keyboard are not necessary to define a device as an IoT embedded system, although these peripherals can also be attached.

Types of IoT embedded systems

An embedded system for IoT will have some type of software installed on it to allow it to perform its function. This could be simply firmware or an embedded operating system. It is the software that empowers the IoT embedded system to communicate with other IoT embedded system devices.

In the context of embedded systems, we also talk of embedded devices. These terms are sometimes used interchangeably. However, embedded devices are usually part of a larger embedded system for IoT use cases. To differentiate: An embedded device is a highly specialized device intended for one or very few specific purposes. It is also called a Single Purpose Device or Dedicated Device. These devices are IoT embedded systems that have been created specifically to operate one particular program or perform a task that is specific to a certain type of business.

Examples of single-purpose IoT embedded systems include:

Kiosk devices
POS solutions
Self-checkout systems

Embedded systems for IoT use cases are widely used in retail, consumer products, automotive, healthcare, and many other industries.

The difference between IoT and embedded systems

The difference between IoT and embedded system in one word is connectivity. Even a pacemaker is an embedded system. By letting that device communicate to the outside world, such as transmitting heart-rate data to a central database, it turns out into an IoT embedded system.

The other difference between an embedded system and IoT is that IoT refers more to a class of devices that represent the newly connected world. But an embedded system refers very specifically to the hardware used in these devices.

The primary difference between an embedded operating system and IoT is that the IoT operating system needs to support connectivity. The capability of IoT networking of embedded system devices is what differentiates plain embedded systems from IoT embedded systems.

Every IoT use case will have an embedded system as part of it, which is why the term for it is IoT embedded system. First, there is an embedded system, and then come the features that turn it into an IoT device.

Embedded systems can be highly sophisticated in themselves, such as robotic systems used in manufacturing or data warehouses. But until that device can communicate and has connectivity with other IoT embedded systems, its function as a pure embedded system remains limited to it’s single intended purpose.

The distinction between embedded system and IoT embedded system is only one of its ability to communicate.

The significance of embedded systems in IoT

When considering an embedded system vs IoT, it might be helpful to consider some real-world applications. Let's look at several examples that further clarify the distinction between an embedded system and IoT.

The first thing to understand is the difference between an embedded operating system and IoT: Both IoT embedded system devices and non-IoT embedded system devices will have some type of software that powers them, whether this is simply firmware and bootloaders, or drivers and a full-on embedded operating system.

That something is an IoT embedded system doesn't automatically mean that it needs a full-blown operating system to run it, although it is more common. There are several operating systems that are suitable for use in embedded systems, for example Linux or Android.

Installing certain modified versions of Android on an IoT embedded system means that the device will be able to ...

- receive Over-The-Air Software updates (OTA updates), and
- to enable Mobile Device Management benefits, which is the ability to manage an IoT embedded system remotely.

Android as an OS is, therefore, a good choice. Furthermore, the introduction of Raspberry Pi opened the door wide for mass-production of IoT embedded systems, especially after the release of its Generation 3 models because that generation introduced built-in connectivity for Bluetooth and WiFi. By installing Android on Raspberry Pi 3 B+ it also becomes possible for this IoT embedded system to avail itself of Raspberry Pi Fleet Management features (like MDM) mentioned above.

Examples of non-IoT embedded system devices

It has become increasingly simpler to create IoT embedded systems.

Although the following examples might lately also have some IoT embedded system versions (such as a smart microwave), the following is a list of devices that have typically not been used in an IoT embedded systems context:

Microwave ovens
Washing machines
Manufacturing robots
Industrial robots and factory machines

However, modern warehouse robots are now typically created as an IoT embedded system that can communicate to a central dashboard via the internet and be managed using MDM services.

Even if there is no need for that particular device to be connected to the internet, opting for an IoT embedded system makes it possible for the device to receive IoT OTA updates so long as the proper operating system has been installed and the required infrastructure is given.

The role of embedded system in IoT

The embedded system in IoT devices is the hardware of the device. All IoT devices are built on top of embedded systems. Building IoT devices on the same hardware used for personal computers would be prohibitively expensive and is also unnecessary. The existence of embedded systems such as the Single Board Computers (SBC) Raspberry Pi, Rock Pi 4B, and others, means that IoT computing devices can be created on a small scale (one to 100,000 devices) while still keeping costs low.

Here are several use cases of very popular types of IoT embedded system devices:

POS terminals

Point-of-Sale terminals and self-service checkout terminals are a common use of an IoT embedded system. These touch-enabled IoT embedded systems connect to payment gateways via the internet to process payments. These devices have a scanner attached to them, and can also receive input from a pin pad and card reader.

Vending machines

An IoT embedded system in the form of a vending machine can send messages alerting employees on the ground when items need to be refilled. The devices receive input from a coin slot that should be able to read the type of coin inserted. Further inputs include a touch screen and credit card reader.

Ticket scanners

Event venues can have a ticket scanner IoT embedded system in place that reads ticket information from visitors, compares it with a local or online database, and then engages a turnstile to let the visitor in.

These IoT embedded systems can be further programmed to send attendance data to an analytics platform, which can then process the data and provide detailed insight to event organizers.

Putting into practice the embedded system for IoT

It is possible for anyone to create their own embedded system for IoT using just a few simple tools.

To create an IoT embedded system, you will need the correct hardware and software.

Hardware for an IoT embedded system

First and foremost, you will need the hardware for creating an embedded system for IoT. By then installing the appropriate software on this hardware and add needed peripherals, you can turn it into an IoT embedded system.

There are numerous choices of hardware for an embedded system for IoT, including:

✔ Various **Raspberry Pi** models from Generation 3 and above.

✔ **Rock Pi 4B**, a single-board computer with similarities to the Raspberry Pi 4 but with a considerably enhanced CPU and GPU performance.

✔ **Intel NUC** — this is a powerful mini PC with a highly customizable CPU, memory and storage setup. It is an excellent choice for a wide variety of use cases, especially those requiring a lot of power.

There are several other IoT embedded system hardware configurations you could use!

Software for an IoT embedded system

Because an IoT embedded system connects to the internet, its software must minimally be able to deal with these connections.

Usually, the choice of software for an IoT embedded system comes down to an operating system such as Android or Linux.

Installing such software on an IoT embedded system can be challenging for several reasons:

- The extremely varied hardware combinations of these systems mean that OS support doesn't come "out of the box" for them. You have to find versions of Android or Linux that have been specifically ported to work on that particular hardware configuration, or develop it yourself.
- Ideally, the ported version of the software also includes support for remote IoT embedded system updates and management. To achieve this, the ported version of the OS must also have a robust infrastructure behind it that deals with creating new builds and pushing those builds out stably to IoT embedded devices.

Advantages of using Android for an embedded system for IoT

Android has several clear advantages over other operating systems when used on an IoT embedded system.

Android has a high-quality user interface and its touch-screen support is familiar to users. The enormous popularity of Android means that hundreds of thousands of apps exist which can be added to enhance that particular IoT embedded system.

But probably the most compelling reason to choose a ported version of Android for your IoT embedded system is that Android can be modified to support remote management of devices as well as IoT OTA updates.

This support requires special adjustments in the Android OS itself but it also requires an enormous infrastructure in the backend that actively works on new versions, compiles them, and then pushes them out to devices. Emteria has built this infrastructure as part of its emteria.OS, which is a modified version of Android designed specifically to work on IoT embedded systems.

How to install Android on an embedded IoT system

Android will not run out of the box on any hardware configuration. Based on the Android Open Source Project (AOSP) a new version needs to be developed to work on that configuration.

Emteria.OS is the ideal choice when looking for an embedded Android version because it already supports popular IoT embedded system solutions and has the infrastructure behind it for regular and stable software updates.

To **install emteria.OS on your IoT embedded system** is as easy as doing the following:

1. Select an appropriate embedded system hardware configuration for your IoT embedded system, such as one of the Raspberry Pi devices from Generation 3 and up
2. Register on the emteria website
3. Download either the Linux or Windows version of the emteria.OS installer
4. Insert an SD card into your computer
5. Double-click the installer to create the Android IoT embedded system image
6. When prompted for where to install this image, select the SD card drive
7. After the installation is complete, remove the SD card and install it into your chosen IoT embedded system (such as the Raspberry Pi)
8. Once you've booted your IoT embedded system up, the Emteria OS will run through the initial setup and you will be prompted to configure things such as WiFi, language, and so forth.
9. To begin using the IoT-embedded system, you will need to activate the OS. There is a free version of it.
10. Because Google doesn't support IoT embedded systems the Google Play Store is not available, you can install a separate app repository to download apps. We recommend using F-Droid which is all about open-source apps and is extremely secure. You can also upload and distribute your own apps via a private app repository within F-Droid.

Emteria's Device Hub for an IoT embedded system

One of the key benefits of using emteria as the chosen OS for your IoT embedded system is that emteria also provides a mobile device management cloud dashboard called Device Hub. Through this dashboard, it is possible to provision and manage your IoT embedded system completely remotely.

By using emteria's Device Hub to manage your IoT embedded system, you can tweak virtually every setting on your IoT embedded system, including screen orientation, orientation lock, branding, managing apps, changing system app preferences, toggling WiFi, toggling Bluetooth, and so on.

Using emteria's Device Hub brings all the benefits of MDM into the hands of fleet managers.

Easy to configure

IoT embedded systems are everywhere. IoT embedded systems touch so many facets of life today that imagining a life without them is virtually impossible.