Contents

Prototyping an IoT Solution 2
Hold up, what is IoT, Really 2
Before You Begin 2
Prototyping Step by Step: From the IoT Architecture Perspective 2
The Things 3
The Communication Technology/Network 6
IoT Cloud 8
Dashboard 12

Prototyping an IoT Solution

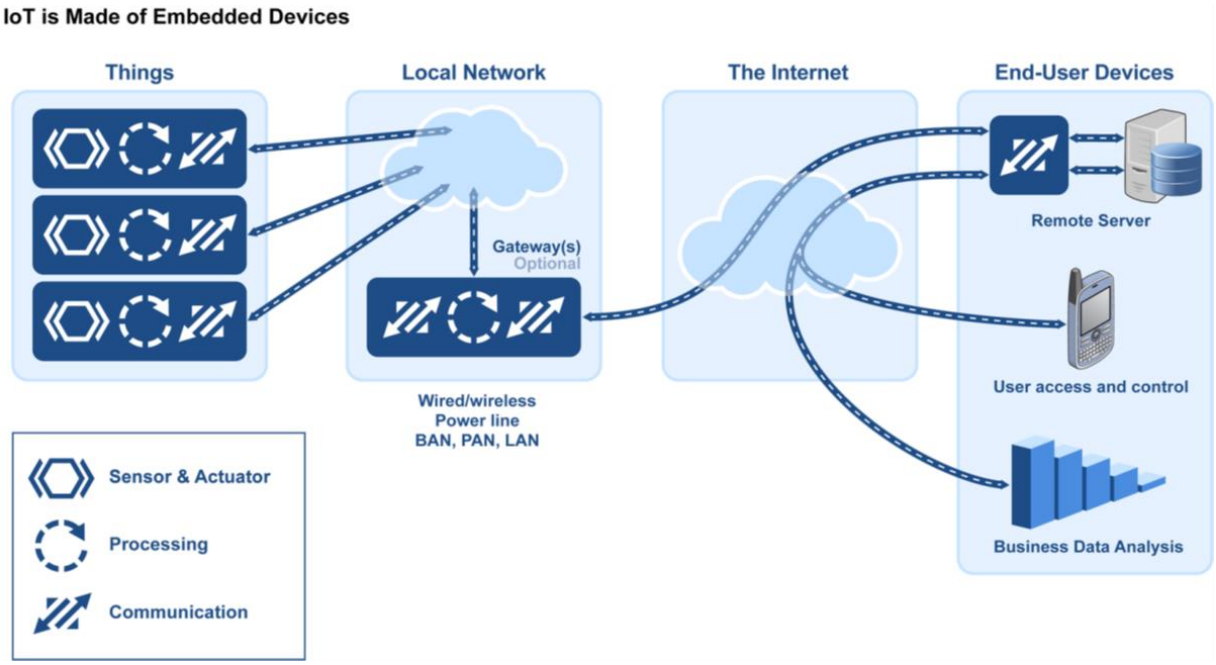
Hold up, what is IoT, Really

According to Wikipedia the Internet of Things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. The term IoT refers to a system of computing devices in the physical world which have been connected to the internet. These devices can either send or receive data from the internet.

Before You Begin

One of the most important process in the designing of an IoT solution (and in fact, any solution for that matter), is to define the problem. Before you decide on the devices to use, or the frameworks to use, think about what you are trying to solve. Well defined problems allow for well-defined solutions.

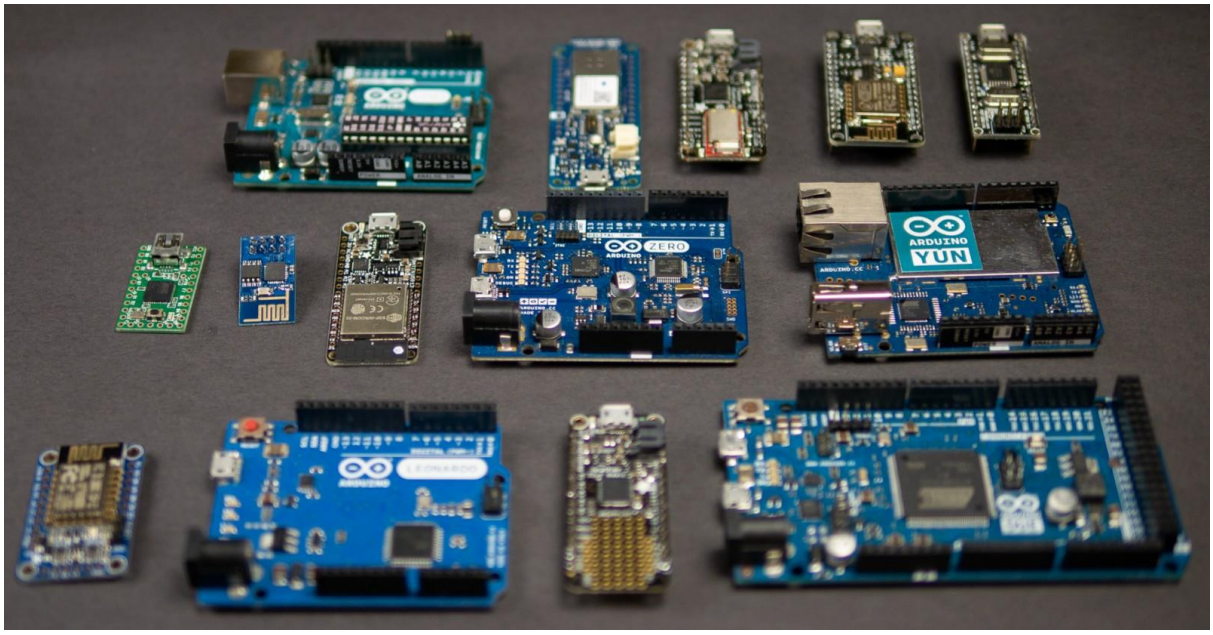
Prototyping Step by Step: From the IoT Architecture Perspective



The Things

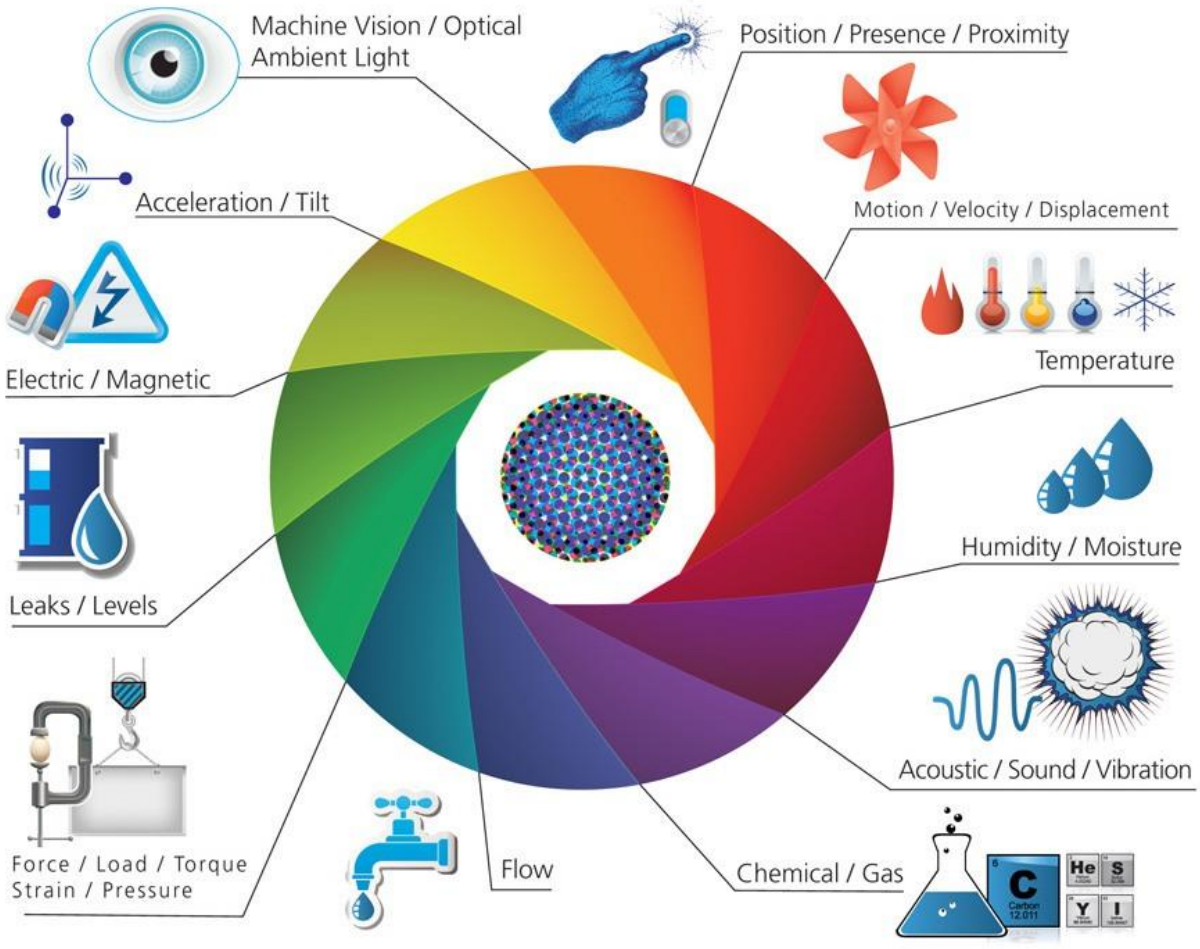
These are the end devices directly in contact with the physical environment. The thing is usually a sensor or actuator with a microcontroller as well as a communication module along with the electronic circuitry necessary for it to function. There are a host of hardware equipment you can use to build a 'Thing'. Some of these devices incorporate both sensing/actuating, processing and communication capabilities. Arduino and Raspberry Pi. Arduino is an open-source electronics platform based on easy-to-use hardware and software. Raspberry Pi is a small and affordable computer that lends itself well in building IoT devices prototypes

Arduino Compatible Boards



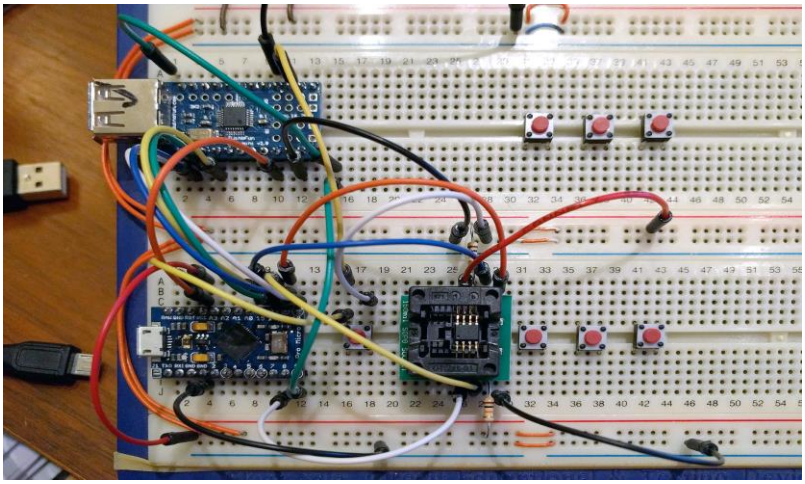
Raspberry Pi

Sensors/Actuators

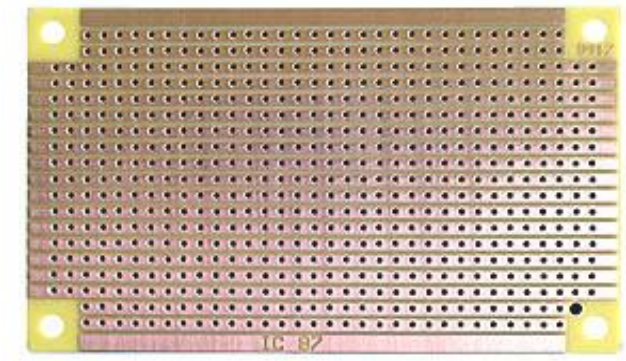


Building the Things

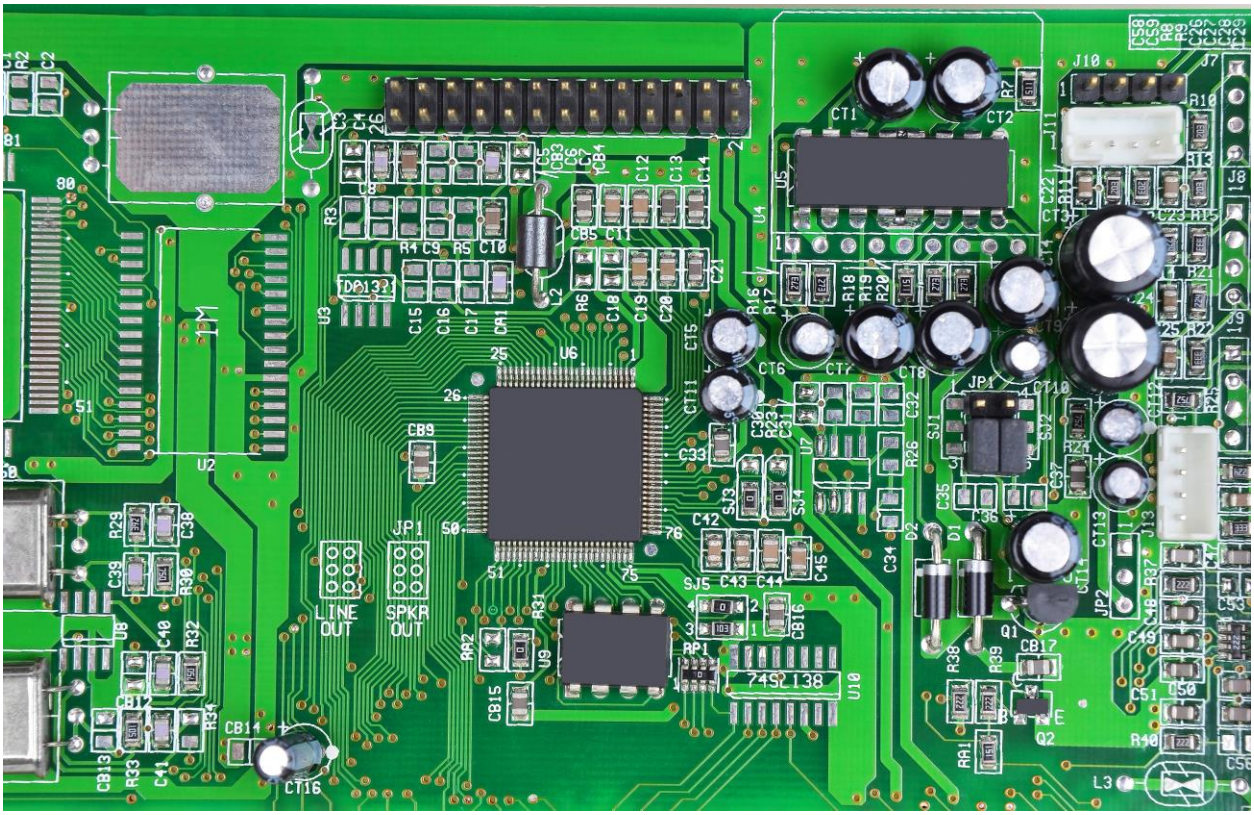
The Breadboard



Protoboards



PCBs (Printed Circuit Boards)



Languages – C/C++, Python, Node.js

The Communication Technology/Network

This refers to the mode at which these ‘things’ communicate to the internet. Such technologies include Ethernet, Wi-Fi, GSM, LoRaWAN, Bluetooth etc. Devices may

combine several communication technologies in order to send and receive data from the internet. For example, Bluetooth cannot be used directly to connect to the internet. The sensor node might be connected to a Bluetooth gateway which is in turn connected to an Ethernet network and thus able to communicate with the internet.

You have to consider several factors in designing the IoT network

- -Power Consumption
- -Range
- -Data Rates

Low Power Wide Area Network technologies like Sigfox and LoRaWAN lend themselves very useful for IoT device communication.

IoT Cloud

This is where the fun begins. The cloud basically refers to computing resources that can be accessed over the internet. A lot of things happen in the cloud. Data is stored, cleaned, analysed. This is where the back-end services run. The back-end applications can do virtually anything you desire them to do, from controlling the IoT devices, to storage, to event processing, to data analytics to machine learning.

There are basically two approaches to designing the cloud environment/Device Backend

I know what am doing/Do it Yourself

Of course you could always build your own computing infrastructure, roll up your own servers in your data centre/local computers that will house the applications you build to interact with IoT devices. You have complete control of your infrastructure and you have to write applications to interface each part of your system.

The Public Cloud Option

The better option is to use the public cloud as the solutions backend for you IoT devices. There are a host of cloud providers that provide specific tools and platforms tailored towards rapid development of not only prototypes but production grade IoT solutions. These cloud providers usually provide platforms that:

-Device endpoint for connectivity and telemetry ingestion, device management (including command and control) capabilities e.g Azure IoT Hub.

-**Stream Processing** – process large streams of data records, and evaluates rules for those streams

-**Business Process Integration** - Facilitates executing actions based on insights garnered from device telemetry data during stream processing. Integration could include storage of informational messages, alarms, sending email or SMS, integration with CRM, and more.

-Storage – Devices are sending huge amounts of data, you need to design for storage. As with storage options in regular applications, there are various ways to store data

from IoT devices, from the traditional SQL databases, to No-SQL databases such as Mongo, to blob storage options e.g AWS S3, Azure's Storage Account, Google's Cloud Storage .

-**Analytics and Machine Learning Platforms**- IoT devices are collecting massive amounts of data from various devices. This data is almost useless if not analyzed. Cloud vendors provide platform services that enable you to easily analyze your IoT data: These include Stream Analytics, Data Lake Storage and Analytics from Azure IoT, AWS IoT Analytics, Google's Data Proc and ElasticSearch which is an open source search engine based on the Lucene library and can be used to analyse and aggregate data from anywhere. Elasticsearch comes with Kibana that can be used to visualize data in ElasticSearch

Cloud vendors also provide easy to use machine learning services such as Azure ML and AWS SageMake.

-**Application Development Platforms**-While this is not specifically tied to an IoT offering, this is a convenient platform as a service that major cloud vendors sell. This is the ability to deploy applications without needing to provision worry about infrastructure of where the application will be running. These include Azure's App Service, Google's App Engine, and AWS' ElasticBeanstalk Services. You could even build serverless apps where you write code and deploy to a serverless platform that runs your code on demand, driven by the reaction to events and triggers happening in near real time. E.g AWS Lambda, Azure Functions, Google's Cloud Function.

There is nothing fundamentally different between an IoT application and a normal application such as a web app (IoT applications are also web apps). You use the same languages and the same frameworks to write these applications.

You can use platforms such as Node-RED to quickly prototype an IoT backend application without needing to write code." Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click."

Examples of Cloud Services for IoT

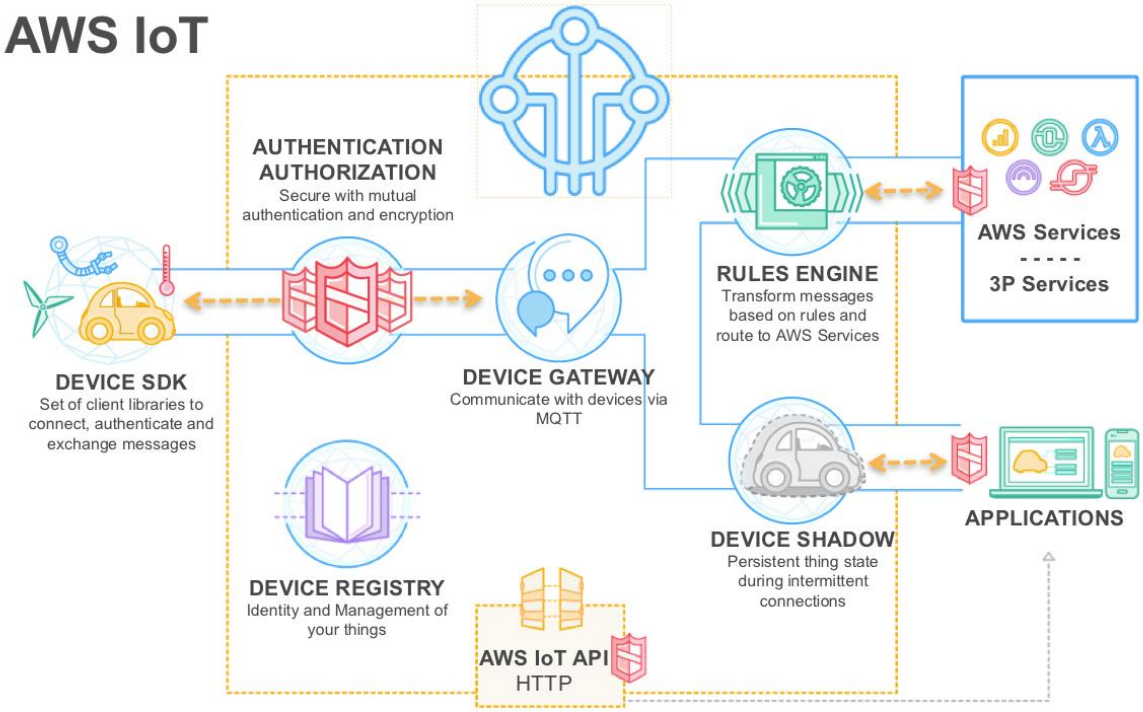
Azure IoT

The Azure Internet of Things (IoT) is a collection of Microsoft-managed cloud services that connect, monitor, and control billions of IoT assets. In simpler terms, an IoT solution is made up of one or more IoT devices and one or more back-end services running in the cloud that communicate with each other.

AWS IoT

AWS IoT is a platform that enables you to connect devices to **AWS** Services and other devices, secure data and interactions, process and act upon device data, and enable applications to interact with devices even when they are offline.

AWS IoT can integrate with other AWS services such as Amazon S3, Amazon Dynamo DB, Amazon CloudWatch, Amazon ElasticSearch Service.



Other IoT Platform providers include: Google IoT, IBM Watson Platform.

Some platforms while not providing fully fledged IoT Suites like Azure or AWS, do provide convenient services to enable data collection, visualizations and analysis.

These include:

- theThings.iO https://developers.thethings.io/docs/getting-started is a platform that provides a complete backend solution for IoT Markers and IoT App Developers through an easy and flexible API. theThings.iO is hardware agnostic. This means that we don't force you to use a specific hardware, provided that it is capable to use any of our supported protocols: HTTP, Websockets, MQTT or CoAP.
- ThingsBoard https://thingsboard.io/ ThingsBoard is an open-source IoT platform for device management, data collection, processing and visualization for your IoT projects.
- Adafruit IO https://io.adafruit.com/ Adafruit.io is a *cloud service* – that you can connect to it over the Internet. It's meant primarily for storing and then retrieving data.On top of that, you can display your data in real-time, Connect projects to web services like Twitter, RSS feeds, weather services, etc. and Connect your project to other internet-enabled devices
- ThingSpeak - https://thingspeak.com/ Is an open IoT platform to collect sensor data privately to the cloud, analyze and visualize data and trigger an action.

Dashboard

This is the final piece of the puzzle in developing your IoT solutions. This is what the end user sees and interacts with. Here is where you visualize the real-time data and analytics from your sensors. It may also act as the where the users may interact with the 'Things' to be able to control and manage them or read their status information. It is your 'window' into your IoT system. It may range from dashboards provided by services such as Power BI, and ThingsBoard or custom made interfaces made with frameworks/libraries such as React, Vue or Angular.

The End: Questions ?