DIGITIZING PRODUCTS:
CREATING DEMONSTRATORS
FOR FUTURE EDUCATION

Conveyor Demonstrator

Demonstrator classification and documentation

Dornbirn, March 2023

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

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The DigiDemo project consortium's members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. © 2023 by DigiDemo Consortium.

The content of this document represents the views of the authors only and is their sole responsibility; it cannot be considered to reflect the views of the European Commission, the Education, Audiovisual and Culture Executive Agency (EACEA) and/or any other body of the European Union. The European Commission and the Agency do not accept any responsibility for use that may be made of the information it contains.

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.

1 Contents

Document authors

	First name Last name	Institution
Key author	Horatiu O. Pilsan	FHV
Further authors	Robert Amann	FHV

Revision history

Version	Date	Author(s)	Description
1.0	2022-11-10	Horatiu O. Pilsan	Initial draft
1.1	2023-03-08	Horatiu O. Pilsan	Added sustainability educational information and did some minor corrections

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)
PCB	Printed Circuit Board
PTP	Precision Time Protocol
A/D	Analog to Digital (Converter)
IEEE	Institute of electrical and Electronic Engineers
CAD	Computer Aided Design

List of figures

List of tables

1 Introduction

This demonstrator is intended to be used in the degree program to practice the integration of different heterogenous software element into one project. Both time and event-based systems are to be combined implementing both sequential control, closed loop control, user interface and communication.

2 Overview

The key properties of the conveyor project are:

Table 1: Specification of key properties of the conveyor project

Key Property	Value
EQF level	7 (Master)
Year of study	1 (in Master)
Domain	Mechatronics & IoT
Objective	Hands-on
Workload	4 ECTS
Keywords	Conveyor belt, embedded system, motor control

The conveyor belt project is part of a course in the 2nd semester of the mechatronics master program at FHV. The motor of the lab board powers a conveyor belt. The conveyor belts can either be operated locally or tied together to set up a closed chain. The purpose is to transport a load along the chain. The boards are interconnected via Ethernet. The movement of the conveyor belt shall follow a given profile. The speed of the motor shall be controlled in a closed loop. The physical setup can be seen in Fig. 1, the structure in Fig. 2.

The students' task is to design, implement and test the software needed for the setup. Details of the project organization can be found in the document "MEM2-ES_Project.pdf" in the folder "10_Documents". This document contains the information for the students at the beginning of the course. Further information on implementation details can be found in the documents "Additional_Requirements.pdf" and "Implementation_Guide.pdf" in the same folder.

For any open questions regarding the demonstrator please feel free to contact the author:

Prof. (FH) Horatiu O. Pilsan (E: hp@fhv.at, T: +43-5572-792-3505)

Figure 1: Physical setup of the conveyor demonstrator

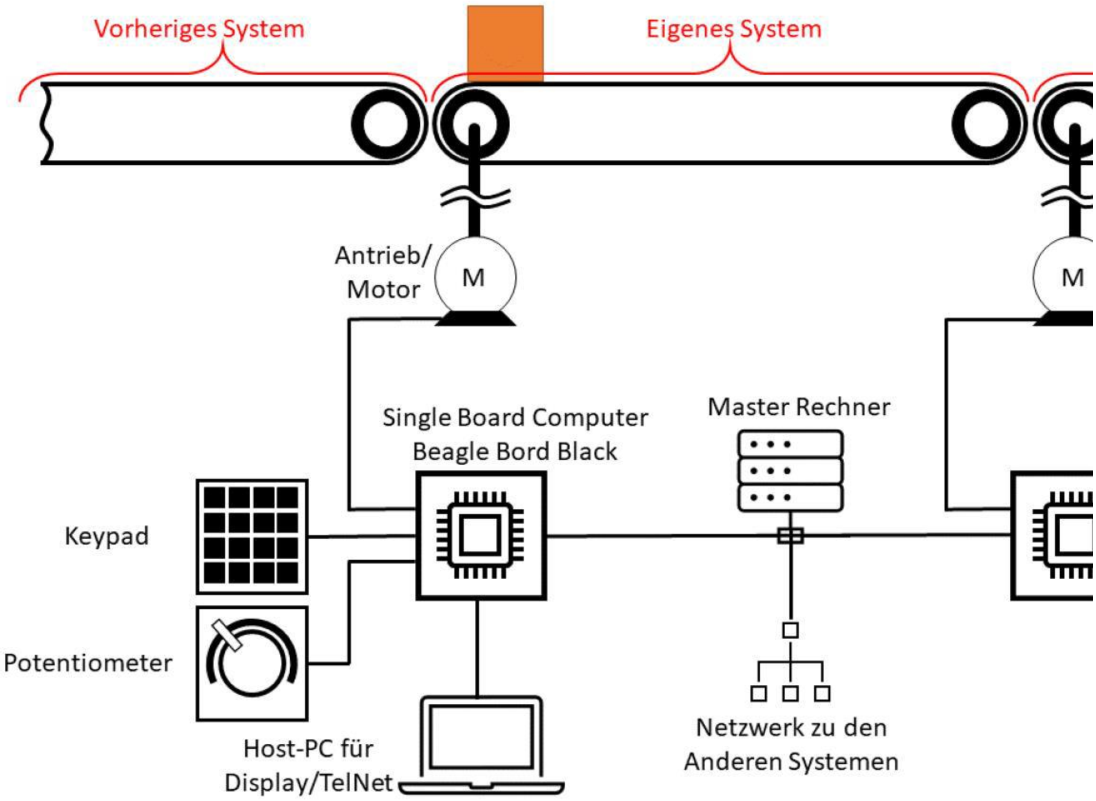

Figure 2: Structure of the conveyor demonstrator

3 Description of fulfilment of characteristics

The demonstrator fulfils the characteristics required as follows:

Table 2: Description of fulfilment of demonstrator characteristics

Characteristic	Description
Teaching improvement	This project allows the students to go through the entire process of software design and implementation including sensor and actuator integration, operating system use, data transmission and control system application.
Sustainability awareness	Mainly energy reduction is addressed, by ensuring that power saving modes of the system are used.
Replicability	The demonstrator is implemented using off the shelf components, which are easy to replicate. For the PCB all the data is supplied for replication.
Industry needs	As the demonstrator is a conveyor belt, lots of application in industry exist.
Interdisciplinarity	The demonstrator requires mechatronic students to integrate sensors and actuators with electronics and software, including an operating system and data communication.

4 Classification according to the dimensions

The demonstrator can be classified according to the dimension as follows:

Table 3: Classification of the conveyor 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	700
Costs (operation)	EUR	20
Workload (implementation)*	Hours	150

Dimension	Property	Value
Workload (operation)*	Hours	4
Size	m	0.15 x 0.12 x 0.11
Weight	kg	<1
Special requests	no/yes, if yes: which	no

* The workload refers to the effort needed by the teacher for implementation and operation of the demonstrator. As for the workload of the students please refer to chapter 5.

5 Educational information

5.1 Course information

The project is embedded in the course in which approx. 30 contact hours are used for input needed by the project. The remaining workload is covered by the project.

ECTS Description of the course in which the project is integrated

Course content

- The phases of the software development process
- Textual gathering of requirements
- UML diagrams for capturing requirements
- Requirements management
- UML state diagrams
- Methods for software quality assurance
- Types of software test
- Time synchronization according to IEEE1588
- Implementation of a project
- Develop an application-layer protocol for a given problem

Learning outcomes

The students can

- list the phases of a software development process and explain their properties.
- capture the requirements for a system both textually and graphically and manage them.
- set up a static software model using a UML class diagram.
- model the behavior of a system using state machines.
- enumerate and explain the methods used for software quality assurance.
- list the types of software test and explain the differences between them.
- explain the methods used to verify the real-time behavior of a system.
- explain the mode of operation of a time synchronization protocol.
- deploy a run-time framework.
- implement and test a hybrid (time/signal-flow and event/control-flow) system.
- apply code generation for the two components of the system.
- equip this type of system with a communication stack.
- provide a time synchronization for such a system.
- explain the nature of network protocols.
- define an own but simple application-layer protocol.

Planned learning activities and teaching methods

Interactive lectures, self-directed learning, laboratory exercises, practical project work

Assessment methods and criteria

Project submission, written examination

Recommended or required reading

Cooling, Jim (2003): Software-Engineering for Real-Time Systems. Harlow u.a.: Addison Wesley

Weilkiens, Tim (2006): Systems Engineering mit SysML/UML. 1. Auflage. Heidelberg: Dpunkt-Verlag

Partsch, Helmuth (1998): Requirements-Engineering systematisch. Heidelberg: Springer

Li, Qing (2004): Real-Time Concepts for Embedded Systems. San Francisco: CMPBooks

Hein, Michael; Reisner, Mathias (2001): TCP/IP gepackt. Bonn: mitp-Verlag

Mode of delivery (face-to-face, distance learning)

Face-to-face

5.2 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. It is suggested that the students in groups answer the questions and then present the result of the discussion on class.

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

General considerations

- Does the making or use of the product create pollution?
- How can the energy consumption of the demonstrator be reduced?

Recycling

- Can the demonstrator, or parts of it be recycled?
- Can all the materials be recycled – are they "pure" or do they need special treatment?
- How can the recycling of the demonstrator or its parts be improved?

Product integrity

- Which parts of the demonstrator need maintenance?
- Can an update of components (which?) be foreseen to enlengthen its lifespan?
- Is the software of the demonstrator easy to update?

6 Organizational information

For the organization of the project the following apply:

- Project duration: the project spans an entire semester, starting in the first days of March and ending late June with the final presentations.
- Team size: The project teams' optimal size is of 2 people.
- Preparatory and follow-up activities. No special activities are needed. Prior to the start of the project the demonstrators must be set up and checked by a lab technician.

7 Description of the technology and the setup

7.1 Demonstrator Documentation

The overview of the files supplied together with this description of the demonstrator is given below. It includes all files (e.g. CAD and PCB files) needed to replicate the demonstrator.

Figure 3: File overview of the conveyor project documentation

File Structure:	**Conveyor Setup**	
	Filename/Folder	Comment
00_Notes		
	FileStructure.xlsx	Description of the files = this document
	Price_Calculation.xlsx	
01_Pictures		
10_Documents		
	MEM2-ES-Project.pdf	Description of the project task for students
	Additional_Requirements.pdf	Additional requirements of the project
	Implementation_Guide.pdf	Guide for the students to implement the project
	SystemTopology.pdf	Overview of the system with borders to neighbours
20_ExportImport		
30_AdditionalEquipment		
	4708.pdf	Datasheet of the trimmer potentiometer
	1384416.pdf	Datasheet of the power supply
	1648836.pdf	Datasheet of the DC/DC converter
	AD8233.pdf	Datasheet of the amplifier on the PCB
	BBB_SRM.pdf	Manual of the Beagle Bone Black board
	MAY1270.pdf	Datasheet of the A/D converter on the PCB
	Maxon_Motor_ENC.pdf	Datasheet of the motor and the incremental encoder
	MC33HB2001.pdf	Datasheet of the H-bridge driver on the PCB
	sn74lv07a.pdf	Datasheet of the hex driver
	txb0104.pdf	Datasheet of the level shifter on the PCB
	vs-murb2020cthm3.pdf	Datasheet of the diode
40_Simulation		
	Gesamtmodell_Kaskade.pdf	Simulink model of the cascaded control and plant
	Gesamtmodell_Simple.pdf	Simulink model of the simple control and plant
	Kaskadenregler.pdf	Simulink model of the cascaded control only
	Motor_Modell.pdf	Simulink model of the motor
	Cascaded-Control.zip	Simulink file of the cascaded control and the generated C-code
	Simple-Control.zip	Simulink file of the simple control and the generated C-code
50_CAD		
	...	Several files describing the case of the setup
60_ElectricalLayouts		
	...	Schematic and layout files of the PCB
70_Programming		
	HardwareFuncions.zip	Functions provided to the students to access the peripheral component
	Master-Files.zip	Files for the master board, which generates IEEE1594 clock
	PtpFiles.zip	Files provided to students to implement the PTP protocol
	StateMachinFiles.zip	Files of the state machine framework provided to the students

The hardware (Beagle Bone Black) runs Linux Debian on it. For the project implementation in C++ Microsoft Visual Studio 2019 has been used since this is licensed at FHV. To implement the behavior using state charts (as requested in the task), a framework is supplied in which the students only need to fill in the transitions in an array. In the end of the project, they are requested to integrate a PTP (Precision Time Protocol) stack. For this purpose, the files of the PTP client are provided to the students and the PTP server is implemented on an additional board. The control files for the motor control are provided as Simulink files, the C code is generated using the Simulink embedded coder. In order to access the hardware (A/D converter, incremental encoder, motor H-bridge) the files needed by the students are provided.

7.2 Price Calculation

Details of the price calculation can be found in the table below:

Table 4: Price calculation of the conveyor demonstrator

Quantity (Unit)		Description	Price (per Unit)		Price (total)
1	pc	Maxxon Motor + Encoder	€ 255,00	€/pc	€ 255,00
1	pc	BeagleBoneBlack	€ 150,00	€/pc	€ 150,00
1	pc	PCB + Parts mounted on it	€ 200,00	€/pc	€ 200,00
1	pc	potentiometer	€ 5,00	€/pc	€ 5,00
1	pc	keyboard	€ 10,00	€/pc	€ 10,00
1	pc	case	€ 50,00	€/pc	€ 50,00
1	pc	miscelaaneous small material	€ 30,00	€/pc	€ 30,00
					€ 700,00

References