

Chapter 12: Towards the Internet of Things

tw rev. 16.10.2024

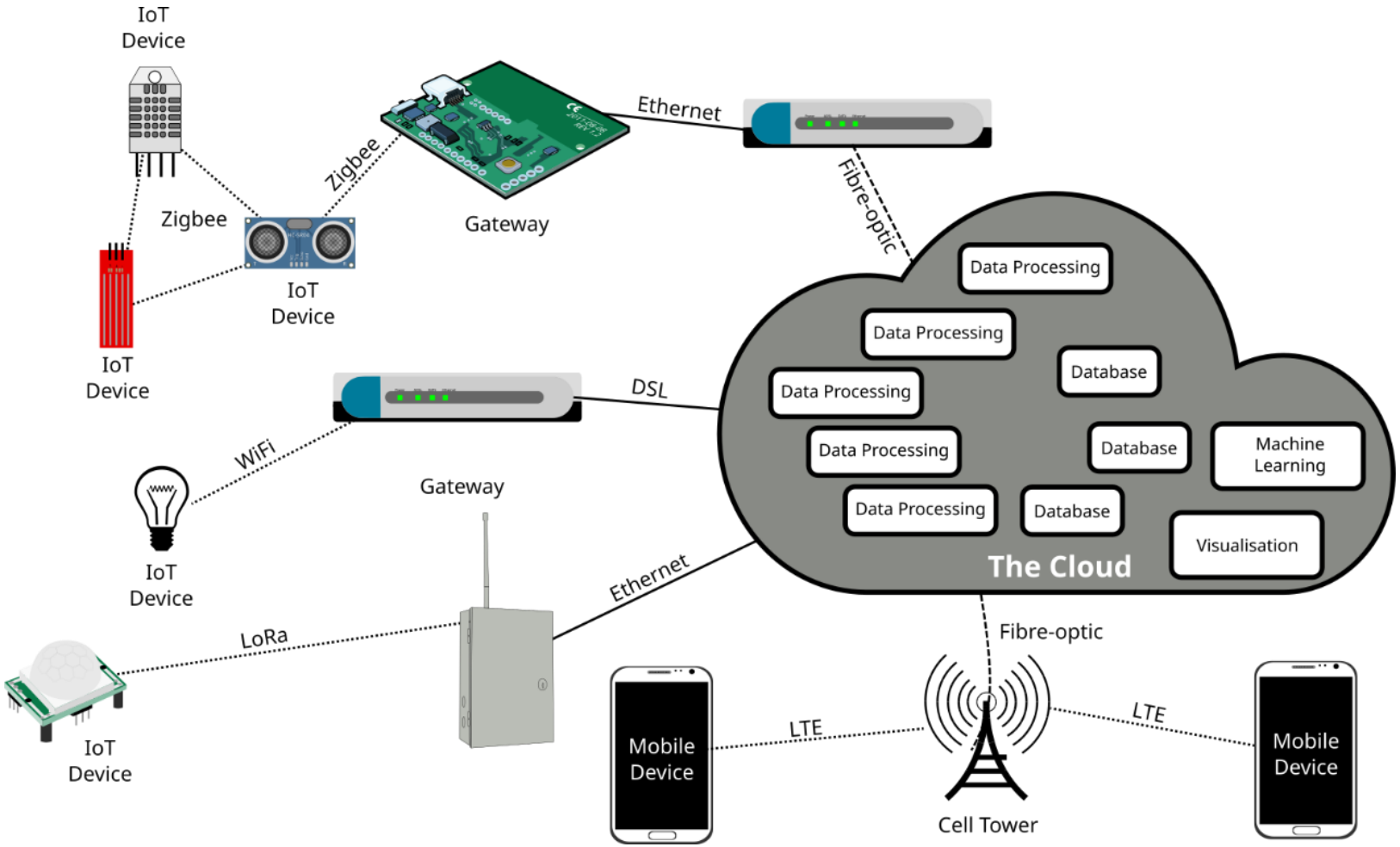
If you use or reference these slides or the associated textbook, please cite the original authors' work as follows:

Wilmshurst, T., Spink, T. & Toulson, R. (2024). Fast and Effective Embedded Systems Design – From Bits and Bytes to IoT, with the ARM Mbed (3rd edition), Newnes, Oxford, ISBN: 978-0323951975.

www.embedded-knowhow.co.uk

What is the Internet of Things?

The Internet of Things (IoT) is a growing network of physical objects that are connected to the internet and collect and exchange data. At its core, the IoT is made up of four main components: sensors and actuators, connectivity, processing, and applications.



Applications of the IoT

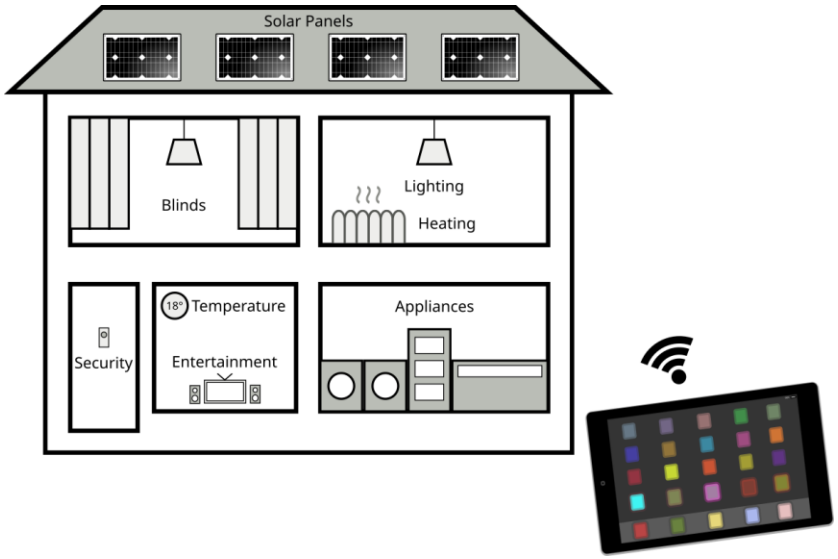
Smart homes: Automation and control in a home, e.g. for lighting, temperature, and security. A home can have a multitude of different sensors and actuators that work together and provide various benefits to the users.

Smart cities: Improves the efficiency and sustainability of urban areas. Connected streetlights can dim or brighten based on the presence of pedestrians or vehicles, and smart waste management systems can optimise rubbish-bin collection routes.

Industrial automation: Improves the efficiency and productivity of industrial processes. Connected machines can alert maintenance workers when they need servicing, and smart inventory systems can automatically reorder supplies when they run low.

Transportation: Improves the efficiency and safety of transportation systems. Connected vehicles can communicate with each other and with infrastructure to avoid accidents and reduce congestion. Public transport systems can optimise routes and schedules.

Medicine: Used for remote monitoring of patients' vital signs, remote medication management, in surgery to help guide surgeons during complex procedures, and to support clinical decisions in conditions that are difficult to diagnose.



A little IoT History

In the 1980s researchers and engineers began exploring the idea of connecting everyday objects to the internet. A group of students at Carnegie Mellon University connected a vending machine to an early version of the internet – ARPANET. The vending machine could report on its stock levels and temperature.

In 1999, Kevin Ashton, a British technology pioneer, coined the term "Internet of Things" in a presentation he gave to Procter & Gamble.

In the 2010s, the widespread adoption of smart phones and the development of cloud computing and big data analytics made it easier for businesses and individuals to connect devices to the internet and extract value from their data.

In the 2020s, the IoT has become an increasingly integral part of our lives, with billions of connected devices in use around the world.

Core Components in the IoT: Sensors and Actuators

Sensors allow IoT systems to gather data about the environment, enabling them to make decisions and take actions based on this data. In the context of the IoT, actuators are often used to control devices or systems based on the data collected by sensors.

Sensors:

- **Temperature sensors:** measure temperature in a range of environments, often used in smart thermostats, industrial equipment monitoring, weather stations.
- **Humidity sensors:** measure the amount of water vapour in the air, often used in home or office, weather stations, greenhouses.
- **Pressure sensors:** measure the pressure of gases or liquids, often used in industrial processes, weather stations, and other applications where pressure is important.
- **Motion sensors:** detect movement or changes in position, often used in security systems, industrial control, or smart home automation systems.
- **Light sensors:** These are used to measure the intensity of light, often in the form of a photometer or photodiode.

Actuators

- **Motors:** used to drive mechanical equipment, often in the form of a servomotor or stepper motor.
- **Solenoids:** used to control the flow of a fluid, often in the form of a valve or pump.
- **Relays:** These are used to control electrical circuits, in the form of a switch or contactor.
- **LEDs:** used as actuators to provide visual feedback or status indication.

Core Components in the IoT: Connectivity

Example technologies include:

- **Wi-Fi:** Wi-Fi is the ubiquitous wireless networking technology that allows devices to connect to the internet over short distances. It is commonly used in the IoT because it is widely available, well supported, and easy to use.
- **Bluetooth LE:** allows devices to communicate over short distances, used to connect devices that are close to each other, such as smart home devices or wearable devices.
- **Near Field Communication (NFC):** a short-range wireless technology used for device-to-device communication in mobile payments and access control. It is well-suited for applications that require high security and low data rates.
- **Cellular networks:** used to connect devices to the internet over long distances, using mobile phone towers. Used in the IoT to connect devices that are not in range of a Wi-Fi or Bluetooth network, such as within smart cities or connected vehicles.
- **LoRaWAN:** allows devices to communicate over long distances, typically several kilometres, using low-power radio waves and with low data rates, with low power requirements. Used in smart cities and connected agriculture.
- **Zigbee:** a wireless networking protocol designed for short-range low power applications. It allows devices to communicate over short distances, typically within 100 meters, and is often used in smart home and industrial automation applications.
- **Z-Wave:** a wireless networking protocol designed specifically for the IoT. Similar to Zigbee in that it allows devices to communicate over short distances and is often used in smart home applications.

Core Components in the IoT: Processing – Edge Computing

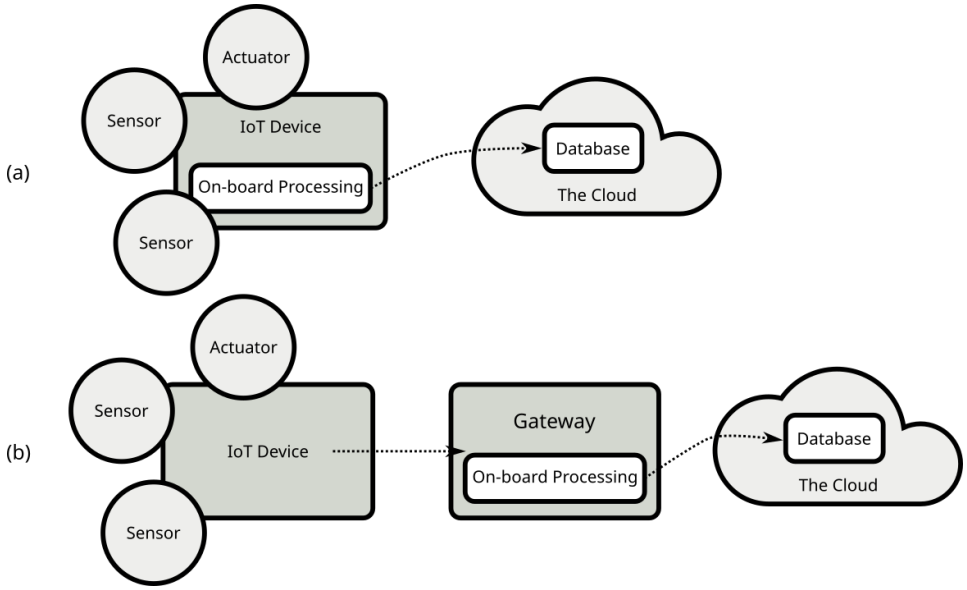
There are two approaches to IoT processing: edge and cloud computing.

Edge computing involves processing data at or near the source, rather than sending it to a centralised location. This can be useful in the IoT because it allows devices to make decisions and take actions based on local conditions, without the need for a constant connection to the internet.

Edge computing is used where low latency or high data volumes make it impractical to send data to the cloud for processing, e.g. a self-driving car. It can also be useful in situations where a constant connection to the internet is not available, such as in remote or rural areas.

The use of edge computing retains a connection to a remote server, but it doesn't depend on this for its real-time operation.

Figure (a) shows data processing happening on the IoT device itself, with a direct connection to a simple database in the Cloud. Figure (b) shows data processing happening on a gateway. In this example, the IoT device might communicate with the gateway via Bluetooth LE, and then the gateway communications with the Cloud over Wi-Fi.



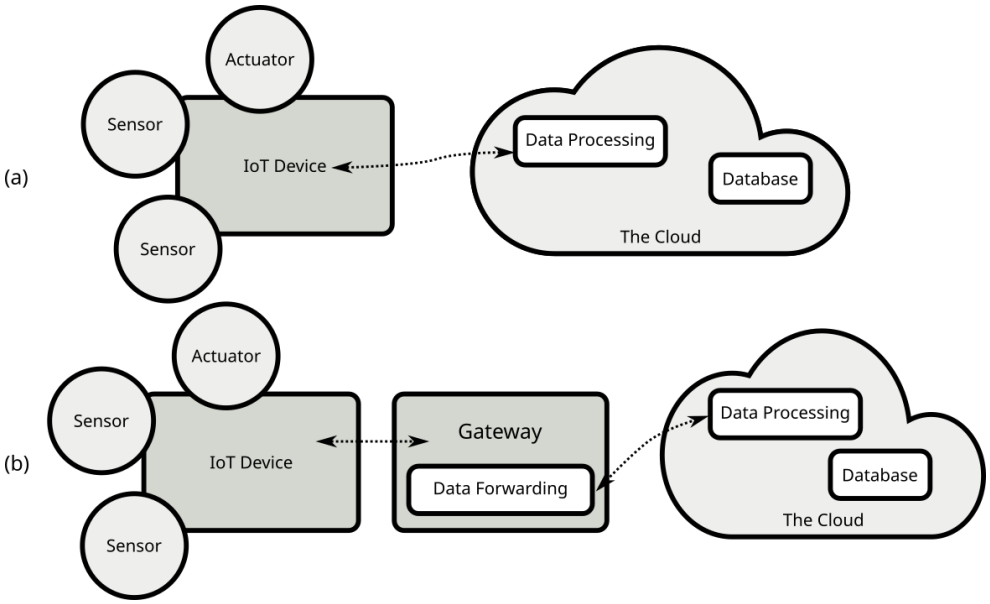
Core Components in the IoT: Processing – Cloud Computing

Cloud computing involves storing and processing data on remote servers, accessed over the internet. The IoT device uses the processing power and storage capacity of these servers, without the need for expensive hardware on the device itself.

Cloud computing is used in applications where the data volumes or processing requirements are too high to be handled by the device itself, e.g. in situations where it is not practical or cost-effective to install and maintain local processing infrastructure. A further benefit is that software running in the cloud can be updated quickly and easily, so that all users benefits immediately from the update, without having to send changes to every individual IoT device.

In Figure (a), the IoT device has a direct connection to the Cloud (e.g., over Wi-Fi), and sends data from its sensors directly to cloud applications that perform the processing. The Cloud might send instructions back to the IoT device, to cause its outputs to change.

In Figure (b), a gateway is involved, but simply acts as a data forwarder. The IoT device generates data, and forwards it to the Cloud through the gateway. Similarly, the gateway will pass on any messages intended for the device, from the Cloud.



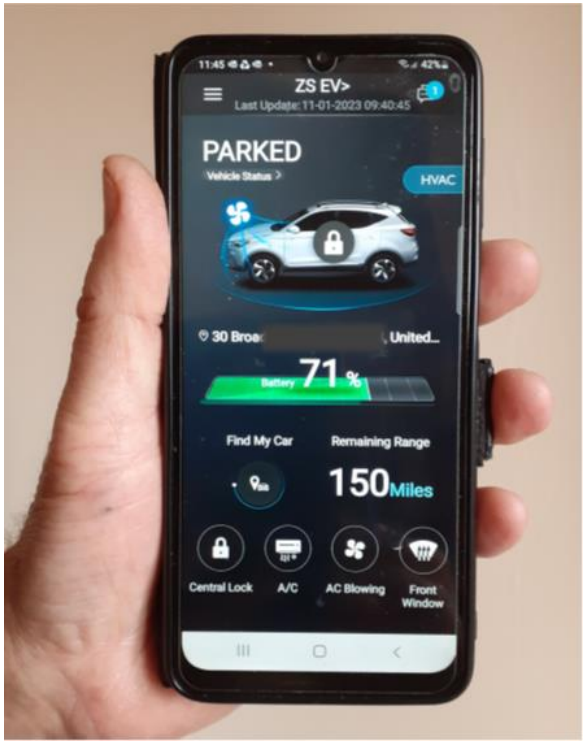
Core Components in the IoT: Applications

Mobile and web apps are used to control and configure IoT systems. These apps allow users to remotely access and manage their IoT devices, often through a smart phone or tablet.

Mobile apps are particularly useful for controlling IoT systems because they provide a convenient and user-friendly interface that can be accessed from anywhere. They provide a range of other features and functions, such as monitoring and tracking, alerts and notifications, and automation. A mobile app can also act as the gateway for an IoT device, as well as the user interface.

In addition to mobile apps, web applications are commonly used to control and configure IoT systems. Like mobile apps, these web-based applications allow users to remotely access and manage their IoT devices through a web browser on a computer or other device.

Web applications are particularly useful for controlling IoT systems because they can provide a more comprehensive and feature-rich interface than mobile apps. They can also be accessed from any device with a web browser, which can be convenient for users who need to manage their IoT devices from multiple locations.



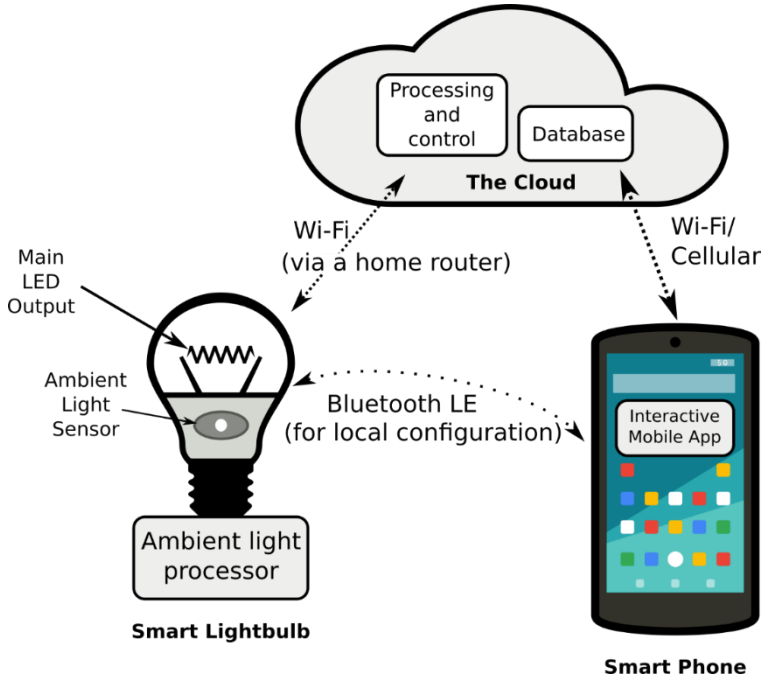
Case Study 1: Smart Lightbulb

Smart bulbs are equipped with sensors and connectivity that allow them to be controlled and configured remotely, through a smart phone app or voice assistant. comprising:

- **Sensors:** used to gather data about the environment, such as a light sensor to measure the ambient light level or a temperature sensor to measure the ambient temperature.
- **Processing:** a processor responsible for collecting sensor data and adjusting the light output of the bulb, forwarding data to the Cloud, and receiving any control instructions.
- **Connectivity:** a connectivity layer responsible for transmitting the data collected by the sensors and receiving commands from the user or other devices.
- **Actuation:** the actual light bulb, almost always high-powered LEDs, often capable of changing color, so that the user can set a “mood” to suit the environment.

Once configured, a smart phone can access the system either internally, when the user is on the home Wi-Fi, or remotely over a cellular connection.

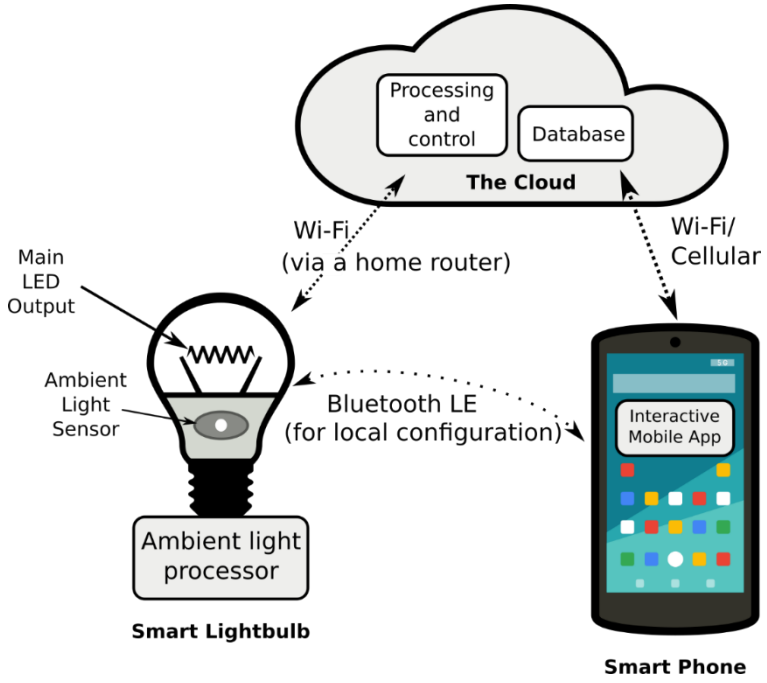
The figure shows an example of a Smart Lightbulb system, in which the bulb has a direct connection to the cloud, using a home Wi-Fi connection. Little additional hardware is required by the end-user.



Case Study 1: Smart Lightbulb

One example of a smart lightbulb is the Philips Hue. This can be controlled through a smart phone app or a voice assistant, such as Amazon Alexa or Google Assistant. The app allows users to adjust color and brightness, set schedules for when the bulb should turn on or off, and create "scenes" that change the lighting based on time of day or user's mood.

Smart lightbulbs can also help save energy. For example, the Philips Hue has a "sleep" function that gradually dims the light over a set period of time, helping users fall asleep more easily. It also has a "wake-up" function that gradually increases the light in the morning, helping users wake up more naturally.



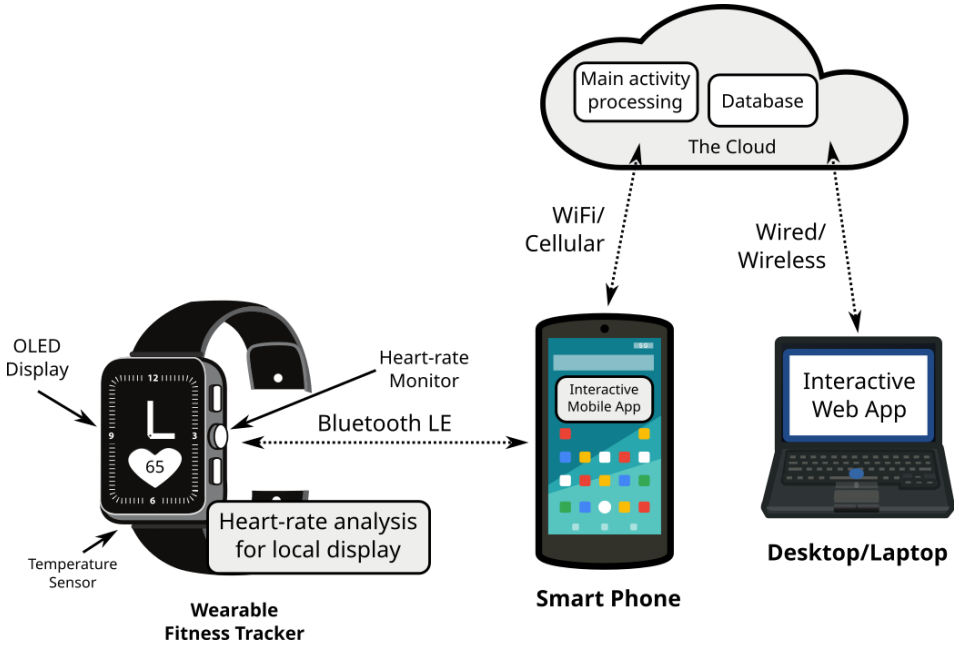
Case Study 2: Wearable Fitness Tracker

A wearable fitness tracker consists of several key components:

- **Sensors:** a variety of sensors used to collect data about the user's activity, such as an accelerometer to track movement, a heart rate sensor, and a GPS tracker.
- **Processing:** a processor responsible for analysing the sensor data and providing user feedback. It may also store data, manage device power, and link with other devices.
- **Connectivity:** responsible for transmitting sensor data to a central server or other device.
- **Actuation:** used to perform actions in response to sensor data, e.g. feedback to the user.

The figure shows a tracker using a hybrid cloud/edge style architecture. It communicates with a mobile app, over a Bluetooth LE connection, and transmits the collected sensor data. It performs some local processing, so it can show the user's measured heart rate.

The mobile app forwards this data to the Cloud, which analyses and processes it, returning a summary back to the mobile app to show to the user. There is also an interactive web application that the user can access from a desktop or laptop to visualise their data.

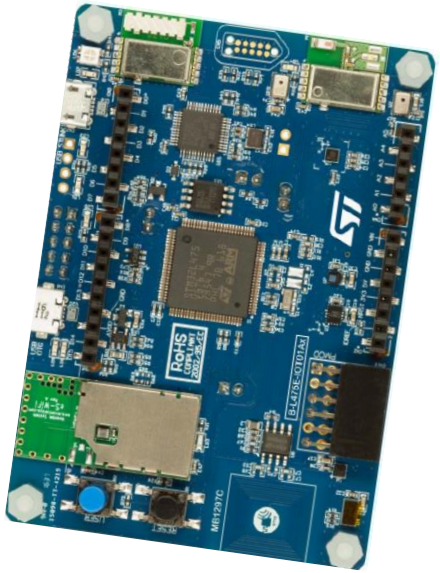


Building a Full-Stack IoT System: an Activity Tracker

Note that full details on how to implement this project, including program examples and configuration guidance, appear in the book but are not replicated in this presentation.

We develop here a full-stack IoT system. Full-stack means everything from the embedded device, the software running in the cloud, through to a mobile app to visualise the data. The resources needed are:

- The ST IoT discovery board and a suitable Mbed development environment
- Access to a Wi-Fi network with an internet connection – note that University networks may not be suitable, due to the type of security they employ.
- A Google Cloud account.
- An Android mobile device in developer mode.
- The Android Studio development environment.



We use the Android ecosystem as it has a lower barrier to entry than other development platforms. There are also many tutorials available online.

We depart from C/C++ and use instead Java and JavaScript, with all code examples provided in full.

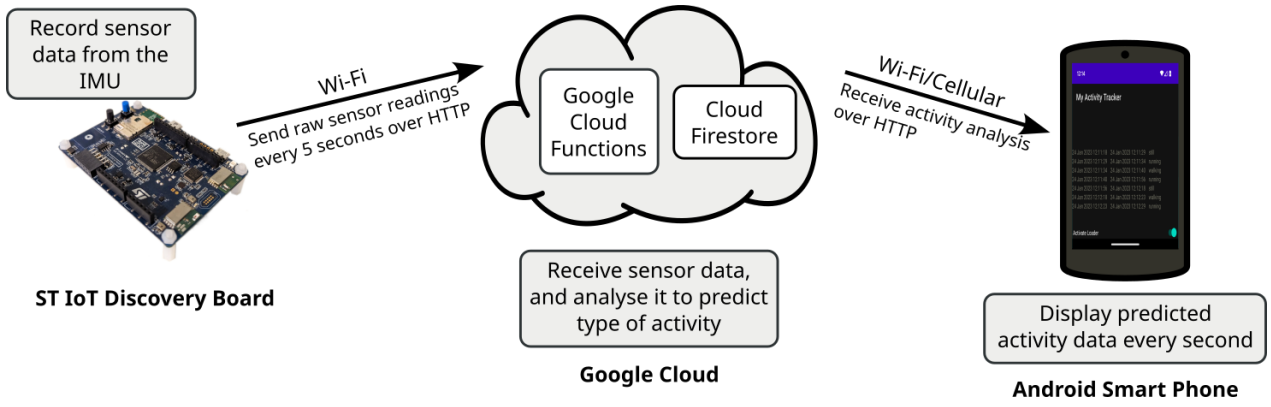
Activity Tracker Overview

The tracker will periodically send sensor data from the onboard Inertial Measurement Unit (IMU) sensor to the cloud. This will process it and make the results available to an Android App in close to real-time. The project is in three phases:

1. The Cloud
2. The Embedded Device
3. The Mobile App

The figure gives an overview. The ST IoT Board is programmed to connect to Wi-Fi, and record sensor readings from the on-board IMU every second. Every five seconds, the previous five sensor readings will be sent to Google Cloud over an HTTP connection, via a Google Cloud Function. This processes the sensor readings and deduces the type of movement (sitting, walking, or running) that occurred. The result is stored in a Cloud Firestore database.

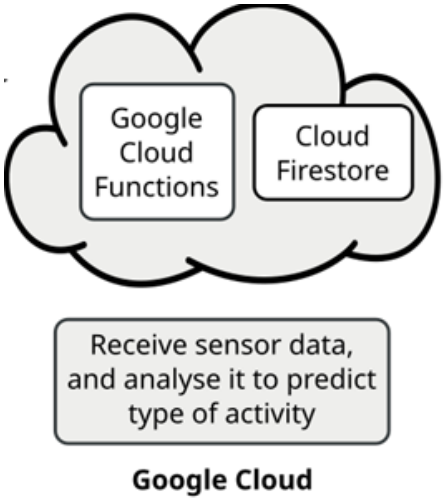
Every second the Android App calls a different Google Cloud Function to retrieve the list of movements from the database and display an aggregated version of these in an on-screen list.



Getting Started with Google Cloud

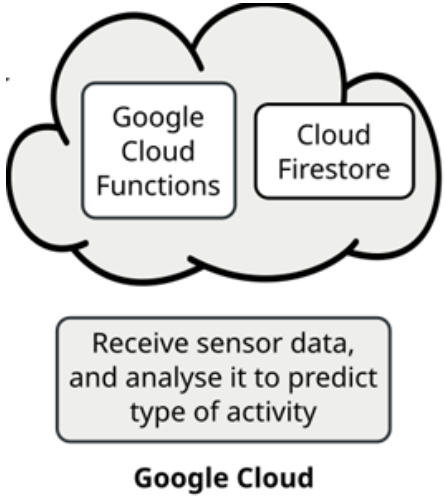
The first phase of the project is to set up the Cloud environment, and create the functions necessary for receiving, processing, and accessing data – along with some basic data storage.

A “Cloud Function” is a piece of code that lies dormant in the Cloud, and runs in response to a triggering event. In this case, the Cloud Function is written in JavaScript, and triggered with an HTTP request. The Cloud Functions stores and retrieves data from the “Firestore” database service, a cloud-hosted database.



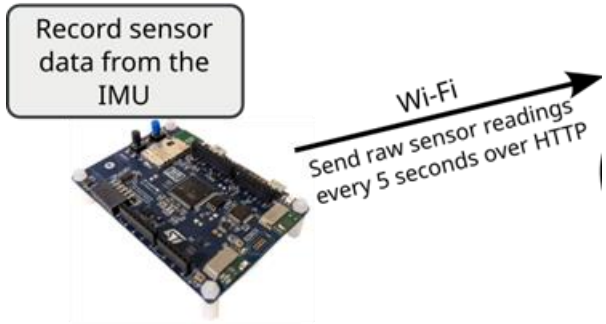
Cloud Functions

We develop two Cloud Functions for dealing with data. Both will be used by some part of the IoT system. The **processSensorData( )** function is called by the IoT discovery board, when the embedded device is uploading sensor data to the Cloud. The **getActivityData( )** function is called by the Android App, to retrieve the processed activity data.



Preparing the Embedded Device

The embedded device is programmed to get it to communicate with the Cloud. For this to work on the IoT Discovery board, you'll need a Wi-Fi connection. A home Wi-Fi connection is usually suitable for this, but University networks generally are not, as they often use WPA Enterprise encryption, which is not easily supported by the device. Some Universities do however maintain separate networks, for just such experimentation as this.



ST IoT Discovery Board

Sending Movement Data

It's now a case of sending the data to the Cloud, by triggering the **processSensorData( )** function created earlier. This function expects a batch of sensor data, so the code sends blocks of five samples. Sensor readings are collected every second, and then every fifth collection the batch is sent to the Cloud.

Building the Mobile App: Designing the User Interface

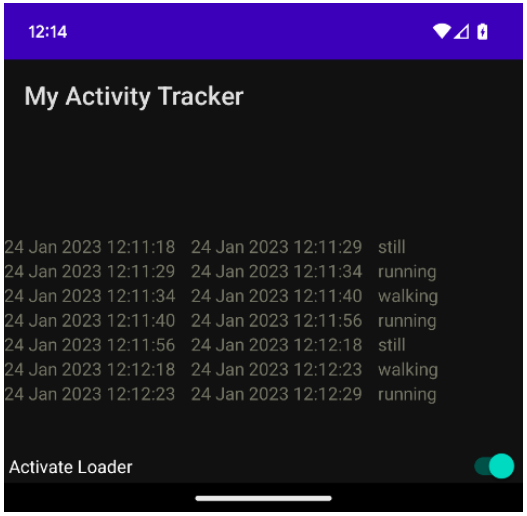
The final part of this project is building the Android App that will retrieve the predicted activity data and display it in a simple user interface. Building an entire mobile application can be a complex task, so we only touch on the basics of Android development. In this project, we will be developing in the Java programming language, although the Kotlin language is also available to use.

To get started with Android development, you will need an Android phone that's in developer mode, and an installation of Android Studio on PC.



Completing the Project

When all the (complex) stages of configuration and programming have been completed, and all parts powered and running turn, every five seconds new data reaches the phone and is added to the end of the table. Different movements can be simulated on the board, by giving it a shake. Two or more successive time periods of the same activity are merged into one line, as the Figure shows.



Chapter Review

- The Internet of Things (IoT) is a global network of connected devices, sensors, and other physical objects.
- The four main components of an IoT system are: Sensors & Actuators, Connectivity, Processing, and Applications.
- Sensors collect data about the environment, such as temperature, light, and motion. Actuators are devices that perform actions.
- Processing in IoT systems includes edge and cloud computing, the first is data processing at or near the source of data, cloud computing is data processing in a centralised location.
- Connectivity in IoT systems enables communication between devices and servers, it can be device-to-device or device-to-server communication.
- Apps are the mobile or web applications that let users interact with an IoT system, and offer a way to configure, control, and visualise the data.
- Public cloud services, such as Google Cloud, Microsoft Azure, and Amazon AWS make building scalable IoT systems easier and more cost effective for developers. They provide the various resources on-demand, and offer a wide range of services.
- Android Studio and the Android development ecosystem make it easy to get started with mobile app development, much like the Mbed Studio IDE makes it easy to get started with embedded development!