DIGITIZING PRODUCTS:
CREATING DEMONSTRATORS
FOR FUTURE EDUCATION

Pipe Temperature Sensor

Demonstrator classification and documentation

Aalborg, December 2022

Co-funded by the Erasmus+ Programme of the European Union

Project consortium

Dissemination level

Code	Access granted to	
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Legal Disclaimer

About the DigiDemo project

Environmental challenges and digital transformation are two of the main drivers changing the world and the way business will be is done in the future. Therefore, it is essential to enable future employees to address these drivers. The skills and competences needed to develop digitalized products and awareness of the environmental challenges are therefore crucial for the European workforce and industry to continue being competitive in a future green economy and to maintain jobs across Europe.

The DigiDemo project addresses these challenges by developing demonstrators especially for higher education allowing to improve mainly mechanical engineering studies by integrating skills and competences allowing them to understand, develop and commercialise connected products. The results will be publicly available and can be used by every institution interested in integrating this type of training in their cursus.

Table of Contents

Document authors

	First name Last name	Institution
Key author	Mogens Holm Iversen	UCN
Further authors	Steffen Rahbek Vutborg	UCN
	Torben Momme Holm	UCN
	Esben Skov Laursen	UCN

Revision history

Version	Date	Author(s)	Description
1.0	2022-12-05	Iversen	Initial draft
1.1	2022-12-08	Iversen	Creating exercises for the Software Development part of the demonstrator
1.2	2023-01-19	Vutborg	Elaborating on the overview of the demonstrator from an electronics standpoint.
1.2	2023-01-24	Vutborg	Added Educational part for the electronics/embedded part.
1.3	2023-01-26	Holm	Elaboration on learning points in relation to mechanical and production issues.
1.4	2023-03-20	Vutborg	Finalising electronics part.

Document status

Status description
Final version

Abbreviations

ESTA	ESTA Belfort (France)
FHV	Fachhochschule Vorarlberg (Austria)
FiV	Fagskolen i Viken (Norway)
UCN	University College Nordjylland (Denmark)
UBB	Universitatea "Babes-Bolyai" din Resita (Romania)
MCU	Microcontroller unit
IoT	Internet of Things
LiPo	Lithium-ion polymer
LAN	Local Area Network
MQTT	Message Queuing Telemetry Transport
BLE	Bluetooth Low Energy
WiFi	Wireless network (Wireless Fidelity)
UN SDGs	United Nations Sustainable Development Goals
IDE	Integrated Development Environment
P2P	Peer to Peer
DBMS	Database management studio
JDBC	Java Database Connectivity

List of figures

List of tables

1 Introduction

This demonstrator focusses on creating a possible solution for water spill in common households across Denmark. Clean water is a precious resource and early detection of spillage seen from a sustainability point of view is an extremely good idea. Furthermore, it is also something the consumer can feel on the household economy. According to calculations based on numbers by The Danish Energy Agency from 2021 [1] the estimated cost ranges from 66 €/year (a gently dripping faucet) to 2000 €/year (a leaking toilet). These numbers can then be multiplied by the number of installations in the household. Also note that the calculations are based on prices from 2021 and the price increase due to the recession and huge inflation in the Euro zone are not included in the numbers.

One way of detecting water spillage, and the basis of this demonstrator, is to measure the surface temperature of a water pipe and compare it to the ambient temperature. The theory is that while there is no flow in the pipe and the water is still standing, the temperature of the water will converge towards the ambient temperature.

Figure 1 is a photo of the embedded unit prototype. It has an ESP32 based development board, a lithium-ion polymer battery, an on/off switch as well as two temperature sensors bundled into one compact package.

Figures Figure 2, Figure 4 and Figure 3 show photos of the embedded device in its 3D printed enclosure from different angles. It features strapping points for mounting on a pipe in such a way that the bottom temperature sensor makes solid contact with the pipe. There is a hole in the lid, allowing the sensor to protrude through it for better ambient sensing once the lid is installed.

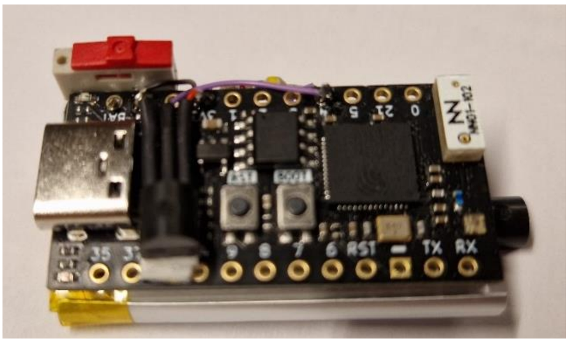


Figure 1 : Picture of the embedded device prototype

Figure 2: Device in enclousure - top view

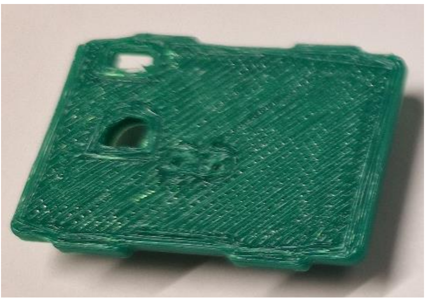


Figure 4: Enclosure lid


Figure 3: Device in enclousure - bottom view

2 Overview

The key properties of the focus project are:

Table 1: Specification of key properties of the focus project

Key Property	Value
EQF level	5 (Technician)
Year of study	2
Domain	Mechatronics
Objective	Hands-on
Workload	5+ ECTS (depending on setup)
Keywords	Internet of Things (IoT), Micro Controller Unit (MCU), Temperature, Distributed Programming, Embedded Programming

The demonstrator is centred around an embedded microcontroller with two temperature sensors and a lithium-ion polymer (LiPo) battery. The embedded unit should be encased in an enclosure that shields the unit from water ingress and physical contact, while allowing the unit to be strapped to a pipe with one temperature sensor making good thermal contact with the surface of the pipe. Furthermore, the enclosure needs to expose the other temperature sensor to the ambient air and allow the battery to "breathe"; I.e., expand, contract and vent.

The microcontroller collects temperature measurements from both sensors at a fitting regular interval and transmit these measurements over Wi-Fi to a Message Queing Telemetry Transport (MQTT) server on the local area network (LAN). To save power, several measurements can be bundled into a bulk transmission, thereby saving on transmission and connection overhead. It is possible that further power can be saved by migrating to Bluetooth Low Energy (BLE) in place of the Wi-Fi link.

A service is then run on the server, taking in measurements from one or multiple embedded units, then analysing these measurements and issue an alert if there are signs of water spillage.

The demonstrator is thought to be interdisciplinary from mechanics over electronics to software development. There are a series of reflection question with regards to sustainability for all disciplines included

3 Description of fulfilment of keywords/characteristics

The demonstrator will improve teaching, by showing digitalization and value-adding of a well-known object.

The demonstrator will be replicable, as most of the hardware will be based on readily available modules, and further instructions on system fitting will be provided.

The demonstrator will be interdisciplinary, covering the following fields: Electronics integration, embedded programming, power efficiency, thermal dynamics, mechanical design.

Table 2: Description of fulfilment of demonstrator characteristics for the focus project

Characteristic	Description
Teaching improvement	The focus project allows the students to go through the entire process of development of a mechatronic product starting with requirements, through design, implementation, test, and integration of both mechanical and electronic components with software. They work in interdisciplinary teams.
Sustainability awareness	The project covers considerations on the following UN SDGs: 6 – Clean water and sanitation. 11 – sustainable cities and communities. 12 – Responsible consumption and production.
Replicability	The demonstrator is implemented using off the shelf components, which are easy to replicate.
Industry needs	Similar products are existing on the market.
Interdisciplinarity	The demonstrator requires both electrical and mechanical engineers to cooperate in implementing a mechatronic system. Microcontroller programming is also part of the project. Furthermore, a monitoring system have to be implemented with security considerations built in.

4 Classification according to the dimensions

Table 3: Classification of the focus project according to the dimensions

Dimension	Property	Value
Value chain	development	☑
	production	☑
	sales	☑
	after-sales-support	☐
	end-of-life	☐
Chain of technology	mechanical structure	☑
	sensors	☑
	electronic circuits	☑
	edge device	☑
	data transmission	☑
	cloud	☐
Sustainability	energy reduction	☑
	material reduction	☐
	better materials	☐
	better production	☐
	reparability	☐
	recycling	☐
Physicality	physical setup	☑
	simulation	☐
Degree of student freedom	demonstrated	☐
	guided	☑
	coached	☑
	autonomous	☑
Transportability	fixed	☐
	transportable	☑
	portable	☐
Costs (implementation)	EUR	300 – 1500
Costs (operation)	EUR	NA
Workload (implementation)	Hours	20 – 50
Workload (operation)	Hours	30 – 60

Dimension	Property	Value
Size	m	< 1
Weight	kg	< 1
Special requests	no/yes, if yes: which	no

5 Educational information

5.1 Embedded development

The demonstrator is to be used by students with a minimum of one semesters experience in low/mid-level programming, as well as knowledge about digital electronics. The following description assumes that C++ is used for embedded unit. The learning goals are written based on Biggs work[2].

5.1.1 Intended Learning Outcomes

The demonstrator is thought to be used over several teaching sessions including exercises. Each session consists of 4 modules of 45 minutes each. It is expected that the student after working with the electronics/embedded part of the demonstrator can:

- Assemble electronics circuits
- Explain the working method of a MQTT based published/subscriber architecture.
- Program a ESP32 based microcontroller to connect to a MQTT broker over a WiFI network, and transmit data to the broker.
- Sample temperature data from two DS18B20 temperature sensors over the OneWire protocol.
- Program the microcontroller to utilise deep-sleep modes in order to save on energy.
- Measure power concumption and estimate battery life.

All above learning outcomes is intended as supported activities, I.e. prepared lectures on the theoretical topics as a base for practical exercises.

1) Production of the embedded unit

Note: *This exercise can be performed by the lecturer if electronics assembly is not desired.*

a) Assemble the electronics parts as indicated by the diagram in Figure 5. Take extreme care not to short anything when connecting the battery. This step is recommended as a supervised activity. Physical size and layout are a priority. The unit should be compact and have one sensor free to connect to a pipe and the other in free air. Figure 1 and Figure 3 can be used as a reference.

b) Using the Arduino IDE; upload a blank sketch to verify that the system is operational.

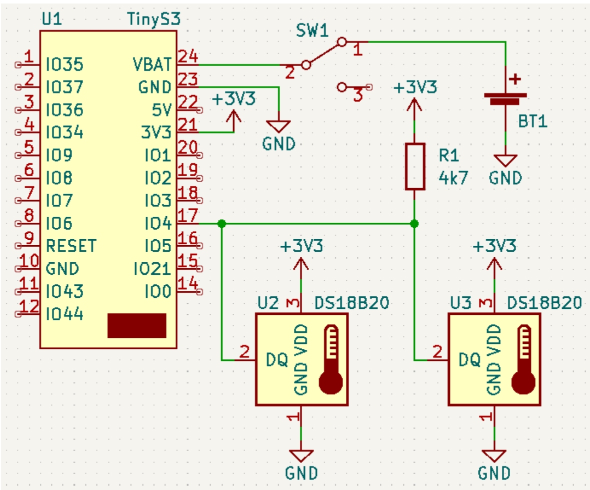


Figure 5: Electronics schematic

2) Programming the unit

a) Using the OneWire and DallasTemperature Arduino libraries to read temperatures of the sensors.

b) Using the WiFi (and optionally the WiFiManager) library to establish a wireless connection.

c) Using the ArduinoMqttClient library to connect to the broker unit.
d) Combine the functionality of exercises a, b and c into one program.
e) Expand the program with deep sleep functionality.
f) Measure and compare the power consumption with and without deep sleep and estimate the battery life in both cases.

5.2 Mechanical development

The demonstrator is to be used by students with a minimum of one semesters experience in mechanical mechanisms and understanding of stresses in materials.

5.2.1 Intended Learning Outcomes

The students are encouraged to understand the situation of the customer, especially how systems like this demonstrator are supposed to be fit-and-forget, but still must be accessible for some maintenance, ie. power resupply or system hardware resetting.

Also, the considerations of temperature versus time, in relation to surface contact and flow of ambient air, together with insulation properties of the housing, will be interesting discussion points in the lessons, even if the students at this level has had limited study of thermodynamics.

The students should develop a useful housing for the electronics, which will both be visually desirable and fulfills the geometric and thermal limitations. Evaluation of production cost in relation to tooling investment will be an important part of the learning outcome.

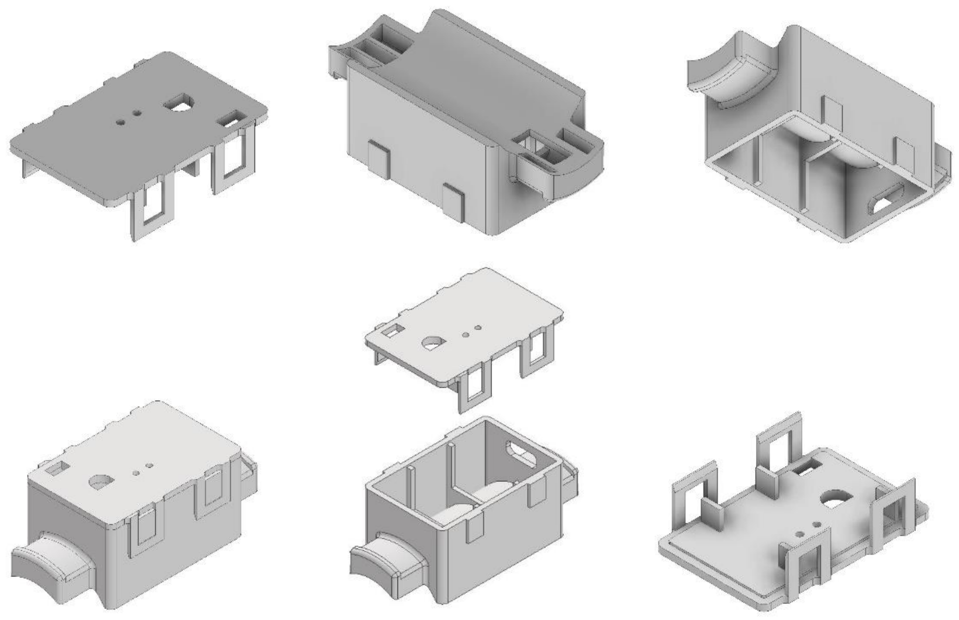
5.2.2 Example of mechanical design

Shown here is an example of a solution for low-volume production of the housing, based on additive manufacturing, to avoid the need for costly tooling investment.

The design already has several considerations regarding injection molding at a later stage, ie. wall thicknesses, flexible hinges, supporting structures for the hardware etc.

The design is adapted to the present hardware, with holes for the 2 temperature sensors for good response, 2 holes for resetting and access to on/off switch and to battery recharging without opening the lid. Each end has a strapping holder, for easy fitting to pipe with Nylon tie straps.

Pipe Temperature Sensor



5.3 Software development

The demonstrator is to be used by students with a minimum of one semesters experience in object-oriented programming (or equivalent experience if another programming language is used). The following description assumes that Java is used for programming the application that is either executed on a PC or a Raspberry Pi. The exercises can easily be transformed to use C#, Android programming, or Flutter. The demonstrator can be used in several different subjects for students enrolled in computer science classes. The work is described with a series of learning goals which is written based on Biggs work[2]. They are followed by a series of exercises the students are supposed to work with. The relevant exercises depend on what subject is being taught for the specific class.

5.3.1 Intended Learning Outcomes

The demonstrator is thought to be used over several teaching sessions including exercises. Each session consists of 4 modules of 45 minutes each. It is expected that the student after working with the entire software demonstrator part can

- Explain how to set up a connection to a MQTT broker in Java
- Implement a connection to a MQTT broker in Java and receive data from a thermal sensor
- Create a relational model for a database for storing data from a thermal sensor
- Optional create a generic model for a database to store data from a generic sensor package
- Connect to a database and persist the data received in a database
- Creating a program which will monitor the surface temperature and ambient temperature entrance pipe(s) into the household
- Create an alarming mechanism which will trigger when the difference goes outside a preprogrammed threshold
- Implement a programmable threshold mechanism

❖ Use version control for storing the different increments of the application

The students must be taught GUI, threads, Java connection framework, and persistence. Persistence is assumed to be created with a SQL based DB. It is assumed that the students as part of the curriculum is taught normalization theory. Then they are asked to solve the following exercises. It is up to the teacher to decide how to insert exercise solving in each session. It is assumed that the students will solve these exercises in groups of 3 – 5 students. It is recommended to place the software under version control since it is an incrementally developed application created over several sessions. Git is assumed to be used here, but any version control system can be used. If the students have knowledge of tagging in version control, then that is highly recommended.

1) Create a domain model for a situation where there in a house are sensors mounted on the ingoing water pipes. Transform the domain model into relational tables for an upcoming database using normalization rules. The DB must be on 3rd normalform unless an argument for another form can be presented by the students. Now setup a database using a database management solution and create the tables using SQL scripts.
2) Create an application which will display the values from the sensors in a GUI designed by the students. It should be possible to see the values from each sensor sorted in a meaningful way. At this level the data should be passed to the application from a TryMe class (a TryMe class is a simple class which will pass test data to the system without final integration). After the display part of the application is tested a connection to the database must be created using JDBC and the data from the TryMe class must also be stored there. Test it by inspecting the database in the DBMS. Implement the following alarms as visual alarms:
    a. If the ambient temperature is different from the surface temperature.
3) Optional. The groups of students should now be able to add functionality to the program which will connect to the database, read data for a configurable period (1st iteration could be a fixed time) and then display that data in a graph in the application allowing the household owner to see a report over the recorded data.
4) With the base application developed the students can now start to work on real data. There are two approaches here.
    a. Use a physical installation in a University Lab.
    b. Create a Digital Twin Simulator.

First step is in both cases requires a connection to the MQTT broker which contains the data from the sensor installation. Data is collected using the library which is part of the SW package for the demonstrator. For the first test of the connection let the application print the received data in the console using stdout. Next step is to display the data in the already created GUI.

5.4 Sustainability – Questions for Reflection

The questions below are meant as inspiration for the teacher using the demonstrator. The aim is to strengthen the student's awareness regarding sustainability through reflecting upon questions related to the demonstrator. The questions can be used during the lecture or as the starting point for individual or group work, connected to the lecture. The questions are thought to be independent of the subject where the demonstrator is used as they are relevant for sustainability considerations in general. The questions are thought to be answered by the students in groups and the result of the discussions are then presented on class.

The questions below are divided into three main categories, (1) general considerations, (2) recycling, and (3) product integrity, where the last two categories are addressing the specific demonstrator.

(1) General considerations

- Does the making or use of the product create pollution?
- Can the material and energy used to make or use this product be reduced?
- Can nature regenerate the resources used to make or use this product?
- Can the product be created using alternative materials or sourced from other parts of the world?
- Can the components be sourced from sustainable manufacturing?

(2) Recycling

- What are the main materials used for the pipe temperature sensor?
- Can all the materials be recycled – are they "pure"?
- How should the parts be assembled so they can be reused?

(3) Product integrity

- What are the key aspects to obtain a durable product, in terms of construction and material?
- What are the most important components to be able to maintain?
- Do you foresee the need to update any components in order to maintain the integrity of the product (e.g., the sensors)?
- Could the same functionality be obtained using a mobile phone or another existing piece of technology?

6 Organizational information

6.1 Electronics, software, and mechanics development

The following requirements apply.

- Project duration: 4 sessions during a semester. Each session consists of 4 modules of 45 minutes per module.
- Group size: Ideally in smaller groups, depending on the number of available platforms. Groups can timeshare and run programs on it in turn.
- Prerequisites: None.

7 Description of the technology and the setup

7.1 Bill of Materials

The following table will outline the materials needed to produce the demonstrator. Note this is the equipment for one broker node (Raspberry Pi) and one embedded unit (ESP32). The total price will automatically update if the quantity of embedded units is changed.

Table 4: Classification of the focus project according to the dimensions

Quantity (Unit)	Description	Price (per Unit)	Price (total)
1 pc	Raspberry Pi 4B, 4 GB	64 €/pc	64 €
1 pc	Black Aluminum Case for Raspberry Pi 4B	30 €/pc	30 €
1 pc	16 GB micro SD card	12 €/pc	12 €
1 pc	USB type-c Powersupply	14 €/pc	14 €
1 pc	Embedded Unit (see below)	33 €/pc	33 €
			153 €
Embedded Unit:			
Quantity (Unit)	**Description**	**Price (per Unit)**	**Price (total)**
1 pc	TinyS3 - ESP32-S3 Development Board	18.5 €/pc	18.5 €
1 pc	Lithium Ion Polymer Battery - 3.7V 350mAh	6.5 €/pc	6.5 €
2 pc	DS18B20 Digital temperature sensor	4 €/pc	8 €
			33 €

Cables are not included.

8 References

[1] Energistyrelsen, “Om vandforbrug | Energistyrelsen.” https://sparenergi.dk/offentlig/bygninger/forbrug/om-vandforbrug (accessed Dec. 05, 2022).

[2] John Biggs and C. Tang, *Teaching for Quality Learning at University*, 3rd ed. McGraw-Hill/Society for Research into Higher Education & Open University Press, 2007. Accessed: May 15, 2017. [Online]. Available: http://www.citeulike.org/group/4810/article/3680928

[3] “Aalborg Forsyning: Privat.” https://aalborgforsyning.dk/privat/fa-hjaelp-af-os/regninger-og-betaling/ny-varmeafregning/ (accessed Jan. 09, 2023).