|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | |  |  | | --- | --- | | |  | | --- | | **ESCUELA POLITÉCNICA SUPERIOR DE MONDRAGON UNIBERTSITATEA**  *MONDRAGON UNIBERTSITATEKO GOI ESKOLA POLITEKNIKOA*  MONDRAGON UNIVERSITY FACULTY OF ENGINEERING | | |  |
|  | **Trabajo presentado para la obtención del título de**  *Titulua eskuratzeko lana*  Final degree project for taking the degree of |  |
|  | **GRADO EN INGENIERÍA EN INFORMÁTICA** *INFORMATIKAKO INGENIARITZA GRADUA* DEGREE IN COMPUTER ENGINEERING |  |
|  |  |  |
|  | |  | | --- | | **Título del Trabajo** *Lanaren izenburua* Project Topic | | RUNTIME VERIFICATION FOR SPATIO-TEMPORAL PROPERTIES OVER IOT NETWORKS | |  |
|  |  |  |
|  |  |  |
|  | |  |  | | --- | --- | | **Autor** *Egilea* Author   **Curso** *Ikasturtea* Year | OIHANA GARCIA ANACABE   2021/2022 | | |

|  |
| --- |
| **Título del Trabajo** *Lanaren izenburua* Project Topic |
| **RUNTIME VERIFICATION FOR SPATIO-TEMPORAL PROPERTIES OVER IOT NETWORKS** |

**Nombre y apellidos del autor**   
*Egilearen izen-abizenak*   
Author's name and surnames   
GARCIA ANACABE, OIHANA   
  
**Nombre y apellidos del/los director/es del trabajo**   
*Zuzendariaren/zuzendarien izen-abizenak*   
Project director's name and surnames   
EZIO BARTOCCI   
ILLARRAMENDI, MIREN   
  
**Lugar donde se realiza el trabajo**   
*Lana egin deneko lekua*   
Company where the project is being developed   
TU WIEN   
  
**Curso académico**   
*Ikasturtea*   
Academic year   
2021/2022

**El autor/la autora del Trabajo Fin de Grado, autoriza a la Escuela Politécnica Superior de Mondragon Unibertsitatea, con carácter gratuito y con fines exclusivamente de investigación y docencia, los derechos de reproducción y comunicación pública de este documento siempre que: se cite el autor/la autora original, el uso que se haga de la obra no sea comercial y no se cree una obra derivada a partir del original.**

*Gradu Bukaerako Lanaren egileak, baimena ematen dio Mondragon Unibertsitateko Goi Eskola Politeknikoari Gradu Bukaerako Lanari jendeaurrean zabalkundea emateko eta erreproduzitzeko; soilik ikerketan eta hezkuntzan erabiltzeko eta doakoa izateko baldintzarekin. Baimendutako erabilera honetan, egilea nor den azaldu beharko da beti, eragotzita egongo da erabilera komertziala baita lan originaletatik lan berriak eratortzea ere.*

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Laburpena

(Laburpen amaieran ipini dokumentuaren amaierarantz informazio gehiago dagoela euskaraz \*Erreferentzia bat sartu atal horretara)

Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Table of contents

[1. Introduction 4](#_Toc99101438)

[1.1. Problem definition 4](#_Toc99101439)

[1.2. Objectives 5](#_Toc99101440)

[1.3. Project phases 5](#_Toc99101441)

[1.4. Product specifications and requirements 5](#_Toc99101442)

[2. State of the art 6](#_Toc99101443)

[3. Product specifications and requirements 6](#_Toc99101444)

[3.1. Description of the service 6](#_Toc99101445)

[3.2. Resources and materials 6](#_Toc99101446)

[3.2.1. Hardware 6](#_Toc99101447)

[3.2.2. Software 6](#_Toc99101448)

[3.3. Tests and trials 6](#_Toc99101449)

[3.4. Conditions for the implementation of the project 6](#_Toc99101450)

[3.5. Legal aspects 6](#_Toc99101451)

[4. Objectives 6](#_Toc99101452)

[5. Use cases 6](#_Toc99101453)

[5.1. Office use case 6](#_Toc99101454)

[5.2. Wiener linien use case 6](#_Toc99101455)

[6. Development (subject to change) 7](#_Toc99101456)

[6.1. Middleware 7](#_Toc99101457)

[6.2. Services SOA 7](#_Toc99101458)

[6.3. Design pattern 7](#_Toc99101459)

[6.4. MQTT/REST 7](#_Toc99101460)

[6.5. Robustness 7](#_Toc99101461)

[6.6. Buffer 7](#_Toc99101462)

[6.7. Thingy 7](#_Toc99101463)

[7. Problems and solutions 8](#_Toc99101464)

[How to manage the time and the buffer 8](#_Toc99101465)

[8. Economic memory 8](#_Toc99101466)

[9. Conclusions and future lines 8](#_Toc99101467)

[9.1. Conclusions 8](#_Toc99101468)

[9.2. Future lines 9](#_Toc99101469)

[10. Personal evaluation of the experience(?) and the project 9](#_Toc99101470)

[11. Sarrera, ondorioak eta etorkizuneko ildoak 10](#_Toc99101471)

[11.1. Sarrera 10](#_Toc99101472)

[11.2. Ondorioak 10](#_Toc99101473)

[11.3. Etorkizuneko ildoak 10](#_Toc99101474)

[12. Appendix A STREL 11](#_Toc99101475)

[13. Appendix B MQTT 12](#_Toc99101476)

[14. Appendix C REST 13](#_Toc99101477)

[15. Appendix D Data bus 14](#_Toc99101478)

[16. Appendix E Gantt chart 15](#_Toc99101479)

[17. Bibliography 16](#_Toc99101480)

# Introduction

This chapter is the introduction to the Bachelor’s Degree Final Project “Runtime verification for spatio-temporal properties overt IoT networks”. In this section, the concepts involved in the project are defined. Additionally, the project definition, scope of the project, planification and the product specification and requirements are explained.

## Problem definition

IoT (**Internet of Things**) is the area of computer science that collects the challenges of connecting millions of smart devices and sensors and making them accessible via the internet. This field is growing fast. The forecast is that the number of connected devices by 2030 will be 25.44 billion worldwide [1]. These devices are already part of several fields (e.g., e-health services, smart cities, e-farm, and intelligent transportation systems (ITS)), being a big part of the digitalization of society to build a smart world.

Among the systems that can exploit an IoT infrastructure, a noteworthy category is **Cyber Physical Systems** (CPS), where physical systems are monitored and/or controlled by a computational core. They interact with physical processes through sensors and actuators. The increasing numbers of IoT devices and intelligent systems made CPS influence society. They can be found in different sectors such as self-driving cars, home equipment and medical devices [2] [3]. The following definition is the most famous one for the term “Cyber Physical Systems”:

“Cyber-Physical Systems are engineering, physical and biological systems whose operations are integrated, monitored, and/or controlled by a computational core. Components are networked at every scale. Computing is deeply embedded into every physical component, possibly even into materials. The computational core is an embedded system, usually demands real-time response, and is most often distributed. The behaviour of a cyber-physical system is a fully-integrated hybridisation of computational (logical) and physical action."

(Helen Gill, US National Science Foundation) [4]

Monitoring is an activity related to the wider category of **Runtime Verification** (RV), which purpose is to observe information from a system while it is operating and analyse the behaviour to detect if it satisfies or violates certain properties. Monitoring the status of a CPS at runtime can give precise information to ensure reliability, safety, robustness and security [5] [6].

This project focuses precisely on the challenges when doing monitoring on CPS over IoT and provides an implementation of a service to monitor data collected by sensors at runtime. It is closely related to some aspects of Helen Gill’s definition. The IoT devices are in the physical part where they are spatially distributed and networked. The data will be collected, both across space and time. One main task of the project is to connect the sensors with the monitor so they can share information (i.e., networking). Finally, this data will be sent to MoonLight to monitor everything in real-time.

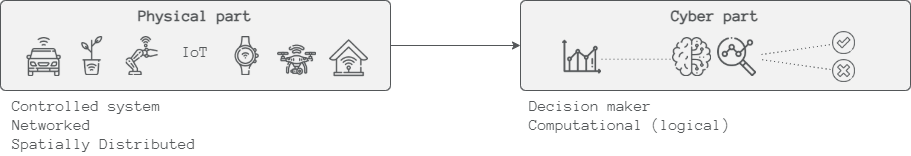


Figure 1‑A Project outline

## Objectives

The objective of this project is to feed MoonLight (the provided monitor) with live data. Moonlight previously has been tested with a Matlab interface since this tool has several CPS models and analysis tools available [7]. With that being said, this is the first time for the monitor to receive real data, so the purpose of this project is to help MoonLight be able to monitor different use cases and see how it works with actual real time data.

Therefore, the main objective is to implement a middleware. This middleware will enable the monitor to communicate with different applications in a distributed network. To achieve this, the middleware not only will provide some services but also will be capable expand by adding more services.

To implement this, several and diverse goals have to be met. On the one hand, the monitoring of spatio-temporal properties will be carried out using logic-based specification languages, so the language used, STREL, must be comprehended in order to implement it on the use-cases defined in the project. On the other hand, the communication of the sensors with the monitor has to be established, to achieve this, two types of protocols will be used (i.e., MQTT and Bluetooth Low Energy).

networking monitor at runtime

Middleware has to be escalable, customizable

## Project phases

The project duration is of eight months, from November 2021 to June 2022. To achieve the objectives, the project has been divided into some tasks and scheduled to manage the work. The development has been divided into the next phases:

### Introduction to the project and department

The project is held in Technische Universität Wien (TU Wien), Austria's largest research and educational institution in the field of technology and natural sciences. The group is the Cyber-Physical System department. This phase has consisted of learning the subjects involved in the project and the tools employed during the development. The principal tool used was Moonlight, therefore, most of the time was dedicated to this, to learn the concepts to comprehend it (i.e., CPS and STREL) and how to use it.

### Product development

After studying the project bases and making some trial examples with the moonlight framework, the product development started. Starting from the middleware, continuing with the networking and the sensors, and finishing with the layout/framework, the development was carried out during the entire process.

A planification was scheduled to make sure that the objectives were achieved before the deadline. There was a beta release planned for the beginning of April. The beta release consisted in a product that was close in look, feel and function to the final version of the project. This beta release happened one week after the planned date, however, the product had more fancy stuff. The product not only had the main functions of the final product done (i.e., all the flow, beginning in the sensors until the moonlight monitor) but also it was able to fully support SOA (Service Oriented Architecture). The services implemented in the Middleware were the one in charge with the MQTT connections and the Online Moonlight service, able to monitor the input data. Besides, the Data Bus did …

In the resting month HTPP, … wre implemented in the middleware. In terms of hardware, the Thingy52 and the ESP were able to do … (Previously, I had some demo programs with the Thingy)

Some objectives of the project had changed from the beginning to the end.

During the development, some aspects of the project were changed. For example, at the beginning the middleware was supposed to be developed to be one directional (i.e., from the sensors to the monitor), but finally, we decided to be bi-directional and add another step (i.e., receive the data from the sensors, send it to the monito, get the results and send it to another service) and with a SOA architecture. Luckily, the middleware I was developing was as general as possible and the tests done had a good coverage, so to change the architecture of the software hadn’t take so long and I was able to extend the project.

At the beginning of the project a Gantt chart was created, but some tasks took longer or shorter than expected or due to the changes in the objectives, it wasn’t followed al pie de la letra. As time passed, the track of the project was being recorded in a gantt chart that you can see in the appendix E – Gantt chart (introduce the reference), there you can see the actual work done.

### Other tasks

* Tests have hight importance in a software project. Tried to do a TDD (Test-driven Development). But I was not used to do it, so sometimes, I wrote the production code before the tests. In any case, the tests where very important during the development and I was doing them during the entire project
* Documentation: I was developing the documentation during the whole time
* Meeting and others: During all the project I had weekly meetings with my supervisors, in order to keep a track of what I am doing and make some questions, do/ask for suggestions… And some other extra meetings too if needed. In the first months they let me attend the IoT master’s course, that helped me with the IoT sensors and MQTT protocol.

## Product requirements

This thesis consists largely in developing a software project, due to this, the main resources used are software, nevertheless, some hardware devices were used too. Down below there is a list of the requirements needed during this project divided into two groups, software and hardware:

### Software

#### Moonlight[[1]](#footnote-1)

This is the principal resource, a lightweight Java-tool monitor, it can monitor temporal, spatial and spatio-temporal properties of distributed complex systems, like Cyber-Physical Systems. It has two different monitoring approaches, offline and online. Finally, supports the specification of properties written with the Reach and Escape Logic (STREL).

#### GitHub Actions and SonarCloud

GitHub Actions is a continuous integration and continuous delivery platform that allows for automating the build, test and deployment pipeline. This project is in a GitHub repository, so is possible to run a workflow. Every time a push is done, the Middleware is built, tested and analysed in SonarCloud.

SonarCloud automatically analyzes branches, the code quality and code security. By knowing the bugs, code smells, security hotspots and coverage and the fast feedback it was possible to maintain the code cleaner.

#### Programming languages

Nearly all the project is developed in Java. As previously mentioned, Moonlight is implemented as a Java program and so is the developed Middleware. The version used is Java 17, the last Long Term Support release.

C is used to develop the program of the devices of this project. A language that can be considered one of the most widely used programming languages in IoT.

#### Gradle

Gradle is a build automation tool. It is open-source and it is focused on flexibility and performance. Its features include the management of the dependencies.

#### Zephyr

The Zephyr Project is a scalable open-source real-time operating system (RTOS) supporting multiple hardware architectures (more than 350 boards), between them Thingy52 can be found. The Zephyr projects are CMake-based. CMake is an open-source, cross-platform family of tools designed to build, test and package software.

#### Integrated Development Environment (IDE)

First, for the middleware development, IntelliJ IDEA was used. This tool provides code completion, static code analysis and refactoring. This makes the development workflow experience smoother.

Then, Visual Studio Code was for the Thingys. This IDE provides just the tools a developer needs for a quick code-build-debug.

Finally, Arduino IDE was also used. This makes it easy to write code and upload it to the board, in this case to the ESP32.

### Hardware

#### Thingy52

The Nordic Thingy52 board is an easy-to-use prototyping platform. It is designed for helping to build IoT demos, with power optimization and several sensors. It is based on the nRF52832 Bluetooth 5 system on chip (SoC). In this project, the Bluetooth Low Energy (BLE) and environmental sensors such as temperature, humidity and air quality (CO2 and TVOC) are used.

#### Segger J-Link

SEGGER J-Links are the most widely used debuggers on the market thanks to its many supported CPUs and compatibility with the popular environments. The J-Link EDU Mini is a reduced version to allow students and educational facilities to debug different devices. USB-powered, can communicate at high speed.

#### ESP32

ESP32 is low-cost, ultra-low power consumption system on a chip microcontroller. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI/SDIO or I2C/UART interfaces. In this project case, uses Bluetooth to communicate with the Thingy sensors and the Wi-Fi to publish it in the MQTT broker.

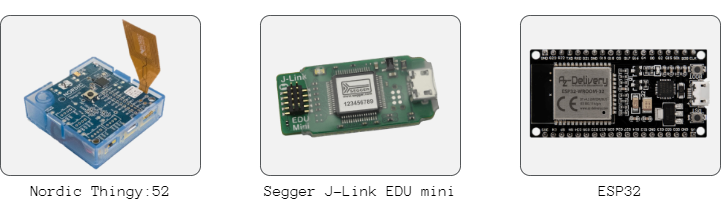


Figure 1‑B Hardware

# State of the art

CPS and others

STREL and others

Middleware types

# Development (subject to change)

## Monitor

Moonlight online approach

Monitoring tools for CPS are generally agnostic of the spatial configuration of the entities such as sensors and computational units.[Moonlight document]

In this section a novel online (out-of-order) monitoring algorithm for STREL is presented. Differently from the standard offline approach, where all the data is available at the beginning of the execution, online monitoring is performed incrementally, when a new piece of data is available. In this case, the uncertainty related to the absence of information must be taken into account. For that aim, the machinery of imprecise signals can be exploited to represent the uncertainty, where the result of the monitoring process, whether it is a satisfaction or a robustness signal, is refined as soon as new updates of the input arrive [Online Monitoring of Spatio-Temporal Properties for Imprecise Signals]

we introduced an online monitoring algorithm for STREL that exploits imprecise signals that can be refined by updates arriving in any order, and that can monitor updates on different locations in parallel. We implemented the proposed methodology in the Moonlight monitoring tool.

## Middleware



Egin horrelako zerbait nire adibidea erabiliz

<https://docs.oracle.com/cd/E21764_01/core.1111/e10103/intro.htm#ASCON110>

## Services SOA

## Design pattern

Builder 🡪 <https://refactoring.guru/>

## MQTT/REST

DR1 Lightweight communication methods

DR2 Interoperability.

DR3 Non-blocking event propagation. Events may arrive at unknown rates

DR4 Scalability(??)

Edge-based Runtime Verification for the Internet of Things

## Robustness

Error handling

Maintainability

## Buffer

Collecting binary data bits into groups that can then be operated on as a unit,

automatic buffering.

It helps devices to manipulate data before sending or receiving.

## Hardware

### Thingy

Kconfig Json importatu ahal izateko 🡪 zephyrrena

CMakeList

Prj.conf 🡪 sensoreak enable egin ahal izateko

<https://github.com/google/eddystone/blob/master/protocol-specification.md>

### ESP 01

# Problems and solutions

#### How to manage the time and the buffer

Overriding problems, how to save them

#### Dealing with missing values

1. The very first values

Discard it directly

1. The missing values during the program

Time Chain splitter

#### Coordinate the sensors

Moonlight is prepared to monitor starting from the time 0

Time table to coordinate the sensors

#### Duplicated values from sensors -Maybe DELETE-

#### Other minor problems

* MoonlightRecord had some problems: Escape + online monitor + MoonlightRecord = infinite loop

The null values didn’t throw errors, wrong error handling.

I reported this issues -> Ennio created Tuple.class, that had everything MoonlightRecord was offering but with these errors fixed: I just used Tuple from that moment on.

* I was having problems with Windows + Zephyr project -> I used Linux for the coding of the sensors

for the Thingy sensors, I used a Linux virtual machine. IntelliJ IDEA does not support C/C++ officially and it requires quite a lot memory, that’s why I used Visual Studio Code.

# Use cases

## Office use case

## Wiener linien use case (?¿?¿?¿?¿?¿?¿)

# Economic memory

The majority of the cost of a software project is in long-term maintenance. [clean code liburua]

Intellij IDEA: unibertsitateko kontuari esker dohainik

# Conclusions and future lines

This is thechnical

## Conclusions

a. Reflexiones técnicas: relacionadas con los objetivos del proyecto b. Reflexión sobre las implicaciones sociales, de salud y seguridad, medioambientales, económicas e industriales   c. Reflexión sobre la aplicación de conocimientos relativos a cuestiones económicas, organizativos de gestión (gestión del riesgo y del cambio) en el contexto industrial y comercial.

## Future lines

“Smart Home Automation System Using on IoT” dokumentuan rosas dagoenari begirada bat bota /!\

Legal aspects: General Data Protection Regulation (GDPR):

Error processing: Maybe there are things that are assumed that can fail and they are not handled, for example, the times of the sensors are always ascendant.

# Personal evaluation of the experience(?) and the project

Proiektua egiten nola sentitu naizen aipatu

Esperientziari dagokionez: A) Unibertsitatea: nola sentitu naizen, IoTko kurtsoak, astero egiten diren hitzaldietara joaten utzi, liburua utzi irakurteko… B) Beste herrialde batera joan: Leku berriak ezagutu, bertoko kulturatik ikasi, bakarrik bizitzea eta independentzia.

# Sarrera, ondorioak eta etorkizuneko ildoak

Atal honetan sarrera, ondorioak eta etorkizuneko ildoak atalen laburpen bat egingo da euskaraz.

## Sarrera

## Ondorioak

## Etorkizuneko ildoak

# Appendix A **STREL**

Titulua aldatzerako orduan kontuz! Formatua galdu gabe/!\ Aurkibidean polit ikusteko

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Tabla A1 …

# Appendix B **MQTT**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Appendix C **REST**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Appendix D **Data bus**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Appendix E **Gantt chart**

# Bibliography

|  |  |
| --- | --- |
| [1] | T. Insights, «Statista,» December 2020. [En línea]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. |
| [2] | L. Nenzi, E. Bartocci, L. Bortolussi, M. Loreti y E. Visconti, «Monitoring Spatio-Temporal Properties (Invited Tutorial),» de *Runtime Verification*, Springer International Publishing, 2020. |
| [3] | D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique y E. Bartocci, «A Roadmap Toward the Resilient Internet of Things for Cyber-Physical Systems,» *IEEE Access,* vol. 7, pp. 13260-13283, 2019. |
| [4] | H. Gill, US National Science Foundation, 2006. |
| [5] | C. Tsigkanos, M. M. Bersani, P. A. Frangoudis y S. Dustdar, «Edge-based Runtime Verification for the Internet of Things,» *IEEE Transactions on Services Computing,* 2021. |
| [6] | M. Illarramendi, L. Etxeberria, X. Elkorobarrutia, J. Perez, F. Larrinaga y G. Sagardui, «MDE based IoT Service to enhance the safety of controllers at runtime,» 2019. |

1. [GitHub - MoonLightSuite/MoonLight: a light-weight framework for runtime monitoring.](https://github.com/MoonLightSuite/MoonLight) [↑](#footnote-ref-1)