Analysis of optimal IoT system development: Component modelling and a view on node assembly cost

Jeltsin Neckebroek

Supervisors: Prof. dr. ir. Sofie Verbrugge, Prof. dr. ir. Didier Colle
Counsellors: Ir. Frederic Vannieuwenborg, Ludwig Stroobant

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Industrial Engineering and Operations Research

Department of Information Technology
Chair: Prof. dr. ir. Daniël De Zutter
Faculty of Engineering and Architecture
Academic year 2015-2016

UNIVERSITEIT GENT

Analysis of optimal IoT system development: Component modelling and a view on node assembly cost

Jeltsin Neckebroek

Supervisors: Prof. dr. ir. Sofie Verbrugge, Prof. dr. ir. Didier Colle
Counsellors: Ir. Frederic Vannieuwenborg, Ludwig Stroobant

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Industrial Engineering and Operations Research

Department of Information Technology
Chair: Prof. dr. ir. Daniël De Zutter
Faculty of Engineering and Architecture
Academic year 2015-2016

UNIVERSITEIT
GENT

Acknowledgements

This thesis would not have been made possible without the assistance of several people. Therefore, I would like to use the beginning of my dissertation to thank them.

Firstly, I would like to thank my supervisors, Prof. dr. ir. Sofie Verbrugge and Prof. Dr. Ir. Didier Colle for offering me this interesting subject to write my masters thesis about.

I also want to thank my thesis advisors, ir. Frederic Vannieuwenborg and ing. Ludwig Stroobant for their open and joyful attitude when evaluating my work which made the creation of this document a true pleasure. I could always count on my advisors, to guide me towards new and interesting trains of thought.

Lastly, I want to thank my parents, for giving me the opportunity to achieve my degree and for being there when I need them the most.

Jeltsin Neckebroek,
May 2016

Analysis of optimal IoT system development: Component modelling and a view on node assembly cost

Jeltsin Neckebroek

Supervisors: Prof. dr. ir. Sofie Verbrugge, Prof. dr. ir. Didier Colle

Counsellors: Ir. Frederic Vannieuwenborg, Ing. Ludwig Stroobant

***Abstract* - The world of Internet of Things (IoT) is growing at fast pace. Both in domestic and in industrial environments, an increasing number of IoT applications can be found. Accompanied with this growing interest in IoT, hardware components (such as sensors, network modules, microprocessors, etc.) become more and more accessible. IoT providers can choose to buy existing, over-the-shelf (OTS) devices or build their own hardware design. Which option is most optimal; Make or Buy? Before making the decision, it is important to analyze the trade-offs and compare the estimated resource costs. The challenge is to develop an IoT system which meets the business requirements, at the optimal financial cost. This thesis will explain an IoT-oriented management methodology, which exists of a series of clear and actionable steps. Starting from the initial business requirements, the reader is guided through the process of developing an IoT solution and estimating the final project cost. This methodology is combined with a building block cost analysis (based on hardware, firmware, assembly and testing) and generates expenditure estimates for both Make and Buy decisions. The effectiveness of the methodologies is proven by applying them to a real-life IoT use-case, situated in the new IBCN Offices.**

***Keywords* – Internet of Things; Project Management Methodology; Cost Estimation; Make-or-Buy**

I. Introduction

The Internet of Things refers to the interconnection of physical objects, by equipping them with sensors, actuators and means to connect to the Internet. Together with the growing number of IoT applications, the accessibility of IoT hardware/software building blocks is growing. In order to create a fully-functioning IoT hardware device (see Figure 1), one can choose to purchase off-the-shelf devices, ready-to-use base-stations combined with sensor modules or to make a proprietary project with separate components from scratch. From that perspective, it is important to have an IoT solution delivery-roadmap which provides feasible guidelines to IoT practitioners on the subject of implementing project strategies. An important part of defining an IoT strategy is to check which choice is economically optimal: 'Make' or 'Buy'. It is possible that purchasing an IoT solution can be favorable in one case, but inadvisable in the other.

II. Objectives

The goal of this Master's thesis is to analyze the development path for a generic IoT project based on a building block methodology and the idea of optimizing device cost at all time. What are the building blocks of an IoT strategy and what are the cost drivers when building or purchasing a device? To answer this question, a generic cost modelling method has to be created which is applicable to a broad range of different IoT projects, but is focused upon IoT characteristics instead of general IT. Using this methodology, an optimal answer should be provided, to the example question: "We would like to develop an IoT device that is able to monitor temperature, movement and acceleration. We need 1.000 devices. There is no WiFi or LAN network available. What technology should we use, which sensors and how much will it cost? Are there alternative solutions and at what quantity are they valid? What would the answer be for 100.000 units?" Note that this problem does not only ask for the total project cost, but for specific technical data as well. Technical and cost-related answers have to be given as a result of the methodology. Based on that, a break-even analysis will show which option ('Make' or 'Buy') is profitable for a predetermined quantity of devices. To show the created management methodology functions in practice, it should be applied to a use-case which will give the formulation of a project-specific IoT strategy and an estimation of the total expenditures as results.

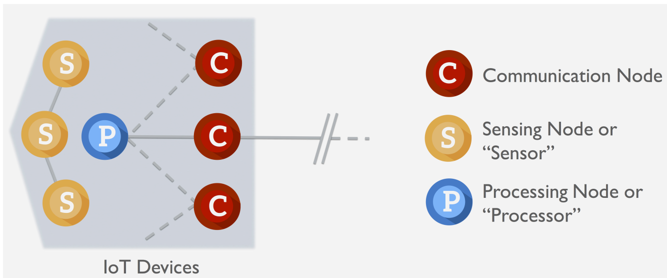


Figure 1 - Node Representation of an IoT Device

III. Management Methodology for IoT

The management of an IoT project, independent of its nature, goal or origin, consists of three stages: Planning, Making and Executing. Each of these phases can be divided into smaller, easy-to-understand sub-tasks (Figure 2). The Planning and Making phases focus on analyzing and converting the business requirements to functional and technical requirements by means of location screening and extraction of the project dimensions. Following these steps, a technology-independent solution proposal is created. This forms the basis for a Proof of Concept (PoC) and is designed before the ‘Make’ or ‘Buy’ question is considered. After performing a market research, listing the different `Buy` alternatives, the Bill of Materials (BOM) of the prototype is created and a first hardware cost estimation can be executed. Based upon the cost building blocks, elaborated in Paragraph IV, a total expected project cost is defined. In this step, it is decided whether the devices will be made or bought. The evolution of the number of users is then forecasted and the relationship with the number of assets and events is defined. Based upon this forecast, a projection of the future cost evolution can be made. Changing volume parameters can shift the preference from the ’Buy’-option towards the ’Make’-option or visa versa. This future growth analysis is followed by setting out a number of milestones concerning project deadlines (production/testing/installation), related to the third phase: Execution. For the ‘Buy’ option, Executing is limited to installing, configuring and maintaining the OTS-bought device. In case of a ‘Make’ decision, the successive steps of Prototyping, Pilot series and Mass-Production have to be followed. Each of these steps is encompassed by continuous evaluation and a feedback loop to the hardware design and future actions steps.

IV. Cost Building Blocks: Hardware, Assembly, Firmware & Testing

In order to obtain a reliable cost estimation for a complete IoT solution proposal, each project has to be divided into 4 cost building blocks (see Figure 3).

Figure 3 - Cost Building Blocks of an IoT Project

Each IoT device is constructed out of one or several components - linked to their proper cost drivers - from the hardware families listed in the thesis. Estimating the hardware cost is based on the sum of the purchasing prices, taking into account volume discounts and services such as shipping, insurance or markups. Assembly costs can be estimated using parametric formulas based on the number of printed circuit boards (PCB), BOM lines, thru-holes, etc. Outsourcing the purchasing of the components results in a 10- 15% markup of the original hardware price. Firmware cost strongly depends on the number of source code lines (KLOC) and is chosen to be estimated using the COCOMO model [1]. The aforementioned expenditures are all of the non-recurring nature, while testing costs are a combination of capital (CAPEX) and operational expenses (OPEX). Testing costs include the operating expense of hiring quality engineers, specialized in hardware and software as well as the cost of buying/hiring testing equipment and paying for certifications and licensing [2]. Technical personnel are required for the maintenance, updating and configuring of the hardware/software on a day-to-day basis. Accumulating the CAPEX and OPEX of these 4 cost building blocks results in an accurate total project cost estimation.

V. Use-Case

The aforementioned IoT management and cost estimation methodologies are applied to a real-life problem situation, situated in the office spaces, meeting rooms and utility zones of the IBCN research building (Technologiepark, Zwijnaarde). The objective of this case study is to setup an instrumentation and measurement campaign to precisely monitor and describe the state and structure of energy usage/comfort readings and to give an overview in time of the consumption of energy and its metadata for the common area of a real office environment and building. Based upon the business requirements, a technically feasible design for the PoC and a prototype BOM are formulated. The total project costs, based on the 4 building blocks, for the ‘Make’ scenario are compared to the one of the ‘Buy’ decision. It can be concluded that it is not economically advisable to produce a proprietary IoT hardware solution under 20.000 energy trackers, which is the single volume parameter in this equation. No break-even point can be found, in the current or future contextual requirements, as seen in Figure 4.

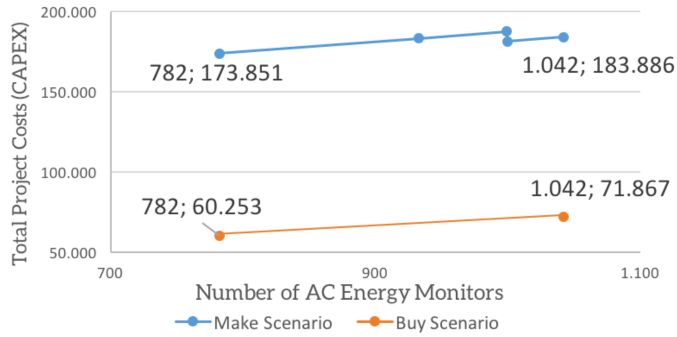


Figure 4 - Total Project Costs (CAPEX) in EUR vs Number of AC Energy Monitors: ’Buy’ versus ’Make’

VI. Conclusion

This thesis has created a unique project methodology, built upon the basis of software development methods but modified to the exclusive needs of IoT. By following the created IoT roadmap, combined with an intermediate technical background of the reader, a project design can be created from scratch, based upon a ’Buy’ or ’Make’ decision. In general, the ’Buy’-decision will remain favorable for low amounts of manufactured units. Due to the overhead costs of firmware and testing, the ’Make’ decision becomes relevant for production numbers over 10k-100k. A widely-applicable and fixed number for this limit cannot be given as it strongly depends on the nature of the IoT project. Therefore, each project should be analyzed using the elaborated methodologies.

References

[1] Putnam, L. and W. Myers, *Measures for Excellence*, Yourdon Press, 1992.

[2] Silicon Labs, *Six Hidden Costs in a 99 Cent Wireless SoC*, http://www.silabs.com/Support_Documents/TechnicalDocs/six-hidden-costs-of-a-99-cent-soc.pdf, 2016

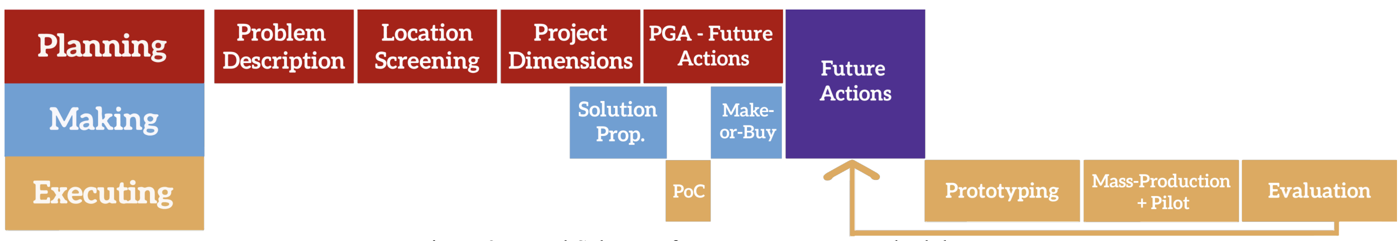


Figure 2 - Total Scheme of IoT Management Methodology

Contents

List of Figures

List of Tables

Abbreviations

AC	Alternating Current
ADC	Analog Digital Converter
ANOVA	Analysis of Variance
BOM	Bill Of Materials
BTLE/BLE	Bluetooth Low Energy
CAPEX	Capital Expenditures
CO2	Carbondioxide
COCOMO	Constructive Cost Model
CT	Current Transformer
DC	Direct Current
DOA	Dead On Arrival
E.g.	Exempli Gratia
EDGE	Enhanced Data Rates for GSM Evolution
EUR	Euro
FTE	Full Time Equivalent
GPRS	General Packet Radio Service
HSDPA	High-Speed Downlink Package Access
HW	Hardware
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol/International Protection Rating
Kbps	Kilobits per second
LAN	Local Area Network
LED	Light Emitting Diode

LOS	Line Of Sight
LTE	Long Term Evolution
mA	Milli-Ampere
M2M	Machine to Machine
MCU	Microcontroller Unit
NFC	Near Frequency Communication
OPEX	Operating Expenditures
OTS	Over The Shelf
PAN	Personal Area Network
PCB	Printed Circuit Board
PoC	Proof of Concept
PoE	Power over Ethernet
PSU	Power Supply Unit
PT	Potential Transformer
RF	Radio Frequency
RFID	Radio Frequency Identification
RTC	Real Time Clock
SCR	Solar Charging Regulator
SD	Secure Digital
SLIM	Software Lifecycle Management
SoC	System on Chip
SW	Software
TE	Thermal Energy
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunications System
UNB	Ultra Narrow Band
USB	Universal Serial Bus
USD	United States Dollar
VHF	Very High Frequency
VOC	Volatile Organic Components
WiFi	Wireless Fidelity
WPT	Wireless Power Transfer

Chapter 1

Context of Master's thesis

1.1 Introduction

The Internet of Things (IoT) refers to the interconnection of physical objects, by equipping them with sensors, actuators and means to connect to the Internet. By technologically enabling this, the goal is to develop new applications and to improve existing ones.

Famous examples of IoT applications include monitoring of personal health through wearables[1], washing machines that enable you to pay per load instead of for the machine, greenhouses that adapt their internal climate to the monitored properties of the crops that grow inside, and stables that adapt feeding and milking schedules to monitored properties of individual cows.[2]

Together with the growing number of IoT applications, the accessibility of IoT hardware/software building blocks grows (sensors, network modules, micro-controllers, batteries, etc.). In order to create a complete IoT hardware device, one can choose to purchase over-the-counter devices, existing, ready-to-use modules or make a proprietary project with separate components. From that perspective, it is from utmost importance to check which choice is economically optimal: 'Make' or 'Buy'.

In order to be able to analyze the costs which the development of an IoT device brings along, a generic project has to be dissected into management building blocks and researched individually. Each building block will be influenced by the 'Make' or 'Buy' decision.

If the 'Make' option is preferred to the 'Buy' option, several questions are asked: Which components have to be chosen? What are their resource costs? How much will it cost to assemble them? Etc.

1.2 Goal of Thesis

The goal of this Master's thesis is to analyze the development path for a generic IoT project based on a building block methodology and the idea of optimizing device cost at all time. This thesis documents the creation of a cost estimation methodology which analyzes the trade-offs between the 'Make' and 'Buy' decisions, by being able to divide any given IoT project into recurring building blocks.

Designing IoT projects by choosing hardware/software components and assembling them comes with a cost. The first way to approach this problem is to list all required technical components which will add up to the total assembly cost of one IoT device (See Section 1.3). A repository of millions of device components could be built which include characteristics such as purchase cost, connectivity requirements and power consumption. However, this thesis will not focus on gathering data from manufacturers or suppliers, as each project will be unique and will have specific demands involving hardware components. Instead, a generic cost modelling method will be created which is applicable to a broad range of different IoT projects, but is focused upon IoT characteristics instead of general IT.

Using this methodology, an optimal answer should be provided, to the example question: 'We would like to develop an IoT device that is able to monitor temperature, movement and acceleration. We need 1.000 devices. There is no WIFI or LAN network available. What technology should we use, which sensors, what will the battery life approximately be and how much will it cost? Are there alternative solutions and at what quantity are they valid? What would the answer be for 100.000 units?'

Note that this problem does not only ask for the total project cost, but for technical data as well. The building blocks created and described in the first part of the thesis divide such problem settings into easy-to-solve sub-problems. Technical and cost-related answers are given as a result of the methodology and a break-even analysis shows which option (make or buy) is profitable for a predetermined quantity of devices. Trade-off remarks are mentioned and best practice advice is given throughout the management methodology.

To show the created management methodology functions in practice, it should be applied to a use-case which will result in forming a project-specific IoT strategy and an estimation of the total expenditures.

1.3 Defining an IoT project

In this thesis, a techno-economical analysis of IoT projects will be executed, consisting of multiple project building blocks. A schematic of a complete IoT network is given by Figure 1.1.

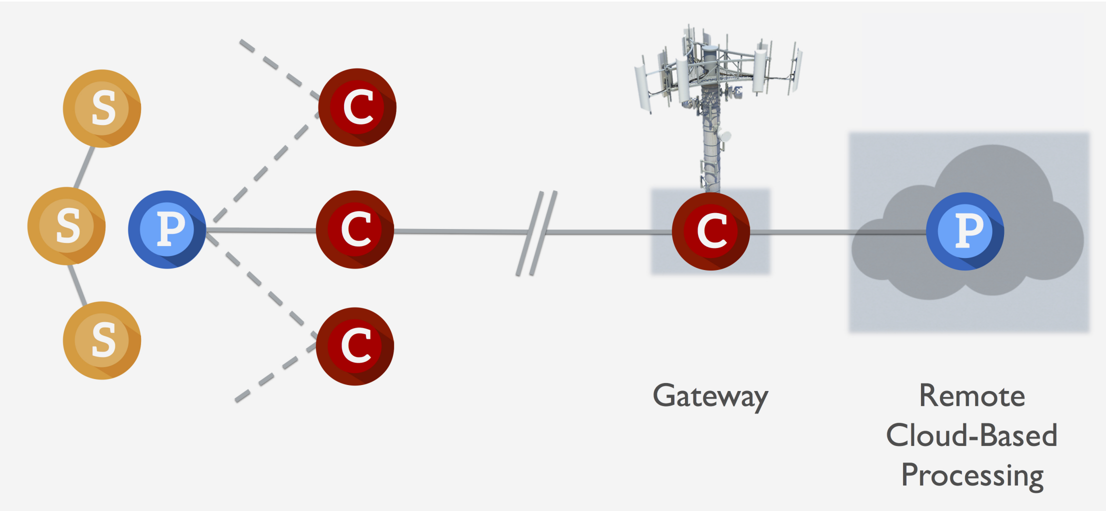


FIGURE 1.1: Building blocks of an IoT network

The focus of this thesis will lay upon the lower level part of the network, which will be referred to as 'IoT device' from now on and can be observed in Figure 1.2.

As it can be observed, one IoT device exists of one or multiple sensor nodes which monitor context parameters. The data logged by these sensors is (partially) processed by a local processor and is then sent to web servers using a predefined communication technology. The process of the data going to the gateway as well as the remote processing in the cloud is beyond the scope of this thesis.

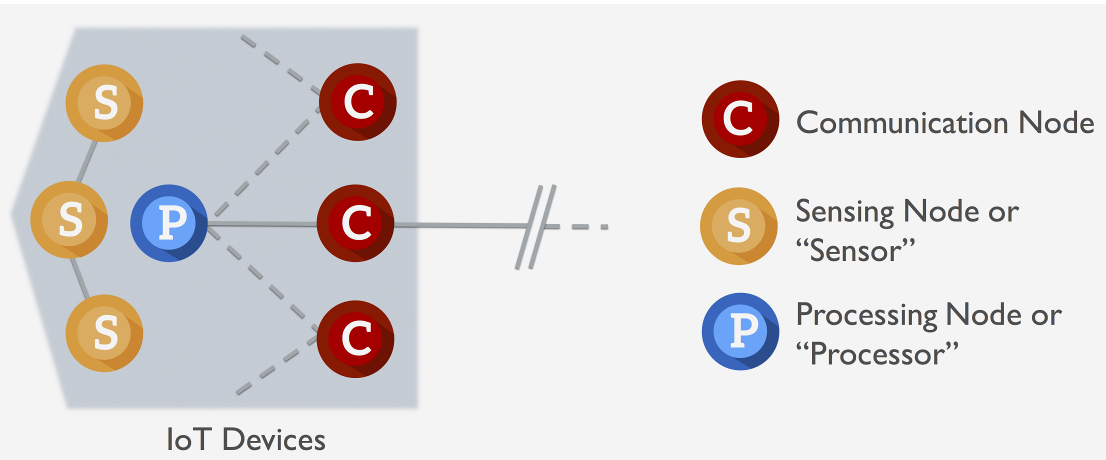


FIGURE 1.2: Focus of thesis - Network schematic

1.4 An introduction to Cost Modelling and Management Methodology

Based on the extensive list of cost drivers for an IoT project, the total estimated cost for the creation of one or multiple devices can be outputted. Cost estimation models, such as the one drafted in this thesis, are mathematical algorithms or parametric equations used to estimate the costs of a product or project. These models function through the input of parameters that describe the attributes of the product or project in question, and possibly physical resource requirements. [3]

These parameters or cost drivers will be listed in Chapters 3 and 4. In order to be able to fully understand the positioning of these cost drivers in the total project scope, a description of the complete management methodology will be given in Chapter 2.

Chapter 2

Management Methodology for IoT Projects

2.1 Introduction

Managing an IoT project, independent of its nature, goal or origin, consists of three stages: Planning, Making and Executing (Figure 2.1).

FIGURE 2.1: 3 Stages of IoT Project Management

In the next paragraphs, each of these phases will be described into detail and divided into smaller, easy-to-understand sub-tasks. The relationship and interaction between the three project management stages can be carried out according to the different methods used in software development. [4]

- Waterfall Model (Picture 2.2)
- Agile Development Model (Picture 2.3)
- V-Model (Picture 2.4)

Which specific approach to choose, depends on the project's time constraints and work circumstances. The classic management approach is the waterfall model, where information requests towards the design team can be considered somewhat 'sluggish'. Once one of the three project stages has been completed sequentially, no reprisal of this level is done. In the waterfall model each step is frozen before the next step is commenced in order to come to a solution, as close to the initial intention as possible.

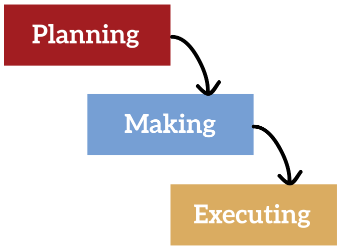


FIGURE 2.2: Waterfall Model

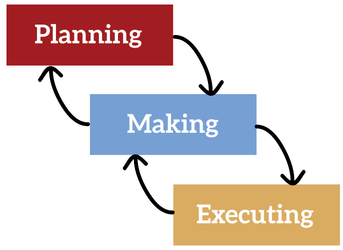


FIGURE 2.3: Agile Model

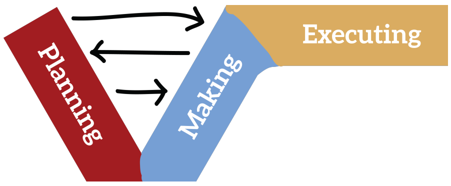


FIGURE 2.4: V-Model

Agile development is more based on making sure the client has a working solution quickly, while continuously improving the project while in use. An important advantage of this model is that it can easily respond to the changing requirements of a project. When developing software, making adjustments to small pieces of code can be considered less cost-demanding than changing existing IoT hardware designs. The further the project approaches completion, adjustments become more expensive (see Chapter 4).

The third model or V-Model can be treated as an extension of the waterfall model. Instead of only moving downstream, the project sub-tasks go upwards again after the ’coding’ phase (for software projects). In IoT hardware development, this bending point will be between the Planning and the Making level. Using this approach, the test user will have a direct involvement in the Planning phase itself.

The main difference between the Waterfall model and the V-model is that, for the Waterfall model, the testing activities are carried out after the development activities are over. On the other hand, in the V-model, testing activities start with the first stage itself. [5] In other words, the Waterfall model is a continuous process, while the V-model is a simultaneous process. Therefore, if the user requirements are clear and fixed from the start, which is preferable, or the magnitude of the project is large, the Waterfall model is the best option. If the requirements of the user are uncertain and keep changing, then the V-model is the better alternative. If the requirements are modified often and time for planning is also limited, the Agile model is recommended. Making changes in the project (software/hardware) in Waterfall model is a difficult and costly task. Often, for smaller systems, it is recommended that one uses the Waterfall model and for the bigger systems, the V-model is used.

IoT hardware development differs from software projects due to the fact that making adaptations to code can be done faster and with a smaller financial effort than modifying hardware designs. Therefore, acquiring clear business requirements from the start benefits the cost optimization in completing an IoT project. The Waterfall method is considered the most appropriate one for IoT project management.

A complete building block scheme of the Planning/Making/Execution methodology can be observed in Figure 2.20, which will be elaborated in the next three sections of this thesis.

2.2 Planning Level

2.2.1 Problem Description

The first step in the planning level can be extracted directly from the client's business model, as a description of the project needs will be given. Together with the client, the problem description can be extended to a technology-independent vision for the IoT solution. This solution will not include jargon, as it has to be understood effortless by all stakeholders. The problem description will list the business requirements of the initiating project, as discussed with the client. An example of a well-defined problem description will be given in Chapter 5.

2.2.2 Business Requirements

From the moment an enterprise decides to initiate an IoT project, the planning stage commences. When starting a project, the client will have a certain idea of his/her needs and wants to receive an estimation of how big the project will be and how much it is going to cost. In dialogue with the design team, the highest level in the project flow - the business requirements - is described.

Business requirements are the critical activities which have to be performed by the project's result in order to meet the premeditated objectives while still remaining solution-independent. Another name to describe business requirements is 'deliverable value'. Clients pay for the satisfaction of their needs as well as for the manner how the solution to their problem is delivered.

Example: the board of directors of DHL (client) decides that they want to get an overview of the real-time trajectories of each company vehicle. This is the value which has to be delivered. The client has needs and is interested in how much it will cost to deliver a solution. The manner of solving the problem is important to the client, as the business requirements have to be met but being given detailed information about the specific solution-technology is redundant.

It has to be noted that business requirements exist and come from the business environment and must be 'discovered', whereas product requirements are human-defined and specified. Business requirements show what the client needs while product requirements show which specifications the technology has to meet to qualify as a project solution. Business requirements are not just high-level but need to be driven down to detail, which is done in the next steps of the Planning Phase. [6]

As mentioned before, listing the business requirements is generally done in the Problem Description phase of the Planning Level. The Planning Level can be split up into 5 major parts. These steps which have to be run through chronologically are illustrated in Figure 2.5.

FIGURE 2.5: Planning Phase - Block Scheme

2.2.3 Location Screening

Following the analysis of the business requirement, the next step is the Location Screening. This is executed together with the client and exists of a tour of the physical location where the IoT devices will be used. Doing such a site survey permits the project team to obtain extra information by means of visual inspection and conducting interviews. This information is crucial as this visit will determine aspects which will have to be implemented in the Project Dimensions (Subsection 2.2.4).

Another effective way of doing this screening, is using documents such as scaled floor plans, electrical schemes or a more detailed description of the business requirements. An example of this technique is given in Chapter 5. Location screening will give answers to following questions[7] [8]:

- Nature of the tracked asset[1]: Stationary of moving?
- Behaviour of asset: Intelligent or dumb?[2]
- Project type: Is there an existing project in use?

[1]Everything of value which can be tracked by IoT devices is considered an asset. E.g. animals, machines, buildings, etc.

[2]Intelligent assets are of electronic nature and programmable

- Environmental conditions
  - Operating temperature?
  - Operating location?
  - Humid/Dusty environment?
- Project size - Project range?
- Outlet power possible?
- Special (contextual) requirements?

The answers to each of these questions are solution-independent and will create a solid basis for the project dimensions to be described. The answers to the project dimensions questions are technology-independent but are no longer solution-independent and influence the total cost of the project. It is important to have unambiguous guidelines, as costs can strongly change depending on the location variables (More information in Subsection 2.3.6).

2.2.4 Project Dimensions

After having executed a location screening, a solution proposal can be designed. This solution description step is part of the 'Making'[3] phase (see Section 2.3) and overlaps with laying out the project dimensions.

In the Project Dimensions step, a checklist of all key aspects which define the project, is created in order to streamline the evaluation of the IoT project dimensions. Before the 'Make-or-Buy' question is even considered, the project dimensions are divided into groups. This checklist will give an overview of the complexity of the project, which will help to sketch an image of the total cost estimation.

Explained in simple terms, it is necessary to know what is needed before it can be determined how much it costs. Table 2.1 gives an list of questions which need to be answered before being able to come up with a solution proposition.

While the location screening step is still solely situated on a purely descriptive level, the project dimensions define the outlines of a concrete solution. More information about the level in solution and technology type definition can be found in Subsection 2.3.2.

2.2.5 Project Growth Analysis

Parallel with analyzing the Project Dimensions, a Solution Proposal is created based on both the Dimensions as assumed Future Growth path, which are the plans for possible expansion of the project. The overlap of both project stages is depicted clearly in Figure 2.6.

FIGURE 2.6: Partial Block Scheme: Overlap of Planning and Making Phase

The Growth Analysis is based on three major parameters, which are all interrelated.

- Amount of Users
- Amount of (to be tracked) Assets
- Amount of Events

[3] The Making phase does not imply that a project cannot be bought. Depending on the answer to the 'Make-or-Buy' question, the project hardware will be bought or created in-house during the Making phase.

Table 2.1: Project Dimensions Checklist

General			
	Project Constraints		
		Timeline	*YEARS*
		Budget	*EURO*
		In-house SW Dev Possible	*YES/NO*
		In-house HW Dev Possible	*YES/NO*
Hardware			
	General		
		Number of Nodes	*NUMBER*
		Integration Complexity	*NEW/RETROFIT*
		Lifetime of Asset	*YEARS*
	Connectivity		
		Required Range	*METERS*
		Required Bandwidth	*Kbps*
		Amount of Messages/Year	*NUMBER*
	Environment		
		Waterproof/Dustproof	*IP CODE[9]*
		Shockproof	*IK CODE[10]*
		Accessibility	*EASY/AV/HARD*
	Processor		
		Amount of Events Processed/Day	*NUMBER*
	Power		
		Power Source	*AC/DC*
		Required Battery Life	*YEARS*
	Sensors		
		Resolution	*DIGITS*
		Connectivity	*BUS-TYPE*[4]
	Circuit		
		In-house Assembly Possible	*YES/NO*
Software			
	Firmware		
		In-house Dev Possible	*YES/NO*
		Updates Mandatory	*YES/NO*
		Update Period	*YEARS*
	Security		
		Crucial	*YES/NO*
		In-House Dev Possible	*YES/NO*

All three of these parameters can be changed separately but it is assumed that the amount of users is the root which influences the rest (see Figure 2.7) or in mathematical terms:

$$\text{Events} = f(\text{Assets}) = f(g(\text{Users}))$$

Example: A certain project requires a group of farmers to track the vitals of their herd of sheep. If the amount of farmers (users) enlarges, more sheep (assets) will have to be tracked which will result in more sent sensor data (events).

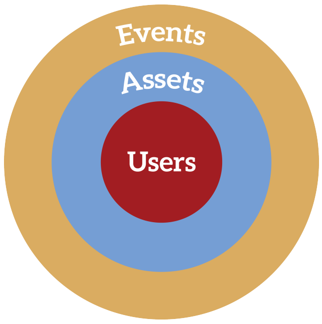


Figure 2.7: Growth Analysis Parameters

A forecast of the number of users is essential as well as the mathematical relationship with the number of assets and events. A prediction for the period up until the end of the first life-cycle (of the most essential hardware) is recommended in order to build a future-proof project.

The next step, Future Actions, is both a part of the Planning as the Making phase (see Figure 2.8). In order to maintain consistency in the methodology road map, the Future Actions is elaborated in Section 2.3, after having explained the Solution Proposal.

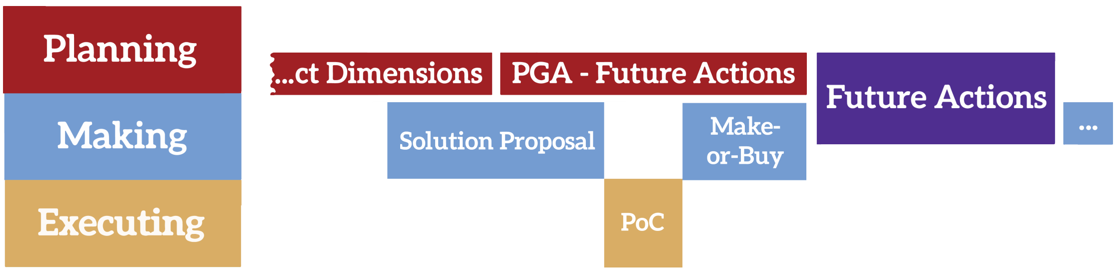


Figure 2.8: Position of the Future Actions step in the methodology

2.3 Making Level

2.3.1 Introduction

Next to analyzing the building blocks of an IoT project, the second main goal of this thesis is to compare the decisions of buying a pre-manufactured IoT device to buying all the components separately and assembling them afterwards. This 'Make-or-Buy' decision will be a paramount question of the Making phase and is to be elaborated carefully in this section.

Before one is capable of answering the 'Make-or-Buy' question, a Solution Proposal is created. A Solution Proposal originates on two different sub-levels: the Functional Requirement Level and the Technical Requirement Level. Both are explained in the subsection below.

2.3.2 Functional Requirements

While in the Planning Level, the business requirements stay solution-independent. The functional requirements which are described in the Making Level, are not. Functional requirements define the inputs, the behavior and outputs of a system, but stay technology-independent. [11] They can be calculations, data manipulation or other specific features which define what a system is supposed to accomplish.

Functional requirements are supported by non-functional/quality requirements. Broadly, functional requirements define how a system is supposed to act and non-functional requirements define how a system shall be.

Example: Demanded results are functional requirements, cost and reliability of the system are non-functional requirements.

2.3.3 Technical Requirements

Generally speaking, Technical requirements are the whole of technical issues which have to be taken into account when successfully wanting to complete a project. E.g. Aspects such as performance, reliability or availability.

Classically, in software projects, technical requirements refer to the programming language or the specific operating systems. In IoT Project Management, the technical requirements will indicate the specific type of hardware (brand of sensor, type of microprocessor, etc.) and firmware used to create the project. Technical requirements are the most narrow as they are no more independent of the solution/technology-type (see Figure 2.9).

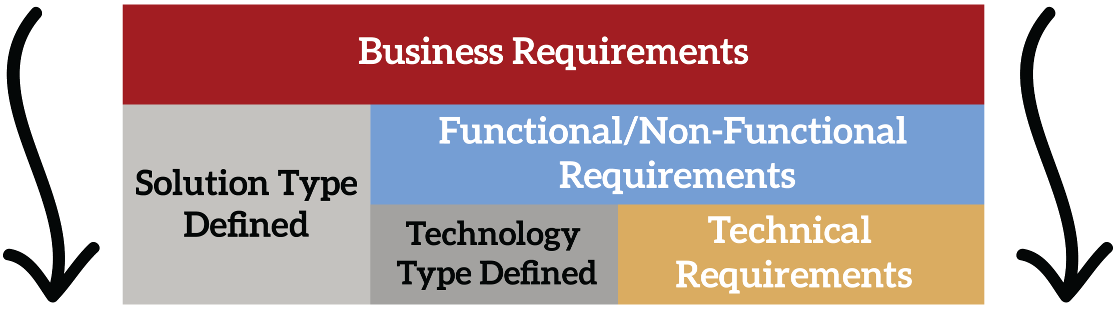


FIGURE 2.9: Hierarchy of the different Project Management Requirements

2.3.4 Solution Proposal

The solution proposal itself can be approached in two different ways: Top-Down and Bottom-Up. For the Top-Down approach, a specific project goal is set and a technical solution is sketched which leads to the design of software and hardware components.

For the Bottom-Up method, these software and hardware components already exist in their final form: sensors, integrated circuits (ICs), firmware, etc. The research question when applying this concept can be formulated as: 'What can be analyzed using the components at hand?'

When composing a solution in reality, a combination of both approaches is used. Hardware platforms, such as Arduino/Raspberry Pi/Intel Edison/Beaglebone/etc[5].. can be used to close the gap between the Top-Down and Bottom-Up methods (See Subsection 2.3.6).

Project solutions are defined, in theory, from the top down, but always use existing hardware components/software libraries, whether they are available off-the-shelf or need to be assembled.

In order to obtain a valid Solution Proposal, the business requirements need to analyzed and the essentials extracted. Before being able to give an answer to the Buy-or-Make question, the next 5 questions need to be answered.

[5]https://www.arduino.cc/; https://www.raspberrypi.org/; http://www.intel.com/content/www/us/en/do-it-yourself/edison.html; http://beagleboard.org/bone

1. What?
    - Which (environmental) properties have to be measured?
    - What has to be put into action?
    - Etc.
2. How?
    - How exact do these measurements have to be?
    - Is the sent data prone to interception?
    - Etc.
3. Where?
    - On how many different locations do these measurements have to be done?
    - Which range does the measurement data have to be sent over to the gateway?
    - Is the measurement done on a distant/outdoor location?
    - Etc.
4. When?
    - What is the project development time span?
    - What is the project life-cycle length?
    - What are the economic conditions?
    - Etc.
5. Who?
    - Who will have to participate in the development/testing/deployment/etc?
    - Is there in-house personnel, experienced in software/hardware/quality engineering?
    - Etc.

The five core questions (What/How/Where/When/Who) will trigger more detailed queries which are specific for each project. When applying this methodology to a use-case, the sub-questions will have to be adapted to the business requirements. The goal of answering this series of questions is to convert the functional requirements to technical requirements in order to come up with a solution suggestion.

The question 'How many different locations do these measurements have to be done on?' will lead to the technical question of 'How many sensor nodes are necessary?'. Similar examples are:

- Which properties have to be measured? → Which type of sensors need to be installed?
- Which physical range is needed? → Which connection technology/radio has to be used?
- How frequent does data have to be sent? → What is the maximum bandwidth of the used connection technology?
- Is the data prone to interception? → Does the data have to be encrypted and to which level?
- Etc.

Stating these technical requirements will lead to the skeleton of the final solution. In the Solution Proposal step, it is not yet the time to start listing the necessary hardware components. Technological constraints such as needed bit rate or sensor accuracy are determined but no decision is made about the model of transceiver, model of sensor, etc.

In order to make a correct choice regarding hardware components, further on in the methodology, detailed information is required. It is recommended to consult data sheets for all proposed peripherals, but an introduction to the key hardware parts of an IoT project is given in Chapter 3. Both the hardware of a project and its cost drivers are listed and analyzed.

2.3.5 Proof of Concept

Defined as the first part in the Executing phase, the Proof of Concept (PoC) sub-step is discussed here in order to retain reading consistency. Based on the detailed technical requirements extracted from the business requirements in the Solution Proposal step, a Proof of Concept solution can be designed.

A PoC is an exercise to test the initial design idea and is focused on experimenting with different technologies. [12] An example of a PoC is testing whether a microprocessor is able to talk to the connectivity radio, using a certain protocol. A PoC should clearly state what it is to be proven and to what degree, in order to be able to feed this feedback into the decision making process.

The most efficient way of creating PoCs is by purchasing base-station hardware and combining them with widely available add-on sensor components. This method is named 'Option 2 of the Buy-decision' (Subsection 2.3.6). Connectivity modules, sensor nodes or other hardware can be interchanged easily to acquire the best compatibility between components.

When technical feasibility validated and potential stumbling blocks are identified, a hardware design is made by a (team of) hardware architect(s). Based on this design, the 'Bill of Materials' (BOM) is created which enlists the needed components in order to create a prototype. These components are based on the functionality of the PoC but will not be the same, as dedicated ICs are used for further development, in combination with modules and/or SoCs (System-on-Chip).

A PoC should always be manufactured, before asking the 'Make-or-Buy' question as it is an ideal exercise to experience where the pitfalls are situated when choosing to manufacture a device autonomously. After having created a technical design for the PoC, the PoC BOM is made. This BOM will differ from the final list of components, as costs for components will go down when increasing volume numbers.

2.3.6 Make-or-Buy

Based on the result of the Solution Proposal step and completing the PoC step, an important decision has to be made: 'Make or Buy'. In order to be able to choose the best option from an economical point of view, both the optimal Make-solution and the optimal Buy-solution have to be compared to one another.

In certain circumstances, it can be the case that one of both decisions is impossible and the project team is forced to choose for Buy or Make, without doing the comparison. E.g.: The needs described in the business requirements cannot be solved by a (single) device currently on the market.

The 'Make-or-Buy' decision is to be made, after first having answered the technical requirement list of questions, based on following criteria:

1. Does a device, which fulfills these needs, already exist on the market and is it in line of the budget?
2. Which kind of hardware is required: standard or custom?
3. Which kind of software is required: pre-integrated or custom?
4. Is it cheaper to make or buy such a device, depending on the quantity of the nodes?

As seen, taking this decision is based on a multitude of factors. Choosing between 'Make'or 'Buy' is not always an issue of budget constraints alone. Contextual requirements (Figure 2.10) influence the choice as not only the goal of the project is crucial but the environment in which the solution is operating as well. The context in which a project is executed is never uniform and hardware should be modified to these requirements, which will have an impact on the total cost of a project.

Example: Developing an IoT project which measures the brightness in a hospital environment requires the same type of sensors (accuracy) and actuators as in an oil industry environment. In retrospect, the functional environment in the latter situation is prone to ignitable concentrations of flammable gases. Replacing all ordinary switches with insulated explosion-proof ones will result in a significant higher cost, due to contextual requirements.

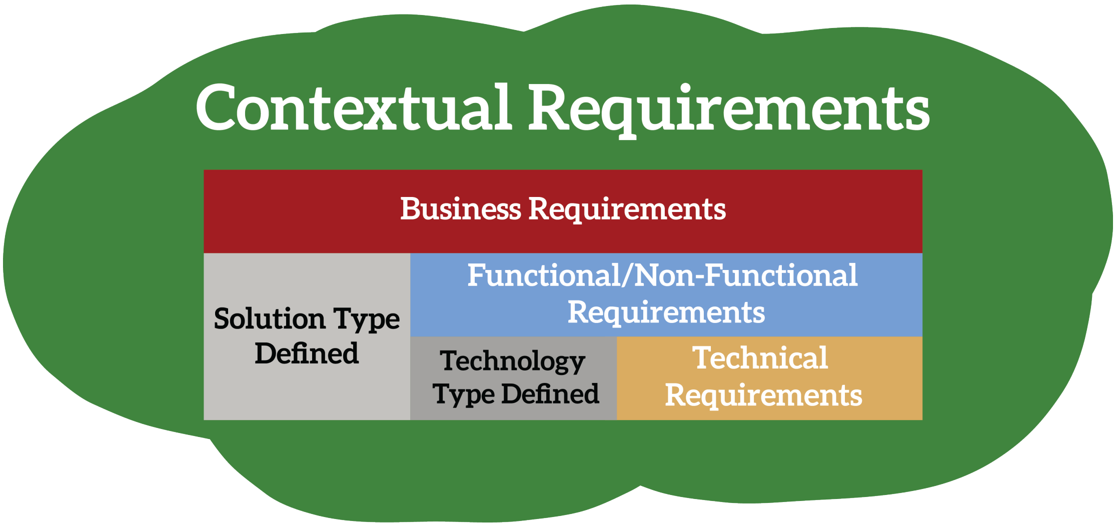


FIGURE 2.10: Project Management situation of Contextual Requirements

If the solution proposal complies with the business and contextual requirements (as well as with corporate policies and guidelines), a financial assessment can be carried out. Note that the scope of this thesis is restricted to solely the purchasing of the hardware (-components) and assembly. Operational costs such as IoT cloud middleware subscriptions, wireless communication contracts with MVNOs (Mobile Virtual Network Operators) or energy consumption itself are not taken into account.

2.3.6.1 Buy Option

In order to compare the costs between the Make and Buy decision, it has to be determined which parts of a project result in financial expenditures. Comparing these expenses in both cases leads to choosing the economical optimal solution.

Each IoT project's total cost can be divided into 4 major parts which is illustrated by Figure 2.11.

FIGURE 2.11: Cost Building Blocks

Each of these 'cost building blocks' are discussed into detail in Chapters 3 and 4. If the 'Buy'-decision would be the optimal choice from an economical point of view - which has to be determined by comparing total project costs - device choices are limited to what the market has to offer.

The scenario of buying a ready-made IoT device can split up into two cases:

- Buying a commercial dedicated, stand-alone device
- Buying a base station and ready-made extensions

While the first option is truly of the pure 'Buy' category, the second one can be considered as slightly leaning towards 'Make'. Ready-made components are still being bought but they have to be assembled later-on, although with a smaller amount of effort when compared to purely 'Make'.

When selecting the first option, the hardware costs are limited to purchasing the device, off-the-shelf (OTS). An extensive market research of all available devices, based on customer reviews, is recommended in order to find the best solution in terms of cost and quality. The purchase price of an OTS IoT device will generally be higher than the sum of its hardware components, due to a series of reasons.

First of all, the assembly, labour and development (testing/prototype) costs are accounted for in the purchasing price, as this is done by the producing company. Customarily, firmware is pre-integrated and also developed by the producer. Again, software development costs lead to a higher purchasing price.

Example: In order to control the temperature in a domestic environment, remotely and with the least amount of effort, an electronic, programmable and self-learning Wi-Fi-enabled thermostat has to bought/built. When looking for an OTS solution,

Table 2.2: Cost Comparison - Buy Choice - No Firmware/Assembly

Buy (OTS)	**Buy (Basestation + Extensions)**
Nest Thermostat - 250 EUR	Raspberry PI Model B+ - 30 EUR
	Watterott RPi Touch Display - 30 EUR
	WIFI Adapter - 8 EUR
	PIR - 9 EUR
	DHT22 - 7 EUR
250 EUR	**84 EUR**

a Nest®(Figure 2.12) Labs Thermostat can be bought, which suffice to the aforementioned needs.

Figure 2.12: Nest®Labs Thermostat
[13]

The second option is to buy a base station as well as ready-made extensions (sensors and actuators) and combining these hardware elements. Both types of components are available OTS but need to be assembled hardware-wise and the extensions have to be integrated with the base station. In comparison to buying stand-alone devices, there is a limited amount of labour/assembly which have to be accounted for.

Example: A programmable, self-learning thermostat device has to be designed using a base-station and ready-made extensions. For this example, a Raspberry PI Model B+ is chosen in combination with a Watterott RPi Touch Display (graphic interface and interaction), a generic WIFI Adapter (connecting to network), a PIR sensor (Passive Infrared Sensor to detect motion) and a DHT22 Temperature/Humidity Sensor. Total purchase costs of this hardware design will be lower than the Nest Thermostat (see Table 2.2), when not accounting for firmware or assembly costs. An extensive case-study and representable example will be given in Chapter 5.

Usually, software to integrate extensions with commercially available base stations is ready-made. It is however possible that this software is not freeware, that customization is needed or that (partial) development is required. After the assembly and integration of the prototype, testing has to be done (See Section 4.4).

Discussions with founders and employees of Productize [6] and Actility[7] made it clear that Option 2 of the Buy-decision is not used in practice, for high production numbers. The base stations with add-ons is only used to develop a viable PoC (Proof of Concept), before assembling prototypes. A PoC shows a (partially) functional design which is used as demonstration. A PoC is therefore a prototype that is designed to determine feasibility, but does not represent deliverables. [12]

The reason Option 2 is not used is that base stations [8] are considered as not reliable enough to work uninterrupted. Technical hiccups such as necessary hard resets or re-flashing of the memory occur more frequently than accepted. Therefor, it is recommended to make a PoC with these devices first and then export the needed functionality to a dedicated hardware design (Subsection 2.3.5). A migration from Buy to Make is then made.

2.3.6.2 Make Option

If the 'Make'-decision is shown to be in favor economically, a wider range of hardware choices can be made. Again, two cases can be described:

- In-House
- Outsourced

These two options are not particular mutually exclusive. It is possible to choose a hybrid option where parts of the 'Make'-process is done in-house and others are outsourced.

For both the first and second option, a new BOM has to be created. In order to obtain this list, it is important to have followed the 'business/(non-)functional/technical requirements methodology' mentioned before meticulously, in order to choose the right components for the problem setting.

This BOM will differ from the one in the PoC step as any base-stations or add-ons are swapped with more low-level type hardware (System-on-Chips, etc.).

After purchasing the hardware, assembly has to be done. It is of utmost importance that the assembly, firmware development and testing of the prototype is done by adequate personnel. If the in-house team is not suited for any of these tasks, it can be decided to outsource them.

A second reason to turn to outsourcing is due to financial reasons. If the total cost of outsourcing a job is lower than the cost of paying in-house personnel (product of

[6] www.productize.be
[7] www.actility.com/en
[8] E.g. Arduino/Raspberry Pi/Intel Edison/Beaglebone/Etc

individual wage per time unit, time spent and number of people). A more into detail analysis accompanied by best-practice numerical data can be found in Chapter 4.

When having chosen for the 'Make'-decision, the work involving assembly, firmware development and testing will be considerably more extensive when compared to the 'Buy'-decision and this will result in extra costs. However, the initial purchase price of the hardware will be lower as only the loose components have to be purchased.

In order to decide between the 'Make' and 'Buy' options, a comparison has to be made based on the financial costs of the four building blocks which are represented in Figure 2.13. Next to the budget constraints of the project, it also has to be repeated that the executing team needs to possess the required skills for each of the task in order to make devices in-house. If not the case, outsourcing is required. If both outsourcing and in-house manufacturing (or hybrid) appears to be more costly, buying is advised.

The crucial factor which influences the total cost of a project, in a non-linear way, is the amount of devices needed. It is recommended to contact multiple hardware suppliers in order to obtain competitive offers for different amounts.

	Buy		Make	
	Stand-alone Off-the-Shelf	Base Station + Extensions	In-House	Outsourced
Hardware	Purchasing Cost	Purchasing Cost	Purchasing Cost	Purchasing Cost
Labour/Assembly	Minimal	Limited	Extensive	Extensive
Firmware	Pre-Integrated	Pre-Int./Custom	Custom	Custom
Testing/Prototype	N.A.	Limited	Extensive	Extensive

FIGURE 2.13: Make-or-Buy decision: Building Blocks

A good example of analyzing the Make-or-Buy question is elaborated in a white paper, drafted by Silicon Labs® [14]. In this document[9], a break-even analysis between using a wireless SoC and a wireless module for a hardware project (Figure 2.14) is made and the hidden costs of using a SoC are revealed. The difference between the SoC and the module is not only the antenna design. The module exists of a fully-characterized PCB (with the SoC) with RF optimization and shielding, timing components (crystals), regulatory approvals, and standards certifications.

When wanting to implement radio-communication into an IoT project, a wireless component has to be integrated onto the main circuit board. The hardware design team

[9]The Make-or-Buy decision can be applied to the entire IoT device or to a single (or limited number of) hardware component(s). This example focuses solely on the radio hardware.

can choose to go with a wireless SoC on the product PCB. It is smaller and cheaper in purchase cost than a wireless module, which is the second option, but the hardware design can bring along hidden costs.

Using a wireless module with the SoC of the first option inside will be larger and more expensive when purchasing. Yet, a majority of the design is already done including a fully-characterized PCB with RF optimization and antenna layout, shielding, timing components, external BOM, regulatory approvals, and standards certifications.

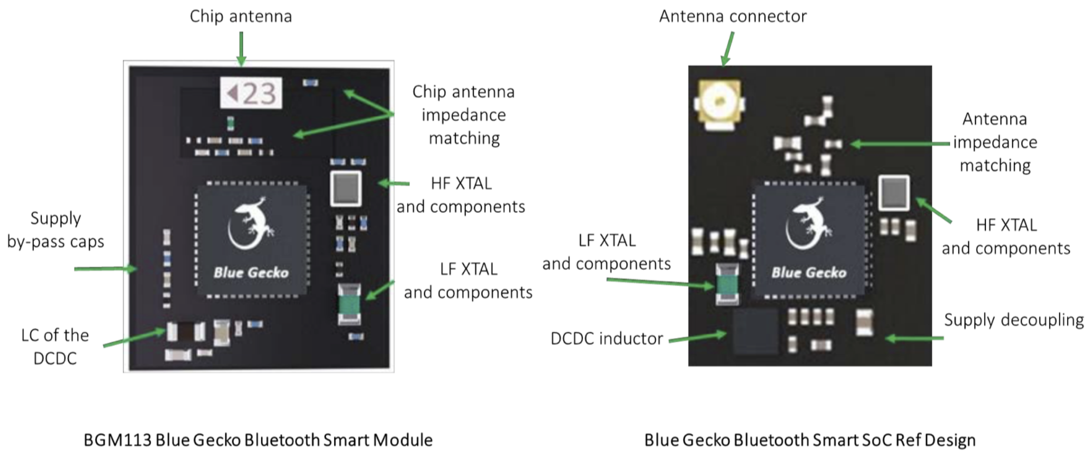


FIGURE 2.14: Example of a Wireless Module vs a Wireless SoC Layout [14]

The key question in this comparison is which is the easiest and most cost effective option? The answer depends on :

- Product (Size and Performance)
- Designer
- Time to Market
- Volume

A break-even analysis (Figure 2.15) is executed in the white paper for a varying volume of units (10.000 - 300.000). From the Silicon Labs Research paper[14], it can be concluded that implementing RF BT SoCs becomes cheaper than using modules when the volume number surpasses 200.000 - 300.000. Under this threshold, buying is considered cheaper than making. Also, using modules will remove unknown risks (hard to quantify in money/time) of designing with a wireless SoC. Using an SoC for a high quantity of devices is not always justified as hidden costs will arise.

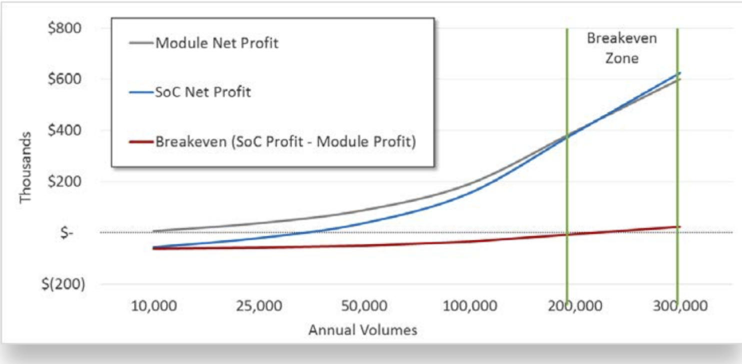


FIGURE 2.15: Break-even Example for using a Wireless Module versus Wireless SoC [14]

A brief overview of these hidden costs mentioned in the white paper is listed below. This list [10] can be generalized and extrapolated for examples other than wireless components, as will be demonstrated in Chapter 5.

- Engineering : Base Salary Engineer + Overhead Costs (33%)
- Equipment : Rent/Buy equipment, software and facilities
- Board Lay-Out : Tweaking board to real-life conditions
- Regulations : Licensing/Approvals take time and money
- Time-to-Market : Each day the product is not on the market is a day of lost revenue
- Supply Management : Sourcing single module is simpler than sourcing BOM to fit SoC. Independence of suppliers is important to guarantee stable production.

Discussions with the founders and employees of Productize and Actility confirm the findings in this aforementioned white paper. If decided to choose for an SoC instead of a module for 1 or several components, a specialized engineer (1 full-time equivalent (FTE)) has to be employed for each field of expertise.

According to Glassdoor.com, The salary for a Belgian RF Engineer is 46-50K EUR/year[11], which does not account for overhead costs such as office space, fringe benefits, etc. A basic rule of thumb is to add an extra 33% on top of the salary, which costs the company 61-67K EUR/year.

[10] A detailed explanation and numerical data for each of these six subjects can be found in the white paper[14]. This numerical data will be used in the case study (Chapter 5).

[11] https://nl.glassdoor.be/Salarissen/rf-engineer-salarissen-SRCHKO011SDAS.htm

It has to be kept in mind that, for example, if the choice is made to integrate SoCs for both wireless communication as processing unit, two separate engineers (2 FTE) have to be hired. Another remark the people at Productize and Actility have, is that an engineer is not laid off after having created a hardware design. Engineers will have to service the deployed IoT installation, full-time or part-time, as duty continuity has to be ensured. This will result in a recurrent cost (salary) when having chosen to go with the Make-decision.

Note: The integration of an RF SoC on PCB exists of optimizing antenna layout, shielding and timing components, so no interference of signals occurs. Each proprietary RF design needs regulatory approvals and standards certifications, which are already included when buying modules. This is the reason modules are generally more expensive and larger than SoCs.

2.3.7 Future Actions

After having made a decision involving the Make or Buy decision of the devices, several other milestones need to be defined in order to be able to transgress towards the next phase. The constraints set in this step will affect the decision domain in the Executing phase. Answers to following questions need to be defined:

- Which hardware supplier?
- When will the initial prototype be finished? (only for Make-decision)
- When will the testing commence/end? (only for Make-decision)
- When will the final device production start? (only for Make-decision)
- When will the device be installed and fully functional?

It is clear that, in case of the Buy-decision, the steps following the Solution Proposal will be significantly less extensive than having chosen for the Make option.

In the Future Actions step, a projection of cost evolution can be made, based on the current cost estimations and the expected growth in users, assets and/or events. Changing volume parameters can shift the preference from the 'Buy'-option towards the 'Make'-option or the other way around.

It is important to have an unambiguous and clear schedule for the entire team, before the actual Executing phase starts. In the next section, the third and final phase of the IoT project management methodology, created in this thesis, will be elaborated.

A discussion can held about which parts of the building blocks, mentioned in the next section, could still be part of the Making phase. It is more important to ensure that each step of the methodology has been followed in practice instead of categorizing each of them from a theoretical point of view.

2.4 Executing

In the third and final phase of the IoT project management scheme (Step 3 in Figure 2.1), the decisions made in the Planning and Making phases will directly influence the action range.

After having made forecasts and assumptions involving the project, the execution phase can commence. If the project team has run through the former two phases meticulously, the final steps should be limited in time and costs.

It is recommended to spend a good amount of time on the Planning and Making Phase as these are not as cost intensive as the Execution Phase and they can significantly reduce the costs of the latter phase (1-10-100 rule - Figure 4.2).

Just as the Planning and Making Phases, the Execution Phase can be divided into a number of sub-steps, following each other in series or being able to be executed in parallel. To create clarity in sketching the building blocks of the methodology, the Execution Phase is divided into 4 steps:

1. PoC (already discussed in Subsection 2.3.5)
2. Prototyping
3. 'Mass'-Production
4. Evaluation

A block diagram of the addition of the third phase is illustrated in Figure 2.16. As it can be observed, the final step in the Executing phase comprises the evaluation of the entire project and its outcomes. Elaboration concerning each step will be provided, respectively in Subsections 2.4.1, 2.4.2 and 2.4.3. The evaluation step is denoted as a single block on the schematic, but it has to be considered as a continuous process. Final project evaluation is executed at the end, yet partial evaluation (fulfillment of business requirements, hardware design, etc.) is done after the PoC, Prototype and Mass-Productions steps.

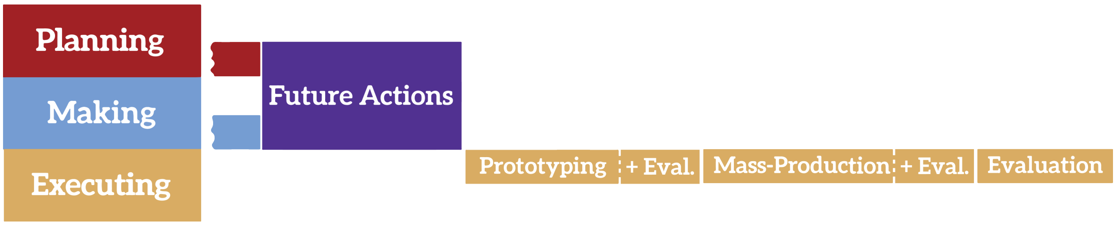


FIGURE 2.16: Addition of the third phase: Executing

2.4.1 Prototyping

2.4.1.1 Introduction

In the next paragraphs, details about the Prototyping step are set down. A prototype is an early model of a product built to test the complete design and to be able to learn from. Prototyping serves to provide specifications for a real, working system rather than a theoretical one.[15] A prototype is configured by certain hardware, which is intended to be used in the Mass-Production step.

Before a prototype can be developed, a more rough version of the device should be created: the Proof of Concept (PoC). The prototyping and mass-production phases are solely reserved for the Make-decision. When having decided to buy the device, these steps are replaced by the installation on asset and deployment according to the manufacturers instructions.

2.4.1.2 Prototype

Both the Prototyping step and the Mass-Production step are very much alike, with the number of constructed devices being the major difference. Elaboration of these steps are focused on the Make-decision, as this would be trivial when choosing to Buy. The Prototyping step can be subdivided into 7 sub-stages, which have to be run through in series, unless explicitly mentioned (Figure 2.17).

1. Ordering Hardware
2. Assembly
3. Developing Firmware
4. On-Asset Preparation
5. Integrating Firmware
6. Installation on Asset
7. Testing

After having created the Prototype BOM, this hardware has to be purchased with the most suitable supplier. The selection process for suppliers has been completed in the

FIGURE 2.17: Breakdown of the Prototyping phase

Making phase. Sufficient components should be ordered to built a number of representable prototype, not necessarily constructed with identical hardware (for testing purposes).

After having acquired access to all the needed hardware, assembly can be initiated. Independent of having chosen to (partially) outsource mass manufacturing or keep it in-house, prototyping should always be executed in-house. If prototyping is kept in-house, total loop time and project time length will decrease significantly. In sub-stage 7, testing will create a feedback loop towards the very first step. Testing should not be seen as a separate step but as a omni-present attitude towards device development (See Section 4.4).

Development of the firmware is placed as the second sub-stage in the Prototyping phase, but can be done parallel with steps 1 to 4. The firmware development team has to ensure the code is finished when integration must commence. The On-Asset preparation can also be done parallel with the before mentioned action, but has to be completed before the device has to be installed in its functional environment. Depending on the contextual requirements, the testing sub-stage can encompass a small or significant amount of time in the execution phase. Testing, for both the Prototyping as Mass-production step, also includes the acquiring of certifications for the used hardware and production processes.

Example: A wearable heart rate monitor is an IoT device which can function in a medical context (hospital) and will directly affects the interpretation of personal health. Testing should be executed more extensively than for the same device which is only being used in a casual manner (fitness tracker).

Certifications such as the European Conformity (CE-marking) are mandatory markings for products sold within the European Economic Area since 1985 (FCC in USA). If the product is not meant to be sold, this certification is not required, as it is used under own risk policy. If licensed radio frequencies (Example: BTLE) are used for data transfer, a design qualification has to be executed by the licensing company and an RF certification fee has to be paid.

In order to finish the project in the least amount of time possible, basic project management rules apply: it is recommended to clearly state deadlines based on the 7-step plan described earlier, for both the Prototyping as the Mass-Production stages.

2.4.2 Mass-Production

2.4.2.1 Introduction

After having completed the manufacturing of the prototype, which has to be approved in the Testing sub-stage, mass-production can be initiated. The Prototyping and Mass-Production steps are interconnected by a cost-cutting phase. The prototype is analyzed for passive components which do not contribute to the working of the device. The final design does not include these redundant parts and hardware as well as assembly costs will drop (assembly costs go up for an increasing number of parts).

Mass-production refers to the stage where production numbers are expanded in order to satisfy the client demand. These production numbers do not imply high lower limits, as the name could suggest.

2.4.2.2 Pilot

Before starting the mass-production of the final product and pushing it to the entire client base, a pilot series is distributed. A pilot (or beta) uses the full production system and tests it against a subset of the general intended audience.[12] The reason for doing a pilot is to get a better understanding of how the product will be used in the field and how to refine the product.

Also, it is possible that there are open-ended questions about scalability that only a live audience using the product can answer. The feedback collected from the pilot is used to implement refinements into a general release. These improvements should be minor and the final product will not differ significantly from the final beta device.

2.4.2.3 Final Products

It is possible that several prototypes were designed and tested, to assess the fitness of use of all of them. When having chosen the most adequate solution, using the pilot series, the mass-produced devices will be copies of the final prototype. A series of steps can be defined which will be similar to the Prototyping step, yet more concise (Figure 2.18).

1. Ordering Hardware
2. Assembly
3. Firmware Adjustments
4. On-Asset Preparation
5. Integrating Firmware
6. Installation on Asset
7. Testing

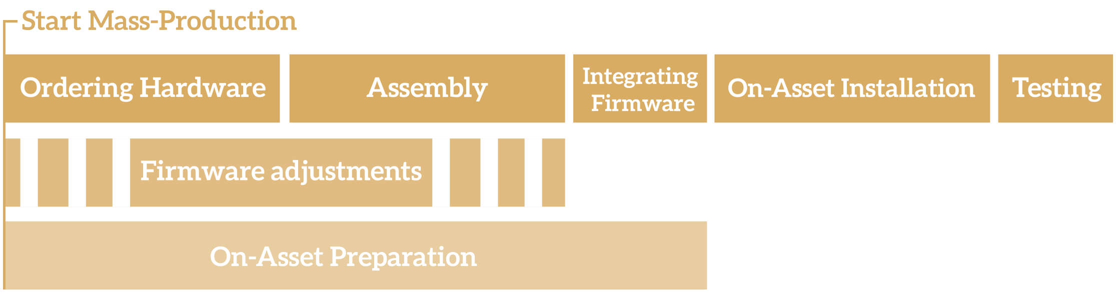


FIGURE 2.18: Breakdown of the Mass-Production phase

Numerous hardware components, similar to the ones found in the final prototype, have to be ordered and acquired. In the mass-production stage, assembly can be executed both in-house or outsourced. Outsourcing enlarges the duration of the feedback loop, but the number of loops will be reduced to a minimum due to earlier extensive testing of the prototype.

Development of the firmware is absent in the mass-production phase, yet it is possible that (minor) tweaks need to be applied to the software, due to scale expansion (more devices working together). This process can be done in parallel as the deadline is defined by the moment that integration of the firmware initiates. The on-asset preparation, which also can be done in parallel, will be identical to the necessary ones executed in the prototype phase.

Integrating the firmware is usually done before installation on the asset. It is however possible, for some devices, to modify or upgrade on-board software using the connectivity radio to transfer data. Integrating firmware is recommended to be done in batch, as this will reduce project time spent. The final testing step is far more limited than in the prototype phase. The hardware design has been approved already, but each final device should be tested for component flaws. An overview of the Prototyping and Mass-Production steps is shown in Figure 2.19.

2.4.3 Evaluation

The final sub-step of the IoT Project Management Methodology covers the evaluation of the project outcome. Evaluation can be defined as using systematic methods in order

FIGURE 2.19: Prototyping and Mass-Production steps

to collect, analyze and use information to answer basic questions about the efficiency and efficacy of a created solution.

Evaluation can and should be done continuously throughout the process, during different stages of the project. On the block diagram, evaluating is depicted as the final step, but after having followed the methodology, 2 types of evaluation have already been described implicitly: 'ex-ante evaluation' and 'mid term evaluation'.

The ex-ante evaluation describes the starting position of the project in order to measure the distance travelled. The mid term evaluation is the action of formally measuring to see if the original environment has changed in a way which impacts on the relevance of the original objectives. Objectives can be reviewed and timescales can be re-negotiated, based on whether the project is on target in terms of its projected outputs. [16]

The ex-post evaluation is final sub-step in the block scheme and is executed after the project is completed. It includes a list of the project's processes and outputs and an analysis of the project's impact on the initial problem settings. Based on this evaluation, the Future Actions block can be influenced, which can be illustrated by a feedback loop.

Future actions are part of the Planning and Making phase and are described before prototyping commences, but they can be adjusted, based on the ex-post evaluation, in order to improve the next generation or production batch of devices. A complete overview of the methodology building blocks is given in Figure 2.20. [12]

2.5 Conclusion

Chapter 2 has outlined a clear path for every IoT practitioner which wants to design an IoT project from the very beginning. The main purpose is to deliver actionable instructions to project teams on how to implement an IoT project creation strategy.

The created methodology look at both the project as product level and is based upon software development methods as well as on the advice and best practices, learned from existing IoT agencies.

[12] A more detailed elaboration about project testing and the related costs can be found in Section 4.4.

Creating the building blocks of this methodology is the result of an iterative process. The theory is tested upon a real-life case-study (Chapter 5) and adjusted/extended where it seemed necessary.

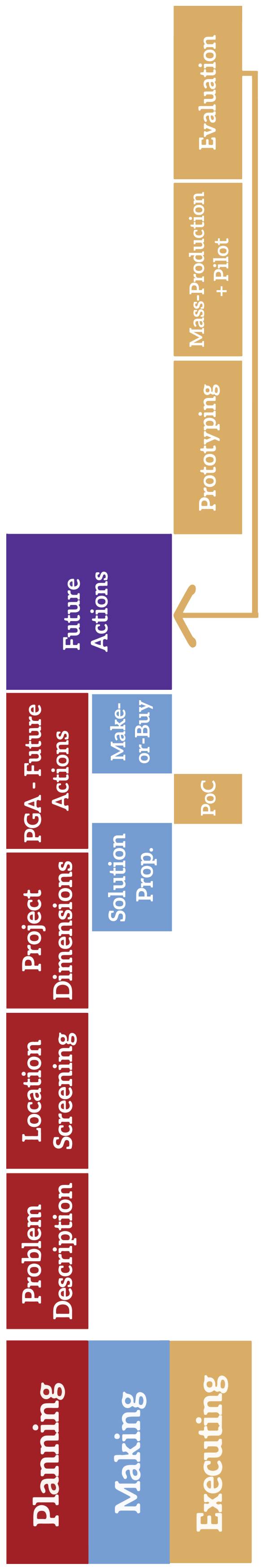


FIGURE 2.20: Total Overview of the Management Methodology for IoT Projects (see Section 4.4 for elaboration about the Testing Phase)

Chapter 3

Hardware Analysis of an IoT Project

3.1 Introduction

In this Chapter, a qualitative description of all hardware components, used in an IoT project is given in order to obtain an estimation of the cost. Next to hardware, an IoT project has three other main cost drivers, as depicted on Figure 3.1.

In this Chapter, the focus will lay upon the analysis of components of a generic project with the goal of dissecting an IoT device and enlisting the hardware part families.

FIGURE 3.1: Cost Building Blocks: Hardware

Each IoT device is constructed out of one or several hardware components, observed in Figure 3.2. Each of these blocks are thoroughly discussed in the next upcoming chapters.

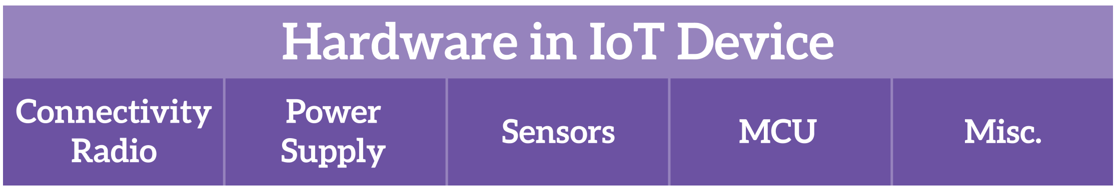


FIGURE 3.2: Hardware Building Blocks

3.2 Network Technologies

3.2.1 Introduction

The Internet of Things or Industrial Internet will not only include new devices such as wearables or specially designed sensors, but will also implement systems which are already installed and are currently operating outside of the 'IoT Cloud'. The creation of new projects or implementation of existing hardware to connect to the IoT cloud[1] will bring along a certain cost for each design. These costs will depend on the fact whether or not the interconnected devices are IP-based and need the ability to connect specific network technologies with the accompanying protocol stack.

Two major challenges in designing IoT projects are scale and connectivity. The IoT is expected to encode 50 to 100 trillion objects and should be able to follow the movement of those objects. Human beings in surveyed urban environments are each surrounded by 1.000 to 5.000 trackable objects. [17]

Each device connected to the Internet must be assigned to a unique IP address, which offers the possibility of identifying and locating each node[2]. Observing the exponential growth of the Internet in the 1990s and 2000s, it was made clear that the original limitation of 4.294.967.296 (2^{32}) addresses, imposed by the original IPv4 protocol, would be insufficient to connect the growing number of devices in the future. Using 128-bit IPv6, a total of 2^{128} unique addresses can be defined.

Connecting IoT nodes can have a static or moving character, depending on the goal of the project. These nodes will always have to be connected to the Internet using a certain type of connection technology. Several existing technologies are well-established and have the potential of fulfilling the majority of requirements for IoT networks.

Cellular networks excel in simultaneously interconnecting multiple devices, absence of interference, relatively high reliability, long range, and capable to service both low-data rate latency-sensitive and high-data rate applications on the same infrastructure.[19] However, existing cellular technologies have shortcomings which (frequently) rule them out for the emerging flood of IoT projects.

Implementing power-efficient, secure and reliable access of the nodes to the Internet using appropriate technology and taking into account the number of nodes per project will involve a development and implementation cost which appears to be unique for each project. By means of a hardware framework, costs for each connection technology and

[1]The cost of deploying cloud services is beyond the scope of this thesis.

[2]A node is a basic unit used in computer science. When defining nodes on the Internet, a node is anything that has an IP address.[18]

linkage to parameters such as the number of nodes, transmission range, etc. a basic cost prediction for an IoT project can be executed.

For the inter-connectivity of IoT nodes, a bewildering choice of technology options is available. The choice of communication technologies will affect the project's hardware requirements and costs and depends on the application's needed security, range, power and data requirements as well as battery life. A list of several available and usable technologies is given in the following sections.

3.2.2 Cellular

Currently, the most globally used communication technology is the cellular system. As mentioned in the introduction, cellular technologies are able to meet most, yet not all, of the requirements for decent IoT networks. In what follows, a brief elaboration of the different cellular technologies is given. [19]

2G/GPRS/EDGE

2G/GPRS/EDGE is relatively power efficient due to its Time Division Multiple Access (TDMA) [3] nature and narrowband 200 kHz channel bandwidth, relatively low-cost, and very long range especially in its 900 MHz band.

2G is not actively maintained and developed anymore. Therefor, the possibility of re-farming or even re-auctioning the frequency bands should be made available, potentially for IoT technologies.

3G/UMTS/WCDMA/HSDPA

3G/UMTS/WCDMA/HSDPA is power hungry by design, compared to 2G, due to continuous and simultaneous (full duplex) receiving and transmitting using Code Division Multiple Access (CDMA) [4] that has proven to be less power-efficient than TDMA. CDMA is best suitable for symmetric traffic, which is not typical for IoT clients.

It is well-known that the battery-life is characteristically shorter when operating in 3G mode compared to 2G mode, either in idle state or during a low data rate.

[3] TDMA is a channel access method for shared medium networks, which allows several users to share the same frequency channel by dividing the signal into different time slots.[20]

[4] CDMA is an example of multiple access, where several transmitters can send information simultaneously over a single communication channel. CDMA employs spread-spectrum technology and a special coding scheme.[21]

Chinese 3G

Chinese 3G or TD-SCDMA[5] was developed in the Peoples Republic of China as a way to avoid patent and license fees associated with other 3G technologies.

TD-SCDMA uses a narrower channel bandwidth and provides lower data rates than WCDMA, but its time-slotted nature provides better power-efficiency, along with less complexity. Although TD-SCDMA is too power-hungry to cover the most constrained IoT use cases, it could be considered the most suitable existing cellular technology for IoT.

4G/LTE

4G/LTE is more power-efficient than 3G, has reduced complexity thanks to its data-only architecture and its limited backward compatibility with 2G/3G. It uses OFDMA[6][22] physical layer in a wide channel bandwidth, typically 20 MHz, for delivering high data rates, 150 Mbps and more.

Interestingly, the requirements for the IoT have been acknowledged and some standardization efforts are aimed at Machine-to-Machine (M2M) lower-complexity and lower-cost. Releases of lower-power version have been foreseen to be commercially available in 2017-2018.

Conclusion

To summarize, two of the main disadvantages of cellular technology are the battery consumption and the cost of the hardware. Although, as mentioned before, some cellular protocols could be suitable for IoT, but existing cellular technologies can be considered cumbersome and overkill.

A third major drawback of existing cellular technologies is that they use licensed frequency bands, which means that a license has to be purchased and users are charged high rates in order to pay for the spectrum licenses (see Section 4.4). With the eye on a rising number of IoT appliances, it is not possible to charge a high amount for each node connected to the network. It can be concluded that cellular technologies suffer from high power consumption and operate on a limited amount of scarce frequency bands.

[5]Time Division Synchronous Code Division Multiple Access
[6]Orthogonal Frequency Division Multiple Access

3.2.3 Bluetooth

Bluetooth is a wireless technology standard and uses short-wavelength radio transmissions to exchange data. Bluetooth is used for data transfer over short distances with data rates up to 1Mbps, between 1-150m and utilizes the globally unlicensed 2.4 GHz short-range radio frequency band, which means no license fees have to be paid.

Bluetooth is a packet-based protocol with a master-slave structure. One master may communicate with up to seven slaves in a piconet[7]. [23]

As a response to the need for low-power applications, the Bluetooth Special Interest Group (SIG) designed Bluetooth Low-Energy (BLE), branded as Bluetooth Smart. BLE is focused on IoT projects as it offers a comparable range to Classic Bluetooth but has a significantly reduced power consumption. In comparison to Classic Bluetooth, BLE is engineered to send small data packets instead of large data transfers.

BLE/Smart utilizes the same 2.4 GHz radio frequencies as Classic Bluetooth, which allows dual-mode devices to share a single radio antenna. LE does, however, use a simpler modulation system. [24]

An example of a dual-mode application involving personal health can be observed in Figure 3.3.

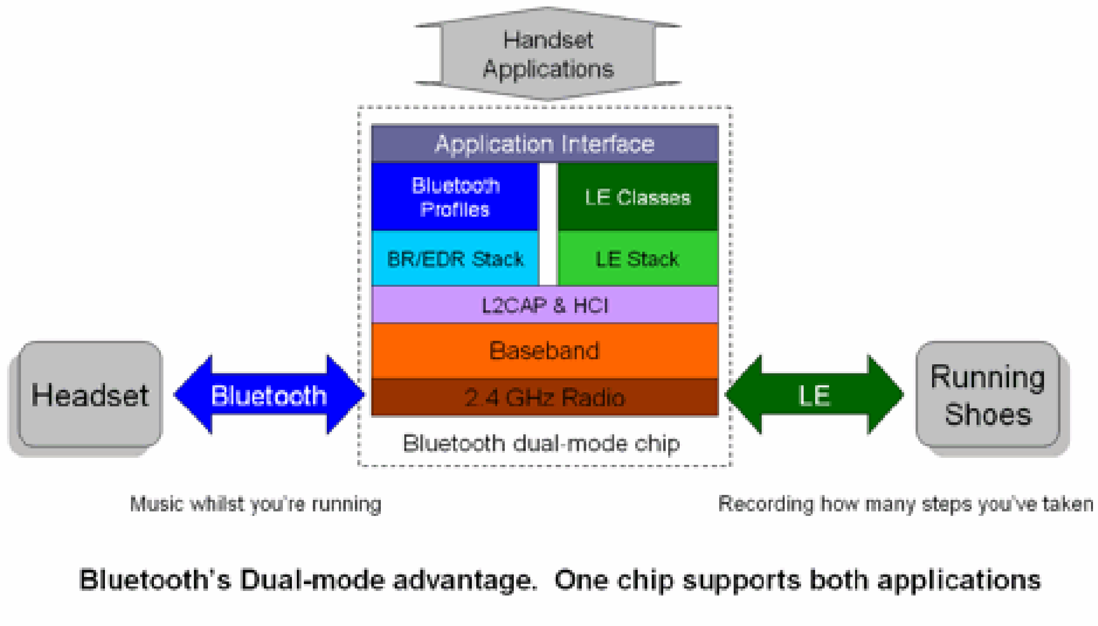


FIGURE 3.3: Example of a dual-mode application - Classic Bluetooth and BT LE

[7]A piconet is a computer network which links a wireless user group of devices using Bluetooth technology protocols.

3.2.4 WiFi

Another possibility to connect IoT projects is to use WiFi, as this connectivity method has been adopted in home environments since several years. Therefor, a wide infrastructure has been deployed which can be used for fast data transfer and large quantities of data.

The standard being used in current domestic and business environments is so-called IEEE 802.11n-2009 or short 802.11n, which uses multiple antennas in order to increase data rates up to 600 Mbps maximum. 150 to 200Mbps is more typical as the throughput depends on the channel frequency used and number of antennas. [25]

IoT projects will seldom require the data rates offered by 802.11n or its successor 802.11ac (theoretical throughput up to 1 Gbps). Also these standards are too power-consuming for many IoT applications.

In the beginning of 2016, a new protocol, called 802.11ah or WiFi HaLow, was unveiled by the WiFi Alliance[8]. The new wireless standard functions of a 900Mhz band (license-exempt bands) and combines the lower frequency with lower power requirements, resulting in a better propagation of signals and a larger effective range (double) compared to 2.4Ghz/5Ghz frequencies as it can penetrate walls and doors more easily. At the highest frequency, WiFi HaLow is expected to reach distances up to 1 kilometer.

A second wireless standard for sub 1 GHz bands is 802.11af which operates in TV white space spectrum, in the VHF and UHF bands between 54 and 790 MHz. 802.11af is not considered an IoT option yet, as it is still in its draft stages, at the moment of writing this thesis. Using the white space spectrum provides an effective way of accessing more radio spectrum in an area where available bandwidth is at a premium and utilising the resource more effectively. [26]

A schematic of the different wireless standards and their frequencies and ranges can be observed in Figure 3.4.

3.2.5 ZigBee

ZigBee is based on an industry-standard wireless networking technology operating which operates at a frequency of 2.4GHz and focusses on applications which require infrequent data transfers at low rates. The technology behind ZigBee enables the creation of personal area networks (PAN's) with small, low-power digital radios and aims to be cheaper and simpler than Bluetooth or Wi-Fi.

[8]http://www.wi-fi.org/discover-wi-fi/wi-fi-halow

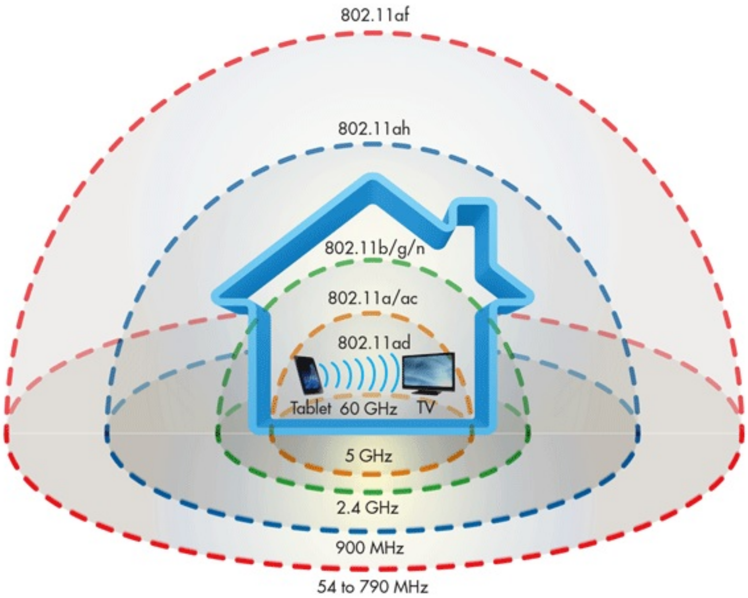


FIGURE 3.4: WiFi Standards - Ranges and Frequencies
[27]

Because of its low power consumption, transmission is limited to maximum 100m. It is possible for ZigBee devices to transmit data over longer distances by passing data through a mesh network using intermediate devices. ZigBee is used in low data rate applications, which makes it ideal for IoT applications as these require long battery life. Data transfers go up to a rate of 250 Kbps.

ZigBee networks support star as well as tree networks, and generic mesh networking. It is necessary for every network to have one coordinator device.

3.2.6 Z-Wave

Z-Wave is a wireless, low-power, radio-frequency communications technology, focussed to allow home automation. It operates on the sub-1Ghz band, which results in non-interference with other wireless standards which operate in the 2.4-GHz/5-GHz range, such as WiFi, BT or Zigbee.

Z-Wave is designed for reliable communication of small data packets (low latency) with rates up to 100kbps. In contrast to Zigbee, Z-Wave supports full mesh networks without having to implement a coordinator node. Z-Wave is also easily scalable and is focussed on fast and simple development because of the easy protocols.

Each network can include up to 232 nodes. Nodes are divided into two sets:: controllers and slaves. Ranges can go up to 30-100m, which is sufficient for domestic environments.

ZigBee and Z-Wave can be applied for the same applications. ZigBee is more versatile because profiles are available in order to minimize development time for frequently used applications. On the other hand, Z-Wave has a larger range.

3.2.7 Sigfox

Sigfox is a wide-range technology, which uses the industrial, scientific and medical (ISM) radio bands (narrow spectrum) and has a range which lays between WiFi and Cellular (3-50 km).

Sigfox uses Ultra Narrow Band (UNB) and is designed to transfer data with speeds up to 1Kbps. As a result of this low data-rate, power consumption is extremely low, e.g. 100 times lower than cellular.

The Sigfox technology can be characterised by its abilities/restrictions to send up to 140 messages/object/day, where every message can be up to 12 bytes.

Sigfox operates at the 900 MHz frequency. In order to use Sigfox, a license for each device has to be purchased.

3.2.8 NFC

NFC (Near Field Communication) enables two electronic devices to connect by bringing them within a 4 cm range of each other. If one of both devices has connectivity to the Internet, the other one is able to exchange data with services such as bank transfers.

NFC uses electromagnetic induction, operating within the globally available unlicensed radio frequency ISM band of 13.56 MHz and has data rates ranging from 106 to 424 Kbps. [28]

Due to its very modest range, NFC applications are limited to contactless transfers where users are in the direct vicinity of the object they interact with.

3.2.9 LoRaWAN

Similar to Sigfox, LoRaWAN also focuses on WAN applications and is designed to function in a low-power environment, ideal for IoT and M2M (Machine-to-Machine) applications. LoRaWAN supports large networks up to millions of devices and has data rates up to 50 Kbps. The range of a networks varies between 2 to 15 km.

LoRaWANs uses the ISM band (868 MHz and 915 MHz) which means the network can penetrate large structures. The technology is designed to connect sensors in harsh environments, over long distances. [29] Using its penetration capability, sensors can connect underground, in basements or under water.

3.2.10 Wired

Every technology mentioned in the past subsections is based on wireless standards. IoT projects can be connected to the Internet using a wired connection, if the physical location is assumed fit. Wired connections require a category 5, 5e or 6 (CAT5, CAT5e, CAT6) Ethernet cable towards each node, starting from the upper-level router or switch.

Purchasing wiring is a costly process. Once the wiring is laid out, the physical location of a node is static and cannot be moved during its lifetime. It is possible to power a node through Ethernet if the equipment supports Power over Ethernet (PoE). For the majority of IoT projects, wireless standards will be preferred.

3.2.11 Comparison

A quantitative comparison of the technologies discussed in previous paragraphs is given in Table 3.1. [30] [31] [32] [33]. Some important remarks are mentioned in the notes below.

Numeric values of indoor and outdoor ranges will depend on the amount of obstacles in the LOS, as this can drastically lower the maximum range, depending on the used frequency. Generally speaking, higher-frequency modes are more absorbed by buildings than their lower-frequency counterparts, which can travel further.

The values for maximum bit rate is based on theoretical research. Transfer speeds in real life situation will be lower due to physical obstacles (energy dissipation) and interference of other signals.

The current consumption displayed in the table is the value of an active transceiver and is based on the typical usage of a generic radio adapter at a voltage of 3.3V.

The values for the current consumption are average numbers for an example environment, based on a wide range of measurements of different projects. Momentary current consumption depends on the transfer frequency of the radio, as sending/receiving will result in large peaks in current draw when compared to the idle state.

Table 3.1: Comparison of the possible connection technologies used in IoT projects

	Max. Range (Indoors)	Max. Range (Outdoors)	Frequency	Max. Bitrate	Current Consumption (Average when Not Idle)
GPRS	3 km (urban)	35 km (rural)	900/1800 MHz	170 Kbps	500mA
EDGE	3 km (urban)	35 km (rural)	900/1800 MHz	384 Kbps	400mA
UMTS	3 km (urban)	150 km (rural)	850/900 1900/2100 MHz	2 Mbps	700mA
HSDPA	3 km (urban)	150 km (rural)	850/900 1900/2100 MHz	600 Kbps	530mA
LTE	3 km (urban)	200 km (rural)	800/1800 2300/2600 MHz	10 Mbps	630 mA
Bluetooth LE	50 m	100 m	2.4-2.5 GHz	1 Mbps	<15 mA
802.11n	70 m	820 m	2.4/5 GHz	135 Mbps	Not suitable for IoT
802.11ac	35 m	250 m	5 GHz	780 Mbps	Not suitable for IoT
802.11ah	125 m	1 km	900 MHz	100 Kbps	Still in development
802.11af	125 m	1 km	54-790 MHz	569 Mbps	Still in development
ZigBee	10 m	100 m	2.4 GHz	250 Kbps	10 mA
Z-Wave	30 m (no hops)	/	900 MHz (ISM)	100 Kbps	40 mA
Sigfox	3-10 km (urban)	30-50 km (rural)	900 MHz (ISM)	1 Kbps	25mA
NFC	4 cm	/	13.56 MHz	424 Kbps	15mA
LoRaWAN	2-5 km (urban)	15 km (rural)	863-870MHz	50 Kbps	40 mA
Wired	100 m	/	Up to 250 MHz	10 Gbps	350 mA (supplied)

3.3 Power Supply

IoT hardware has to be powered by a power supply unit (PSU) which is able to deliver sufficient energy for an extended period of time. Equipping each node with a battery would result in the purchase, maintenance and disposal of billions of batteries. Energy can be harvested in several other ways, focusing on powering remote devices using clean energy.

As already mentioned in previous paragraphs, the ability to place wireless nodes in several remote locations is crucial. E.g. : underneath asphalt, inside walls, in moving vehicles, etc. In multiple of these locations, the installation of power-distribution wires is not possible. In case of batteries, the battery life is an important issue. For this reason, it is important that projects are as power-conserving as possible to extend the time between battery replacements.

It is necessary to match the demanded power of the integrated circuit (IC) with the power generating element. Characteristics such as voltage and current are unique for each power source. In the next paragraphs, several different methods to power IoT projects will be discussed as well as their strengths and weaknesses. A general schematic of the composition of a wireless network node is given by Figure 3.5.

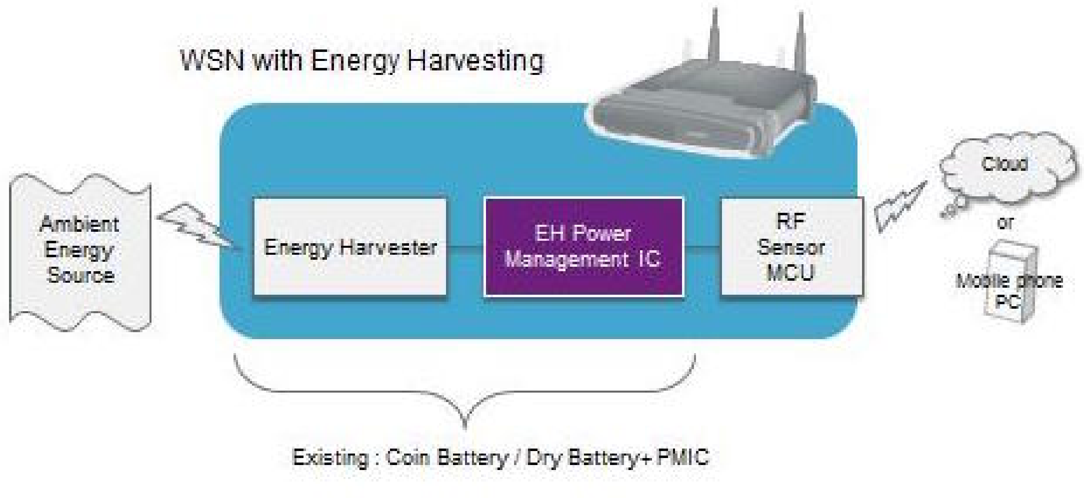


FIGURE 3.5: Composition of a Wireless Sensor Network Node
[34]

3.3.1 Direct Current

3.3.1.1 Batteries

In the paragraph about connection technologies, a wireless standard is chosen. The next step for designers of this project is to determine factors such as transmission strength/-duration as well as the duty-cycle between active and sleep states. [35]

Transceivers go into sleep mode when not in use in order to save power. The device consumes less power when in sleep state which extends the battery life. The frequency of wireless transmissions (duty-cycle) will directly affect the battery life of the product.

Once the duty cycle has been analyzed, different battery options can be analyzed. Generally speaking, smaller batteries have less storage capability than larger ones. Also, the used technology in the battery affects the capacity and cost.

A first kind of frequently-used batteries, next to Lithium, is Alkaline. E.g.: 1.5 volt AA and AAA

- (+) Low cost.
- (+) Relatively high storage capabilities
- (-) Large size
- (-) Prone to leakage

Another technology is Lithium batteries. E.g.: CR2, CR123A

- (+) Smaller in size
- (+) Lower weight
- (-) Low leakage
- (-) Higher costs

Small coin-cell sized batteries are efficient, when comparing storage to size. However, because of their small size, they still have lower storage than AAA batteries, so coin-cell batteries are used in designs for very low power use. A comparison of the different used battery form factors can be observed in Table 3.2 [35].

The advantage of using batteries in IoT projects is that they are relatively cheap to replace compared to other power sources such as solar cells or vibrational energy harvesting. Depending on the power usage of the entire circuit, batteries can have a lifespan up

TABLE 3.2: Battery Form Factors

Type	**2xAAA**	**CR2032**	**CR123A**	**CR2**
Material	Alkaline	LiMnO2	Lithium	Lithium
Voltage	3V	3V	3V	3V
Capacity	1.000 mAh	225 mAh	1.500 mAh	800 mAh
Diameter	10.5 mm (x2)	20 mm	17 mm	15.6 mm
Height	45 mm	3.2 mm	34 mm	27 mm
Weight	24 g	3 g	17 g	11 g

until several years. Batteries cannot be used in projects which are meant to be installed and then forgotten, as battery replacement can be impossible without (damaging) disassembly, unless the project is of temporary nature (monitoring for limited time, throw away device later).

E.g.: Nodes embedded in walls, concrete or in non-removable casings. For these projects, other technologies will come forward.

Batteries will deliver power at a constant voltage up until 80 percent of their lifespan. If the voltage of the battery supply is chosen to be equal to the demand of the remaining hardware, there is no need for voltage conversion using as DC/DC converter.

In the case of using batteries as a power source for IoT projects, the major cost driver will be the amount of times having to purchase batteries, in other words the amount of times of having to replace them during the lifespan of the IoT project (in relation to the lifespan of the batteries).

3.3.1.2 Solar Cells

Powering an IoT project using solar power is a viable option in case of a well-lighted environment and if replacing batteries is not easy-to-do. There is a broad range of usable Solar Power Modules which can deliver energy to projects.

Important to keep in mind that solar modules have to be combined with a Solar Charging Regulator and a battery. If no battery is implemented as a storage medium, the project will only be powered on in case of (sun)light. The remainder of energy not consumed by the project, would be lost. Using a battery pack, solar energy can be stored and used during the night or in case of a cloudy day.

A second necessary component, when using solar cells as a power source, is the Solar Charging Regulator (SCR). This is an electronic controller which switches on the connection to the solar cells if the battery is empty and switches it off again, when the

Figure 3.6: Example of a Solar Charging Regulator [36]

Figure 3.7: Example of a Solar Panel Circuit [37]

battery is fully charged. The power consumption of a typical SCR is negligible (1-3 mA).

An example of both a single regulator and a solar cell panel connected to a battery through a regulator are respectively pictured on Figures 3.6 and 3.7.

The cost drivers, when using solar cells to power a project are the capacity (and thus surface) of the solar cells and regulator as well the storage capacity of the batteries, accounting for their lifespan.

3.3.1.3 Vibrational energy

Piezoelectric patches or modules can convert mechanical energy to electric energy. Similar to using solar cells, a battery is recommended to be used for when mechanical energy is not present and the project needs to be powered.

In between the battery and piezoelectric patch, a rectifier (example in Figure 3.8) with capacitor has to be positioned which converts the fluctuating alternating current (variable AC) to a constant direct current (constant DC). The reason vibrational modules are classified under DC power sources is because the frequency of the input vibrations is not constant.

By applying high repetitions and great amplitudes of a mechanical load on the piezoelectric module (3.9), the energy level is increased. Typical power supplies of one module range around 3.6 volts and 100 mA. Using and charging a super capacitor instead of a standard battery is a good alternative to store a smaller amount of energy which can be delivered in a short amount of time.

Piezoelectric modules are used in applications such as LED signage next to rail roads, monitoring of industrial machinery or powering wireless sensors, all in environments prone to vibrations.

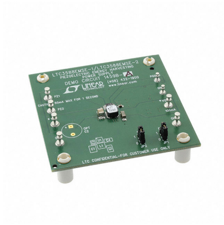


FIGURE 3.8: Full-wave bridge rectifier with a high efficiency buck converter optimized for high output impedance energy sources such as piezoelectric modules
[38]

FIGURE 3.9: Example of a piezoelectric module
[39]

The cost drivers, when using piezoelectric modules to power a project are the capacity (and thus amount) of the modules and rectifier as well the storage capacity of the batteries, accounting for their lifespan. Piezoelectric modules are usually custom-made to fit different power needs.

3.3.1.4 Thermal energy

A thermal energy-harvesting system (TE systems) takes advantage of a temperature difference between two surfaces. As temperature gradients are everywhere, a wide range of equipment operating at temperatures much higher than the ambient environment can be found.

Human bodies are relatively warm considering core body temperature is 37 degrees Celsius. Skin temperatures are typically in the range of 32 degrees Celsius. Using a TE system, a harvester attached to a persons skin offers a temperature difference of around 10 degrees. [40]

Typically, TE systems generate a potential of 10 mV for a temperature difference (ΔT) of 1 Kelvin/degree Celsius. In order to power the hardware in a generic IoT project, it needs to be said that a step-up converter (and power manager) will be needed. Otherwise, ΔT would have to around 300-500 degrees in order to power a generic IoT project which is not always realistic. Also, TE systems do not give a constant, distortion-free signal so it has to be smoothed by a power manager.

In case of using TE systems, it can be assumed that a certain ΔT is constantly present. The exact difference can fluctuate but will be compensated by the power manager (which includes capacitors as a buffer). Therefor, batteries are not considered necessary in these systems. If the source of the ambient energy is not always present, energy storage is a necessity. An example of a thermally powered wireless sensor is given in Figure 3.10.

TABLE 3.3: Comparison between different energy harvesting technologies

Energy Source	**Technology**	**Generated Power**	**Output Voltage**	**Environment**
Light	Solar Cell	30μW -10mW/cm^2	0.7 VDC/cell	Indoor-Outdoor
Vibration	Piezoelectric Module	10μW -500μ/cm^2	80 VAC	Human Action-Mechanical Vibration
Temperature	TE Module	0.5W/cm^2	10mV/K	Thermal Differentiation

The cost drivers, when using TE systems to power a project are the capacity of the Peltier module (Figure 3.11) as well as the capacity of the DC-DC converter (and power manager). Depending on the fact that batteries are needed, the storage capacity is an additional cost driver.

A comparison between the different energy harvesting technologies is made and can be observed in Table 3.3.

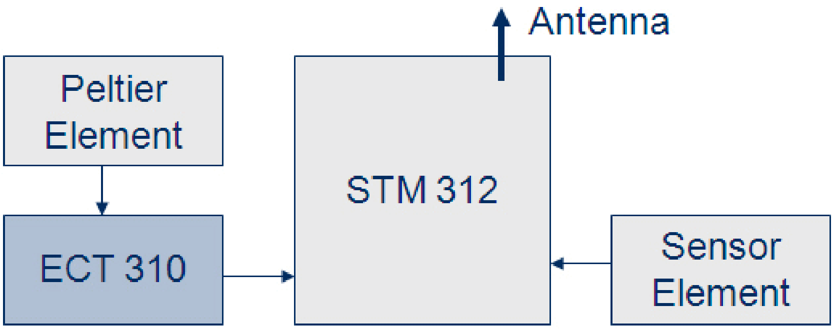


FIGURE 3.10: Block diagram of a thermally powered wireless sensor [41]

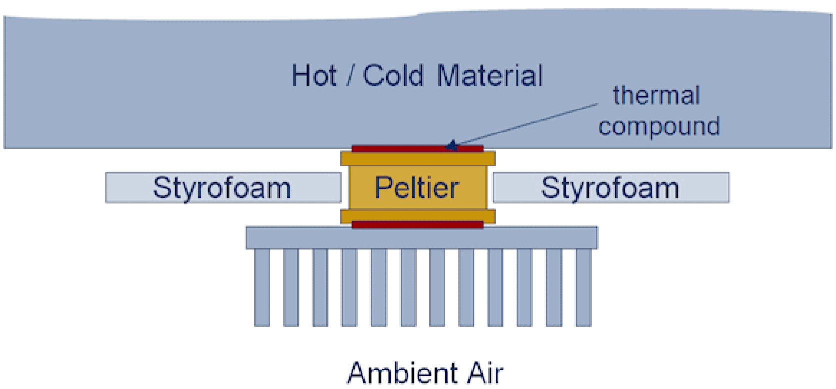


FIGURE 3.11: Thermal insulation surrounding the Peltier element [41]

3.3.2 Alternating Current

The power sources discussed in previous paragraphs are all based on direct current (DC) delivered to the circuit board of the IoT project, whether or not a DC-DC converter is needed. Another possibility is to indirectly power the hardware using alternating current, coming from an AC outlet or received wirelessly using antennas.

3.3.2.1 Outlet

Globally, there are two basic standards for voltage and frequency in the world. One is the North American standard of 120 volts at a frequency of 60 Hz, while the other is the European standard of 220/240 volts at 50 Hz. [42]

Both ranges of voltages are not considered fit for use in electronic circuits such as IoT projects, as they require a direct current and a lower voltage. In order convert an AC signal with a high amplitude to a low DC signal, an AC/DC converter (AC adapter) can be used.

AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from mains power. These adapters can be used to power a circuit directly or to recharge the battery pack of a project.

The most important cost driver, when using mains power in IoT projects, is the power capacity of the AC adapter. Other aspects such as regulation, reliability and ripple-reducing are considerable when choosing a converter but these specifications are a lot harder to quantify. An example of an AC/DC power supply with a US plug and its internals is given in Figure 3.12.

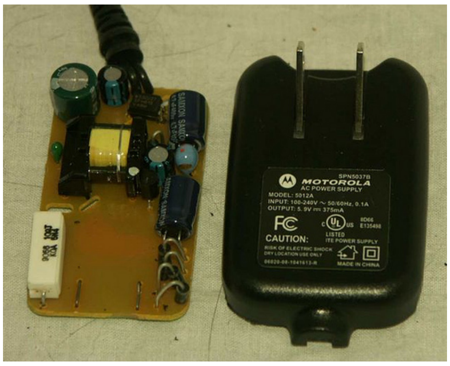


FIGURE 3.12: Example of an AC/DC power supply
[43]

3.3.2.2 Wireless Power Transfer

Wireless power transfer (WPT) consists of transferring energy, comming from an AC source to an electrical load, without the use of cables or discrete conductors. WPT can be considered useful in case of having to power projects where wires are either unwanted, impossible or hazardous.

There are two sorts of WPT: non-radiative and radiative. In the first category, power is transferred by means of the magnetic inductive coupling between coils. This technology is similar as how data is transferred when using NFC or RFID (Radio-frequency identification) tags. Non-radiative power transmission is used to power implantable medical devices or to charging electric vehicles like trains or buses. This technology, being used for IoT, has the limitation of being subject to a short range, yet has a decent efficiency.

A second possible technology, which is still in development, is radiative WPT. In radiative far-field techniques, power is transferred by beams of electromagnetic (EM) radiation, like WiFi signals, microwaves or laser beams. This option can bridge a larger distance than non-radiative WPT, but the power beams have to pointed and transmissions have to be uninterrupted, which is still a shortcoming when using standard WiFi signals. [44]

In order to maximize the efficiency of non-radiative WPT, it is important for the two coils to align with each other. The larger the coils, the easier it gets to capture the majority of the magnetic flux. Therefor, the cost driver is the size of the coils which influences the amount of copper used. Because of the AC nature of the energy transferred to the electrical load, an AC/DC converter is needed as well, which is a second cost driver. An example of a generic block diagram of an inductive wireless power system is given in Figure 3.13.

It is important to mention that in the complete hardware design of an IoT device, the power supply is one of the components which are most prone to failure. Due to frequent heating, leaking capacitors or power surges, switched-mode power supply can halt functionality of the entire device. As IoT devices get more compact, swapping PSUs is considered to not be possible for each design. Therefor, it is recommended to pay attention to the quality of the power supply, as it limits the life span of the entire project.

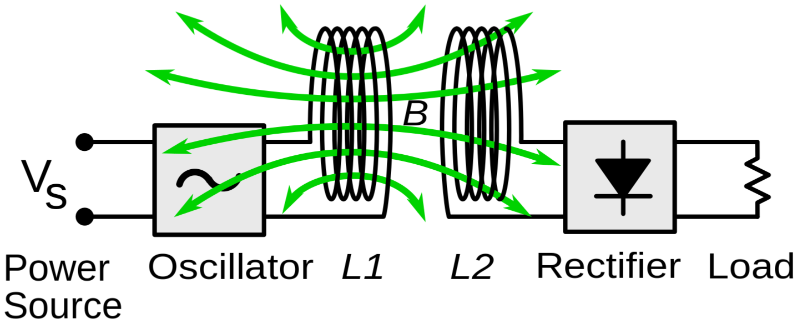


FIGURE 3.13: Example of a generic block diagram of an inductive wireless power system
[45]

3.4 Sensors

One of the most important parts in an IoT device is without doubt the sensor[9]. (Wireless) Sensor nodes will collect environmental information while connected to a network.

Obviously, the greater the number of sensor terminals installed, the greater the accuracy and variety of the data will be. As already mentioned before, big data gathered by these devices can be used to control other devices as well as monitoring and making predictions, both in cloud-based computing services as in business intelligence.

In electronic projects, sensors can be used for a wide variety of applications. As there is an innumerable amount of different sensor types, it would be irrational to list every single one of them. A limited list of examples is given below.

- Acoustic
- Vibration
- Chemical
- Electric Current/Potential/Power
- Air Flow
- Position/Displacement,
- Speed/Acceleration
- Optical
- Pressure
- Etc.

Every type of sensor will have unique cost drivers, which will be discussed when researching a specific case. A cost driver which can defined, independent of which sensor is chosen, is the wiring method. Sensors can be wired using two, three or four wires (Figures 3.14 and 3.15), depending on the accuracy of the needed measurements. Two-lead constructions result in lead-wire resistance getting added to the element resistance,

[9] In the broad definition, a sensor is an object with the purpose to detect events or changes in its environment and to provide a corresponding output.

FIGURE 3.14: Example of a temperature sensor (PT100) with 2 wires [46]

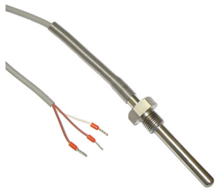


FIGURE 3.15: Example of a temperature sensor (PT100) with 3 wires [46]

which decrease measurement accuracy. Three-lead constructions result in canceled lead-wire resistance error and four-lead wiring will cancel out all resistance inaccuracy. The downside is that installation costs will go up with amount of wires. Also, more wires means more needed space in the housing.

In electronics, next to the wiring, there are two major different family of sensors: analog and digital. Analog sensors produce a continuous analog output signal which is proportional to the property what is being measured. Good examples of analog sensors are accelerometers, pressure sensors or sound sensors.

A second type of sensor is the digital sensor. In these nodes, data conversion as well as transmission takes place digitally. In these digital sensors, the measured signal is converted from analog to digital inside the sensor and can be transferred digitally through cables afterwards.

Converting an analog signal to digital is done through an Analog-Digital Converter (ADC). The main reason analog sensors are chosen in specific cases, is that the choice to pick an ADC (and its configuration) can be done by the end-user. In case of digital sensors, it is assumed that the designer can create the best conversion concerning hardware/firmware. Most of the sensors used nowadays are digital as they are easier to work with because no configuration of the converter has to be done manually.

In terms of costs, analog sensors can be cheaper if the labor costs of developing the ADC firmware is cheaper than purchasing digital sensors combined with the manufacturer's option.

3.5 Microcontroller

A microcontroller (MCU) can be considered as a small computer, positioned on a single IC. An MCU contains a processor core as well as memory and programmable input/output (I/O) peripherals. Because they have a smaller size and a reduced cost compared

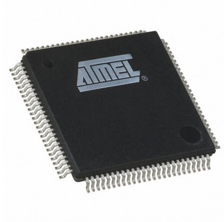


FIGURE 3.16: Picture of the AT91M40800 Atmel MCU [47]

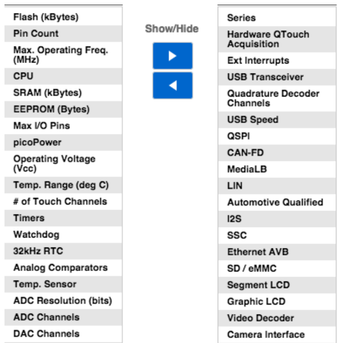


FIGURE 3.17: Picture of Atmel MCU Selector - Parameters [47]

to designs with a separate processor and hardware (modules), it is economical to use MCUs.

When choosing a correct MCU, an extensive list of parameters have to be considered and cost drivers have to be analyzed. The flash memory and maximum operating frequency will have an effect on the purchasing price while the amount of I/O pins will affect both purchasing price as labour time when soldering the project (assembly).

Other parameters will have a smaller impact on the total cost. An example of such decision list can be observed in Figure 3.17.

MCUs for IoT applications are preferred to be as power preserving as possible. Low clock rate frequencies and the ability to go into sleep result in low power consumption.

3.6 Miscellaneous

Every IoT project consists of a connectivity radio, one or multiple sensors, a MCU and a power supply. However, this list is not limitative and it is possible that extra hardware is required, depending on the purpose and environment of the project. For this reason, it is thinkable that a project needs:

- An LCD Monitor/Touch Screen
- Local Storage
- Input buttons/Joystick
- Speaker
- GPS Module
- Etc.

A synopsis of the cost parameters influencing the total cost for each hardware segment can be observed in Table 3.4.

Table 3.4: Synopsis of the Cost Parameters influencing the Total Cost for each Hardware Segment

Hardware		**Cost Parameters**
Conn. Radio		
	2G/3G/4G **BTLE/ZigBee/Z-Wave** **SigFox/LoRaWAN/Wired**	Amount Purchasing Cost Transceiver Subscription Fee Implied Battery Life
Power Source		
	Batteries	Amount + Lifespan Purchasing Cost Batteries Duration of Replacement
	Solar Cell	Amount + Lifespan Purchasing Cost Cells Purchasing Cost Regulator
	Vibrational Energy	Amount + Lifespan Purchasing Cost PE Module Purchasing Cost Rectifier
	Thermal Energy	Amount + Lifespan Purchasing Cost Peltier Module Purchasing Cost Regulator
	Outlet	Purchasing Cost AC/DC Conv.
	WPT	Purchasing Cost Coils Purchasing Cost AC/DC Conv.
Sensors		
	Analog Sensor	Amount + Lifespan Purchasing Cost Sensor Purchasing ADC Programming ADC
	Digital Sensor	Amount + Lifespan Purchasing Cost Sensor
MCU		
	MCU	Amount + Lifespan Purchasing Cost MCU Programming Firmware
Peripherals		
	Input Devices	Amount + Lifespan Purchasing Cost Implementing in Firmware
	Output Devices	Amount + Lifespan Purchasing Cost Implementing in Firmware
Assembly		
	General	Amount of Devices Purchasing Wires Purchasing Solder Purchasing PCB Purchasing Casing Assembly Labour

3.7 Conclusion

In this Chapter, the hardware building blocks and their cost drivers have been elaborated. Each IoT device exists of at least one component of each hardware family, discussed in previous sections. It is important to sketch the hardware skeleton of a device first, before blindly starting to purchase parts.

The choice of a specific hardware component will influence the complexity of the IoT project, the project time, the required manpower as thus the total project cost. E.g.: licensing fees, complex assembly, purchasing cost, etc.

Hardware costs can be perceived as the biggest building block when estimating project expenditures, but there are three other ones, which will have a significant influence on the fixed overhead costs of a 'Make'-oriented project.

Chapter 4

Assembly, Firmware and Testing

4.1 Introduction

In order to obtain a detailed cost estimation of the hardware components utilized in the IoT project, one can simply use the supplier's (preliminary) invoice which states the purchasing and shipping costs. The act of estimating costs to assemble these components, providing the project of firmware and testing it before putting it out for mass-production is not as straightforward as with the hardware analysis.

Basically, cost assessment in these areas is based on rudimentary rules of thumb and experience of precedents.

4.2 Assembly Costs

Each IoT project which has not been bought OTS is composed of hardware components (listed in Chapter 3) which are soldered upon an PCB (Printed Circuit Board). Ordering separate components with a supplier do not deliver a finished product, as this hardware has to be assembled.

Creating the BOM, purchasing components and the assembly method depend on which step of the Executing phase is being run through: PoC, Prototype, Pilot or Final Product.

For the PoC and prototype(s), assembly is frequently done in-house and almost always manually. When progressing to the Pilot and Final Product steps, assembly of the components is outsourced and executed in an automated manner, depending on the desired quantity of devices. In order to assemble PCBs in-house, an experienced electronics engineer has to be hired [1].

If a client chooses to outsource assembly, it is common practice that the assembler also becomes responsible for ordering the hardware components. A 10-15% markup is typically added to the suppliers' hardware price, in order to cover for administration, handling and DOA (dead-on-arrival) testing. Assembly companies usually manufacture PCBs in-house, based on the design plans received from the customer.

Frequently-used pricing parameters for PCBs are [2]:

- Board Quantity
- Board Size (square cm/inch)
- Thickness (inch)
- Layers
- Lead Time
- Shipping Country

Online market research shows that PCBs have a price of around 0.5-1 EUR/inch^2 for order quantities of 100 and up. A more detailed comparison of different suppliers and their prices in relation to the aforementioned parameters can be found here[3].

Depending on the size of the project and the country of out-sourcing, the cost of technical staff (soldering of the PCB) differs.

Example: the salary of an average Chinese worker at Foxconn Electronics is USD 1.50/hour, while the salary of an equally-skilled employee in the USA is around USD 13-20/hour[48].

The financial benefit of manufacturing in China will decrease, yet will still remain larger when adding shipping costs. To obtain a general idea of PCB assembly costs, a cost calculator for China-manufactured boards can be found here[4]. The manufacturer did not want to disclose pricing details, when being contacted, but a rudimentary ANOVA/F-test analysis (see Figure 4.1) revealed these price estimations (for 4-7 days lead time):

$$\text{Total Cost in USD (all boards)} = 367 + 0.3*BQ + 1.75*BOML + 165*DS + 165*LF \quad (4.1)$$

where the first term (367 USD) is the setup cost, BQ the Board Quantity (in pieces), BOML the number of individual components on the board and the last two terms (DS and LF) are binary values which are 1 if the PCB is double-sided/lead-free and 0 if

[1] These methods are decribed based on best practices, as described by multiple hardware suppliers and IoT project development teams.

[2] http://www.7pcbassembly.com/PCB-quote.php

[3] http://www.ladyada.net/library/pcb/costcalc.html

[4] http://www.7pcb.com/PCB-Assembly-Quote.php

not. One should be aware that these assembly costs do not include purchasing of any materials (hardware components or PCB). This formula is rudimentary and does not include the number of thru-holes as parameter. Therefore, it is recommended to use the on-line calculator for projects with an extended number of connection points.

board quantity	bom lines	Double-Sided?	Lead Free?	price USD
1	1000	0	1	2289,82
100	100	0	1	719,02
100	200	0	1	897,89
100	300	0	1	1076,75
100	1000	0	1	2328,78
200	100	0	1	758,37
200	200	0	1	937,24
200	300	0	1	1116,1
200	1000	0	1	2368,13
4000	100	0	1	1879,84
4000	200	0	1	2058,7
4000	300	0	1	2237,56
4000	1000	0	1	3489,59
100	100	1	1	880
1000	100	1	1	1155,45
100	1000	1	1	2489,76
1000	1000	1	1	2765,2
1000	1000	1	0	2513,82
100	100	1	0	800
1000	100	1	0	1050,41
100	1000	1	0	2263,41

SAMENVATTING UITVOER

Gegevens voor de regressie	
Meervoudige	0,999393
R-kwadraat	0,998786369
Aangepaste k	0,998482962
Standaardfou	32,09243539
Waarneming	21

Variantie-analyse

	Vrijheidsgraden	*Kwadratensom*	*Gemiddelde kwadraten*	*F*	*ignificantie F*
Regressie	4	13561615,05	3390403,763	3291,89573	4,2312E-23
Storing	16	16478,79055	1029,924409		
Totaal	20	13578093,84			

	Coëfficiënten	*Standaardfout*	*T- statistische gegevens*	*P-waarde*	*Laagste 95%*	*Hoogste 95%*	*aagste 95,0%*	*loogste 95,0%*
Snijpunt	367,2262148	26,50267743	13,85619305	2,4948E-10	311,043048	423,409381	311,043048	423,409381
board quanti	0,29598121	0,004901	60,39200605	2,6063E-20	0,28559155	0,30637087	0,28559155	0,30637087
bom lines	1,748047226	0,017308886	100,9913167	7,1182E-24	1,71135403	1,78474043	1,71135403	1,78474043
Doubleside	165,4681451	18,79917197	8,801884753	1,5721E-07	125,615681	205,320609	125,615681	205,320609
leadfree	165,6925	22,69277869	7,301551841	1,7765E-06	117,585958	213,799042	117,585958	213,799042

Figure 4.1: Pricing Estimation PCB Board Manufacturer

4.3 Firmware

In electronic systems such as IoT projects, firmware is a type of software that provides control, monitoring and data manipulation of engineered products and systems.[49] Firmware is (semi-) permanently integrated onto hardware and provides the low-level control program for these devices.

When buying OTS hardware solutions, firmware is pre-integrated and frequent updates are usually offered by the manufacturer. Hobbyists design custom firmware to extend the basic functionality of ready-made IoT devices. Manufacturer's warranty usually becomes void after modifying this software, as this was not intended.

Base-stations which are commonly used to engineer PoCs (see Subsection 2.3.5) can be sold without pre-installed firmware. Customers can choose between installing the manufacturers default software, a pre-developed custom firmware or coding their own control program. In this case, the customer is given more freedom to experiment and enlarge the possibilities of the bought base-station.

When choosing for the Make-option, firmware is not provided as loose components are bought, brought together and assembled, possibly in a unique way. For Make-projects, firmware needs to be tailored to the hardware project at hand. As already mentioned, this development can be done in-house or outsourced. Regarding the total cost of writing firmware code, implementation and updates, this strongly differs from project to project.

Out of conversations with IoT agencies, it could be concluded that IoT firmware development is similar to the world of mobile application development, in terms of costs and workload. It has to be kept in mind that the initial development is only part of the process, as there always has to be an expert at hand who can deal with updates, bugs or break-downs of the software.

If the choice is made to outsource the firmware development, best practice dictates that these companies will rely on code re-usability and the use of OTS software components as much as possible. Various algorithms have been developed which calculate a (rudimentary) cost estimation for developing a custom-made firmware.

Three frequently-used methods are SLIM (Software Lifecycle Management), Function Points and COCOMO (Constructive Cost Model). Studies [50] show that COCOMO is the most effective method when having the ability to use historical data of previous firmware projects as starting ground.

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers). Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases. [51]

For all of these forms, the expected SLOC (Source Lines of Code) is the main metric of the cost estimation tool. The second input variable for the basic COCOMO is the type of software class:

- Organic projects : small teams with good experience working with less than rigid requirements
- Semi-detached projects : medium teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects : developed within a set of tight constraints. It is also combination of organic and semi-detached projects.

The COCOMO algorithm will give three output variables: the effort applied (in man*months), the development time (in months) and the people required (in persons). In order to obtain a firmware cost, the effort applied has to be multiplied with the average monthly salary of an experienced programmer (EUR 4000-5000/month[5]).

[5]http://www.payscale.com/research/US/Job=Computer$_{P}rogrammer/Salary$

The more advanced the level of COCOMO is used, the more accurate the estimations will be. An on-line calculator, made by the Computer Science Department of the University of Michigan and based on the work of a respected researcher[52] can be found here [6].

These algorithms will only a give rough estimation of the possible firmware cost. IoT agencies recommend to consult multiple software development companies to request a price quotation. It is not uncommon for these companies to have a preferred set of hardware manufacturers, which makes it easier, faster and cheaper to integrate components (SoCs, sensors, MCUs, etc.) into the project's firmware.

4.4 Testing

The evaluation phase is depicted as the final building block in the entire IoT Management Methodology (Figure 2.20), yet testing is an omni-present process. This procedure checks if the project, being developed:

- meets the business, functional and technical requirements
- responds correctly to all inputs, defined in the requirements
- performs tasks within an acceptable time,
- can be installed and run in its intended environments
- achieves the general result demanded by the client

The main goal of testing is to find bugs, errors or other defects in the project (hardware/software) in order to fix these. The evaluation/testing step is more of an iterative process: when one problem is fixed, it can illuminate other, deeper errors, or can even create new ones. [53]

Keeping this definition in mind, a sole feedback loop from the final phase to the Future Actions phase can be considered incorrect. Testing should be depicted as interaction links between each two adjacent building blocks (Figure 2.20), much like the Agile Project Management Method. Testing can be done on three different levels:

- Unit Testing : Individual Units (eg: Hardware Components) are tested. These tests are cheap and fast, and help to comprehend the abilities and constraints of each component.

[6] http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/gamel/introduction.html

- System Testing : Linking of Individual Units. Do all individual units function cooperatively? Correct assembly and bug-free firmware are the main goals.

- Performance Testing : Raw processing speed, bit rate/band-width, latency and capacity of the project has to be tested in order to ensure maximum reliability during daily-use.

All of these levels of testing are focused on the Making and Executing Phase. Yet, the most cost-influencing evaluation reasoning is situated in the Planning Phase. The basics of quality engineering dictates that preventing errors is better than having to fix them in the first place. A tool to make this principle more comprehensible, is the 1-10-100 rule [54].

The 1-10-100 rule is applied in many scenarios concerned with quality and the cost of correction. The rule expresses cost in any number of units, measured in financial terms, resources or time. The 1-10-100 rule (Figure 4.2) illustrates the importance of maintaining a high standard of hardware and software quality continually rather than occasionally. The rule applied to an IoT project is as follows:

- Checking the quality of hardware/firmware costs the business EUR/USD/Hours 1. This is known as the prevention cost.

- Repairing an error before deployment of the project costs the business EUR/USD/Hours 10. This is the correction cost.

- Repairing an error after the project has been put into motion costs EUR/USD/Hours 100. This is the failure cost.

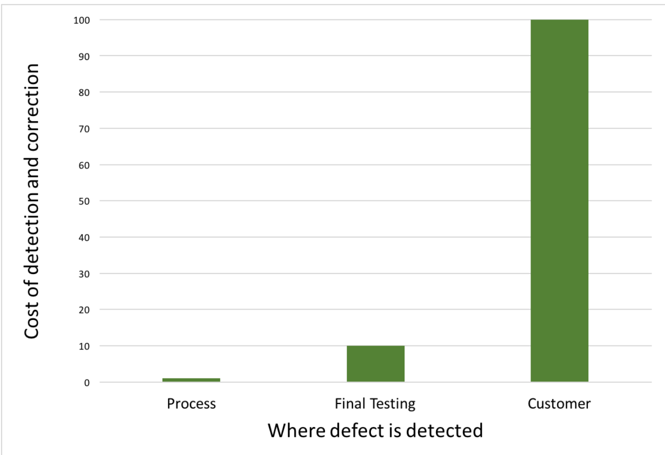


Figure 4.2: 1-10-100 Rule

In order to put this rule into practice, one of the software/firmware engineers and one of the hardware engineers should be educated in quality engineering and has to focus on going through each of the three testing steps for their department. For small to moderate-sized projects, it is not crucial to hire an extra quality engineer. The (part-time) job of quality engineer is an on-going process and cannot disappear after having finished the final product.

Testing has to be done in a lab environment which requires special equipment, software, and facilities to debug IoT project designs. Depending on the size of the project, the cost of buying/renting this equipment will differ (Figure 4.3).

The testing cost building block also includes the cost for regulatory approvals. Connectivity radio modules, functioning in the unlicenced spectrum band, are (most frequently) sold with a Wireless Standard Certification, while SoCs do not. Adding these certifications to a product implies that the product designer must apply for membership in the standards bodies and conduct some product-level regulatory testing.

Regulatory testing costs and type approvals vary by country. Some countries will accept others approvals. [14] A list of these costs can be observed in Figure 4.4. [7]

Lab Equipment	Cost to Own	Cost to Rent/Day
Calibrated traceable gain horn antenna	~$2,500	Included in a single day rental at test facilities. This is generally $1,000-$3,000/day.
Bi-conical antenna	~$2,000	
3D positioner	~$2,000	
Spectrum analyzer	~$6,000	
Wireless testing software with desired modulation	~$1,500	
RF isolated, anechoic room (5m x 5m)	~$20,000	
Wireless standard emulator, sniffer, and debug	~$20,000	

FIGURE 4.3: Wireless Lab Equipment and Facilities
[14]

Certifying Body	Estimated Cost	Module Pre-Certification Applies (Yes / No)	Wireless SoC Certification Applies (Yes / No)
FCC	~$7,900	Yes	No
IC (Canada)	~$7,900	Yes	No
ETSI / CE (Europe)	~$7,900	Yes; some limited testing/re-testing required	No
South Korea	~$4,500	Yes	No
Japan	~$8,600	Yes	No
Bluetooth®	~$8,000	Yes; Add'l membership fee required	No; Add'l membership fee required
ZigBee®	~$4,000	Yes; Add'l membership fee required	No; Add'l membership fee required

FIGURE 4.4: Regulatory and Certification Estimated Costs
[14]

[7]It has to be kept in mind that these costs reflect an estimation when having chosen for the Make-decision and are not applicable to the Buy scenario.

4.5 Conclusion

When applying the IoT Management Methodology upon a real-life use-case, the 4 cost building blocks have to be analyzed in order to obtain a project cost estimation: hardware, assembly, firmware and testing.

- Hardware : The costs are the sum of the (volume-discounted) purchasing prices and extra expenditures such as tax, shipping, insurance, custom/import charges, etc. Component prices can be found in manufacturers' catalogues. Prices for PCBs need to be quoted depending on size and thickness.
- Assembly : Outsourcing the purchasing of the components/PCB results in a 10-15% markup of the original hardware price. The cost of assembly itself can be estimated by Formula 4.1 or by using the quota-calculator found on many specialized websites.
- Firmware : The development cost of firmware always depends on the re-usability of code and on the amount of code which has to be written from scratch. The COCOMO calculation in combination with the average salary of a programmer will give the cost of initial development (no updates/maintenance). Best practice experience from IoT agencies show that firmware for embedded project, with no extraordinary contextual requirements, take up 30-40% of the overhead costs, for device numbers under 10-100k.
- Testing : Testing costs include the operating expense of hiring (a minimum of) two quality engineers, specialized in hardware and software as well as the cost of buying/hiring testing equipment and paying for certifications and licensing. Technical personnel is required for the maintenance, updating and configuring of the hardware/software on a day-to-day basis.

While the Hardware, Assembly and Firmware costs can be categorized as capital expenditures (CAPEX), Testing is considered as a combination of CAPEX and operating expenses (OPEX) as it is an on-going process. Firmware updates, bug-fixes, hardware upgrades, etc. are tasks which have to be performed by dedicated personnel.

Chapter 5

Case Study

5.1 Introduction

In order to apply the aforementioned IoT management methodology to possible real-life problem situations, it can be considered interesting to design a use-case which puts the theory to the test. In this Chapter, an existing problem setting is elaborated and each methodology phase is discussed while applying it to this case[1]. Only the Planning and Making phase are applied to the use-case. The Executing phase focuses on the actual buying or manufacturing of devices which is beyond the scope of the thesis.

The objective of the case study is to setup an instrumentation and measurement campaign to precisely monitor and describe the state and structure of energy usage/comfort readings and to give an overview in time of the consumption of energy and its meta-data for the common area of a real office environment and building.

To define the common area within the project, the case focuses on the floors of the future office building that the UGent IBCN research group will take into use ('Technologiepark Zwijnaarde'). The common area comprises 4 floors of office spaces, meeting rooms and building utility zones.

This measurement campaign will produce information that will allow researchers to base their works on reliable data. This project aims to describe the state of the energy consumption in an office building as carefully as possible in order to obtain more accurate models and forecasts. The specifics of the case are stated more detailed, following the steps of the IoT Project Management Methodology.

[1] Complete Case Study document can be found in Appendix

5.2 Application of the Methodology

5.2.1 Planning Level

5.2.1.1 Problem Description - Business Requirements

The goal of this project is to create a dense, long-term monitoring setup allowing future research to extrapolate the data to a detailed breakdown. The client, in this case the IBCN Research Group, has a specific demand: obtaining data about energy usage/flow and comfort related parameters in a well-defined area. More specific, following data has to be acquired:

- For the office zone (office space and meeting rooms)
  - Electrical energy consumption: per outlet for each desk (4 outlets/desk as identified in Location Screening)
  - Electrical energy consumption: per light circuit
  - Temperature, relative humidity, amount of CO_2, VOCs, Light intensity
- For the building utilities zone
  - Electrical energy consumption: per electrical circuit

5.2.1.2 Location Screening

The physical location comprises 4 floors (9^{th}-12^{th}) of the iGent-Tower in Technologiepark Zwijnaarde (Figure 5.1). A detailed floor plan of the location where the problem setting is situated is attached in Appendix.

When falling back to the IoT Methodology, the location screening step consists of 7 questions which help to dissect the problem setting into multiple subproblems. Answering the general questions found in Table 5.1 will constrain the range of specific choices which can be made in the Solution Proposal step of the Making Phase.

Each of these questions' answers will be evaluated in the next paragraphs. In order to obtain a better view of the problem situation, a detailed description of the location is given, which also include the answer for the final question concerning contextual requirements.

The business requirements demand that comfort readings and detailed energy measurements are obtained from the 'office' areas, while more general measurements have to be

TABLE 5.1: Location Screening: Questions

1	Nature of the to be measured/tracked asset	Stationary
2	Behaviour of asset	Dumb
3	Existing Project	No
4	Environmental conditions	
	Operating temperature	15°C-25°C
	Operating location	Indoors
	Humid Environment	No
	Dusty Environment	No
5	Project Size/Range	27m x 42m (1 floor)
6	Outlets available	Yes
7	Special (contextual) requirements	Yes (see notes)

acquired for the 'utilities' zone. Different location areas have different demands involving measurements.

The office area is defined as the spaces within the building intended for day-to-day use by the occupants of the building, such as:

- Office spaces
- Meeting rooms
- Hallways
- Passages
- Kitchens
- Sanitary
- Staircases
- Photocopy rooms

The building utilities zone is defined to include the spaces that are intended for and to make room for technical equipment. For example, the building utilities zone includes:

- Electrical cabinet rooms
- Technical rooms
- Archive rooms
- Storage rooms

For this project, measurement data of 4 floors is needed. The layout of floors 9-11 is (almost) identical, floor 12 is slightly aberrant to the other ones. A detailed description of the current infrastructure is given in Table 5.2. Important to note is that this layout is subject to change and growth in the future (see Future Actions).

In accordance with the business requirements, energy measurements should be done on outlets and electric circuits, while air quality readings have to be obtained for each separate office room. For both cases, the assets are 'dumb' and stationary, which means the installed hardware can function using outlet current and does not (necessarily) required the use of batteries.

Table 5.2: Detailed Description of current layout - Location Screening

Floor	**9**	**10**	**11**	**12**
Office Area				
	18 x office room 59 x indiv desk 1 x hallway 2 x staircase 4 x passage 2 x conference room 3 x bathroom 1 x photocopy room 1 x kitchen	18 x office room 49 x indiv desk 1 x hallway 2 x staircase 4 x passage 2 x conference room 3 x bathroom 1 x photocopy room 1 x kitchen	17 x office room 30 x indiv desk 1 x hallway 2 x staircase 4 x passage 2 x conference room 3 x bathroom 1 x photocopy room 1 x kitchen 1 x data center	3 x office room 1 x hallway 2 x staircase 4 x passage 1 x conference room 3 x bathroom 1 x photocopy room 1 x kitchen 6 x lab room
Utility Area				
	1 x tech room 1 x storage 1 x archive	1 x tech room 1 x storage 1 x archive	1 x tech room 1 x storage 1 x archive	1 x tech room 1 x storage 1 x archive

Currently, there is no similar project implemented, which does not impose any extra contextual requirements. The surface of 1 floor amounts to approximately 1.134 m^2, having a width and length of 27m x 42 m. Each floor has 4 connected hallways, two measuring 2,5 x 30 m, while the other two are dimensioned by 4 x 8 m. These dimensions play a crucial role in choosing the correct connectivity technology to send measurement data from the IoT device to the gateway as well as the choice of air quality sensors for the comfort readings.

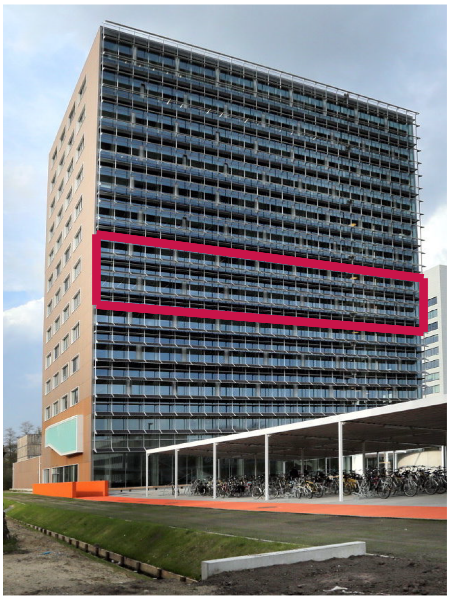


Figure 5.1: iGent-Tower in Technologiepark Zwijnaarde. In red are floors 9-12

TABLE 5.3: Case Study: Project Dimensions Checklist

General			
	Project Constraints		
		Timeline	3-5 years
		Budget	*Minimal*
		In-house SW Dev Possible	*Yes*
		In-house HW Dev Possible	*Yes*
Hardware			
	General		
		Number of Nodes	*See 'HW Constraints'*
		Integration Complexity	*New (No retrofit)*
		Lifetime of Asset	*Maximal*
	Connectivity		
		Required Range	*50 m*
		Required Bandwidth	*464 Kbps*
	Environment		
		Waterproof/Dustproof	*IP20*
		Shockproof	*IK00*
		Accessibility	*EASY*
	Power		
		Power Source	*AC*
		Required Battery Life	*Back-Up Purposes*
	Sensors		
		Resolution	*See 'Req Bandwidth'*
	Circuit		
		In-house Assembly Possible	*Yes*
Software			
	Firmware		
		In-house Dev Possible	*Yes*
		Updates Mandatory	*Maintenance*
		Update Period	*When needed*
	Security		
		Crucial	*No*
		In-house Dev Possible	*Yes*

5.2.1.3 Project Dimensions

As mentioned in the Methodology elaboration, the project dimensions step will convert the findings of the location screening to a list of solution (and even technological) restrictions. The solution proposal will have to comply with each of these limitations, found in the project dimensions checklist (Table 5.3), in order to satisfy the business requirements.

General Constraints: the client demands for a project deployment time of 5 years (preferably less). In this case, the total budget is not fixed. The objective of this use case is to test the IoT Management Methodology and to design the hardware solution with a minimal cost for purchase, assembly and installation.

A great deal of these costs will depend on the choice of keeping both hardware and software development in-house or deciding to outsource it. For this use case, the IBCN personnel (project clients) are qualified to develop the entire project as they are educated in both software and hardware development. Yet, as the clients are no specialists in this particular field, project time and costs could rise due to inefficient work flows. A comparison between outsourcing and in-house development is executed in the 'Make-or-Buy' subsection.

Hardware Constraints: the number of nodes will be equal to the amount of IoT devices obtaining measurements. So each AC outlet in the office area will have to be equipped with an energy data logger, as well as each electric circuit in the utilities area. A comfort readings logger have to be installed for each secluded room of each floor.

After having completed the Location Screening, based on the available documents and a visit to the facility, a total of 782 active[2] AC outlets were estimated to have to be monitored, spread over the 4 floors (319-270-203-125). Each floor has approximately 20 separate electrical circuits[3]. A total of 862 energy monitors should be acquired (Made or Bought).

For the comfort readings, each secluded area of the 'Office' part will receive one monitor. Four comfort monitors will be assigned to the hallway on each floor. A total of 113 (31-31-30-21) monitors should be acquired (Made or Bought).

The number of nodes also affects the choice of connectivity technology (for the Make-decision), together with 3 other questions from Table 5.3 which will determine which wireless protocol to select: Required Range, Required Bandwidth, Power Source.

Required Range: The total diagonal of each floor (rectangular shape) amounts to approximately 50 m. If each sensor node in a single IoT-device is connected to a gateway on the same floor, a maximum range of 50 meters (in obstructed environment) is required.

Required Bandwidth: The electric energy data will have to be monitored and logged at a certain frequency and then transferred to the gateway for further processing or analysis. The sample frequency and send frequency do not have to be equal as the logged data can be kept in memory for a limited amount of time.

The sample frequency of the electric current and voltage measurements should be at least twice as high as the highest frequency measured in the AC-grid[4]. If not, the sample data will not be sufficient to reconstruct voltage and current wave forms. European AC-grids

[2] These are the outlets which actively in use and have to be monitored. This number is lower than the actual amount of installed outlets.

[3] AC outlets and lighting are on different circuits.

[4] Nyquist frequency[55]

distribute electric energy at a frequency of 50 Hz. Due to the intervention of non-linear electric loads (switching power supplies/laptop adapters/electronics/etc), frequencies will appear which are integer multiple of the fundamental frequency [56] (Figure 5.2).

Both the fundamental waveforms as harmonics deliver electric power through an AC-outlet and in order to acquire a complete and scientific correct data set, the IoT-device, unregarded if made or bought, has to be able to measure frequencies up to 20 times the fundamental [57].

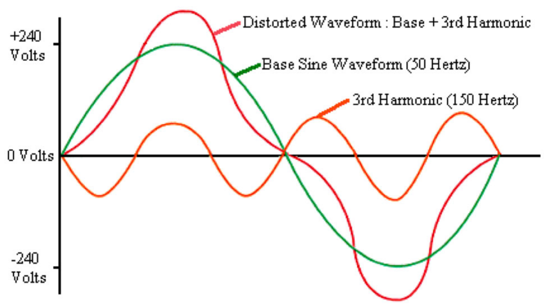


FIGURE 5.2: Example of Harmonics in Voltage Waveforms [58]

Having a fundamental frequency of 50 Hz, the sample frequency of the device should be minimum 2000Hz. 1 energy measurement should be a line of text, containing following data[59]:

[Effective Voltage; Effective Current; Date; Time; Real Power; Apparent Power; Power Factor; Frequency]

Example: [220,020;2,320;08/01/2016;16:02:15;424,7;510,5;0,832;50,002]

As seen in the example, data for the electric current, voltage and power as well as other important meta-data are logged onto the device. When maintaining a resolution of respectively 1 mV/1 mA/100 mW/100 mVA, one message will have a size of approximately 58 bytes. A minimal bandwidth of 116 kB/s or 928 kbps is needed to send the acquired (meta-)data if accuracy up to the 20^{th} harmonic is wanted.

Fourier analysis of commonly used switched power supplies (Figure 5.3) has shown that the importance of harmonics significantly drops, surpassing the 10^{th} harmonic. If this assumption is accepted, the necessary bandwidth drops to 464 kbps.

A bandwidth of 464 kbps is needed if each IoT device would measure and send the effective electric energy values in real-time (1.000 times a second). For data logging and statistical purposes this can be considered overkill. One measurement[5] every 2 seconds will suffice and will still result in accurate measurement data, as electric loads in office environments do not alter significantly. In this way, a bandwidth of only 0.232 Kbps (58 bytes message each 2 seconds) is required.

The environmental monitor only needs to send data to the gateway every 10 seconds, as values for temperature, relative humidity, amount of CO_2, VOCs[6] and light intensity do not change in a blink of an eye. 1 comfort measurement should be a line of text, containing following data[59]:

[Temperature;Relative Humidity;Amount of CO_2;Amount of VOCs;Light Intensity;Date; Time;]

Example: [29,5;50,5;0662;1021;2523;08/01/2016;16:02:15]

When maintaining a resolution of respectively 0.5 degrees Celsius/0.5 percent/1ppm/1ppm/1 lux, one message will have a size of approximately 46 bytes. A minimal bandwidth of 2,6 B/s or 0,0368 kbps is needed to send the acquired (meta-)data.

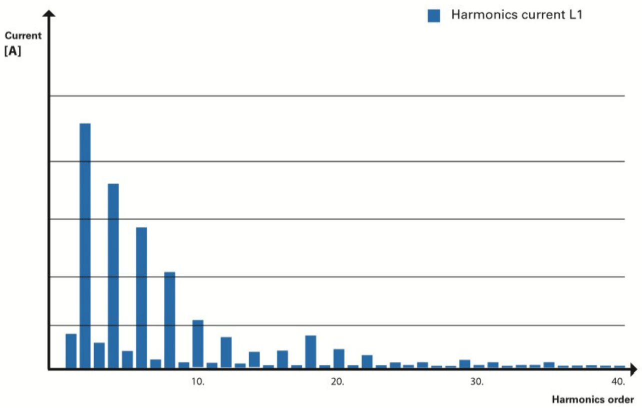


FIGURE 5.3: Fourier analysis of laptop power supply [61]

Choosing a connectivity technology is solely relevant when have chosen for the Make-option or Buy-option with assembly of base station and extensions. When buying TOS products, the manufacturer has decided on the wireless technology and constraints regarding maximum number of nodes, power supply and maximum amount of sent data are fixed.

[5]One measurement will take up 20 ms, the time to measure 1 period of the fundamental frequency

[6]Volatile Organic Compounds can be both human-made and naturally occurring chemical compounds. Some VOCs are dangerous to human health or cause harm to the environment. Harmful VOCs typically are not acutely toxic, but have compounding long-term health effects. Because the concentrations are usually low and the symptoms slow to develop, research into VOCs and their effects is difficult[60].

As already mentioned, no existing project is currently in deployment, so there is no need to take into account retrofitting of connectors or protocols. The general lifespan, in office environment, of an AC outlet is around 50 years. As the business requirements state that no end-date for the monitoring is determined, the IoT device should remain fully-functional as long as possible.

The project will be deployed in an office setting, so an International Protection Marking grade of IP20 or better suffices.

5.2.1.4 Project Growth Analysis

Whilst running through the Project Dimensions, a Solution Proposal is created, which will be elaborated later on. The last step of the Planning Phase sketches the future evolution of the project. In order to obtain a clear image of this projected growth, three major parameters need to be estimated: Amount of Users, Assets and Events.

For this specific project, the amount of users will affect the number of assets in a linear way. There is both a fixed amount as a variable amount of assets. The fixed amount exists of the number of rooms (comfort readings) as all available ones are already included in the 113 assets, independent of the amount of users.

For the energy readings, the amount of users does interact with the number of assets. Again, there is a fixed amount of electrical circuits (80) which do not change when increasing the users amount. Also, the number of active outlets in the hallways, staircases, conference rooms, photocopy rooms and bathrooms do not vary if the user base grows or diminishes. For each user which is added, 4 extra outlets are reserved. The amount of daily events for the energy readings are as follow:

$$\text{Number of Events} = \text{Number of Assets} * 43.200\ \text{Messages/Day} =$$
$$(862 + 4 * \text{Number of Future New Users}) * 43.200\ \text{Messages/Day}$$

The amount of daily events for the comfort readings are as follow:

$$\text{Number of Events} = \text{Number of Assets} * 8.640\ \text{Messages/Day} =$$
$$113 * 8.640\ \text{Messages/Day}$$

The upper limit of users is determined by the available working space in the office rooms. An estimation based on the current occupation and the architectural plans, a maximum of 45 persons can be added[7]. An upper limit of 1.042 energy monitors, divided over 4 floors, are needed if the space is at capacity.

[7] To the utmost extent to the 11th floor.

5.2.2 Making Level

5.2.2.1 Solution Proposal

In the Solution Proposal step, the five fundamental questions, explained in the Methodology, need to be answered: What, How, Where, When and Who? From these questions, the functional requirements will be converted to technical requirements in order to come up with a solution suggestion (Table 5.4).

5.2.3 Proof of Concept

In order to be able to compare the IoT devices, capable of fulfilling the business requirements, on the market and the device(s) which can be made for minimal cost in-house/outsource, a PoC of this project has to be made. The PoC is intended to test if the functionalities offer a solution for these business requirements. The final design of the device will differ from the PoC, regarding the chosen hardware components and smaller technical details.

Based on the technical requirements and the base-station/add-ons principle, a BOM is created:

- Base-Station (+ RTC)
- Temperature Sensor
- Humidity Sensor
- Gas Sensors
- Light Sensor
- Network Module
- Current Measurement
- Voltage Measurement
- PSU

5.2.3.1 Base-Station

For the base-station, the choice between the wide range of OTS solutions was limited to the three most popular devices as they have the largest community support if wanting to consult similar projects for technical help. The choice between these three models (See Figure 5.4) is made based on the price and the necessary hardware capabilities (processor speed and memory).

The choice goes out to the Arduino (Figure 5.5) base-station. The Arduino Uno is built based on an Atmel ATmega328P MCU, which possesses all functionalities needed in the energy and comfort readings monitor. An important feature (which is not included in

TABLE 5.4: Conversion: Functional to Technical Requirements - Use-case

What?		
	Which (environmental) properties have to be measured?	Temp, Rel Humidity, CO_2, VOCs
	Which sensors are needed?	*Temp Sensor/Humidity Sensor/Gas Sensor More information in notes.*
	What is the intention of the measurement?	Forecasting, research and usage reduction
	What is done with the data?	*Processed in forecasting models : Matlab, Excel, etc.*
	Which metadata needs to be recorded?	*Time and Date*
How?		
	How exact do these measurements have to be?	No core scientific purposes
	Which resolution do the sensors need?	*See 'Hardware Constraints'*
	Is the sent data prone to interception?	Possibly
	Is data encryption necessary?	*No. No actuators can be controlled. The information is harmless in wrong hands.*
Where?		
	Is the measurement done on a distant/outdoor location?	The measurement area is indoors, easy-accessible and physically close to power supply/gateway.
	On how many different locations do these measurements have to be done?	*Each active outlet, electrical circuit and enclosed room is a separate node.*
	Which range does the measurement data have to be sent over to the gateway?	*Maximum 50 m (obstructed) Less if gateway is centered.*
	Which connectivity technology can be used?	*Bluetooth LE or LoRaWAN*
	How can the device be powered?	*AC outlets (wired or inductive coupling)*
	Does the device need backup power?	*Yes. In case of power outage, energy and comfort readings are still relevant.*
When?		
	What is the project life-cycle length?	Not specified.
	How much can the project cost?	*Minimal cost.*
Who?		
	Who will have to participate in the development/testing/deployment/etc?	Everybody working on floors 9-12.
	Is there in-house personnel, experienced in software/hardware/quality engineering?	Yes. The question will be if it is cheaper to deploy these people as developers or outsource/buy the project.

Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	$29.95	$35	$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog

FIGURE 5.4: Comparison of Arduino Uno, Raspberry Pi and BeagleBone [62]

the Raspberry Pi) is the integrated ADC converter, which is needed to read the signals from the analog sensors. The MCU works at a frequency of 16 MHz, incited by an external oscillator, which sufficient for the tasks at hand [57]. In order to be able to log

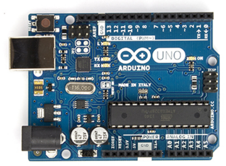


FIGURE 5.5: Arduino Uno [63]

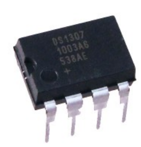


FIGURE 5.6: RTC DS1307N+ [64]

the metadata such as the date and time, a real-time clock (RTC) has to be connected to the Arduino Uno. The DS1307N+ (Figure 5.6) is power-efficient chip which saves its information in the 56 bytes NV SRAM (Non-Volatile Random-Access Memory) when the power supply shuts off.

In the DS1307, an integrated circuit detects if the main power has been shut off and will switch to the back-up power supply (battery - See Subsection 5.2.3.8). The RTC is suitable to run between temperatures of +85 and -40 degrees Celcius.

5.2.3.2 Temperature and Humidity Sensor

There is a broad range of sensors which can monitor temperature and relative humidity separately. The cheapest and most frequently used type, DHT22 (Figure 5.7) can measure both variables simultaneously.

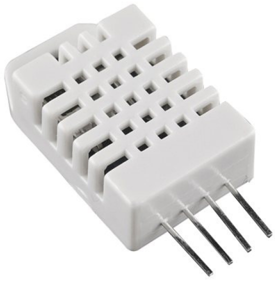


FIGURE 5.7: DHT-22
[65]

FIGURE 5.8: AS-MLV-P2
[66]

This sensor is capable of measuring temperatures between -40 degrees and +80 degrees Celsius, with an accuracy of +/- 0.5 degrees Celsius. The relative humidity can be measured from 0 to 100%, using a capacitive sensor with an accuracy of 5%. The data from the DHT22 can be measured every 2 seconds using a single digital pin (no ADC required).

5.2.3.3 Gas Sensors

The business requirements demand that the IoT device can monitor and log data concerning the relative amount (ppm) of VOCs (Air Quality) and CO_2 in the air.

For the VOC detection, the 'AS-MLV-P2 Air Quality Sensor' is chosen. This analog sensor shows a strong sensitivity to a wide range of reducing gases as can be seen in Figure 5.9. This sensor has a temperature operating range from 0 to 50 degrees Celsius, a humidity range from 5% to 95% and an experienced air velocity from 0 to 4 m/s[8].

As the amount of CO_2 particles in the air have to be monitored separately from the VOC data, a second carbon-dioxide sensor has to be integrated into the PoC. The T6615-10K-ND sensor is chosen as it measures gas particles based on diffusion (no strong air velocity needed) and it can measure up to 10.000 ppm of CO_2. [9] This sensor has identical working conditions as the VOC sensor and can log data with an accuracy of 75 ppm. This sensor can output data in both an analog or digital way so use of the integrated ADC is not essential.

[8]All of these properties comply with the office environment.

[9]Normal Outdoor Level: 350 - 450 ppm. Acceptable level: 600 ppm. Adverse health effects may be expected: 2.500 - 5.000 ppm. Maximum allowed concentration within a 8 hour working period: 5.000 - 10.000 ppm

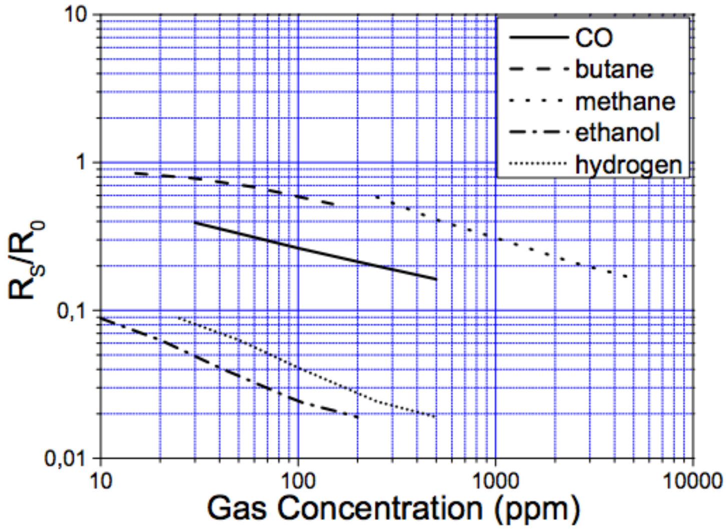


Figure 5.9: Sensitivities for various gas concentrations for AS-MLV-P2 [66]

5.2.3.4 Light Sensor

For the light [67] monitoring part of the IoT device, a low-cost digital-output sensor such as the LTR-303ALS-01 (Figure 5.10) is chosen. The LTR-303ALS-01 converts analog inputs from 0,01 lux to 64.000 lux. [10] The sensor has a temperature operating range from -30 to 570 degrees Celsius and has a function to ignore 50/60 Hz lighting flicker.

Figure 5.10: LTR-303ALS-01 [68]

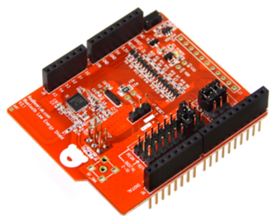


Figure 5.11: Redbear Lab BLE Shield [69]

5.2.3.5 Network Module

As mentioned in the Solution Proposal, Bluetooth LE and LoRaWAN are the two most suitable solutions for this IoT project. Technologies such as Zigbee and Z-Wave do not have the required range to serve as a solution for this problem (DIY Modules), while

[10] Recommended lux levels. Normal Office Work/PC Work: 500 lux. Detailed Mechanical/Drawing Work: 2.000 lux

Sigfox does not offer the required message size and sending frequency.[11] Cellular modems on each device can be considered overkill as purchasing costs and power consumption are tenfold when compared to BT modems.

The choice for BTLE is made, as the cost for a LoRaWAN Arduino shield was about 3 times as high as for a BTLE Arduino shield (99 USD vs 30 USD). The Redbear Lab BLE Shield (Figure 5.11) has an indoor range of 50 meters, is plug-and-play compatible with the Arduino Uno and support BTLE. As this is a ready-made module, the price will be higher than compared to SoCs[14]. This hardware component needs to be looked at in the Prototype phase as it can be replaced by a cheaper alternative, in combination with the know-how of an RF engineer.

5.2.3.6 Current Measurement

The measurement of current can be executed in a variety of ways: using a shunt (resistor), a Rogowski coil, a 50/60 Hz current transformer (CT) or a Hall effect sensor. In this IoT project, current will be measured using a transformer (CT) as the other alternatives are linked to some important disadvantages.

Measuring current using a shunt resistor does not offer galvanic isolation and the entire device is vulnerable to overcurrent. The Hall effect sensor needs an amplifier to measure current, which can lead to amplification of noise signals. Costs for Hall sensors and Rogowski coils are significantly higher than for a current transformer.

A magnetic coupling between two coils and a mutual soft-iron core makes it possible to measure current, using a CT. The first, primary, coil in a CT has a single winding while the secondary has a large amount of wire turns. The greater the amount of windings on the secondary coil, the greater the accuracy of the measurement. The to-be-measured current has to run through the core which creates the primary coil. A visual interpretation of this principle is given in Figure 5.12.

For the PoC design, the CST-1020 by Triad-Magnetics (Figure 5.13) is chosen. This CT can monitor AC current correctly (no saturation effects in magnetic core) up to 20 A. This is essential as the IoT device has to be deployed in both two-phase lighting circuits and AC outlet circuits [12].

[11] Sigfox allows up to 12 bytes per message and up to 140 messages per day

[12] The Belgian regulation for Electrical Equipment (AREI) dictates that a maximum of 16 A of AC current should flow through the wiring of a lighting circuit, while a maximum of 20 A is allowed for AC outlet circuits (non-industrial environment).

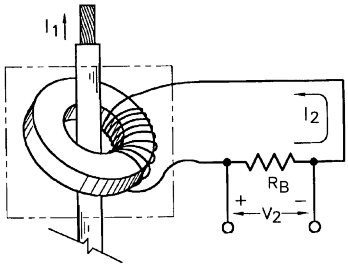


FIGURE 5.12: Principle of a Current Transformer
[57]

FIGURE 5.13: CST-1020 by Triad-Magnetics
[70]

The CST-1020 operates efficiently at the 50/60 Hz range and is designed to snugly fit a PCB. It has to be mentioned that this CT has a permanently closed primary windings which makes it more difficult to quickly move the IoT device from one circuit to another.

5.2.3.7 Voltage Measurement

Several techniques make it possible to measure AC voltage in an electrical circuit. The economically most efficient way is to use two resistor, positioned in series, and applying the concept of the voltage divider [71]. This method does not offer galvanic isolation and put the IoT device at risk, in cases of high voltage peaks. A safer, yet slightly more expensive method to monitor electric potential, is to use a potential transformer (PT) in front of the voltage divider[13].

The PT transforms the grid voltage to a lower potential, using two magnetic coupled coils (Figure 5.14). The voltage divider is connected to the secondary winding, which provides galvanic isolation and lower to-be-measure voltages. For the PoC, the SPK 05509 by Spitznagel (Figure 5.15) is chosen. This PT transforms 220-230 V to 9 V at the 50-60 Hz frequency range.

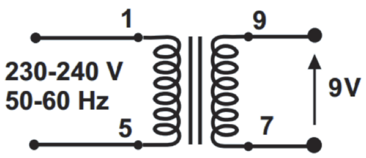


FIGURE 5.14: Principle of a Potential Transformer
[57]

FIGURE 5.15: Spitznagel SPK 05509
[70]

[13]Note that the resistors, needed for the voltage divider, are not included in the BOM. They are essential for the technical design, but their purchasing cost is negligible when compared to the other BOM lines.

5.2.3.8 PSU

As the PoC does not demand the ability of portability yet, which will be the case for the final product, it can be chosen to hard-wire the power supply units of both the energy monitor and the comfort readings device to the AC grid.

For the energy monitor, the design should serve the functionality of being able to measure both AC outlets as bare-bone electrical circuits. Also, the monitor should be able to send data, even when the electrical grid drops in current/potential. The components on the PCB need a constant potential of 5 VDC and a delivered power of less than 0,5 W (sum of components).

For the PoC's, the MYRRA 47122 is chosen, which is a switched-mode power supply[72]. This PSU can handle an input potential between 85-265 VAC and 85-370VDC, a frequency range of 47-440Hz and outputs a constant voltage of 5VDC (output power of 2.75W). This PSU can power the Arduino Uno, the Bluetooth module and the other hardware components on the board without problems.

In order to ensure data transfer if power would drop (blackout/brownout), the circuit is equipped with a backup battery. Power drops are strongly uncommon in Belgium [14], so it is not necessary to install a costly rechargeable battery. 2 Lithium CR123A batteries (See Table 3.2) deliver the correct amount of voltage and enough energy to power the Arduino UNO and its peripherals for approximately 36 hours.

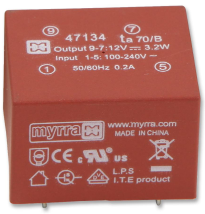


FIGURE 5.16: MYRRA 47122 [70]

FIGURE 5.17: CR123A Battery [70]

5.2.3.9 Conclusion

The business requirements of this use-case dictate the obtaining of measurement data of two different natures: energy and comfort. As the amount of to-be-tracked assets for each segment is different, it is recommended to create two different kind of devices. Both devices can share the basis of their hardware design: MCU/Base-Station, Connectivity Radio and PSU.

[14] Less than 1, every 3 years. Average length 2,5 hours. http://www.elia.be/nl/grid-data/data-download

The PoC of the energy tracker will consist of extra hardware to measure the AC current and voltage, while the environmental tracker will be equipped with gas, light, temperature and humidity sensors.

The PoC will differ from the final product (if chosen to be built in the 'Make-or-Buy' step) as the base-station will be replaced by a dedicated PCB, after the cost-cutting step, containing only the necessary properties of the multi-functional Arduino such as the MCU, RTC, needed amount of input pins, etc.

After having manufactured the PoC, the 'Make-or-Buy' question has to be analyzed and answered. The BOM list, made in the PoC step, will serve as a basis to calculate resource prices and has to be complimented by the firmware development costs as well as assembly and testing expenses.

5.2.4 Make-or-Buy

5.2.4.1 Introduction

In the 'Make-or-Buy' step, the best option from an economical point of view has to be chosen between the optimal Make-solution and the optimal Buy-solution. In order to acquire a total cost estimation for the Buy-solution, a market research has to be executed first to find the possible (if any) OTS solutions. The purchasing cost of these devices is compared to the cumulative cost (hardware/firmware/assembly/testing) for the final products developed in the Make-scenario.

5.2.4.2 Buy Option

A market research is executed and OTS solutions to the business requirements are found, by buying one or several different IoT devices. The basis of the market research can be built upon the initial business and functional requirements, stated in the use-case documents:

1. Electrical energy consumption per outlet in Office zone
2. Electrical energy consumption per light circuit in Office and Utilities zone
3. Temperature, relative humidity, amount of CO_2, VOCs, Light intensity in Office zone
4. Send all data to central gateway

TABLE 5.5: Comfort Readings Tracker - Market Research

	Birdi[73]	**KotoAir[74]**	**Cube Sensors[75]**
Temperature	+/- 0.3 degrees Celsius	Yes; Accuracy Unknown	Yes; Accuracy Unknown
Rel. Humidity	+/- 2%	Yes; Accuracy Unknown	Yes; Accuracy Unknown
CO_{2}	+/- 5 PPM	Yes; Accuracy Unknown	Yes; Accuracy Unknown
VOCs	+/- 2 PPM	Yes; Accuracy Unknown	Yes; Accuracy Unknown
Light Intens.	0.025 Lux/LSB	Yes; Accuracy Unknown	Yes; Accuracy Unknown
Data Transfer	BLE 4.0 - 802.11 b/g/n 2.4 Ghz	BLE 4.0 - Zigbee	BLE 4.0 - Zigbee
Power	USB/AA Battery* 70 mW *Back-Up 1 Year	USB	USB
Price	119 USD/Unit 999 USD/10 Units	125 EUR/Unit	Discont. in May 2016

An extensive market research revealed the fact that currently, there are no IoT appliances for sale which give solutions for all requirements in a single device. The market does offer a range of devices which meet requirements 1-2-4 and 3-4 separately.

For the comfort readings (requirements 3-4), three widely-available solutions are currently offered on the market (See Table 5.5). As CubeSensors will discontinue production in May 2016, it is unwise to choose this device, keeping future growth analysis and device compatibility in mind. The Birdi (Figure 5.18) device offers monitoring accuracies and a connectivity range beyond the client's needs, for a price which is lower than the alternative KotoAir (Figure 5.19).

The purchasing cost for 113 Birdi Smart Detectors, based on a personalized quote, is 11.328 USD including shipping and handling.

For the energy readings (requirements 1-2-4), three widely-available solutions are currently offered on the market (See Table 5.6).

The first three options (Figures 5.20, 5.21 and 5.23) in Table 5.6 are designed to track current, voltage and electric energy straight from a single AC outlet. All 3 devices are plug-and-play and can be moved around from one outlet to another. They are all similar in functionality and accuracy. The choice for this project goes out to the Aeotec

Table 5.6: Energy Tracker - Market Research

	WeMo[76] Insight Switch	**Aeotec[77] SmartSwitch 6**	**smart-me Plug[78]**	**smart-me Meter[78]**
Remote I/O	Yes	Yes	Yes	Yes (32 A mode)
Max. Voltage	230 VAC	230 VAC	230 VAC	85-253 VAC
Max. Current	16 A	15 A	16 A	32/80 A
Frequency	50 Hz	50 Hz	50 Hz	50/60 Hz
Power	3680 W	3450 W	3680 W	7.3/18.4 kW
Data Transfer	WiFi 2.4 Ghz 802.11n	Z-Wave (Gen5) *higher range	WiFi	WiFi
Range	150 meters	150 meters	150 meters	150 meters
Price	50 USD/unit	50 USD/unit	100 EUR/unit	190 EUR/u

SmartSwitch 6 due to the lower cost and flush design, which makes it possible to plug them into a power strip.

The Aeotec SmartSwitch 6 will be used solely for the AC outlet tracking. For the light circuits, market research shows only one option: the smart-me Meter (Figure 5.22). This device is a single phase electricity meter which is hardwired to an electrical circuit and can be mounted into an electrical cabinet. The difference, compared to the first three devices is that the smart-me Meter can handle a larger electric power. Also, the wiring is done in a more rigid and durable way than for the AC outlet monitors.

In total, 80 smart-me Meters are needed for floors 9-12, which amounts to a purchasing cost of 15.200 EUR. 782 Aeotec SmartSwitch devices will bring along an expenditure of 39.100 USD. Combined with the purchasing cost of 11.328 USD for the 113 Birdi Smart Detectors, a total CAPEX of 60.253 EUR[15] has to be made.

Purchasing costs of OTS IoT solutions already include the cost of firmware development, device assembly and factory testing. However, this CAPEX does not include the cost of installation on the asset(s) and configuration of the device. These actions appear to have a momentarily behaviour, as they only have to be executed once.

Yet, discussions with IoT agencies made clear that 1 FTE[16] with sufficient technical skills is needed to do the maintenance[17] of a measurement project of this size. As this particular project is deployed in the offices of a UGent research facility, the annual gross salary for a scale 2 technical employee with 3 years experience [79] is 21.360 EUR.

[15] 1 EUR is 0,8934 USD at time of publishing thesis

[16] This indicates the workload of 1 employed person. This can be divided over several persons.

[17] Updating of firmware, further developing of dashboard for measurement readings, replacing batteries, repositioning devices, etc.

The total cost for the Buy-decision can be found in Table 5.7.

TABLE 5.7: Total Cost: Buy-Decision

	CAPEX	**OPEX**
AC Plugs	782 x 50 USD	/
AC Meters	80 x 190 EUR	/
Comfort Readers	113 x 100,25 USD	/
Installation and Maintenance	/	21.360 EUR/year
Total Cost	60.253 EUR	21.360 EUR/year

FIGURE 5.18: Birdi Smart Monitor

FIGURE 5.19: KotoAir

FIGURE 5.20: WeMo Insight Switch

FIGURE 5.21: Aeotec SmartSwitch 6

FIGURE 5.22: smart-me Meter

FIGURE 5.23: smart-me Plug

TABLE 5.8: BOM Final Product - Energy Tracker

	Function	**Unit Cost (1) USD**	**Unit Cost (10) USD**	**Unit Cost (100) USD**	**Unit Cost (1.000) USD**
ADE7753	IC	1,85	1,85	1,85	1,85
ATmega328P-PU	MCU	3,70	3,30	2,71	1,84
CST-1020	CT	6,06	5,02	4,92	4,49
Spitznagel SPK 05509	VT	7,18	7,02	6,54	6,01
MYRRA 47122	PSU	11,92	11,92	9,75	7,80
CR123A	Battery	4,00	3,82	3,32	2,89
MOD-nRF8001	BTLE	11,18	10,06	8,94	8,94
RTC DS1307N	RTC	3,78	3,58	2,83	2,03
AC Plug Male	PSU	1,67	1,45	1,34	0,89
AC Plug Female	PSU	1,73	1,54	1,39	0,92

5.2.4.3 Make Option

While the total cost of the Buy option was composed of the purchasing and maintenance expenses, more cost building blocks (Figure 2.11) have to be taken into account when evaluating the Make option. The hardware purchasing cost can be based upon the BOM of the PoC, after having made some adjustments while the firmware, assembly and testing expenses are based on the assumptions made in Chapter 4.

As the amount of assets for energy tracking does not equal the number for comfort readings, two seperate device are manufactured. The energy tracker is built to be able to track both AC outlets as lighting circuits.

BOM adjustment The base-station from the PoC step has to be replaced by dedicated hardware which will be more reliable, cheaper and compact than having to purchase a full-blown Arduino for each device. The energy monitor's Arduino can be replaced by a multimeter IC, which is designed to measure current and voltage input signals, and a MCU which will do the calculations before sending data to the radio module. The comfort readings sensor can be directly connected to the MCU. The network module will not be replace by a SoC, as this would only be cost-effective for device numbers over 100k-300K [14]. PCB and enclosure costs will add to the final hardware costs.

A Bill of Materials, combined with the costs, for the Prototype/Final product is given for both the energy as the comfort monitor in Tables 5.8 and 5.9.

Table 5.9: BOM Final Product - Comfort Tracker

	Function	Unit Cost (1) USD	Unit Cost (10) USD	Unit Cost (100) USD	Unit Cost (1.000) USD
ATmega328P-PU	MCU	3,70	3,30	2,71	1,84
DHT22	Temp/Humidity	2,98	2,68	2,68	2,23
LTR-303ALS-01	Light	1,14	0,89	0,60	0,45
AS-MLV-P2	Gas	25,17	21,7	17,16	17,16
CR123A	Battery	4,00	3,82	3,32	2,89
MOD-nRF8001	BTLE	11,18	10,06	8,94	8,94
RTC DS1307N	RTC	3,78	3,58	2,83	2,03
AC Plug Male	PSU	1,67	1,45	1,34	0,89
MYRRA 47122	PSU	11,92	11,92	9,75	7,80

Table 5.10: Cost of Enclosures and PCB - Use Case

	Enclosure Unit Cost (1) USD	Encl U.C. (100) USD	Encl U.C. (1.000) USD	PCB U.C. (1) USD	PCB U.C. (100) USD	PCB U.C. (1.000) USD
Comfort Readings Tracker	7,42	5,10	4,74	44,54	3,02	2,64
Energy Tracker	18,51	12,06	11,60	70,72	9,10	8,71

The cost of the PCB and the electronics enclosure is based upon the space which is occupied by the hardware components. Based upon the components in the BOM, an enclosure of 112 x 112 x 60 mm (L x W x H) is ideal for the energy monitor, while the comfort readings tracker can fit into a casing of 60 x 60 x 35 mm. Costs for enclosures and PCBs can be found in Table 5.10.

Hardware Cost Taking into the account the needed numbers of devices, calculated in the Project Dimensions step, a total hardware cost of **61.801,49 EUR** is calculated. The total hardware cost for the Make-decision can be found in Table 5.11.

Table 5.11: Total Hardware Cost Estimation: Make-Decision

	CAPEX
Comfort Readings Monitor	113 x 57,51 USD
Energy Monitor	862 x 64,68 USD
Total Cost	61.801,49 EUR

Assembly Cost Based on the assembly data and cost calculation tool[18] explained in Chapter 4, assembly costs for both batches of IoT devices can be estimated (See Table

[18]http://www.7pcb.com/PCB-Assembly-Quote.php

Table 5.12: Assembly Cost Estimation - Use-Case

	Board Quantity	**BOM Lines**	**Lead Free**	**Thru Holes**	**Fine Pitch Parts**	**BGA /QFN Parts**	**Cost USD**
Comfort Reading	113	8	Yes	67	7	2	2.225,07
Energy Tracker	862	10	Yes	88	8	0	8.889,08

Table 5.13: Firmware Cost Estimation - Use-Case

Parameters					
A=3,60					
B=1,20					
C=2,50					
D=0,32					
KLOC	1	2	3	4	5
Effort (in persons x months)	3,6000	8,2706	13,4539	19,0009	24,8351
Duration (in months)	3,7667	4,9153	5,7434	6,4143	6,9881
Staffing (in persons)	0,9558	1,6826	2,3425	2,9623	3,5539
Cost in EUR	16.200	37.218	60.543	85.504	111.758

5.12) and account to a total of **9.929,55 EUR**.

Firmware Cost The COCOMO cost estimation tool strongly depends upon the lines of source code and nature of the project, which are the main parameters in this equation. Coding for both devices can be considered as 'Embedded projects'[19]. The number of lines of code (in thousands) remains uncertain.

A range between 1.000 and 5.000 lines of code is visualized in Table 5.13 together with the estimated duration and cost of the project. Discussions with the employees of IoT agencies advice that similar projects, which are based upon reading out basic sensor data and sending this information to a gateway using a radio module, take around 6 months to complete. COCOMO estimates that an embedded project of 3.000 lines of code takes around the same amount of time and will cost approximately **60.543 EUR**.

Testing Cost As explained in the Management Methodology, testing is considered an ongoing process which does not end after the final product has been designed, produced and installed. Testing includes the act of evaluation prototypes as well as maintaining, updating and configuring (newly) installed devices.

[19]Effort=$A * KLOC^B$; Duration =$C * Effort^D$; Staffing=$Effort/Duration$

Table 5.14: Testing Cost Estimation - Use-Case

	CAPEX	OPEX
Installation & Maintenance	/	21.360 EUR/year
CE Certification	7.900 USD	/
In-Process Testing	46.000 EUR	/
Total	53.058 EUR	21.360 EUR/year

Table 5.15: Total Cost Estimation: Make Option

	CAPEX	OPEX
Hardware	61.801 EUR	/
Assembly	9.930 EUR	/
Firmware	60.543 EUR	/
Testing	53.058 EUR	21.360 EUR/year
Total	185.332 EUR	21.360 EUR/year

Just as in the Buy-scenario, a FTE technical employee is needed for the day-to-day operations. For the testing operations, before the final production, two extra quality engineers (specialized in hardware and software) are required. Using specialized equipment, they have to cooperate for the duration of the project (+/- 6 months).

If one or both manufactured devices would have a commercial future (being sold), a CE-certification is required.[20] The total cost estimation of testing can be observed in Table 5.14.

5.2.4.4 Conclusion

Tables 5.15 and 5.16 clearly show that, for this case, the OPEX for both scenarios are identical. The CAPEX for the Make-scenario is around 3 times as high compared to the Buy-decision, taking into account all cost building blocks. It can be concluded that it is economically more attractive to go for the Buy-option.

It has to be noted that the hardware and assembly costs are dependent upon the volume of devices which are produced. The estimated numbers are projections in case that the project would commence at the current room occupation. As explained in Subsection 5.2.1.4, it is possible that the amount of devices need to go up. A reprisal of the total estimated cost is carried out and elaborated in Subsection 5.2.5.

[20]CE marking is a mandatory conformity marking for certain products sold within the European Economic Area

Table 5.16: Total Cost Estimation: Buy Option

	CAPEX	OPEX
Hardware	60.253 EUR	/
Assembly	/	/
Firmware	/	/
Testing	/	21.360 EUR/year
Total	60.253 EUR	21.360 EUR/year

5.2.5 Future Actions

5.2.5.1 Introduction

The Future Actions step focuses upon the transgression from the Making phase towards the Execution Phase. For this use-case, the questions which are normally asked (Production Date? Installation Date? Etc.) are irrelevant as the factual purchasing of devices for this use-case is not included in the scope of this thesis. Instead, it is interesting to see how the cost estimation evolves when expanding the number of users.

5.2.5.2 Cost Evolution: Buy Option

In the current case, 113 comfort trackers (Birdi), 782 AC energy trackers (Aeotec SmartSwitch 6) and 80 circuit energy trackers (smart-me Meter) have to be purchased installed and configured. In Subsection 5.2.1.4, it is made clear that, when occupying the office and technical rooms at full capacity, the number of comfort and circuit energy trackers remain constant while the number of AC energy trackers can rise to 1.042.

For the Buy-option, the only applicable cost building blocks are hardware and testing (maintenance) costs. The operational costs remains the same, as 1 FTE can manage to maintain and configure new and existing devices. A linear evolution of the partial hardware costs (left - for the Aeotec Smartswitch) as well as the total project cost (right - combined with purchasing Birdi and smart-me Meters) can be observed in Figure 5.24.

5.2.5.3 Cost Evolution: Make Option

For the Make-option, the total cost depends on all 4 cost building blocks. Again, the number of comfort trackers will remain constant, while the number of AC trackers has to expand. Enlarging the scale of the project will result in growing hardware and assembly costs, while firmware development costs and testing expenditures will remain constant (as well as the operational costs). A graph of the total hardware costs for the entire

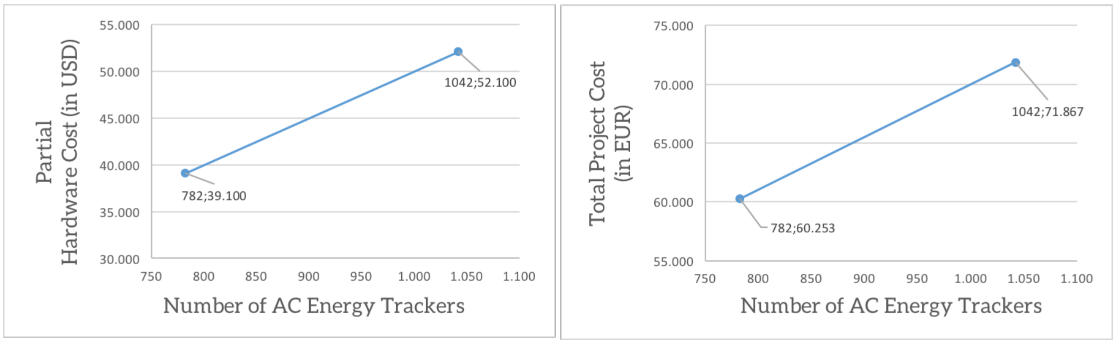


FIGURE 5.24: Partial Hardware and Total Project Costs (CAPEX) vs Number of AC Energy Monitors

project as well as a visualization of the total project costs for the 'Make'-scenario can be observed in Figure 5.25.

As clearly shown in the graphs, a cost drop occurs around the 1.000-units manufacturing point. Initially, the project has a device amount need, lower than the price-drop point. If the 'Make'-scenario would be considered as more profitable, it could be worthwhile producing the whole amount of 1.042 AC Energy trackers, as it would amount to the same cost as producing around 950 units otherwise.

5.2.6 Conclusion

By applying the management methodology on the use-case, the business requirements are converted to functional and technical requirements. A BOM and cost estimation is created for PoC devices.

Furthermore, a market analysis is executed and the 'Make' and 'Buy' scenarios are dissected using the four cost building blocks. Total project costs for both scenarios are calculated, from which it can be concluded that, for this particular use-case, 'Buy' will always be the optimal solution. It has to be taken into account that this comparison is done, solely from an economical point of view and for the current contextual requirements.

Fixed, overhead costs such as firmware and testing lead to a high starting point and therefore, no break-even point can be found in this use-case, for the amount of needed devices. The cost for firmware was based on a rudimentary estimation, as the number of source code lines is unknown. Even if the cost for firmware would be discarded from the equation, the 'Buy'-scenario would come out on top.

If the contextual requirements (maximum number of devices based on physical space) would be eliminated, it is possible to find a break-even point. Assuming the hardware

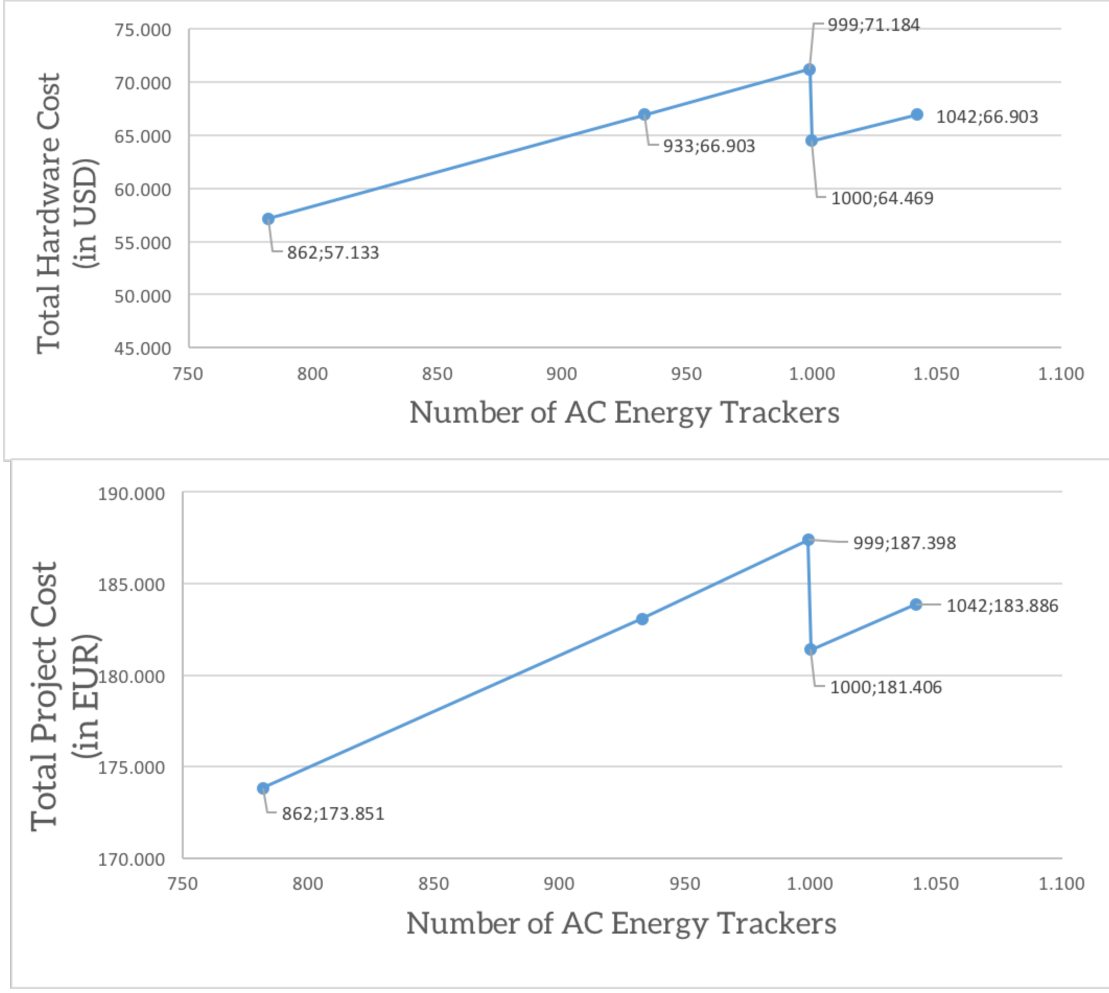


Figure 5.25: Total Hardware and Total Project Costs (CAPEX) vs Number of AC Energy Monitors

costs per unit will not drop any further, after having reached 1.000 units, the 'Make' decision will become favorable when manufacturing approximately 20.000 (≈ 19.684) AC energy trackers.

As a final clear overview, a visual representation of the current project costs for both scenarios and the lack of break-even point is given in Figure 5.26.

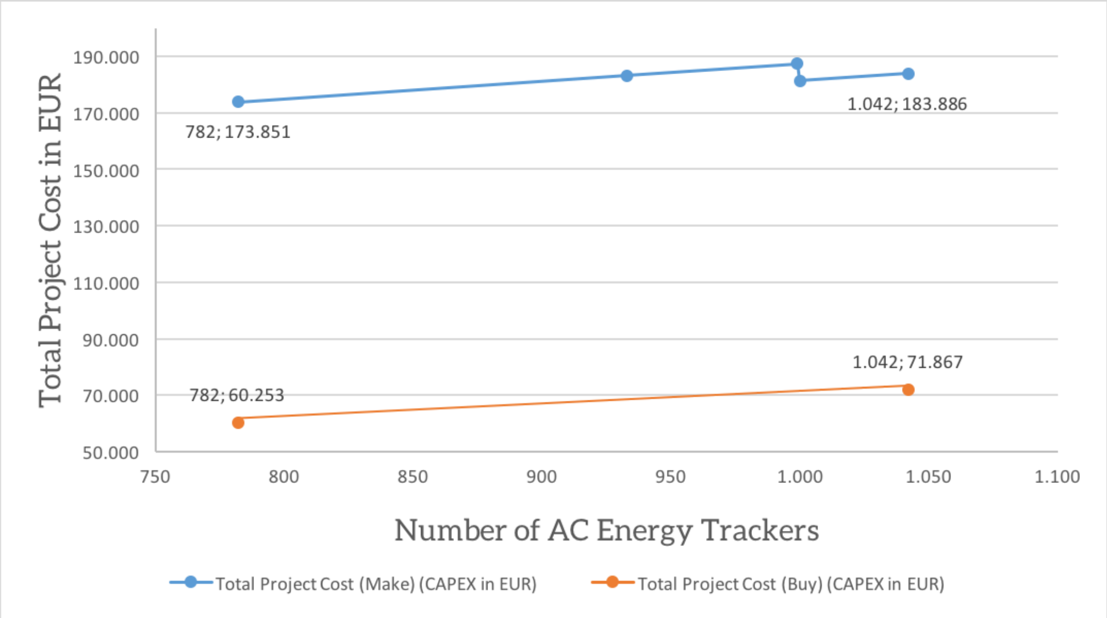


Figure 5.26: Total Project Costs (CAPEX) in EUR vs Number of AC Energy Monitors: 'Buy' versus 'Make'-Scenario

Chapter 6

Conclusion

In the first part of this Master's thesis, a development path for a generic IoT project is created, based on a building block methodology. This is done with the eye on trying to optimize both the device cost as the total project cost at all time, which is not always possible, as seen in the discussed use-case.

Each IoT project can be divided into 3 major phases, which are able to interconnect in a number of ways, similar to software project planning. These three phases - Planning, Making and Executing - can be subdivided into a series of easy-to-understand and actionable sub-tasks. The first and second phase focus on analyzing and converting the business requirements to functional and technical requirements by means of location screening and extraction of the project dimensions. A technology-independent solution proposal forms the basis for a PoC and is designed before the Make or Buy question is considered. After performing a market research, based upon the technical requirements, the BOM of the prototype is created and a first hardware cost estimation can be executed.

Based upon the cost building blocks, elaborated in Chapters 3 and 4, a total expected project cost is defined and a Make or Buy decision is made. The evolution of the number of users is forecasted and the relationship with the number of assets and events is defined. Based upon this forecast, a projection of the cost evolution can be made. Changing volume parameters can shift the preference from the Buy-option towards the Make-option or visa-versa. The future growth analysis is followed by setting out a number of milestones concerning project deadlines (production/testing/installation), related to the final, Execution phase.

This third phase of each generic IoT project strongly depends on the Make of Buy choice. For the Buy option, Executing is limited to installing, configuring and maintaining the

OTS-bought device. In case of a Make decision, the successive steps of Prototyping, Pilot series and Mass-Production have to be followed. Each of these steps is encompassed by continuous evaluation and a feedback loop to the hardware design and future actions steps. By following this IoT roadmap, combined with an intermediate technical background of the reader, a full-blown project design can be created from scratch, based upon the client's business requirements and grounded by the 'Buy' or 'Make' decision.

In the second part of this thesis, the creation of a cost estimation methodology for these IoT projects is documented. Project expenditure estimates are based upon 4 building blocks: hardware, assembly, firmware and testing. For the hardware part, an exposition of the basic electronic components in an IoT device is given, by means of a generic hardware catalog and corresponding cost drivers. For the 3 remaining cost building blocks, cost estimation methods are explained, based upon best practice rules and industry calculations. Combined with the 'Buy' or 'Make' analysis from the management methodology, it shows that in frequent cases, the solution with the minimal device hardware cost ('Make') will not necessarily be the optimal choice, when comparing the total project cost to the 'Buy' decision.

The third and final part of this document applies both the project management as the cost estimation methodology to a real-life use-case, provided by the thesis' supervisors. The objective of this case study is to setup an instrumentation and measurement campaign to precisely monitor and describe the state and structure of energy usage/comfort readings and to give an overview in time of the consumption of energy and its metadata for the common area of a real office environment and building. The business requirements are analyzed and project propositions are created for both the 'Make' and 'Buy' scenarios. Capital and operating expenditures for both decisions are estimated and the trade-offs are analyzed. The purchase of OTS AC energy trackers, circuit energy trackers and comfort readers is compared to the self-production of hardware solutions, combined with the accompanying firmware, assembly and testing costs. Applying the cost estimation method to the use-case shows that the 'Buy' decision is more cost-effective than the alternative, for device numbers under 20.000.

In general, the 'Buy'-decision will remain favorable for low amounts of manufactured units. Due to the overhead costs of firmware and testing, the 'Make' decision becomes relevant for production numbers over 10k-100k. A widely-applicable and fixed number for this limit cannot be given as it strongly depends on the nature of the IoT project. Therefore, each project should be analyzed using the proposed methodologies.

In this thesis, a unique project methodology is created and tested on a single use-case. Both the management as the cost estimation methods have proven their worth while solving this case-study of moderate proportions. This document focused on the IoT device itself and did not pan its scope towards the gateway and cloud-processing nodes of projects with larger dimensions. It can be considered interesting to combine the created cost estimation methodology with the higher levels of IoT projects, in order to obtain a more complete OPEX-estimation of subscription fees for data transfers, cloud-server renting, etc.

Case studies with devices numbers over 10-100k will prove the value of economies of scale in hardware purchasing, which is not the case in the example examined in this thesis. Also, in order to obtain a more accurate cost estimation for "Make"-decisions, it can be considered valuable to construct a third methodology which solely focuses on IoT firmware costs and its influencing parameters. A close cooperation with IoT software companies seems key to succeed in this future project.

Appendix

USE CASE: Energy Lab @ IBCN

"Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it." — H. James Harrington

Revision: v1.04
Date: 04/12/2015

Part I: Measurement setup

Scope

The scope of the project is to contribute to an **increased understanding of the overall energy consumption for different types of (ICT) equipment**, including the consumers' behavior and comfort levels, to identify use and demand trends and to **allow a complete building envelope energy performance assessment**.

Objectives

The objective is to **setup an instrumentation and measurement campaign to precisely measure and describe the state and structure of energy usage and to give an overview in time of the consumption of energy and its metadata for the common area of a real office environment and building**. To define the common area within the project, we focus on the floors of the future office building that the IBCN research group will take into use. The common area including among others the offices, building utilities and datacenter.

The campaign will produce reference information that will allow research teams, and organizations that work in the modelling and forecasting of energy consumptions, to base their works on reliable data and on sane basis. No pertinent action can save the cost of a sharp analysis of the initial situation. This project aims at describing as carefully as possible the state of the energy consumers in an office building. The descriptive approach is one of the most important contributions of this project.

The campaign will include monitoring of (ICT) equipment, HVAC, comfort parameters, outside weather conditions, etc.

The campaign will generate and support research and will be kept as open as possible, will be easily reproducible in different environments and able to grow based upon new requirements. It is scheduled to start as soon as the future office building for IBCN is effectively taken in use. No end date will be set.

General characteristics

As stated in the objective, the goal of the project is **to create a dense, long-term monitoring setup allowing future research to extrapolate the data to a detailed breakdown**.

Initially, the campaign will **focus on energy usage and flow and on comfort related measurements**.

The **datasets** will be **augmented with metadata**, increasing their value and ease of use, including device registrations and explicit equipment inventories. The combination of energy related data, utilization statistics, and meta-data will allow us to answer several open questions:

- What is the contribution of e.g. computing systems to an enterprise's overall electricity consumption and waste, and how is this cost distributed across different components of the computing infrastructure?
- If existing research makes power analyses based on isolated research lab measurements: how do different assumptions and methodology techniques hold in a larger enterprise setting?
- We will heavily instrument the infrastructure as we do not know what we will find; Once that we have an understanding of the data, how would one design a measurement infrastructure to achieve good accuracy with the least effort? The answers to these questions form a fundamental contribution of this project.
- Detailed examination of where energy goes reveals in what parts the electricity is spent. Based on this, decisions can be made.
- The deployment and data studies should eventually expose the relative importance of device coverage versus duration of deployment. We want to be able to answer questions like 'what to measure' and over what time scale.

Research partners involved

- IBCN-GreenICT is the initiator of the project.

Current needs for the project

- Lay out a foundation to build on
  - We need supporters
    - Within the next days and weeks we will expose the project proposal to people at key positions to validate if there is a firm foundation.
    - If there is an interest for supporting the project we will ask these people to:
      - Generate some proposals towards research projects based on or making use of the data set coming from this project. (Validation of the foundation)
      - Provide a list of possible new supporters
- We need users of the data set. These will come from project supporters.
- Help us find new possibilities for research based for the measurements
- How can we initiate a knowledge build up so we can share to the industry
- What to measure and in what resolution (spatial and temporal)

Current questions

How to design a metering network: buy or make

- What kind of data do we want to capture:
  - Energy consumption;
  - Power consumption;

- Presence detection;
    - …
- At what level of detail do we want to capture data:
    - Per outlet;
    - Per electrical circuit;
    - Per office;
    - Per desk.
- At what resolution do we want the data capturing in terms of:
    - Precision (e.g. Wh or kWh);
    - Time resolution (e.g. Data per second, minute, etc.).
- Internet of things (how to connect all sensors and gather the data within real time embedded devices, possibly using dynamically code updating etc.);
- How to make it future proof and industry usable.

Part II: Potential project outcomes

Research value

Research value of this project is manifold. We do not focus on one domain:

- Every environment has a tremendously diverse set of devices that exhibit huge variations in workload and configuration and exist under several overlapping administrative domains. Improving the efficiency of such a system requires detailed data of both energy consumption and energy waste. **Characterize energy data** from micro to macro scale (e.g. from individual to building). How the individual energy consumption data points relate to the full building energy use (e.g. extrapolation). This will help to visualize (e.g. hot zones) into exact detail **where the energy is going** (e.g. which devices) and how much of it is spent usefully and wastefully.
- Evaluate the **potential savings** that can be achieved by substituting the solutions in place by energy efficient solutions.
- What are **opportunities to reduce the energy waste** of certain systems? For example, many people leave their computers on overnight, even when they are not needed, or have power-hungry PCs for undemanding tasks such as document-processing and web browsing. Observations like these have motivated recent research into green computing work[1] [2] [3].
- Very sharp mapping of the **lighting energy consumption** by monitoring every light source in every area. And therefore evaluating the efficient solutions and control algorithms.
- Discover how using different equipment affects the energy consumption and, ideally, help to cut the electricity consumption.
- Correlation e.g. **utilization vs energy**
- **Improve building modeling**, fitting part of a building with sensors to measure its operational characteristics will make it useful as a living laboratory to calibrate existing energy models and develop new, more accurate models.
- **Visualization of the energy flow** that will enable researchers to identify discrepancies in the predicted versus actual energy balance.
- **Before and after studies** like evaluation of control algorithms or the effect of campaigns like e.g. 'Turn off the lights when you leave.'
- **Error detection** at a whole new building automation level e.g. in heating or water systems.
- **Quantification of energy variation** between:
    - device classes
    - within device classes
    - For individual devices. This analysis identifies simple optimizations, such as changes in display settings which could lead to significant energy savings.
- **Waste identification**:
    - Where can we reduce and how can this lead to new usage scenarios.
    - Where do we have to rethink (system design)?

In addition to the improvement of building physics/building envelope and building systems (heating, cooling, etc.), the targeted use of room and building automation contributes significantly to the

[1] Yuvraj Agarwal, Stefan Savage, Rajesh Gupta, SleepServer: a software-only approach for reducing the energy consumption of PCs within enterprise environments. in: USENIX Annual Technical Conference, 2010.

[2] Tathagata Das, Pradeep Padaala, Venkata Padmanabhan, Ramachandran Ramjee, Kang Shin, LiteGreen: saving energy in networked desktops using virtualization. in: USENIX Annual Technical Conference, 2009.

[3] Sergiu Nedevschi, Sylvia Ratnasamy, Jaideep Chandrashekar, Bruce Nordman, Nina Taft, Skilled in the art of being idle: reducing energy waste in networked systems, in: Proc. Networked Systems Design and Implementation, 2009.

energy-efficient operation of buildings. This potential is based on the application of bus systems and automation technologies. Building automation in particular is a necessary instrument for maintaining the energy-efficient operation of buildings through continuous energy and building management. In order to analyze and advance this topic systematically this project could aid in the performance of a variety of experimental and theoretical research work. The campaign could help to understand the interplay between the essential parameters needed for the sustainable operation of a building. These essential parameters can be found in:

- Aspects of use;
- Energy efficient operation of the building;
- Building envelope;
- And building systems (including technical systems used for heating, cooling, ventilation and the power supply of the building)

The project will also aid in coming to better methodology guidelines for the green computing community.

Related project outcome

Potential project outcomes are:

- **Educating and making users aware** about their energy use, e.g. how do users use energy (do we still know how to turn off things?). The goal could also be to make information and tools available for people to manage their energy consumption and encourage people to use less energy (e.g. 'your personal energy trainer').
- Standby powers are very particular, because there is no service at all associated with their demands. Their consumption appears like wasted energy, and most of them are probably avoidable, generally at a relatively low cost. Therefore **standby power** could be **monitored and analyzed** in order to define as precisely as possible the nature and the extent of the associated consumption.
- Better oversight and **management of the energy use** with a real-time data set.
- Exploring the usage of the building and its content.
- Lead the way to **future instrumentation and measurement campaigns** in:
  - Households
  - Data centers
  - Office buildings
  - Industry (production lines)
  - Railway (tram and/or train)
- **Identify policies that prevent energy conservation**: such as a nightly backup policy that requires desktops to be kept on overnight even though backups only take an hour. Policies that will lead to better energy use and awareness.
- **Provide a test bed** to roll out future experiments
- **Provide more accurate data** for forecasting with predictive models.
- **'It allows to know and to understand.'**

The (green) ICT research community can benefit from the availability of more extensive power measurements. Also non-research related objectives can be deduced e.g. an energy savings campaign to reduce the University's energy consumption by a certain amount, this amount coming from profound and in depth research.

Part III: Tentative cost estimation

The instrumentation and measurement setup within the future office building will be spread over two distinct zones, the office & building utilities zone and the datacenter zone. For each zone one or more scenarios will be defined. Each scenario will describe the measurements included within that scenario. For all scenarios, several types of implementation will be reviewed.

Office & building utilities zone

The building itself, where the instrumentation and measurements will take place, is a multi-floored building. The definition of the office & building utilities zone will be general and applicable throughout the entire building. The cost estimation will give an overview per number of instrumented floors.

The office zone is defined to include the spaces within the building intended for day-to-day use by the occupants of the building. For example, the office zone includes:

- Office spaces;
- Meeting rooms;
- Hallways;
- Kitchens;
- Sanitary.

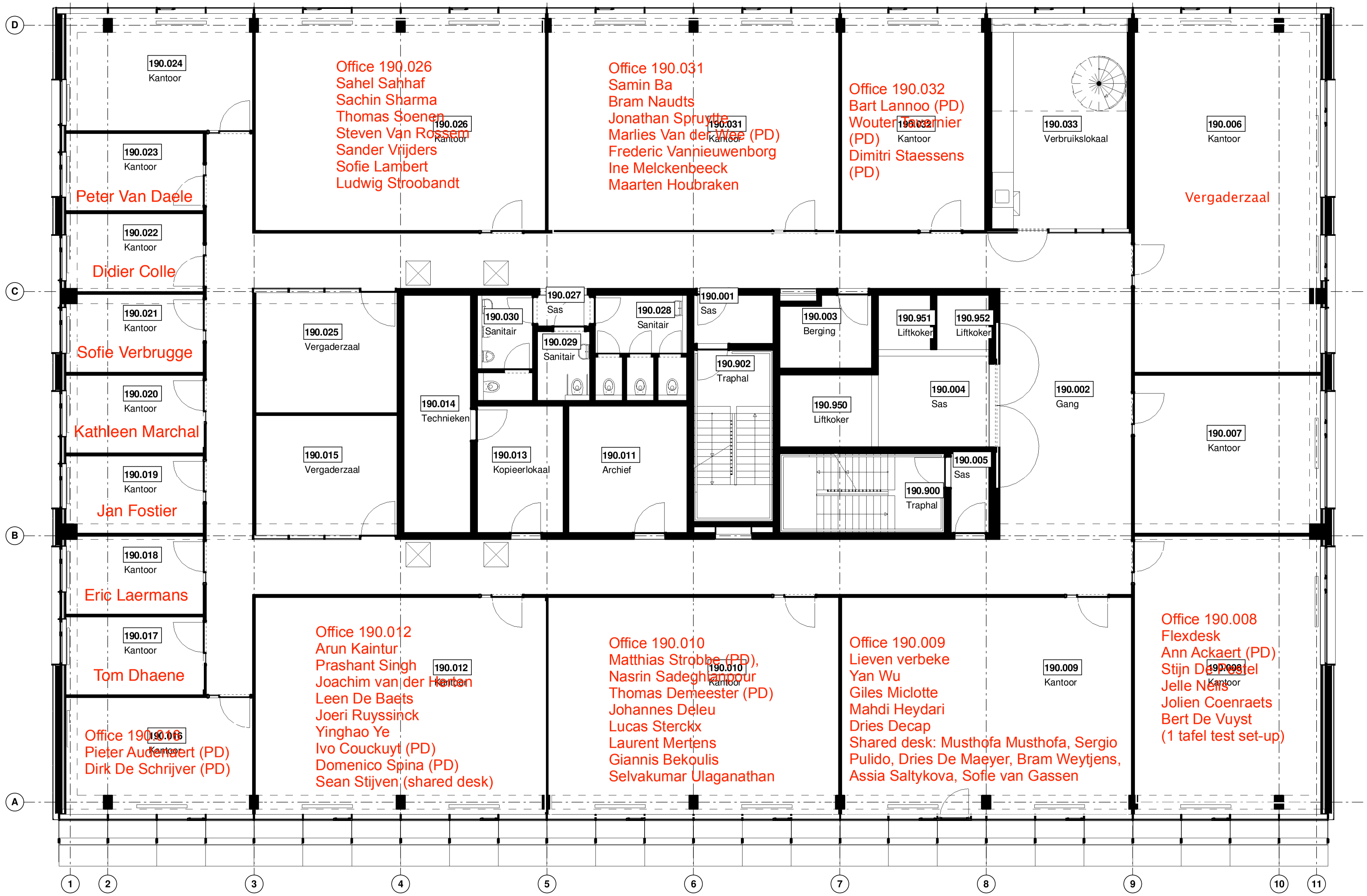
The building utilities zone is defined to include the spaces within or outside the building that are intended for and to make room for technical equipment. For example, the building utilities zone includes:

- Electrical cabinets;
- Technical rooms.

Scenario

The scenario includes measurements of:

- For the office zone:
    - In the office spaces and meeting rooms:
        - Electrical energy consumption per outlet for each desk, 4 outlets per desk;
        - Comfort readings
            - Temperature, relative humidity, amount of CO2, VOCs, Light intensity.
- For the building utilities zone:
    - Electrical energy consumption per electrical circuit (outlets & lighting).



BB	EA05-EA06-TT-DC	Grondplan 09	
	Zwijnaarde	Datum: 31-10-2012	DETAILPLAN
	Project: A/00682, F.I.: 60.33, Archief DGFB: O A	Schaal: 1/125	
	EVR-ARCHITECTEN BVBA	Formaat: A3	A 0 0 0 PD 09 -
	i.s.m. BUREAU BOUWTECHNIEK NV	Opp.: 0.12 m²	

TEK: PP H:\SAMENWERKING\evr-Architecten Autodesk Revit Architecture 2012

BB	**EA05-EA06-TT-DC** Zwijnaarde Project: A/00682, F.I.: 60.33, Archief DGFB: O A EVR-ARCHITECTEN BVBA i.s.m. BUREAU BOUWTECHNIEK NV	Grondplan 10 Datum: 31-10-2012 Schaal: 1/125 Formaat: A3 Opp.: 0.12 m² DETAILPLAN A 0 0 0 PD 10 -

TEK: PP

H:\SAMENWERKING\evr-Architecten

Autodesk Revit Architecture 2012

BB

EA05-EA06-TT-DC

Zwijnaarde

Project: A/00682, F.I.: 60.33, Archief DGFB: O A

EVR-ARCHITECTEN BVBA
i.s.m. BUREAU BOUWTECHNIEK NV

Grondplan 11

Datum:	31-10-2012
Schaal:	1/125
Formaat:	A3
Opp.:	0.12 m²

DETAILPLAN

A 0 0 0 PD 11 -

TEK: PP   H:\SAMENWERKING\evr-Architecten   Autodesk Revit Architecture 2012

INTEC - IBCN - Design
ELIS - CAS

BB	EA05-EA06-TT-DC Zwijnaarde Project: A/00682, F.I.: 60.33, Archief DGFB: O A EVR-ARCHITECTEN BVBA i.s.m. BUREAU BOUWTECHNIEK NV	Grondplan 12 Datum: 31-10-2012 Schaal: 1/125 Formaat: A3 Opp.: 0.12 m² DETAILPLAN A 0 0 0 PD 12 -

TEK: PP H:\SAMENWERKING\evr-Architecten Autodesk Revit Architecture 2012

Bibliography

[1] Wikipedia. Wearable technology — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Wearable_technology&oldid=716901574. [Online; accessed 28-May-2016].

[2] R.M. Dijkman, B. Sprenkels, T. Peeters, and A. Janssen. Business models for the internet of things. *International Journal of Information Management*, 35(6):672 – 678, 2015. ISSN 0268-4012. doi: http://dx.doi.org/10.1016/j.ijinfomgt.2015.07.008. URL http://www.sciencedirect.com/science/article/pii/S0268401215000766.

[3] Wikipedia. Cost estimation models — wikipedia, the free encyclopedia, 2015. URL https://en.wikipedia.org/w/index.php?title=Cost_estimation_models&oldid=647516667. [Online; accessed 23-February-2016].

[4] S Balaji and M Sundararajan Murugaiyan. Waterfall vs. v-model vs. agile: A comparative study on sdlc. *International Journal of Information Technology and Business Management*, 2(1):26–30, 2012.

[5] Buzzle. Waterfall model vs. v model, 2016. URL http://www.buzzle.com/articles/waterfall-model-vs-v-model.html. [Online; accessed 28-May-2016].

[6] ComputerWorld. Business requirements, 2006. URL https://books.google.be/books?id=WAKmeS-MOo8C&lpg=PA8&ots=98U4SWSR99&dq. [Vol. 40,Nr. 22].

[7] Markus Lffler Michael Chui and Roger Roberts. The internet of things, 2010. URL http://www.mckinsey.com/industries/high-tech/our-insights/the-internet-of-things. [Online; accessed 03-March-2016].

[8] Jim Morrish Dirk Slama, Frank Puhlmann and Rishi M Bhatnagar. Enterprise iot, 2015. URL http://enterprise-iot.org/book/enterprise-iot/part-ii-igniteiot-methodology/igniteiot-solution-delivery/planbuildrun/iot-project-initiation/initial-solution-design/analysis-projections-planning/. [Online; accessed 03-March-2016].

[9] Wikipedia. Ip code — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=IP_Code&oldid=721655296. [Online; accessed 28-May-2016].

[10] Wikipedia. En 62262 — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=EN_62262&oldid=710708584. [Online; accessed 28-May-2016].

[11] Karl E Wiegers. Software requirements: Practical techniques for gathering and managing requirement through the product development cycle. *Microsoft Corporation*, 2003.

[12] Poc, prototype, or pilot? when and why. URL http://www.contenthere.net/2007/03/poc-prototype-or-pilot-when-and-why_92.html. [Online; accessed 2-May-2016].

[13] Nest labs thermostat. URL www.nest.com/index.html.

[14] *Six Hidden Costs in a 99 Cent Wireless SoC*. Silicon Labs, 2016. URL http://www.silabs.com/Support_Documents/TechnicalDocs/six-hidden-costs-of-a-99-cent-soc.pdf.

[15] PC Magazine. Prototyping definition. URL http://www.pcmag.com/encyclopedia_term/0,1233,t=prototyping&i=49886,00.asp. [Retrieved 2012-05-03].

[16] Jenny Hughes and Loek Nieuwenhuis, editors. *A Project Manager's Guide to Evaluation*. Evaluate Europe Handbook Series. ITB Institute Technology and Education, 2005. URL http://www.pontydysgu.org/wp-content/uploads/2008/02/EvaluateEuropeVolume1final.pdf.

[17] Jean-Baptiste Waldner. Nanoinformatique et intelligence ambiante. *Inventer l'Ordinateur du XXIeme Sicle*, page 254, 2007. ISSN 2-7462-1516-0.

[18] Wikipedia. Node (computer science) — wikipedia, the free encyclopedia, 2015. URL https://en.wikipedia.org/w/index.php?title=Node_(computer_science)&oldid=689564808. [Online; accessed 10-November-2015].

[19] sleeper. Review of existing cellular network technologies: The pros and cons, 2015. URL https://iotee.wordpress.com/2015/03/16/internet-of-things-connectivity-option-cellular-network-technologies/.

[20] Wikipedia. Time division multiple access — wikipedia, the free encyclopedia, 2015. URL https://en.wikipedia.org/w/index.php?title=Time_division_multiple_access&oldid=688487450. [Online; accessed 11-November-2015].

[21] Wikipedia. Code division multiple access — wikipedia, the free encyclopedia, 2015. URL https://en.wikipedia.org/w/index.php?title=Code_division_multiple_access&oldid=688359316. [Online; accessed 11-November-2015].

[22] Wikipedia. Orthogonal frequency-division multiple access — wikipedia, the free encyclopedia, 2015. URL https://en.wikipedia.org/w/index.php?title=Orthogonal_frequency-division_multiple_access&oldid=686603111. [Online; accessed 11-November-2015].

[23] Quang-Dung Ho Yue Gao Gowdemy Rajalingham Tho Le-Ngoc. Wireless communications networks for the smart grid. *Springer*, page 33, 2014. ISSN 9783319103471.

[24] Wikipedia. Bluetooth low energy — wikipedia, the free encyclopedia, 2015. URL https://en.wikipedia.org/w/index.php?title=Bluetooth_low_energy&oldid=690327658. [Online; accessed 16-November-2015].

[25] Wirevolution. How does 802.11n get to 600mbps?, 2007. URL http://www.wirevolution.com/2007/09/07/how-does-80211n-get-to-600mbps/. [Online; accessed 17-December-2015].

[26] Xiaojun Feng. Enabling co-channel coexistence of 802.22 and 802.11af systems in tv white spaces, 2011. URL http://www.cse.ust.hk/~xfeng/doc/xfeng-ws-coexistence-ICC13.pdf. [Online; Retrieved 2014-01-19.

[27] Link Labs. Examining the future of wifi: 802.11ah halow, 802.11ad (and others), 2015. URL http://www.link-labs.com/future-of-wifi-802-11ah-802-11ad/. Online; Retrieved 2014-01-19.

[28] Wikipedia. Near field communication — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Near_field_communication&oldid=704462368. [Online; accessed 13-February-2016].

[29] LoRa Alliance. Lora technology, 2016. URL https://www.lora-alliance.org/For-Developers/LoRaWANDevelopers. [Online; accessed 10-February-2016].

[30] Harpreet S. Dhillon, Howard Huang, and Harish Viswanathan. Wide-area wireless communication challenges for the internet of things. *CoRR*, abs/1504.03242, 2015. URL http://arxiv.org/abs/1504.03242.

[31] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen. A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi. In *Industrial Electronics Society. IECON. 33rd Annual Conference of the IEEE*, pages 46–51. IEEE, 2015.

[32] Paolo Baronti, Prashant Pillai, Vince W.C. Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655 – 1695, 2015. ISSN 0140-3664. doi: http://dx.doi.org/10.1016/j.comcom.2006.12.020. URL http://www.sciencedirect.com/science/article/pii/S0140366406004749. Wired/Wireless Internet Communications.

[33] Gabriele Devita, Alan Chi Wai Wong, Mark Dawkins, K Glaros, U Kiani, Franco Lauria, V Madaka, O Omeni, Johannes Schiff, Ananthasayanam Vasudevan, et al. A 5mw multi-standard bluetooth le/ieee 802.15. 6 soc for wban applications. In *European Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014-40th*, pages 283–286. IEEE, 2014.

[34] 2015. URL http://m.edntaiwan.com/ART_8800521986_3000005_TA_2cc19a62.HTM. Online; Retrieved 14-02-2016.

[35] battery-life-in-connected-wireless-iot-devices, 2015. URL https://www.silabs.com/Support_Documents/TechnicalDocs/battery-life-in-connected-wireless-iot-devices.pdf. Online; Retrieved 16-02-2016.

[36] Maplin Media. URL http://images.maplinmedia.co.uk/solar-charging-regulator.jpg?w=283&h=283&r=4&o=xdwC9np8zcfD28RssxHtSp98TW0j&V=DQh1.

[37] Amazon. URL http://ec2-52-22-100-29.compute-1.amazonaws.com/wp-content/uploads/2013/03/usbdcsolarchargeruse_MED.jpg.

[38] Linear Technology. URL http://cds.linear.com/docs/en/datasheet/35881fc.pdf.

[39] Piceramic. URL http://www.piceramic.com/product-detail-page/e-821-608000.html.

[40] Don Scansen. Thermoelectric energy harvesting, 2011. URL http://www.digikey.com/en/articles/techzone/2011/oct/thermoelectric-energy-harvesting. [Online; accessed 10-February-2016].

[41] Jim O'Callaghan. Eocean inc, 2011. URL http://www.sensorsmag.com/networking-communications/energy-harvesting/thermoelectric-energy-harvesting-8708. [Online; accessed 25-February-2016].

[42] Wikipedia. Ac power plugs and sockets — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=AC_power_plugs_and_sockets&oldid=706546455. [Online; accessed 26-February-2016].

[43] WestFW. Ac/dc converter: non-wall warty?, 2010. URL http://forum.arduino.cc/index.php?topic=15891.0. [Online; accessed 27-February-2016].

[44] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Shyamnath Gollakota, and Joshua R Smith. Powering the next billion devices with wi-fi. *arXiv preprint arXiv:1505.06815*, 2015.

[45] Wikipedia. Wireless power transfer — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Wireless_power_transfer&oldid=706525021. [Online; accessed 27-February-2016].

[46] 2016. URL directindustry.com. [Online; accessed 27-February-2016].

[47] Atmel. At91m40800 summary, 2016. URL http://www.atmel.com/Images/1348S.pdf. [Online; accessed 29-February-2016].

[48] Foxconn wages. URL http://www.forbes.com/sites/timworstall/2012/05/17/why-wages-at-apples-foxconn-are-so-low. [Online; accessed 4-May-2016].

[49] Wikipedia. Firmware — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Firmware&oldid=716212861. [Online; accessed 7-May-2016].

[50] Chris F Kemerer. An empirical validation of software cost estimation models. *Communications of the ACM*, 30(5):416–429, 1987.

[51] Wikipedia. Cocomo — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=COCOMO&oldid=716952074. [Online; accessed 9-May-2016].

[52] Roger S Pressman. *Software engineering: a practitioner's approach.* Palgrave Macmillan, 2005.

[53] Wikipedia. Software testing — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Software_testing&oldid=719160889. [Online; accessed 10-May-2016].

[54] George Labovitz, Yu Sang Chang, and Victor Rosansky. *Making quality work: A leadership guide for the results-driven manager.* John Wiley & Sons, 1992.

[55] Wikipedia. Nyquist frequency — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Nyquist_frequency&oldid=712111111. [Online; accessed 1-May-2016].

[56] Wikipedia. Harmonics (electrical power) — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Harmonics_(electrical_power)&oldid=708184431. [Online; accessed 1-May-2016].

[57] Mathieu De Zutter. Ontwerp van een robuuste datalogger voor elektrische energie voor kleinschalige hernieuwbare energie-installaties in ontwikkelingslanden, 2014. URL http://www.unicat.be/uniCat?func=search&query=sysid:24859153. [Online; accessed 1-May-2016].

[58] Harmonic distortion analysis, 2016. URL http://www.selectricity.com/harmonicdistortionanalysis.html. [Online; accessed 1-May-2016].

[59] Wikipedia. Ac power — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=AC_power&oldid=716736200. [Online; accessed 1-May-2016].

[60] Wikipedia. Volatile organic compound — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Volatile_organic_compound&oldid=719938498. [Online; accessed 16-May-2016].

[61] Janitza. URL http://www.janitza.com/umg-96rm-p-m-cbm-el-features.html. [Online; accessed 1-May-2016].

[62] 2016. URL http://makezine.com/2013/04/15/arduino-uno-vs-beaglebone-vs-raspberry-pi/. [Online; accessed 17-May-2016].

[63] 2016. URL https://www.arduino.cc/en/Guide/MacOSX. [Online; accessed 19-May-2016].

[64] 2016. URL http://cktplus.com/components/semiconductor/RTC-I2C-DS1307-Real-Time-Clock-RTC-with-32K-Memory. [Online; accessed 19-May-2016].

[65] 2016. URL http://www.robotshop.com/en/humidity-temperature-sensor-dht22.html. [Online; accessed 19-May-2016].

[66] As-mlv-p2 air quality sensor data sheet, 2016. [Online; accessed 19-May-2016].

[67] Wikipedia. Lux — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Lux&oldid=713838521. [Online; accessed 28-May-2016].

[68] 2016. URL http://www.digikey.es/product-detail/en/lite-on-inc/LTR-303ALS-01/160-2161-2-ND/4847335. [Online; accessed 19-May-2016].

[69] 2016. URL http://redbearlab.com/bleshield/. [Online; accessed 19-May-2016].

[70] 2016. URL http://www.digikey.com/product-detail/en/triad-magnetics/CST-1020/237-1620-ND/4009889. [Online; accessed 21-May-2016].

[71] Wikipedia. Voltage divider — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Voltage_divider&oldid=718312368. [Online; accessed 21-May-2016].

[72] Wikipedia. Switched-mode power supply — wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/w/index.php?title=Switched-mode_power_supply&oldid=715132342. [Online; accessed 21-May-2016].

[73] 2016. URL http://getbirdi.com. [Online; accessed 21-May-2016].

[74] 2016. URL https://koto.io. [Online; accessed 21-May-2016].

[75] 2016. URL https://cubesensors.com/. [Online; accessed 21-May-2016].

[76] 2016. URL http://www.belkin.com/us/p/P-F7C029/. [Online; accessed 21-May-2016].

[77] 2016. URL http://aeotec.com/z-wave-plug-in-switch. [Online; accessed 21-May-2016].

[78] 2016. URL https://smart-me.com/. [Online; accessed 21-May-2016].

[79] UGent. Salarisschalen personeel ugent, 2016. URL http://www.ugent.be/nl/vacatures/categorieen-salarisschalen/salarisschalen/atpes2.htm. [Online; accessed 20-May-2016].