

IoT Technologies for Embedded Computing: A Survey

Farzad Samie, Lars Bauer, Jörg Henkel
Chair for Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), Germany
{farzad.samie, lars.bauer, joerg.henkel}@kit.edu

ABSTRACT

Emergence of Internet-of-Things brings a whole new class of applications and higher efficiency for existing services. Application-specific requirements, as well as connectivity and communication ability of devices have introduced new challenges for IoT applications.

This paper provides an overview of IoT technologies required from an embedded design perspective and specific properties associated with IoT in embedded systems' landscape. We investigate essential technologies for development of IoT systems, existing trends, and its distinguishing properties. By discussing the key characteristics, main application domains, and major research issues in IoT, this paper provides a comprehensive IoT perspective for embedded system design.

1 Introduction

The Internet of Things (IoT) is a multidisciplinary paradigm in which many of the objects that surround us will be networked and connected to the Internet in order to provide new services and increase the efficiency [1, 2]. Recent and ongoing advances in the technologies such as wireless communication, ultra-low power processors, embedded sensors and actuators, Radio Frequency IDentification (RFID), mobile phones, and cloud/fog computing has enabled the emergence of IoT [3]. Although not all those technologies are needed for each and every IoT application, they all facilitate the proliferation of IoT by providing an essential prerequisite [4, 5]. While RFID enables low-cost object identification, and while ultra-low power system-on-chips (SoC) enable portable battery-operated embedded devices, cloud computing and fog computing can be used to offload computations and services to the local or global servers, providing additional resources for handling large-scale data or performing more complex operations [6, 7].

Connectivity (wired or wireless) is what distinguishes embedded IoT systems from conventional embedded

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CODES/ISSS '16, October 01-07, 2016, Pittsburgh, PA, USA

© 2016 ACM. ISBN 978-1-4503-4483-8/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2968456.2974004>

systems. In a broader sense and vision, IoT is a global infrastructure of *heterogeneous, networked* embedded devices and objects [8]. Communication ability, and in particular the Internet connectivity, lets devices and smart objects (also known as machines) communicate and interact with (i) other machines and devices, or (ii) humans [4, 9, 10].

IoT covers diverse application domains that include wireless sensor networks (WSN), machine-to-machine (M2M), RFID, Cyber Physical Systems (CPS), Mobile Computing (MC), etc. [9, 10, 11]. There have been many research efforts on IoT from the perspective of networking, object identification, data access (security and privacy) [3, 5], however, it has gained less attention from the perspective of embedded computing.

The diversity of IoT applications and technologies makes it difficult to present a general comprehensive statement for the requirements of IoT in hardware and software. Therefore, **the IoT embedded designer faces questions whose answers are challenging as the solutions can be contradictory, e.g.:**

- Which wireless communication technology 1) covers the required range, 2) provides the required data rate, 3) is still (ultra) low-power and meets energy constraints?
- What trade-offs to make between 1) Quality of Service (QoS) and energy consumption, 2) on-board processing and computation offloading, etc.?
- How to handle the uncertainty and unpredictability of IoT systems (mainly caused by communication)?

This paper explores the IoT and its technological enablers from this point of view. Besides investigating essential technologies for IoT and existing trends, this paper provides distinguishing properties of IoT for embedded domain in addition to a comprehensive IoT perspective for embedded systems which, to the best of our knowledge, lacks in the existing surveys like [1, 4, 5, 7, 8].

Paper structure: in Section 2, a high-level overview of IoT applications and devices is presented. Then, we provide a brief presentation on hardware and software characteristics of IoT in Section 3. The electronic design automation (EDA) tools to support IoT are presented in Section 4. Wireless technologies and their challenges for connectivity of IoT devices are discussed in Section 5, while Section 6 concludes the paper.

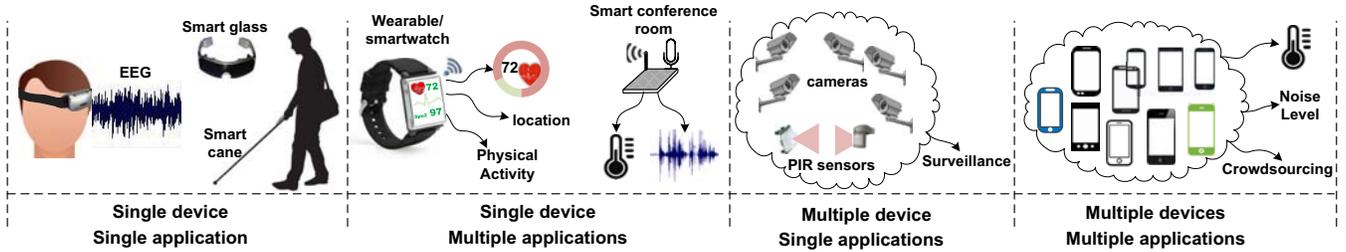


Figure 1: IoT systems may exploit single/multiple devices to implement single/multiple applications

2 Properties of Devices and Applications

2.1 Application Areas

IoT can impact various application domains either by enabling new services, or by improving the efficiency of existing ones [12]. Among the possible applications, we provide a review of their main categories (that cover a wide range of different requirements, technologies, development challenges) and futuristic applications. Indeed, IoT applications are not limited to these categories, and a huge number of applications can be envisioned. However, their requirements, properties, and design challenges have similarities with those presented in our categories. The main challenges and requirements are discussed in the following sections.

Healthcare. IoT has shown a great potential for enabling and improving healthcare services [13]. IoT-based healthcare systems enable long-term monitoring of personal health status in real-time anytime, anywhere. They acquire vital biosignals including electrocardiogram (ECG) –electrical signal of heart–, electroencephalogram (EEG) –electrical signal of the brain–, and electromyogram (EMG) –electrical signal of muscles–, body motion, etc. The real-time data can be stored, processed, or transmitted to a remote device (e.g. cloud server) for further processing and diagnosis [14, 15]. Ultra-low power design and real-time constraints are among the challenges for these applications.

Assisted Living. Assisted living aims at offering solutions for helping (i) elderly, (ii) chronically ill, and (iii) disabled people [1]. For instance, a wearable IoT device can leverage online city maps together with a smart cane to detect and avoid obstacles, access buildings, navigating indoor and outdoor, etc.

Smart Building and Home. IoT provides connectivity for embedded devices which can enable applications for reducing the costs, increasing personal comfort, and improving safety and security in buildings and homes [11].

Smart City. In a smart city, distributed IoT devices equipped with different sensors are used to improve the transportation and traffic management, monitoring the air quality (e.g. pollution, temperature, humidity, etc.), smart parking, smart lighting, and smart watering gardens [16, 17].

Smart Industry. IoT-enabled solutions for automation, control and monitoring may improve industry by

lowering operational and maintenance cost, and increasing quality of service [5], for industrial domains such as supply chain management, transportation and logistics, and automotive [12, 18]. An example is remote monitoring of machinery (e.g. in plant, wind turbine, etc.) for predictive maintenance [5, 10].

2.2 Applications vs. Devices

The combination of IoT applications and their underlying hardware or device introduces some design challenges which need to be address either at software application or at the hardware level. The relation between the number of devices and number of provided service and applications can be classified into four categories:

- **One-to-One:** One IoT device is used for a single service. For instance, an IoT-based healthcare monitoring device that captures real time biosignals [19, 20].
- **One-to-many:** One single IoT device provides multiple services. One example is a wearable device like a smart watch that has several sensors and can keep track of user’s physical activity, heart rate, location, etc. [4]. Another example is a smart conference room which uses a single device for multiple applications including detecting the start/end of a meeting, analyzing the environmental condition of the room (e.g. temperature and luminance), and processing the acoustic signals to record the proceedings of the meeting, etc. [21]. For this category, a decision that needs to be made by the designer is the management of shared resources. The solutions range from conservatively choosing the underlying hardware (processor, memory, wireless ratio, etc.) which support the worst case accumulated usage, to dynamically managing and scheduling the hardware usage.
- **Many-to-one:** In this class, spatially distributed devices provide a single service. For instance, distributed smart cameras are exploited for video surveillance in [22]. This category usually has two properties that need to be considered by designers to optimize the system: 1) high communication between devices and 2) large amount of redundancy.
- **Many-to-many:** In some IoT applications multiple devices are shared between multiple applications and services. Smart Citizen [23] consists of multiple IoT embedded devices that are geographically distributed to gather information for the applications that report temperature, humidity, noise level, and air pollution. Shared WSNs belong to this category, too [21, 24].

Figure 1 shows these categories with some examples. Although shared IoT devices reduce the cost of hardware and maintenance, they introduce new challenges like binding, allocation, online and runtime resource management, guaranteeing the timing constraints of applications that share a single device, etc. [11, 25, 26].

3 Hardware & Software Architecture for IoT

The general operation stages of an IoT application include 1) data acquisition, 2) data processing, 3) data storage, and 4) data transmission. The first and last stages exist on every application, while the processing and storage may or may not exist in some applications (see Figure 2).

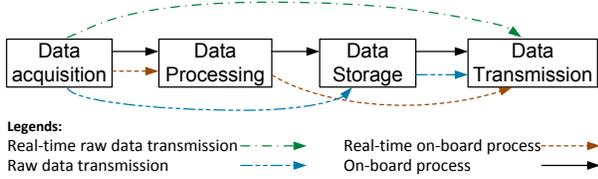


Figure 2: General stages of IoT applications

3.1 Efficiency at Different Stages

- **Data acquisition:**

At HW level: Low power integrated and on-chip sensors with Micro-Electro-Mechanical Systems (MEMS) technology can reduce the energy consumption of data acquisition. On-chip accelerometers, gyroscopes, microphones and biosensors are some examples.

At SW level: Energy efficient sensing schemes should effectively exploit spatial and temporal characteristics of the input data, in order to collect as less samples as possible while the required signal quality is retained [7]. The reduced amount of input data (depends on number of samples and resolution) affects the energy for transmission and storage, too. Compressed sensing (CS) [27] is a novel techniques in which the signal can be reconstructed from much fewer samples than Nyquist theory, at the cost of accuracy loss. As long as the input data has the sparseness property, a smaller number of samples can capture the required information, and CS can reduce the volume of collected data without significant loss of information.

Many IoT applications have the data sparsity property and can exploit the CS paradigm. In health monitoring applications and wireless body sensor network, CS has been investigated and studied extensively.

- **Data processing:**

At HW level: Energy efficiency in a processing unit can be achieved by 1) ultra-low power processors [28] and 2) efficiently customized co-processors [29]. In [25], a heterogeneous dual-core processor is proposed and fabricated based on the big.LITTLE architecture. An ultra low power near-threshold processor alongside with a high performance processor in addition to a task scheduling framework brings energy

efficiency for IoT applications.

- **Data storage:**

At HW level: Energy reduction in memory has received significant industrial and academic attention in embedded system design community [30], but there are some characteristics specific to IoT applications that can be exploited for further improvements in energy efficiency of memory in IoT embedded devices. For instance, many IoT applications inherently tolerate errors in data which opens up new possibilities for hybrid memory architectures composed of an error-free portion (for more reliable data and operations) and an error-prone one (for less important data). A hybrid memory based on this property has been designed for a low power biomedical signal processors in [31]. For wearable IoT devices that need to have flexible and curvilinear forms, flexible non-volatile memory (NVM) has been proposed [32].

- **Data transmission:**

At HW level: Data transmission can be improved by integrating radio transceivers into SoCs, providing low power multi-radio chips, etc. Main existing technologies, trends and challenges in connectivity and communication are discussed in Section 5.

At SW level: In order to reduce the amount of data to be stored or transmitted, new data compression techniques, specially for the streams of data, are needed [33].

Figure 3 shows a general architecture of the main components of an IoT SoC platform [34]. An IoT embedded device has many –if not most– of these components, e.g. at least one RF component for the connectivity.

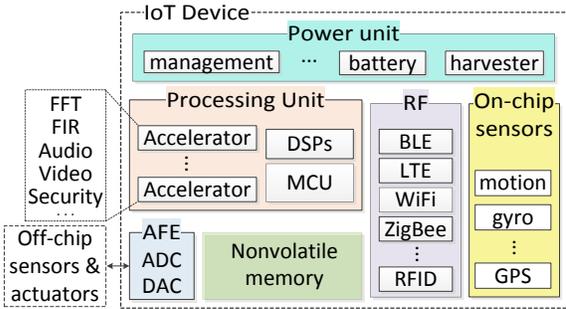


Figure 3: General architecture of an IoT embedded device

3.2 Different Computing Layers

An enormous amount of data, including streams of data, audio, or video, will be generated from IoT devices which is also known as Big Data [35]. The problems associated with emerging Big Data (e.g. massive storage and huge processing power demand, high latency, etc.) necessitates the migration of computation and processing to different underlying computing layers available in the IoT chain [14, 26]. Figure 4 shows different processing and computing layers starting from IoT embedded device up to the Cloud servers.

The collected data can be processed on either IoT device, or gateway, etc. Hence, for IoT applications that

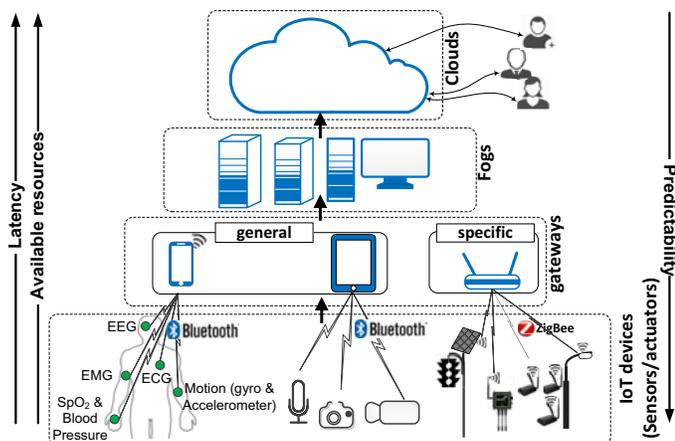


Figure 4: Different computation layers. The available resources (e.g. memory, processing power) and networking latency increase from bottom to top.

involve data processing, a major challenge is to decide where the computation should be done (i.e. computation offloading) [26, 36]. The decision depends on many parameters and factors including the system objectives (e.g. real-time requirement, energy efficiency, etc.), and system specifications (e.g. energy consumption for data processing and data transmission on IoT device, communication bandwidth, transmission delay, etc.). As shown in Figure 4 different computing layers and platforms include:

- **Device centric:** The microcontroller in an IoT device can be exploited to perform the computation. The main challenges are the scarce resources on IoT devices (e.g. on-chip memory, processing capabilities) and to meet the low-power requirements. A key decision concerns whether to perform the computation on the IoT embedded device or to offload it to other layers of computation. This decision making becomes more challenging when the answer should be determined at runtime (i.e. depends on the operation, input data, energy source’s status, and other runtime parameters) [26].
- **Gateway centric** (e.g. smartphone): IoT gateway devices which are used to settle the heterogeneity between different networks and Internet usually have more computational power [36, 37] (e.g. ARM Cortex-A and Cortex-M MCUs are proposed as gateway processor in ARM, Freescale and Texas Instrument solutions). Gateway devices like smartphones can be used to perform data processing as proposed in [38]. This scheme has been used for medical and healthcare monitoring application in [19], where the captured ECG signal is transmitted to the smartphone to be processed. The key challenge here is to guarantee the availability and deadline constraints, which is difficult due to the unpredictability of the wireless communication and latency, especially when the number of IoT devices increases.
- **Fog centric:** The concept of fog computing is the ex-

tension of the cloud computing paradigm, which can help the IoT domain to restrain the Big Data problem [6, 35]. Fogs provide more computational power compared to IoT embedded and gateway devices (due to their location and distribution) [14, 39]. An ECG feature extraction for a healthcare monitoring application is presented in [14] and shows the benefits of the fog computing concept.

- **Cloud centric:** [7, 18] Cloud computing provides a solution for handling Big data and processing them. It requires massive data storage volume, huge processing resources to deliver a high quality of service (QoS) and to help decision making [5]. However, as the number of IoT devices increases, and consequently, the amount of stream data increases, the cloud computing solution faces problems and challenges including the scalability, high energy cost, latency, bandwidth, and availability [13, 35].
- **Hybrid approach:** Since each of the aforementioned choices have their advantages and disadvantages, solutions that exploit a multi-layer and hybrid computing approach are more efficient. An example: classification methods, including Support Vector (SV) machines, are widely used for anomaly detection or gesture recognition in personal healthcare monitoring applications [20]. A set of SVs are stored in memory which will be used at runtime for classification. When the number of SVs is large (i.e. needs large memory on IoT device), an efficient solution is to store some SVs on the gateway (e.g. smartphone), instead of increasing the memory size at design time. In this solution, if a feature is far from the SVs that are stored on the IoT device, it will be transmitted to the smartphone, and compared with other SVs. Such a hybrid approach enables the applications to increase their efficiency and meet their requirements by leveraging the advantages of each computing level (e.g. larger memory, more computational resources, lower latency, etc.). However, the main challenge to address in a hybrid approach is to find the efficient balance between local processing and computation offloading, and to find the perfect timing for offloading (in real-time applications).

3.3 Approximate vs. Exact Computing

The emerging paradigm of approximate computing leverages inherent resilience of applications and relaxes the requirement of exact equivalence between the specification and implementation to gain more efficiency [40, 41]. Most of IoT applications are interacting with the physical world with noisy input data [11]. Therefore, they are inherently dealing with approximation. For instance, the first stage of approximation happens in the Analog-to-Digital conversion which introduces a quantization error. Although these applications tolerate some errors, the final output or QoS should be in a certain range.

Many applications can exploit the error tolerance

property to trade the output quality for computational effort (e.g. energy consumption, performance, etc.). Some open challenges exist in the domain of approximate computing for IoT. The tolerable error can be accepted and exploited at different hardware components and different software parts:

- **Data acquisition:** The quality of input data, during the data acquisition, is determined by resolution and sampling rate (or frame rate). For instance, the ECG signal in [14] is capture at 360 samples per second with 11-bit resolution. When the IoT application tolerates error, reducing the quality of input data is one way to take advantage of it to reduce the energy consumption or delay.
- **Data processing:** The approximation can be done also in underlying hardware by designing inexact hardware units for specific arithmetic operations (e.g. adders, multipliers, DCT, FFT, etc.) [41]. It can also be done at the software level by methods like stage skipping [40].
- **Data storage:** The tolerable error can be exploited at the memory unit to reduce the size of required memory, reduce the number of access to memory, or reduce its energy consumption. One example of approximation in memory is presented in [31].

Indeed, hybrid schemes, where multiple stages exploit approximation, are also possible. The main challenge here is to decide 1) at which stage, and 2) how much approximation should be applied in order to minimize the computational effort while meeting the QoS requirements.

4 IoT Electronic Design Automation (EDA) Tools

EDA tools need to address challenges in two aspects of IoT design: 1) the ever-increasing demand for designing embedded devices, and 2) the growing complexity of IoT SoCs.

The proliferation of IoT market leads to small design companies with relatively specific and innovative products in a smaller scale environment which means the emergence of IoT companies with a system design (i.e. not chip design) background [42]. Their main concerns will be reduced design time and risks [43]. These concerns, in addition to the lack of chip design expertise,

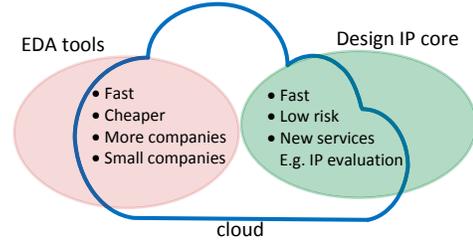


Figure 5: Trends in design tools & EDA for IoT

make using Intellectual Property (IP) cores a promising solution to manage design complexity and integration risk. It is predicted that design IP cores (various hardware blocks that are already pre-designed, pre-verified and tested) will be driving the IoT development, and we will witness an explosive growth in the IP companies [44].

The IoT specific characteristics are 1) interaction with physical world (by means of sensors and actuators) and 2) communication ability (intra and inter devices). Both of these properties involve with mixed-signal and analog. Wireless radio, sensor ADC, analog filters and amplifiers seem to be the most minimal requirement of IoT platforms. This makes the design flow increasingly more complex, especially the verification and debugging of mixed-signal SoCs [44, 45].

There are two complementary trends in EDA to fulfill the aforementioned IoT demands [44] (as shown in Figure 5):

1. Design tools to support the growth of IoT devices (i.e. scalable, cheap, fast designs): These tools provide traditional EDA services but at a lower cost, higher scalability, shorter design time. For instance, IBM and SiCAD are offering a cloud service with IBM EDA tools running on high performance cloud platforms that makes the modeling, verification, and simulations faster and more efficient. Similarly, Silicon Cloud International (SCI) provides cloud-based “workflow as a service” for ASIC design, targeting IoT system developers.
2. Design tools for IoT specific core (e.g. analog or RF IP cores): It includes energy efficient and small IP cores that provide wireless communication and analog front ends for the IoT SoCs.

With the increase in IP demand, some new cloud-based opportunities will arise. For instance, the customer will

Table 1: EDA tools for IoT design

(a) IP based design

(b) ASIC design tools

EDA Tools \ IP Cores	Cadence	Synopsys	ARM	Silvaco	Tanner Mentor Graphics	IBM SiCAD	SCI	Web-based Cloud tools
Radio	2G, 3G, LTE, WiFi 802.11b/g/n/ac/ad	BLE	RFID, BLE, WiFi, 802.15.4	×	✓	✓	✓	AMS
Coprocessors	noise, image, video, vector processing	vision processor, sensor fusion, CNN	NA	✓	✓			MEMS
Analog Front End	ADC, DAC	ADC, DAC	ADC, DAC	✓	✓			
On-chip Sensors	fingerprints, motion, noise level, gyroscopes	NA	NA					

* CNN: convolutional neural network

* AMS: Analog-Mixed Signal, MEMS: Microelectromechanical systems.

be able to evaluate the IP core online (without downloading it) by running simulations and checking the performance and power before purchasing it [46].

Table 1 summarizes the tools and features that some EDA companies provide for IoT design. However, EDA tools have not caught up with other IoT trends like approximate computing.

5 Connectivity

5.1 Wireless Communication Technologies

Different wireless communication technologies can be used for (i) connecting the IoT device as local networks, and (ii) connecting these local networks (or individual IoT devices) to the Internet.

- **NFC** [47]: It is a short-range wireless communication technology that enables the data transmission between devices in a close proximity to each other (~ 20 cm). It has a tag that can contain small amount of data. This tag can be read-only (similar to RFID tags for identification purposes) or can be re-writable and be altered later by the device.
- **Bluetooth**: This technology witnesses an increasingly ubiquitous presence including in smartphones, tablets, laptops, headsets, etc.
 1. **Classic Bluetooth**: It offers a high enough throughput and bandwidth which makes it suitable for data stream applications (e.g. audio). However, it has several limitations including limited number of nodes in the network (up to seven slaves) or topology.
 2. **Bluetooth Low Energy (BLE)**: It is also known as Bluetooth smart and is designed and enhanced for short-range, low bandwidth, and low latency IoT applications. The advantages of BLE over classic Bluetooth include lower power consumption, lower setup time, and supporting star topology with unlimited number of nodes.
 3. **Bluetooth 5.0 (BT v5)**: It increases the range and doubles the speed of low energy connections while increasing the capacity of broadcasts.
- **ZigBee**: A small-size, low-cost, low-power wireless specification that can support different network topologies (e.g. mesh, star, tree). It offers a wide transmission range, depending on the output power. Although ZigBee has established in some industrial applications and WSN nodes, it faces some market barriers, especially with the emergence of attractive alternatives like BLE that provides higher bandwidth at a lower energy consumption.
- **WiFi**:
 1. **Conventional WiFi (IEEE 802.11 b/g/n)**: The main advantages are high bandwidth and availability in urban districts. Its high energy consumption makes it unsuitable for ultra-low-power IoT devices.
 2. **Low-power WiFi (802.11 ah) or HaLow**: Compared to conventional WiFi, it is intended to extend the range of transmission with less data rate, and to reduce the energy consumption for IoT

applications. In addition, it will suffer less interference with existing wireless networks as it uses a different frequency band (i.e. 0.9 GHz).

- **Cellular network**: Widespread mobile networks like 3G and LTE provide reliable high-speed connectivity to the Internet. However, they have a high power consumption profile and they are not suitable for M2M or local network communication.
- **Low Power Wide Area Network (LPWAN)** [48]: These technologies are suited for low power applications with very long range transmission. They support up to 10 Km distance between end-nodes and gateway. However, it comes at the cost of very low data rate (<1 Kbps). Main technologies of LPWAN include SigFox, LoRaWAN and Weightless, which operate in sub-GHz bands. One of the challenges for LPWAN is the lack of a globally available band for LPWAN in sub-GHz.

Table 2 summarizes some characteristics of wireless technologies which are needed to be considered in the design process of IoT devices. This table shows the typical value of these parameters. Indeed, most of them depend on the design constraints of the IoT devices. For example, by increasing/decreasing the transmission power (and the size of antenna), the transmission range can be further increased/decreased. Table 3 shows the suitability of each wireless technology for different application domains.

Each of these communication technologies has its advantages and disadvantages. For instance, Bluetooth, WiFi, and ZigBee may face interference due to the co-existence of other devices working at the same frequency band (i.e. 2.4 GHz), especially with the increasing rate of IoT devices. The interference can lead to severe drop in data rates, which consequently may increase the energy consumption of IoT devices, reduce the QoS, and result in missing the deadline in real-time applications (i.e. affecting other optimization goals of the system). On the other hand, NFC needs sender and receivers to be close to each other.

Hybrid communication schemes seem to be the best fitted solutions for IoT applications. The emergence of integrated transceivers which include multiple communication technologies on a single chip has opened doors for more efficient wireless communication. For instance, BLE is not intended for continuous data streaming applications, but it is highly efficient for sending small, discrete data chunks (e.g. states, temperature, heart rate). Bluetooth classic and WiFi, on the other hand, offer higher throughput and efficiency for streaming application with higher data rate demands (e.g. ECG monitoring, audio).

5.2 Timing of Communication

The timing of data transmission schemes in IoT applications can be classified into three different categories.

- **Continuous**: The IoT devices send or receive data continuously (e.g. real-time health monitoring).

- **Sporadic:** The IoT device collects and stores the data and then transmits it whenever the connection is available.
- **On-demand:**
 1. **User driven:** The IoT device can be requested by the operator to send the collected data [49].
 2. **Event driven:** The communication is done once a specific event happens.

For designing an efficient IoT system, the timing of data transmission matters, especially for managing low power modes (deep sleep, standby, active). For instance, if the data transmission is sporadic, a predictor based on learning techniques can be used to dynamically manage the low power modes of the wireless transceiver.

5.3 Bandwidth & Data Rate of IoT Sensors

In this section, we provide a comprehensive overview of the main sensor applications in IoT. Then we derive the typical data generation rate of different sensors according to the reported application scenarios. The required wireless transmission bandwidth and the appropriate technology is annotated, too. Then we estimate the required processing power to perform the typical operations on the captured data, and we report the appropriate IoT hardware core to support it.

- **Ambient/Object Temperature, Humidity:** Different applications including smart homes, smart cities, smart industry, etc. use these sensors.
- **Accelerometers and Gyroscopes:** These sensors are widely used in industrial machinery, medical and fitness devices, as well as wearable devices for monitoring physical activity, fall detection, navigation, or for structural health monitoring of machinery or buildings [50].

- **Magnetometer:** It usually comes along side with accelerometers and gyroscopes and is used for finding the direction for navigation.
- **Light:** This sensor is usually used for saving energy consumption by adjusting the lighting (e.g. smart-phones, smart homes, smart cities).
- **Chemical Sensors:** These sensors are mainly used for measuring the air quality (e.g. CO, NO2, SH2, and CO2) in smart city application, environment monitoring, or smart factories [23].
- **Location:** These sensors are mainly based on GPS technology (seldom work indoors). Smart industry and transportation use this sensor for tracking and localizing the objects.
- **Imaging:** Besides normal imaging sensors, infrared sensors are emerging in IoT market to capture the temperature differences of the objects. The feature is useful for security applications to detect intruders or getting thermal image of home or objects to detect insulation or possible leaks.
- **Acoustic:** Analog or digital microphones are used for security and surveillance, and noise pollution for smart cities [23].
- **Ultra Violet (UV):** It measures the strength of UV radiation for healthcare, wellness, and smart city applications. It also indicates the strength of sun's exposure for smart farming and smart agriculture.
- **Ultrasonic:** It is useful to detect the obstacles and get the distance. Assisted-living and smart industry can benefit from this sensor to help visually impaired people or preventing the moving robots to collide, respectively.

Table 2: Communication technologies for IoT applications, and their properties

		NFC	Bluetooth	BLE	BT v5	ZigBee	HaLow 802.11 b/g/n	LP WiFi 802.11ah	LPWAN	Cellular network	
Range	indoor	<0.2m	1–100 m	~100 m	<300 m	<20 m	<70 m	<700 m	<10 Km	3G	LTE
	outdoor									>5 Km	>5 Km
Bit rate [Mbps]		0.424	1–3	1	2	0.25	>1	0.15–40	<0.05	0.17	75–300
Throughput [Mbps]		0.22	1.5	0.30	1.5	0.15	2-50	>0.1	<0.05	NA	
freq. [GHz]		0.014	2.4–2.5	2.4–2.5	2.4	2.4	2.4/5	0.9	sub-GHz	0.8–1.9	2.1
Network topology		p2p	scatternet	star, scatternet	NA	star, tree, mesh	star	star	star	NA	

Table 3: Suitability of communication technologies for IoT application domains

		Application domains					Local Network (M2M)
		Healthcare	Smart Cities	Smart Building	Automotive	Industry	
wireless tech.	NFC	medium	high	low	very low	very high	medium
	BLE	very high	low	low	very low	low	high
	ZigBee, BT v5	medium	high	very high	low	high	high
	WiFi b/g/n	low	high	medium	medium	low	high
	HaLow	high	very high	high	high	high	very high
	LPWAN	low	very high	high	high	very high	high
Cellular networks (3G, LTE, etc.)		low	high	high	high	medium	very low

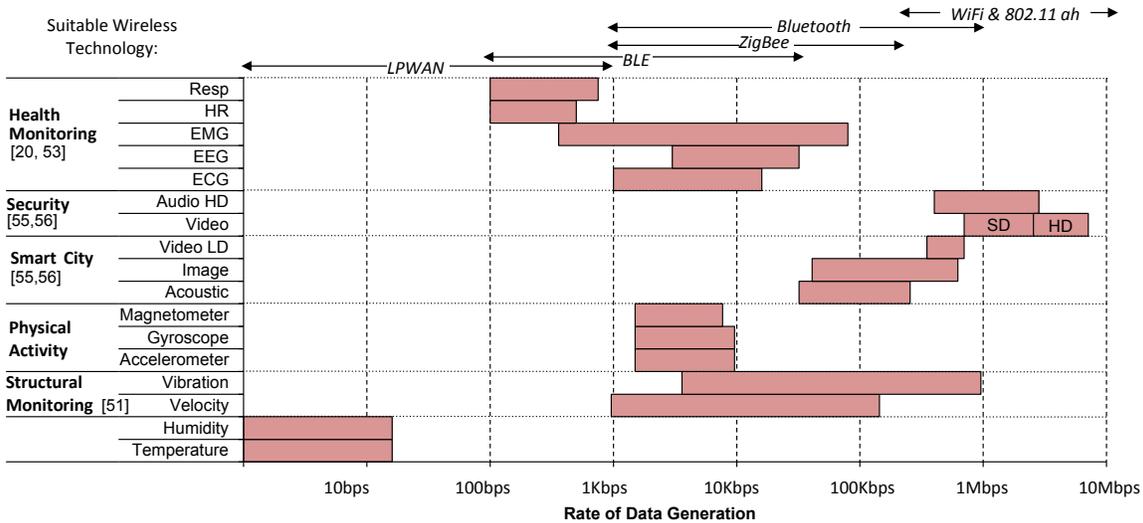


Figure 6: Data rate generation by different sensors in typical IoT applications

- Health Monitoring Sensors:** As mentioned in Section 2.1, healthcare applications of IoT [13, 51] include, but are not limited to:
 - Heart Rate: To detect the heart rate (and consequently heart rate variability).
 - ECG: To measure the electrical activity of the heart which conveys essential information about the status of heart and the function of its muscular contractions [20].
 - EMG: To measure the electrical signal causes by muscular activity [49] for gesture recognition, detection of neuromuscular diseases, etc. [20].
 - EEG: To capture the electrical voltages which represent the brain activity.
 - Blood Pressure
 - Respiration Rate
 - SpO2: The arterial oxygen saturation or the amount of oxygen dissolved in blood.
 - Skin Conductivity: To measure the conductivity of skin to detect psychological or physiological arousal, or the moisture level of the skin [52].
- RF radio modules:** Although RF modules are primarily used for communication, they can be used as localization sensors. The strength of received signal can be used for indoor localization and navigation [53].

5.4 Analysis & Insight

In Figure 6, we illustrate the typical data rate of different IoT sensors in different applications. We derived those numbers based on the typical resolution of captured data and typical sampling rates in different case studies and applications [54, 55].

In Figure 7, we estimated the required processing capability to perform the basic operations and processing tasks on the captured signals of different sensors in different IoT applications. For instance, to process the EEG sensors on the IoT device, we first need to subdivide the signal to different bands (e.g. Delta, Theta, etc.) by filters. Then, different bands are transformed to

frequency domain using Fast Fourier Transform (FFT). We calculated the average number of CPU cycles to perform those operations per sample and then obtained the required frequency to process the data at the derived data rates.

Figure 8 shows the minimum required frequency for a microcontroller to transmit the unprocessed data to the Internet (through the gateway). We obtained those numbers under the assumption that the microcontroller needs to encrypt the data and add error detection codes to it before transmission. We used Advanced Encryption Standard (AES) and cyclic redundancy check (CRC) for this purpose, implement them and measured the average execution time and number of cycles to perform these operations on one byte of captured data. Then, using the derived typical data rate, we estimated the required CPU cycles to transmit the data.

The number of CPU cycles and execution time for the typical operations (e.g. FFT, FIR, AES, CRC) are obtained from our experiments on the Atmel Atmega328 microcontroller which is used in IoT platforms including Arduino boards. Table 4 summarizes the measurements for those operations.

Table 4: Typical data processing operations in IoT applications and their exec. time (per Byte)

Operation	Exe. time [μ s]	# of cycles	Code Size [B]	
FFT (256 p)	18	586	3850	
FIR	1492	23872	1750	
AES	Enc	18.25	292	3666
	Dec	22.75	364	
CRC	3.75	60	500	

6 Conclusions

The emergence of IoT has added a new dimension to the embedded systems by enabling communication between devices and connectivity to the Internet. The existing challenges in designing conventional embedded systems, including energy efficient hardware and software, efficient memory architecture and management, real-time constraints, etc., need to be addressed by con-

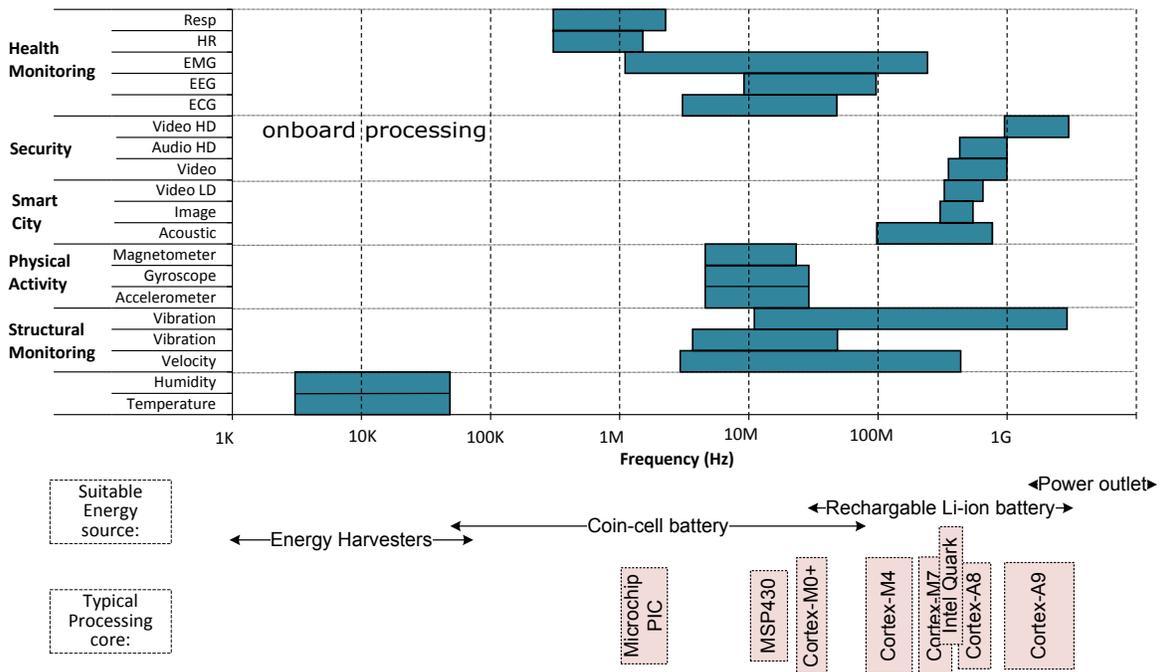


Figure 7: The number of cycles (i.e. required frequency) to fully process the IoT sensors, and their expected power source

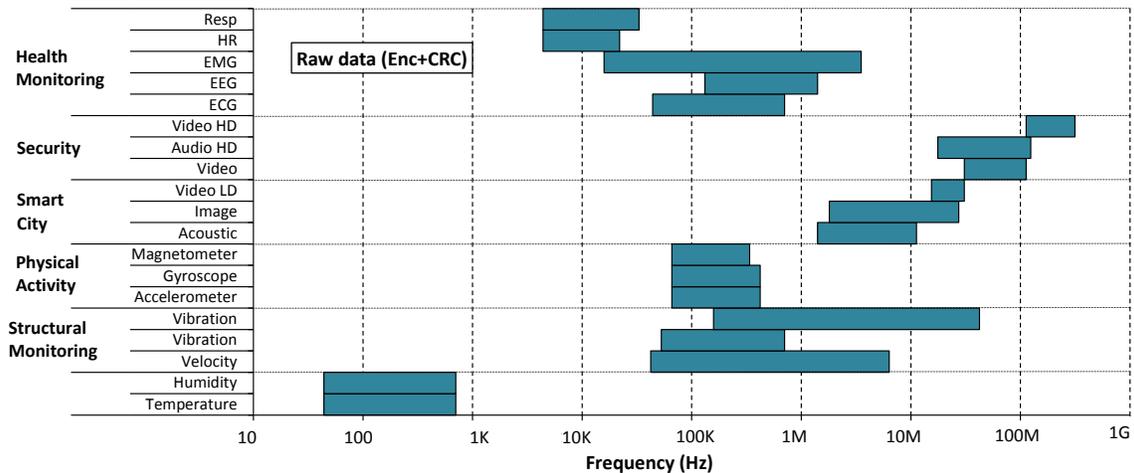


Figure 8: The number of cycles (i.e. required frequency) to encrypt and transmit the captured data from IoT sensors

sidering new properties and requirements of IoT including connectivity. Moreover, new research problems arise due to specific IoT requirements including the connection of IoT devices, large number of devices, coexistence of connected devices with wireless radios in the same frequency band (i.e. interference problem), Big Data, etc.

We presented an overview of the technological enabler of IoT development from the embedded design perspective. We categorized IoT applications and devices based on different criteria and parameters, and highlighted the associated properties with each category. The suitability of different technologies for IoT is investigated, too.

References

- [1] L. Atzori and A. Iera and G. Morabito. The Internet of Things: A survey. *Computer networks*, 54(15), 2010.
- [2] D. Bandyopadhyay and J. Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.
- [3] B. Guo, D. Zhang, Z. Yu, et al. From the Internet of Things to embedded intelligence. *World Wide Web*, 16(4), 2013.
- [4] D. Miorandi and S. Sicari and F. De Pellegrini et al. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.
- [5] I. Lee and K. Lee. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 2015.
- [6] F. Bonomi and R. Milito and J. Zhu et al. Fog computing and its role in the Internet of Things. In *MCC workshop on*

- Mobile cloud computing*, pp. 13–16, 2012.
- [7] J. Gubbi and R. Buyya and S. Marusic et al. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 2013.
 - [8] A. Gluhak, S. Krco, M. Nati, et al. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, 49(11):58–67, 2011.
 - [9] G. Wu, S. Talwar, and K. Johnsson, et al. M2M: From mobile to embedded internet. *IEEE Communications Magazine*, 49(4):36–43, 2011.
 - [10] L. Da Xu and W. He and S. Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.
 - [11] J. Stankovic et al. Research directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1):3–9, 2014.
 - [12] C. Perera and A. Zaslavsky and P. Christen et al. Context aware computing for the Internet of Things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.
 - [13] M. Hassanalieragh, A. Page, T. Soyata, et al. Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. In *International Conference on Services Computing (SCC)*, pp. 285–292, 2015.
 - [14] T. N. Gia, M. J. A.-M. Rahmani, and T. Westerlund, et al. Fog computing in healthcare Internet-of-Things: A case study on ECG feature extraction. In *Int'l Conf. on Computer and Information Technology (CIT)*, pp. 356–363, 2015.
 - [15] M. U. Ahmed, M. Björkman, and A. Causevic, et al. An overview on the internet of things for health monitoring systems. In *IoT Technologies for HealthCare*, 2015.
 - [16] CityPulse: Real-time IoT stream processing and large-scale data analytics for smart city applications. Online www.ict-citypulse.eu.
 - [17] R. Tönjes, P. Barnaghi, M. Ali, et al. Real time iot stream processing and large-scale data analytics for smart city applications. In *European Conference on Networks and Communications (EUCNC)*, 2014.
 - [18] N. Kaur and S. K. Sood. An energy-efficient architecture for the Internet of Things (IoT). *IEEE Systems Journal*, 2015.
 - [19] F. Samie and L. Bauer and J. Henkel. An approximate compressor for wearable biomedical healthcare monitoring systems. In *CODES+ISSS*, pp. 133–142, 2015.
 - [20] S. Benatti, F. Casamassima, B. Milosevic, et al. A versatile embedded platform for EMG acquisition and gesture recognition. *IEEE trans. on biomedical circuits and systems*, 2015.
 - [21] F. Samie and L. Bauer and C.-M. Hsieh et al. Online binding of applications to multiple clock domains in shared fpga-based systems. In *DATE*, pp. 25–30, 2015.
 - [22] S.-Y. Chien, W.-K. Chan, Y.-H. Tseng, et al. Distributed computing in IoT: System-on-a-chip for smart cameras as an example. In *ASP-DAC*, pp. 130–135, 2015.
 - [23] Smartcitizen. Online: <https://smartcitizen.me>.
 - [24] I. Leontiadis and C. Efstratiou and C. Mascolo et al. SenseShare: transforming sensor networks into multi-application sensing infrastructures. In *Wireless Sensor Networks*, 2012.
 - [25] Z. Wang, Y. Liu, Y. Sun, et al. An energy-efficient heterogeneous dual-core processor for Internet of Things. In *ISCAS*, pp. 2301–2304, 2015.
 - [26] S. Kim. Nested game-based computation offloading scheme for mobile cloud IoT systems. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–11, 2015.
 - [27] S. Li and L. Da Xu and X. Wang. Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186, 2013.
 - [28] K. Ma, Y. Zheng, S. Li, et al. Architecture exploration for ambient energy harvesting nonvolatile processors. In *HPCA*, pp. 526–537, 2015.
 - [29] J. Kwong and A. P. Chandrakasan. An energy-efficient biomedical signal processing platform. *IEEE Journal of Solid-State Circuits*, 46(7):1742–1753, 2011.
 - [30] M. Ueki, K. Akeuchi, T. Yamamoto, et al. Low-power embedded ReRAM technology for IoT applications. In *Symposium on VLSI Circuits*, pp. 108–109, 2015.
 - [31] D. Bortolotti, H. Mamaghanian, A. Bartolini, et al. Approximate compressed sensing: ultra-low power biosignal processing via aggressive voltage scaling on a hybrid memory multi-core processor. In *ISLPED*, pp. 45–50, 2014.
 - [32] M. T. Ghoneim and M. M. Hussain. Review on physically flexible nonvolatile memory for internet of everything electronics. *Electronics*, 4(3):424–479, 2015.
 - [33] A. Pande and E. Baik and P. Mohapatra. Efficient health data compression on mobile devices. In *ACM MobiHoc workshop on Pervasive wireless healthcare*, pp. 25–30, 2013.
 - [34] D. Kelf. IoT: A Return to Our Favorite EDA Requirements. Online: www.eetimes.com/author.asp?doc_id=1322009, 2014.
 - [35] B. Zhang, N. Mor, J. Kolb, et al. The cloud is not enough: saving iot from the cloud. In *USENIX Conference on Hot Topics in Cloud Computing*, pp. 21–21, 2015.
 - [36] C. Perera, P. P. Jayaraman, and A. Zaslavsky, et al. Mosden: An Internet of Things middleware for resource constrained mobile devices. In *HICSS*, pp. 1053–1062, 2014.
 - [37] S. K. Datta and C. Bonnet and N. Nikaein. An IoT gateway centric architecture to provide novel m2m services. In *World Forum on Internet of Things (WF-IoT)*, pp. 514–519, 2014.
 - [38] T. Zachariah, N. Klugman, B. Campbell, et al. The Internet of Things has a gateway problem. In *Mobile Computing Systems and Applications (HotMobile)*, pp. 27–32, 2015.
 - [39] A. Al-Fuqaha, M. Guizani, and M. Mohammadi, et al. Internet of things: a survey on enabling technologies, protocols and applications. *IEEE Communications Surveys & Tutorials*, 2015.
 - [40] V. K. Chippa and S. T. Chakradhar and K. Roy et al. Analysis and characterization of inherent application resilience for approximate computing. In *DAC*, 2013.
 - [41] J. Han and M. Orshansky. Approximate computing: An emerging paradigm for energy-efficient design. In *European Test Symposium (ETS)*, pp. 1–6, 2013.
 - [42] P. Singer. Meeting the IoT Design Challenge. Online: <http://semimd.com/blog/2015/11/02/meeting-the-iot-design-challenge/>, November 2015.
 - [43] R. Quinell. ASICs for the IoT Cross the Horizon. Online: www.open-silicon.com/asics-iot-cross-horizon/, June 2015.
 - [44] G. Moretti. Internet of Things (IoT) and EDA. Online: www.chipdesignmag.com/sld/blog/2014/04/08/internet-of-things-iot-and-eda/.
 - [45] G. Qu and L. Yuan. Design THINGS for the Internet of Things - An EDA perspective. In *ICCAD*, pp. 411–416, 2014.
 - [46] B. Bailey. EDA's Clouded Future. Online: semiengineering.com/edas-clouded-future/, June 2015.
 - [47] V. Coskun and B. Ozdenizci and K. Ok. A survey on near field communication (NFC) technology. *Wireless personal communications*, 71(3):2259–2294, 2013.
 - [48] Link Labs. A comprehensive look at low power, wide area networks. Online: <http://info.link-labs.com/lpwan>, 2016.
 - [49] V. Shnayder, B.-r. Chen, and K. Lorincz, et al. Sensor networks for medical care. In *SenSys*, pp. 314–314, 2005.
 - [50] S. Kim. Wireless sensor networks for structural health monitoring. Msc, University of California at Berkeley, 2005.
 - [51] L. Catarinucci, D. De Donno, L. Mainetti, et al. An iot-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2012.
 - [52] S. Coyle, K.-T. Lau, N. Moyna, et al. BIOTEX-Biosensing textiles for personalised healthcare management. *IEEE Trans. on Info. Tech. in Biomedicine*, 14(2):364–370, 2010.
 - [53] F. Palumbo and P. Barsocchi and S. Chessa et al. A stigmergic approach to indoor localization using bluetooth low energy beacons. In *Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2015.
 - [54] Understanding bitrates in video files. Online: help.encoding.com/knowledge-base/article/understanding-bitrates-in-video-files, 2013.
 - [55] G. Siantikos and D. Sgouropoulos and T. Giannakopoulos et al. Fusing multiple audio sensors for acoustic event detection. In *ISPA*, 2015.