Integrating Digital Twin Technology into the Master Next Level Engineering: A Case Study on the Leaphy Platform

Bas de Graaf

HU University of Applied Siences Utrecht, Padualaan 99, 3584 CH Utrecht, the Netherlands

Abstract: This study focuses on offering a tangible Digital Twin (DT) learning experience in the Master Next Level Engineering (MNLE) program at the University of Applied Sciences Utrecht (HU), focusing on the Leaphy platform. MNLE blends system and data perspectives to prepare students for emerging technologies. One emerging topic is robotics, which includes engineering, electronics, and programming. Robotics offers an opportunity to challenge students in data-driven engineering. However, a gap exists in tools bridging Hands-On, virtual and remote-control engineering for students with minimal background in these areas. The proposed solution involves creating a DT. Using a DT makes working with a (physical) system and data possible. This research uses the education Leaphy robot, a well-described open-source robot. Leaphy represents the physical system of the digital twin, connecting theoretical knowledge with practical application. The research investigates DT levels, the creation process, industry applications, and social implications. Perspectives from the Institute of Design and Engineering (IDE) lecturers provide insights into applications and educational perspectives. The methodology outlines a structured framework with the V-model. The resulting product is an open-source website enabling simulation and real-time monitoring of physical systems, facilitating data collection and control.

Keywords: Digital Twin, Next Level Engineering, Leaphy, Robotics, Education

1. Introduction

This study aims to integrate a Digital Twin (DT) into the Master Next Level Engineering (MNLE) program at the University of Applied Sciences Utrecht (HU) (Hogeschool Utrecht, n.d.) using the Leaphy platform (Stichting Leaphy, n.d.). The objective of bridging theoretical knowledge and practical application is to equip students with the necessary skills for emerging technologies, particularly robotics. There is a notable gap in tools for bridging between Hands-On, virtual and remote-control engineering for students (Alsaleh et al., 2022). Students find it hard to visualise the data streams.

Leaphy: As an open-source platform, the Leaphy robot provides an accessible, affordable and practical basis for creating the DT, ensuring that students can engage with the physical and digital engineering aspects, as shown in Figure 1.

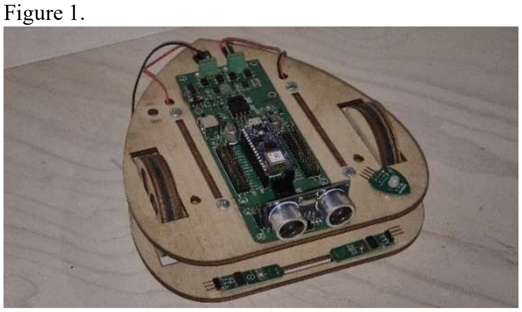

Figure 1 Leaphy Original V2

Students and teachers at Coderius College in Amersfoort developed Leaphy, an affordable and accessible robot gaining popularity in Dutch secondary education without commercial or university backing (NVON | Robotica Voor Iedere Leerling, n.d.). Over 2,000 units of Leaphy have been produced (NVON, n.d.).

Research based on Leaphy has shown that computational thinking is a growing skill among students, its assessment remains contentious. A pilot study found a positive correlation between computational and logical thinking but no significant impact from a robotics course (Veenman et al., 2022).

Technological advancements are increasingly bringing robotics into classrooms. However, high teacher workloads and knowledge gaps continue to be issues. Near-peer teaching has proven effective in improving students' STEM attitudes and skills while alleviating teacher workload (De Vink et al., 2022).

Other topics focused on the persistent gender gap in STEM attitudes, highlighting that boys score higher in technology and engineering attitudes (Van Wassenaer et al., 2023).

Digital twin: A DT should be able to mimic real-life situations accurately, letting users predict outcomes and test different scenarios. It needs to adapt to changes in real-time data and allow users to explore hypothetical situations for better decision-making. Overall, it should help optimise performance, manage risks, and improve decision-making in various fields like manufacturing (Tao et al., 2022). A DT system features a continuous data exchange between the physical asset and its digital twin, allowing bidirectional interaction, as shown in Figure 2.

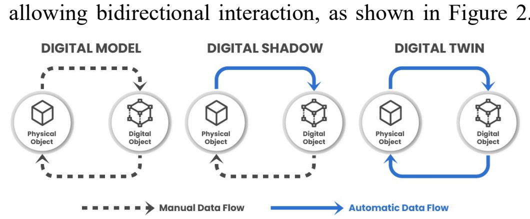

Figure 2 Digital Twin Levels (Andrade, 2023)

Changes in one can directly affect the other. This constant, fully integrated data flow characterises a Digital Twin (Andrade, 2023).

The MNLE program blends system and data perspectives to prepare students for new technological challenges. Robotics, including engineering, electronics, and programming, prepares students for various MNLE fields. This research aims to fill this gap by introducing DT, an educational tool that challenges students' problem-solving skills and practical knowledge.

A Digital Twin (DT) offers advantages over other digital tools in education, particularly in disciplines like engineering. Research by Liljaniemi and Paavilainen (2020) highlights that DTs enhance students' understanding of complex concepts, increase motivation, and improve learning outcomes, as seen in courses such as "Simulation in Control System Design." Unlike traditional digital tools, DTs enable real-world simulations without physical constraints, better-preparing students for engineering challenges and making them more competitive in the job market. However, it is crucial to balance digital learning with in-person interaction, as excessive reliance on technology may diminish the value of face-to-face engagement with educators and peers (Liljaniemi & Paavilainen, 2020).

Despite challenges like IT issues and resource shortages in education, students perceive DT technology positively due to its ability to enhance engagement and skill development. The prospects of DTs in terms of professional relevance and learning facilitation are highly regarded. The MNLE learning outcomes from Appendix VII further emphasise the importance of applying systems engineering, data science, and interdisciplinary approaches in education. As shown in Appendix I, interviews with IDE lecturers shed light on specific requirements for DT implementation, helping tailor the technology to meet educational needs effectively (Liljaniemi & Paavilainen, 2020).

Integrating DT technology through the Leaphy platform will provide a hands-on learning experience that connects theoretical concepts with real-world applications. This research focuses on the technical process of creating DT and its potential as a learning tool. By exploring the creation and implementation of DTs, this study aims to provide a holistic view of their impact on the learning experience. The primary research question is:

"How can a Digital Twin be created as an educational tool in the Master Next Level Engineering program?"

To address this overreaching question, the following sub-questions will be explored:

1. *What essential features of physical systems must be included in a Digital Twin?*
2. *What key learning outcomes must be incorporated into the Digital Twin?*
3. *How can a Digital Twin framework be built for the Leaphy platform?*

2. Methods and Materials

A modified version of the traditional V-Model integrates DT technology into both the development and testing phases, as illustrated in Figure 3. The V-Model follows the conventional steps of the traditional model. This integration enables simulation, data collection, problem detection, and system planning before manufacturing (Wagg et al., 2020). The V-model is a structured framework that guides creating a DT, from identifying the educational and technical requirements to maintenance after deployment.

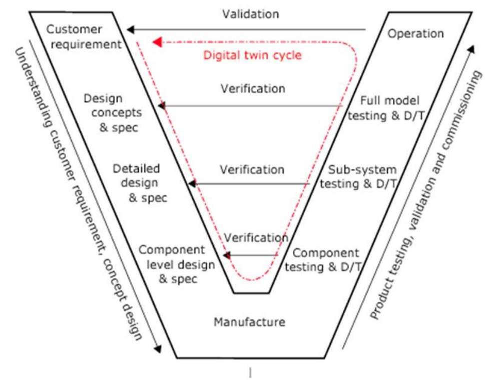

Figure 3 V-model Digital Twin (Wagg et al., 2020)

The process begins with initial planning and scope definition, where the project's scope and objectives are outlined, and key questions and goals are set. After this, relevant stakeholders are identified and involved to ensure their perspectives are considered.

The next step involves conducting thorough research on digital twins. This research aims to understand what digital twins are, how they are used, when, and why they should be used.

2.1 Customer Requirements

Subsequently, the IDE conducts interviews to gather insights on the need for a digital twin and its potential applications. Voice memos are recorded during these interviews, and the main outcomes of each expert are summarised.

After collecting these insights, the educational and technical requirements necessary for the DT to achieve its goals are defined. These interviews assess the needs of the MNLE across the IDE program—including mechanical engineering, industrial administration, electrical engineering, information technology, and building and environment.

A Requirements list is created and prioritised using the MoSCoW method (MOSCOW, 2022). This step includes validating the requirements with stakeholders to confirm that the list meets everyone's needs.

2.2 Design

The overall DT system design is crafted in a high-level design, focusing on the system architecture, including its interaction with the physical Leaphy model, data collection and exchange, and its application in an educational context. The next step is the detailed architectural design, which includes the preferable open-source software and components compatible with the hardware from Table 1, user interface design, data storage solutions, and communication protocols.

Table 1 The initial set of hardware

Hardware	Amount
The Leaphy Original frame	10
DC motor	20
IR Sensor	20
Ultrasonic sensor	10
Leaphy Delphy shield	10
Leaphy Delphy I2C Module	10
Leaphy Delphy Motor Module	10
RGB LED	10

The module design phase breaks the architectural design into smaller, manageable modules or components. This phase is for developing simulation tools, real-time feedback mechanisms, and integration with external data sources. With the components defined, the implementation phase starts, focusing on coding and developing each component and integrating them with the Leaphy platform.

2.3 Implementation

The implementation phase focuses on translating the detailed design into a functional DT. This begins with coding and developing each component and module as specified in the design. The next step is integrating these individual components into a complete system, ensuring that all parts work together seamlessly and communicate effectively.

System configuration is then carried out, which includes setting up communication protocols, configuring data storage solutions, and ensuring compatibility with the hardware components. Thorough testing and debugging are conducted to identify and fix any issues.

2.4 Validation

Unit testing follows, where individual components or modules undergo thorough testing to ensure their correct functioning before integrating into the more extensive system. The full system test phase shows the various DT components coming together, ensuring operation and meeting initial requirements.

The operational test involves validating the DT with end users, such as students and faculty from the MNLE program, to gather feedback on usability, effectiveness, and educational impact. The acceptance testing phase is the final evaluation based on system testing feedback, adjusting as necessary to confirm the DT's readiness for full educational deployment.

3. Results

This chapter outlines the results of the development of the Leaphy DT, followed by the methodology. It covers requirement analysis, system design, detailed design, implementation, integration testing, system testing, acceptance testing, and iterative improvements based on feedback. These steps ensured that the DT aligns with the educational and technical needs of the MNLE program.

3.1 Requirement Analysis

The requirement analysis phase defined the Leaphy platform's DT needs. This included understanding the system it would represent, the operational environment, and constraints. The requirements were prioritised using the MoSCoW method: Must have (M), Should have (S) and Could have (C) (MOSCOW, 2022). As summarised in Appendix I, the requirements were gathered through interviews with stakeholders, including IDE lecturers and MNLE program coordinators, to ensure alignment with educational goals and technical capabilities. The complete requirement list is shown in Table 2.

Table 2 Requirement list DT

Nr	Pr.	Requirement
1	M	Align DT with specific MNLE learing outcomes
2	M	Ensure DT compatibility with the Leaphy original model
3	M	Enable bi-directional data exchange
4	M	Provide simulation tools for system testing and problem-solving
5	M	Achieve platform accessibility stands
6	M	Implement real-time feedback mechanisms
7	S	Conduct usability testing with students and faculty
8	S	Facilitate DT applications across IDE
9	C	Implement ML and RL simulation tools for DT
10	C	Ensure DT scalability for future expansion

The MNLE learning outcomes are divided into four categories: system, data, change and impact, as shown in Appendix VII.

A. The student is leading in applying systems engineering, modelling, and prototyping within an interdisciplinary system development process.
B. The student is leading the process of applying data science within an interdisciplinary system development process.
C. The student plays a leading role in an interdisciplinary system development process by navigating a broader context of socio-technical change.
D. The student makes a meaningful impact with an applied research project that supports an interdisciplinary system development process.

These MNLE learning outcomes indicate a strong emphasis on interdisciplinary skills and leadership. Students must master technical competencies in systems engineering, data science, change and impact. The focus on impact through applied research highlights the program's commitment to practical, real-world problem-solving.

3.2 System Design

The high-level design is divided into two main components, as shown in Figure 4: the digital twin components and the learning experience. Four learning outcomes are included, where systems and data are more integrated in the digital twin and change and impact the hackathon.

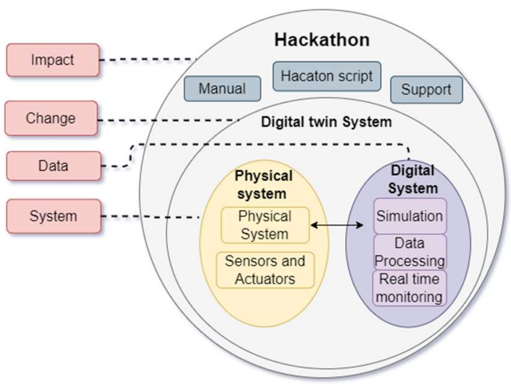

Figure 4 High-level Design DT

The system architecture comprises the physical Leaphy Original V2 robot and a digital model. The Leaphy robot, equipped with sensors and actuators, functions as the physical system. The digital model is a virtual representation of the robot, simulating its behaviours and interactions. The communication layer facilitates bidirectional data exchange between the robot and its digital counterpart. The digital twin user interface enables students to interact with the digital twin for real-time monitoring and control.

3.3 Detailed Design

The detailed design phase focused on specifying each component for the DT and preparing for the hackathon week. This involved defining hardware, software, communication, and the interface for the DT. As part of the DT learning experience, the hackathon will be designed iteratively with the stakeholders, refining each day with specific tasks, requirements, guidelines, and daily planning.

3.3.1 Hardware

A line follower robot is being developed to create the physical system. It detects the path by sensing the line and follows it, adjusting its direction by steering left or right based on the track. The system uses two infrared sensors to detect a black line on a white background. Additionally, an ultrasonic sensor is used for distance measurement to detect objects in the path. These sensors gather environmental data. The actuators consist of two DC motors that provide movement; the hardware is illustrated in Figure 5.

Figure 5 Hardware Leaphy Original V2

3.3.2 Software

The full DT system has three different software components: the Leaphy system, the master module, and the user web interface. The master module facilitates wireless communication between the Leaphy and the website, as illustrated in Figure 6.

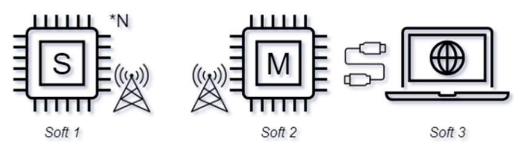

Figure 6 Slave, Master, Web interface Communication

Starting with the Leaphy (slave) module, this software controls the physical system by following a line while updating the current status to the master.

The master module manages incoming status updates from multiple Leaphy modules and updates the website accordingly. Although, in theory, the master module could be removed, and a laptop could handle the wireless communication directly, in practice, not all laptops are compatible with the latest communication hardware. Due to its reliable serial communication, the master module ensures a more seamless plug-and-play experience.

The third component is the simulation engine, which started with Scratch to create a line-following robot simulation, as shown in Figure 7. Scratch is an open-source block game simulation program that allows fast prototyping for games (Scratch, n.d.). Initially, Python was used for the simulation, but it proved insufficient for simple visualisations. After extensive research on forums, communities, websites and online videos, LeopardJS was found (*Leopard*, n.d.), allowing Scratch projects to be exported to JavaScript and providing more coding flexibility, enabling the addition of advanced functionalities and hosting on a local web server.

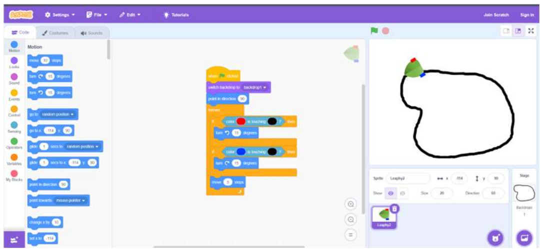

Figure 7 Scratch line follower game (Scratch, n.d.)

Simulation Algorithms were chosen to calculate the robot's new position. Colour interactions (e.g., red for sensor touch and black for line detection) were used to calculate speed and direction based on environmental changes, as shown in Figure 8.

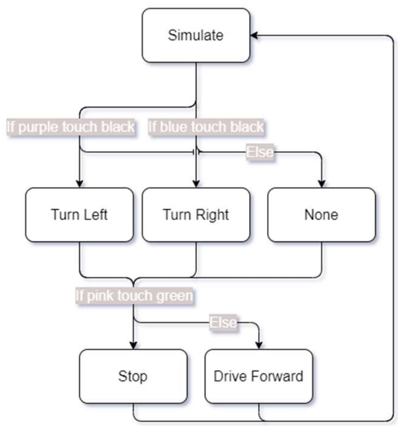

Figure 8 Colour Interactions Simulation

3.3.3 Communication

The robot state data model includes attributes like position, orientation, speed, and sensor readings in JavaScript Object Notation (JSON) format. The command data model includes command type and parameters.

Communication Protocols were initially attempted using Bluetooth but proved infeasible due to its one-to-one connection limitation (Devices Over Bluetooth, n.d.). The ESP32 was chosen to support the Painless Mesh networking library, enabling effective communication among multiple robots (*Painless Mesh*, n.d.).

Web socket is used for data transmission, ensuring minimal latency and reliable real-time communication (Serial Prot Websocket, 2020). The data can be logged using a CSV web socket (Export CSV, n.d.). Commands are sent to the robot, and acknowledgements and status updates are returned to the user interface. I

There are three feasible ways to update the DT representation with limited resources: 1) updating the model by transferring forward, stop, left, and right commands; 2) adding encoders to the wheels to measure the distance travelled; and 3) using checkpoints along the track to provide a global impression of the last known location. A combination of these methods can also be used to enhance accuracy.

3.3.4 Interface

The user interface module was initially planned for integration into the LeaphyEasybloqs website (Leaphy Easybloqs, n.d.). However, it was later decided to host the DT on a dedicated site for two reasons: first, the steep learning curve of Angular, and second, the Leaphy Foundation's switch to Svelte as the new framework (Angular, n.d.) (*Svelte*, n.d.). The interface, developed in JavaScript, focuses on simulation, real-time monitoring, and data logging. Users can adjust parameters such as speed, turn degree, and the number of Leaphys for the line follower simulation. Table 3 provides a complete overview of the software used in the project.

Table 3 Software Overview

Software	Description
Scratch	Game simulation development
Easybloqs	Block programming and controlling the robot
Python	Custom data processing and simulation tools
Angular	Developing web interfaces Easybloqs
Java	developing web interfaces Digitaltwin
LeopardJS	Exporting games to javascript
Git	Version control and collaboration

3.3.5 Learning experience

The following sections summarise each day of the hackathon. A complete script with further details is provided in Appendix II.

Mo: Begin with an introduction to the Leaphy Line Follower robot and digital twin concepts. Program the robot to follow a line using IR sensors and optimise performance using Easybloqs coding environment. In the afternoon, focus on movement control, obstacle detection, status indication, and surface colour detection.

Tue: Develop an interactive system. Program the robot to pick up and place objects at designated spots, navigate between locations, and identify objects with the colour sensor. In the morning, focus on object manipulation. In the afternoon, cover navigation, object identification, and control operation sequences through a central control system.

Wed: Manage data and detect maintenance needs. Modify robots to detect issues like smaller wheels or dirty sensors. Implement sensors for maintenance detection and analyse collected data. In the morning, cover data collection and storage. In the afternoon, focus on data analysis, visualisation, real-time monitoring, and control adjustments.

Thu: Address change management and human factors. Reflect on technical work and consider real-world implications by solving a hypothetical disruption scenario. In the morning, focus on the human impact of automation. In the afternoon, handle system upgrading, workforce training, community engagement, and disruption management.

Fri: Conclude with impact evaluation and professional skills development. Reflect on the week's activities and prepare presentations to showcase the work. Analyse the system's performance, evaluate the impact of changes, develop professional skills, and present outcomes to stakeholders.

3.3.6 Design Test

Four test plans were created to validate the design during the V-model's test phase. These tables are designed to validate four key aspects: the Leaphy (slave) module, the master module, the web interface, and the learning outcomes. The detailed tables can be found in Appendix III through V.

3.4 Implementation

The implementation phase translated the detailed design into a functional system. This involved setting up hardware and software components, realising communication protocols, and ensuring the user interface was operational.

3.4.1 Hardware

To integrate the hardware components, the ESP32 microcontroller was connected to the infrared line sensors and ultrasonic sensors for obstacle distance detection, and the DC motors were connected to motor drivers. The printed circuit board minimised interference and ensured reliable connections. Each sensor and motor were tested individually to ensure proper functionality before further integration. This is demonstrated in the video by Bas de Graaf (2024) on creating a Leaphy Line follower.

3.4.2 Software

The software development involved creating the simulation engine using LeopardJS and converting Scratch projects to JavaScript. Simulation scenarios were created for warehouses, bus stations, and plain fields, as shown in Figure 9. Algorithms for position and sensor interactions were developed to accurately simulate the robot's movements based on colour triggers and responses. The simulation engine was hosted on a local webserver to allow easy access and iterative testing. Later, the website was hosted on a web server to make it accessible to everyone (Leaphy Digital Twin, 2024).

Figure 9 Warehouse Forklift simulation

3.4.3 Communication

Communication was established using mesh networking with ESP32 and WebSocket for data transmission wireless. ESP32 microcontrollers were configured with the painless mesh networking library to enable communication among multiple robots, as shown in Figure 10.

The master node communicates with the computer through serial communication, ensuring data transfer between the physical robots and the digital model. The WebSocket detects data from the serial port and transmits it to the website, updating the digital twin model. The system was tested by sending commands and receiving status updates, ensuring communication, real-time control, and monitoring among the robots and the control system.

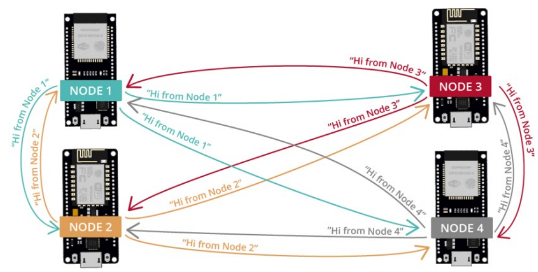

Figure 10 Painless mesh (Santos and Santos, (2020))

Data is captured while the robots are in motion, using JSON for data interchange and SQLite (SQLite, 2024) for local storage. Another socket was implemented on the website to store the data and convert it into a CSV file with one button press.

3.4.4 Interface

The user interface included controls for adjusting simulation parameters, real-time monitoring of robot status, and data logging for performance analysis. It was designed with intuitive controls and real-time monitoring features. The interface also included data logging capabilities to review historical performance and make informed adjustments, as shown in Figure 11.

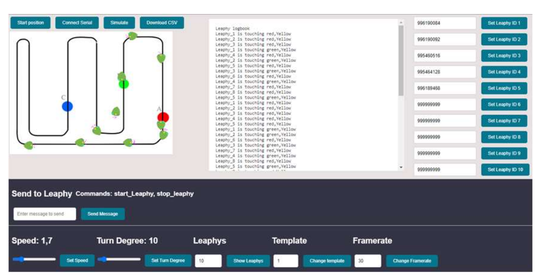

Figure 11 DT website (Leaphy Digital Twin, 2024)

For full access to the software project, refer to Bas de Graaf's work on GitHub (De Graaf, n.d.).

3.4.5 Setting up the system

Figure 12 shows a setup flowchart to provide a step-by-step guide for setting up the Leaphy robot with the DT. The flowchart helps identify the actions to take if an error occurs, troubleshoot the Leaphy robot, and provide systematic procedures for error correction, facilitating a structured building strategy.

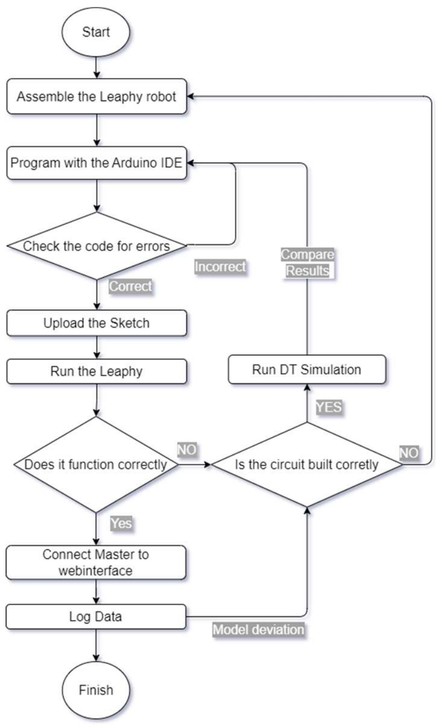

Figure 12 Flowchart Set-up DT

3.5 *Integration Testing*

This phase validates the individual component and the interaction between the slave modules, master module, and web interface to confirm functionality. The results are stated in Appendix III through V.

The Leaphy robots, functioning as slave modules, responded well during integration testing (Appendix III). Commands such as "start" and "stop" were executed correctly, with the robots performing the intended actions. Sensor data collection, including line-following and obstacle detection, was accurate and transmitted to the master module. The Leaphy robots could follow the line and navigate the entire track. Real-time updates were consistent and reliable, confirming proper functionality.

The master module successfully managed communication between the Leaphy robots and the web interface (Appendix IV). It received data from multiple slave modules and updated the web interface in real time. The mesh networking ensured stable and reliable connections among all nodes. Tests confirmed that the master module handled data transmission smoothly and maintained data integrity.

The web interface provided a platform for user interaction with the DT system (Appendix V). It displayed real-time data from the Leaphy robots and allowed users to control their movements and settings. Features like starting and stopping the robots, adjusting speed, and viewing data updates from the nodes worked as intended.

These results confirmed that the individual components can work together effectively. The next phase will validate whether the entire system functions as a complete DT.

3.6 *System Testing*

In the previous phase, the individual components were verified to work as intended. This phase involves evaluating the complete Digital Twin (DT). The potential learning outcomes for the MNLE will be stated to provide context for filling out the rubric.

The hardware functions correctly, and the software operates as designed. However, the interface struggles with synchronisation. Initially, three methods were proposed to synchronise reality with the digital environment: updates in all directions (forward, backwards, left, right), second encoders, and three checkpoints.

The first method, which was based on updating in all directions, was the easiest to implement but proved unreliable. It quickly led to incorrect x and y coordinates and direction representations within seconds. The second method, using encoders, was more challenging to implement and still carried risks of travelled distance and losing grip, causing synchronisation problems on the track. The third method, updating position by fixed grid points, was the most feasible. When an object reaches a predefined point, its position is updated accordingly. This third method is implemented, and the foundation is completely established.

3.7 *Acceptance Testing*

The acceptance process involves evaluating the system in collaboration with the project initiator and presenting the results in the MNLE, ensuring alignment with predefined criteria. This validates that the system meets technical specifications and the learning outcomes. The validation process included the following steps:

1. **Execution**: Conducted the tests according to the plans, involving end-users to gather practical feedback.
2. **Evaluation**: Analysed the test results to identify discrepancies or issues, ensuring the DT's performance met all requirements.
3. **Iteration**: Addressed any identified issues through iterative development and re-testing until the DT met all acceptance criteria.
4. **Agreement**: The project initiator and lecturers reviewed and agreed upon the test plans and plans to ensure alignment with educational and technical goals.

3.7.1 Digital twin

Integration testing is conducted to ensure that all individual components work, as shown in Appendix III to V. This was successful, and the complete system works together. This includes verifying the communication between the Leaphy robots (slave modules), the master module, and the web interface. Extensive testing ensures that the system functions correctly. Usability testing with students and faculty ensures that the DT is user-friendly and meets the initial requirements, as shown in Table 4.

3.7.2 Learning outcomes

The MNLE program has specific learning outcomes: systems, data, change, and impact, as shown in Appendix VII. The digital twin supports these by enabling students to engage in practical, real-world scenarios that enhance their understanding and application of interdisciplinary concepts. For example, the digital twin mimics an automatic warehouse with forklifts.

Interviews with lecturers provided valuable insights into the digital twin's needs and expectations (Appendix I). Key points include the importance of real-time data interaction, the application of interdisciplinary skills, and the ability to simulate real-world scenarios to improve understanding and problem-solving capabilities. A requirement list (Table 1) was built based on the interviews.

To guide and provide the learning experience, a hackathon script outlines a week-long series of activities that engage students with the digital twin (Appendix II). On Monday, students are introduced to the Leaphy Line Follower robot and digital twin concepts. They learn basic programming and optimise the robot's performance. Tuesday focuses on developing an interactive system where students program the robot for object manipulation and navigation tasks. Wednesday involves data management and maintenance detection, where students modify robots to detect issues and analyse collected data. Thursday emphasises change management and human factors, reflecting on technical work and understanding real-world implications. Finally, on Friday, students analyse system performance, evaluate the impact of changes made throughout the week, and present their findings.

Appendix VI details how the DT system learning outcomes for 'data' and 'system' are achieved. The topics 'change' and 'impact' are not part of the DT system but are integral to the hackathon experience. In systems engineering, students follow the V-model to design, develop, and validate systems (1.1). For instance, during the hackathon, they create a digital twin of the Leaphy robot, moving iteratively through the design and implementation phases (1.2, 2.2, 3.1). This practical application enables students to apply systems engineering principles effectively, thus enhancing their comprehension of interdisciplinary system development (2.1, 3.2).

In data science, students collect real-time data from the Leaphy robots' sensors, such as infrared and ultrasonic sensors (1.1). They preprocess and analyse this data to optimise the robots' performance (1.2, 2.2). Through these activities, students apply data science techniques, ensuring data validity and reliability, which helps them make informed decisions and solve problems effectively (2.1, 3,1, 3.2, 4.2).

In addressing change management, students participate in activities focusing on the human impact of automation and technical changes. For instance, they engage in discussions and scenarios that consider the broader socio-technical context, such as the potential consequences of automation on the workforce. This helps them develop skills to navigate and manage change effectively in real-world settings.

Regarding impact, students conduct applied research projects during the hackathon. They analyse system performance and evaluate the impact of changes made throughout the week. Presenting their findings to stakeholders demonstrates their ability to make meaningful contributions to interdisciplinary system development processes. These projects emphasise the practical application of their knowledge and skills,

Table 4 Requirment Checklist

Nr	Pr.	Requirement	Result	Fufulled
1	M	Align DT with specific MNLE learning outcomes	Learning outcomes are aligned and accepted as documented (Appendix VI)	√
2	M	Ensure DT compatibility with the Leaphy original model	Full compatibility ensured with the Leaphy Original model	√
3	M	Enable bi-directional data exchange	Bi-directional data exchange successfully enabled	√
4	M	Provide simulation tools for system testing and problem-solving	Comprehensive simulation tools have been provided for testing and problem-solving	√
5	M	Achieve platform accessibility standards	DT Website meets all required accessibility standards	√
6	M	Implement real-time feedback mechanisms	Real-time feedback mechanisms have been successfully implemented	√
7	S	Conduct usability testing with students and faculty	Usability testing conducted with both students and faculty (Appendix IV)	√
8	S	Facilitate DT applications across IDE	DT application focus is currently limited to the MNLE	x
9	C	Implement ML and RL simulation tools for DT	ML and RL simulation tools are not supported at this time	x
10	C	Ensure DT scalability for future expansion	The platform is designed to allow for future expansion	√

ensuring they are prepared to tackle complex engineering challenges in their future careers.

Three requirements have not been fulfilled: the DT only focused on the MNLE, and there are ways to use it in other studies that have not been validated yet. Due to the choice of hardware, Machine Learning (ML) and Reinforcement Learning (RC) are not currently supported. The acceptance testing phase confirmed that the DT aligned with educational and technical requirements, ensuring its readiness for deployment in the MNLE program.

3.8 Feedback and Iteration

Weekly meetings with the project initiator were essential for updates and direction discussions. Feedback from IDE lecturers and MNLE was integrated to meet educational needs. Continuous testing, feedback, and adjustments refined the DT system.

Discussion

This study aimed to integrate DT into the MNLE program at HU using the Leaphy platform. The primary research question was:

"How can a Digital Twin be created as an educational tool in the Master Next Level Engineering program?"

Addressing the Main Research Question:

The development of the Leaphy DT involved creating a framework that integrates essential features of physical systems with key learning outcomes. The DT facilitated real-time data exchange and interaction between the physical Leaphy robot and its digital counterpart. This approach successfully bridged theoretical knowledge with practical application, aligning with the MNLE program's educational goals.

Sub-Questions Analysis:

1. **Essential Features of Physical Systems in a DT:** The Leaphy DT incorporated essential features such as real-time data exchange, accurate sensor data collection, and the ability to simulate physical behaviours in the digital model. These features ensured that students could interact with and understand the physical system through its digital twin.
2. **Key Learning Outcomes:** Key learning outcomes included applying systems engineering principles, data science skills, and understanding change. The Leaphy DT provided a platform for students to engage with these outcomes through hands-on activities and real-time feedback.
3. **Building a DT Framework for Leaphy:** The framework was built using a structured V-model approach, integrating hardware components like the ESP32 microcontroller, sensors, and actuators with software tools such as LeopardJS and WebSocket for real-time communication. The process included stages of requirement analysis, system design, detailed design, implementation, integration, and testing, ensuring a comprehensive development cycle.

Technological Implementation and Challenges: The implementation faced synchronisation issues between the physical and digital models. Methods like updating positional data through fixed grid points were tested, but further refinement is needed. These challenges highlight areas for future research and improvement.

Educational Impact and Interdisciplinary Integration: The Leaphy DT impacted education by integrating theoretical and practical components, aligning with the interdisciplinary nature of the MNLE program. Interviews with IDE lecturers confirmed the DT's effectiveness in enhancing educational experiences and preparing students for real-world engineering challenges.

4 Conclusion

Integrating a DT into the MNLE program at HU through the Leaphy platform has the potential to be a valuable educational tool. The Leaphy DT supports students' understanding of complex engineering concepts, facilitates real-time data interaction, and provides a hands-on learning experience that bridges theoretical knowledge with practical application.

Contributions and Economic Implications: The Leaphy DT demonstrates a practical application of DT in engineering education. It suggests that digital twins could enhance teaching methods and help students develop technological skills. Economically, using open-source platforms like Leaphy could make this technology more accessible and affordable, benefiting educational institutions with limited budgets.

Acknowledgement

I would like to acknowledge the support and assistance provided by Stichting Leaphy Robotics. I am working as a volunteer for the foundation; I do not receive any financial compensation. I declare that no financial support or relationships may pose a conflict of interest related to this work. Additionally, I would like to thank those who provided technical assistance during the research.

Funding Information

The authors acknowledge the financial support and resources Stichting Leaphy and The HU provided. Stichting Leaphy provided the necessary hardware during the project's test phase, and The HU covered the project's overall costs for the implementation phase.

Author Contributions

Bauke Steenhuisen, Steven Haveman, Olivier van Beekum, and the thesis team (Rick Bakker, Sergio Castillo Watson, Naut van Goor, and Jakob Steger) assisted with the project. Their expertise and dedication were invaluable to the successful completion of this work.

Conflict of Interest

The authors declare no conflicts of interest regarding this work's research, authorship, and publication in this manuscript. There are no financial, personal, or other associations that have influenced or could be perceived to influence the content of this manuscript.

References

1 Alsaleh, S., Tepljakov, A., Kose, A., Belikov, J., & Petlenkov, E. (2022). ReImagine Lab: bridging the gap between Hands-On, virtual and remote control engineering laboratories using digital twins and extended reality. IEEE Access, 10, 89924–89943. https://doi.org/10.1109/access.2022.3199371

2 Andrade, A. (2023, January 26). The 3 levels of the Digital Twin Technology – 2023. Vidya. https://vidyatec.com/blog/the-3-levels-of-the-digital-twin-technology-2/

3 Angular. (n.d.). Angular. Retrieved May 27, 2024, from https://angular.dev/

4 De Graaf, B. (n.d.). GitHub - Basie147/Scratch_Leaphy. GitHub. Retrieved May 27, 2024, from https://github.com/Basie147/Scratch_Leaphy

5 De Graaf, B. (2024, May 9). Leaphy Line follower [Video]. YouTube. https://www.youtube.com/watch?v=ofu_sMj_EAQ

6 De Vink, I. C., Tolboom, J. L. J., & Van Beekum, O. (2022). Exploring the effects of near-peer teaching in robotics education: The role of STEM attitudes. Informatics in Education. https://doi.org/10.15388/infedu.2023.10

7 Devices over Bluetooth. (n.d.). Retrieved May 27, 2024, from https://docs.arduino.cc/tutorials/nano-33-ble-sense/ble-device-to-device/

8 Export CSV. (n.d.). Google for Developers. Retrieved May 27, 2024, from https://developers.google.com/drive/api/guides/manage-downloads

9 Hogeschool Utrecht. (n.d.). Master of Engineering - deeltijd | Hogeschool Utrecht. Retrieved May 27, 2024, from https://www.hu.nl/deeltijd-opleidingen/master-of-engineering

10 Leaphy Digital Twin. (2024). Retrieved May 27, 2024, from https://digitaltwin.leaphyeasybloqs.com/

11 Leaphy Easybloqs. (n.d.). Retrieved May 27, 2024, from https://leaphyeasybloqs.com/

12 Leopard. (n.d.). Retrieved May 27, 2024, from https://leopardjs.com/

13 Liljaniemi, A., & Paavilainen, H. (2020). Using Digital Twin Technology in Engineering Education – course concept to explore benefits and barriers. Open Engineering, 10(1), 377–385. https://doi.org/10.1515/eng-2020-0040

14 MOSCOW Prioritization. (2022, December 21). https://www.productplan.com/glossary/moscow-prioritization/

15 NVON | De lieve bruggenbouwer. Een zelfbouwrobotje met ongekende mogelijkheden. (n.d.). https://www.nvon.nl/nvox/de-lieve-bruggenbouwer-een-zelfbouwrobotje-met-ongekende-mogelijkheden#:~:text=Een%20zelfbouwrobotje%20met%20ongekende%20mogelijkheden,-Coen%20van%20der&text=Soms%20kan%20een%20nlt%2Dopdracht,nog%20een%20beetje%20aardig%20uitziet.

16 NVON | Robotica voor iedere leerling. (n.d.). https://www.nvon.nl/nvox/robotica-voor-iedere-leerling

17 Painless Mesh. (n.d.). Retrieved May 27, 2024, from https://www.arduino.cc/reference/en/libraries/painless-mesh/

18 Santos, S., & Santos, S. (2020, November 23). ESP-MESH with ESP32 and ESP8266: Getting Started | Random Nerd Tutorials. Random Nerd Tutorials. https://randomnerdtutorials.com/esp-mesh-esp32-esp8266-painlessmesh/

19 Scratch. (n.d.). Retrieved May 27, 2024, from https://scratch.mit.edu/

20 Serial Prot Websocket. (2020, August 12). Chrome for Developers. https://developer.chrome.com/docs/capabilities/serial?hl=nl

21 SQLite. (2024, May 23). https://www.sqlite.org/

22 Stichting Leaphy. (n.d.). Stichting Leaphy. Retrieved May 27, 2024, from https://www.leaphy.nl/

23 Svelte. (n.d.). Retrieved May 27, 2024, from https://svelte.dev/

24 Tao, F., Xiao, B., Qi, Q., Cheng, J., & Ji, P. (2022). Digital twin modelling. Journal of Manufacturing Systems, 64, 372–389. https://doi.org/10.1016/j.jmsy.2022.06.015

25 Van Wassenaer, N., Tolboom, J., & Van Beekum, O. (2023). The Effect of Robotics Education on Gender Differences in STEM Attitudes among Dutch 7th and 8th Grade Students. Education Sciences, 13(2), 139. https://doi.org/10.3390/educsci13020139

26 Veenman, K., Tolboom, J. L. J., & Van Beekum, O. (2022). The relation between computational thinking and logical thinking in the context of robotics education. Frontiers in Education, 7. https://doi.org/10.3389/feduc.2022.956901

27 Wagg, D. J., Worden, K., Barthorpe, R. J., & Gardner, P. (2020). Digital Twins: State-of-the-Art and future Directions for modeling and simulation in Engineering Dynamics Applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems. Part B. Mechanical Engineering*, *6*(3). https://doi.org/10.1115/1.4046739

Figures

Tables

Abbreviation

Abbreviation	Description
DT	Digital Twin
MNLE	Master Next Level Engineering
HU	University of Applied Sciences Utrecht
IDE	Institute of Design and Engineering
JSON	JavaScript Object Notation
IR	Infrared
DC	Direct Current
ESP32	A Wi-Fi and Bluetooth microcontroller
CSV	Comma-Separated Values
ML	Machine learning
RL	Reinforcement learning

Appendix I Interviews

Arjen Boesveld, Mechanical engineer

Digital twins gather data about a physical object, similar to a medical file or the information Google collects about an individual. This idea covers various tools like dashboards, 2D models, and replicas.

In engineering education, digital twins have practical applications. For example, mechanical engineers can simulate factors like torque and friction to analyse the behaviour of motor-driven wheels. Electrical engineering can use digital twins to program Arduino-based controls before implementing them physically.

Digital twins offer interdisciplinary educational value by bridging theory and practice. They provide an additional perspective to traditional learning methods, allowing for simulation and optimisation in design processes and post-design phases. This enhances comprehension and efficiency in learning complex concepts.

Plans are underway to integrate the Leaphy robot into electrical engineering curricula and potentially other engineering disciplines. This integration will use digital technology.

Twins for various educational goals, from simulating control systems to optimising robot behaviour.

However, some disciplines, particularly less technical ones like built mechanical disciplines, are not fully utilising digital twins' potential. In these areas, applications could extend to maintenance data analysis using Building Information Modeling (BIM) models.

Bart Bozon, Electrical / Master next-level engineering

Digital twins are models or software representations that mimic real-world processes, not necessarily needing to be exact copies. The importance of interaction within these models was discussed, stressing the inclusion of features allowing for the simulation of real-world interactions without replicating them precisely.

Reinforcement Learning (RL) and Machine Learning (ML) techniques were considered for efficiently simulating and training models. The conversation addressed the creation of digital or hybrid environments to facilitate effective reinforcement learning, enabling models to learn and adapt more quickly than in physical reality. The objective is to refine these models for real-world use by training them in a simulated environment.

Edge AI was explored, highlighting the practice of training AI models on PCs due to resource limitations and then deploying these trained models onto devices like robots for task execution. This approach enables real-time processing and decision-making on edge devices without continuous communication with a central server.

The integration and application of these technologies in educational settings, particularly in engineering courses, were discussed. This included developing user interfaces for programming and testing models, simulating environments to teach complex concepts, and the potential for these technologies to transform the teaching of subjects like complexity science and data science.

Considerations regarding hardware and software were also addressed, such as selecting appropriate hardware (like Sense or Raspberry Pi for quicker interaction) and designing software interfaces to facilitate the integration of digital twins, reinforcement learning models, and Edge AI into educational curriculums and projects.

Koen Smit, Information Technology

The Digital Twins Lab, established in 2020 in collaboration with the province of Utrecht, aims to connect students and academic programs with real-world issues, particularly in area development, by utilising digital twins for various applications.

Examples of digital twin applications include creating replicas of buildings, streets, neighbourhoods, or entire regions to aid planning and development. Specific projects involve scanning the Utrecht train station to monitor elevator and escalator statuses in real time using a digital dashboard.

In digital twin communication, changes in the physical world are mirrored in the digital version and vice versa. For instance, sensor-equipped city trash cans optimise collection routes, while electronically lockable bins deter vandalism.

Digital twins offer significant educational value, particularly in ICT and technical disciplines, by providing simulation, testing, and learning opportunities. They allow students to experiment with hardware and software integrations safely and analyse sensor data meaningfully.

Challenges include the varied definition of digital twins and the importance of aligning educational goals with their use. Additionally, there is potential overlap between disciplines like electrical engineering and ICT in their approach to digital twins and embedded systems.

Future directions involve exploring how tools like the Leaphy robot can enhance learning experiences by providing a hands-on understanding of complex systems, optimising designs, and integrating sensors and modules.

Steven Haveman, Master next-level engineer

Digital twins are digital representations that evolve alongside a system or product's design and development process. They encompass various forms of representation, such as dashboards or 2D or 3D models, and are integrated into the overall process rather than being standalone entities.

Systems engineering emphasises a model-based approach, favouring digital models over traditional document-based methods. Digital twins are a "knowledge cloud," providing relevant information to support design and decision-making.

The "digital thread" concept involves the digital twin evolving continuously throughout the design process, from initial conception to the realisation of the physical product. This ongoing digital presence enhances design and helps anticipate potential issues.

Digital twins play a crucial role in the design and optimisation process, enabling simulation and testing before physical implementation. They also support incremental development and ongoing optimisation even after deploying the product.

Digital twins can be utilised post-deployment for predictive maintenance and continuous efficiency improvement, leveraging real-time data to adjust and enhance performance.

Expectations of digital twins can vary across engineering disciplines, demonstrating their adaptability to different requirements and contexts.

They facilitate continuous improvement by providing insights into how a system or product behaves in reality compared to its simulated model, enabling real-time adjustments and enhancements.

Real-world examples of digital twin applications include predictive maintenance, optimising autonomous vehicle routes, and managing factory layouts to avoid disruptions in automated vehicle paths.

Emma Koster, Mechanical engineer

At Alliander, digital twins are applied in infrastructure and energy systems to understand how different infrastructures impact each other and simulate scenarios to prevent cascading failures.

In higher education, digital twins are valuable teaching tools. They allow students to experiment and learn without real-world risks. Through safe experimentation in a digital environment, they help students understand complex systems, validate theories, and enhance comprehension.

For instance, in the energy lab, a digital twin could simulate the behaviour of systems like heat pumps, ventilation, and radiators, offering bidirectional data flow for real-time adjustments and simulations.

Digital twins also serve as learning and evaluation tools, enabling students to replicate real-world measurements and better understand systems. They bridge theory and practice in education.

In mechanical engineering, digital twins have potential uses, such as creating virtual prototypes and evaluating systems like robot vehicles used in student projects.

Looking ahead, digital twins have the potential to impact education significantly. However, despite their efficiency and safety benefits, there is concern about losing the hands-on aspect critical to applied sciences and engineering disciplines.

Hubert Schuit, Electrical engineer

Hubert shares his experiences with digital twins, stemming from his involvement in industrial automation projects. He discusses the evolution of simulation in electronic design, highlighting tools like LT Spice and the importance of accurately modelling static and dynamic behaviours for predicting and validating design functionality.

The discussion focused on defining digital twins, with Hubert noting the historical use of simulations in electronics and exploring whether early simulations could be considered digital twins. They also discussed the dual role of digital twins in marketing and operational contexts.

Hubert describes the practical applications of digital twins in visualising and simulating infrastructure systems like bridges and viaducts, emphasising their marketing potential in attracting customers. He also discusses the transition from analogue to digital systems, where digital twins can surpass original analogue designs.

Regarding educational implications, the speakers consider using digital twins in electrical engineering education as simulators for reality. They highlight the importance of digital twins in enabling design and testing before physical realisation and the shift towards integrating them into embedded control systems.

The conversation explores integrating machine learning with digital twins, speculating about developing "learning" digital twins to optimise performance based on explored possibilities. This leads to a discussion on the balance Between using digital twins as simulators and their role in actual control systems.

Hubert suggests that a practical digital twin in electrical engineering should accurately simulate and predict real-world behaviours, aiding in design and validation processes. The discussion also touches on the origins of digital twins, tracing back to data management and validation techniques developed during the Apollo moon missions.

Joop Zuur, Build and environment

Joop shares an anecdote about the origin of the term "digital twin," which traces back to NASA's moon lander project, which was used to predict feedback delays in communication between Earth and the moon. This story sparks a discussion on the definition of digital twins. Joop suggests it as a digital representation of a physical object or system applicable across different scales, from city planning to building management.

The conversation explores applications of digital twins in the built environment, distinguishing between urban-scale models and those focused on individual buildings or infrastructure components. Joop explains how digital twins incorporate sensor data for real-time tracking and decision-making, highlighting the importance of historical data for predictive maintenance and operational adjustments.

Transitioning to the technical aspects, they discuss using software like Autodesk and programming languages like Python to create digital twins. They consider starting with 2D representations before progressing to more complex 3D modelling, weighing the trade-offs between development effort and the added value of spatial dimensions.

Future directions include brainstorming potential applications of digital twins in mobility, infrastructure management and built environment education. They contemplate using "live" robots to simulate traffic patterns, building operations, and environmental sensing, recognising the educational value of such projects.

Joop suggests various tools and platforms to support the digital twin project, such as Autodesk Forge for integrating sensor data into 3D models and A-Frame for web-based 3D visualisations. The conversation concludes with the first speaker expressing interest in further exploring these technologies and considering how to integrate digital twin functionalities into the "live" platform.

Johan Looijenga, Industrial / Master next-level engineer

Johan describes a digital twin as a broad concept representing a digital model of a system. This model aims to simulate the system's behaviour under various conditions. This simulation helps anticipate the impact of changes in the system, providing a real-time, accurate representation of its current state.

While Johan has experience with systems that monitor object statuses, he distinguishes them from true digital twins, which involve more complex interactions and simulations.

In education, Johan sees potential in integrating basic programming and sensor data into the technical business administration curriculum. This integration can enhance students' understanding of technical processes and manufacturing, bridging the gap between theoretical knowledge and practical experience. Digital twins are precious in embedding sensors and generating dashboard visualisations from real-world data.

In next-level engineering applications, digital twins could recognise patterns in sensor data, supporting data science objectives such as machine learning for object recognition and system optimisation.

Appendix II: Digital Twin Hackathon Script

Monday: Introduction and Line Following

Daily Assignment:

Introduction to the Leaphy Line Follower robot.

Task: Program the robot to follow a line using IR sensors. Adjust the settings to optimise line-following performance.

Requirements: Ensure the robot can navigate a simple track and avoid obstacles.

Guidance: Use the provided Easybloqs coding environment and adjust the robot's parameters.

Morning: General Introduction

1. Welcome and overview of the week's activities.
2. Introduction to digital twin concepts and the Leaphy Line Follower.

Afternoon: Line Following and Movement Control

3. Line Following: Lines are followed by infrared (IR) light to follow lines on surfaces by detecting their reflectivity.
4. Movement Control: Can drive forward, backward, left, or right.
5. Obstacle Detection: Measures distances to avoid obstacles in the system's path.
6. Status Indication: Displays colours to indicate system status.
7. Surface Colour Detection: Identifies surface colours for advanced navigation or tasks.

Tuesday: Creating an Interacting System

Daily Assignment:

Task: Program the robot to pick up and place objects at designated spots.

Requirements: The robot should navigate between locations and correctly identify objects using the colour sensor.

Guidance: Implement a central control system to manage the robot's tasks and ensure precise movements.

Morning: Object Manipulation

1. Pick up Objects: Grab and lift various objects or imaginary boxes from one spot using a gripper.
2. Place Objects: Positions and releases objects at specific destinations.

Afternoon: Navigation and Control

3. Navigate Between Locations: Moves between different spots within its working area using sensors and programmed instructions.
4. Detect and Identify Objects: Recognizes items using the colour sensor.
5. Control Operation Sequences: These are managed through a central control system that plans and organises the robot's tasks and movements.

Wednesday: Data Management and Maintenance Detection

Daily Assignment:

Task: Modify Leaphy robots to detect maintenance needs like smaller wheels or dirty sensors.

Requirements: Implement sensors to identify potential maintenance issues and analyse collected data.

Guidance: Use real-time monitoring to adjust and ensure the robot operates efficiently.

Morning: Data Collection and Storage

1. Data Collection: The robot's sensors and activities collect information.
2. Data Storage: Stores collected data for later analysis on the robot or in the cloud.

Afternoon: Data Analysis and Visualization

3. Data Analysis: Analyses collected data to find insights, predict maintenance needs, and improve operations.
4. Data Visualization: Shows data in easy-to-understand formats like dashboards or reports for decision-making.
5. Real-Time Monitoring: Allows operators to see what the robot is doing right now, making it easier to make quick adjustments.
6. (Real-Time) Control: Adjusts system settings in real-time.

Thursday: Change Management and Human Factors

Daily Assignment:

Task: Reflect on the technical work and consider its real-world implications. Solve a hypothetical disruption scenario.

Requirements: Understand the lifecycle of changes and consider organisational resistance and dynamics.

Guidance: Engage with a guest speaker to discuss real-world applications and challenges.

Morning: Non-Technical Change Activities

1. Focus on understanding the human impact of automation.
2. Discuss the consequences of automation, such as a truck driver missing out on conversations with a forklift driver.
3. Reflection on the technical work done earlier in the week.

Afternoon: Solving Human Disruptions

4. System Upgrading: Makes the systems better without causing problems.
5. Workforce Training: Teaches employees skills that match our goals.
6. Community Engagement: Works with others to build good relationships.
7. Disruption Management: Plans for and deals with unexpected events.

Friday: Impact Evaluation and Professional Skills

Daily Assignment:

Task: Reflect on the week's activities and prepare a presentation to showcase your work.

Requirements: Analyse the system's performance and evaluate the impact of changes made throughout the week.

Guidance: Use presentation skills to communicate your findings to the class effectively.

Morning: Professional Skills Development

1. Focus on reflection or presentation skills.
2. Discuss the importance of professional skills in technical fields.

Afternoon: Impact Evaluation

3. System Analysis: Reviews current system structure and operations to understand capabilities and integration points.
4. Change Evaluation: Measures the impact of changes on system performance and efficiency.
5. Outcome Presentation: Showcases results to stakeholders through reports, presentations, or demos.

Appendix III: Unit test Leaphy Original V2

Test Case ID	Description	Steps	Expected Result	Fulfilled
1.1	Verify Mesh Network Initialization	1. Initialize the mesh network. 2. Check the serial output for successful initialisation messages.	Mesh network initialises without errors and prints startup messages.	√
1.2	New Connection Callback	1. Add a new node to the network. 2. Check the serial output for new connection messages.	A new connection message appears in the serial output, indicating that a new node has joined.	√
1.3	Changed Connection Callback	1. Simulate a topology change. 2. Check the serial output for changed connection messages.	A changed connection message appears in the serial output indicating topology change.	√
1.4	Node Time Adjusted Callback	1. Adjust the node time. 2. Check the serial output for time adjustment messages.	A time-adjusted message appears in the serial output.	√
2.1	Send Broadcast Message	1. Call the sendMessage function. 2. Check the serial output and other nodes for received messages.	The message is broadcast to all nodes and appears in their serial outputs.	√
2.2	Receive Message Callback	1. Send a message from another node. 2. Check the receiving node's serial output.	The receiving node prints the received message in the serial output.	√
3.1	Serial Input Handling	1. Send a command via serial input. 2. Check serial output and pin state changes.	The command is correctly received and processed, updating the pin state as indicated in the serial output.	√
3.2	Stop Command Handling	1. Send the "stop_leaphy" command via serial input. 2. Check the pin state and serial output.	Pin 14 is set to LOW, and the stop command is broadcast over the mesh network.	√
3.3	Start Command Handling	1. Send the "start_leaphy" command via serial input. 2. Check the pin state and serial output.	Pin 14 is set HIGH, and the start command is broadcast over the mesh network.	√
4.1	Randomised Task Interval	1. Check the taskSendMessage interval. 2. Verify that the interval is correctly set and messages are sent at the specified interval.	The task interval is correctly set to 0.1 seconds, and messages are broadcasted at this interval.	√

Appendix IV: Unit test Master module

Test Case ID	Description	Steps	Expected Result	Fulfilled
1.1	Verify Mesh Network Initialization	1. Initialize the mesh network. 2. Check the serial output for successful initialisation messages.	Mesh network initialises without errors and prints startup messages.	√
1.2	New Connection Callback	1. Add a new node to the network. 2. Check the serial output for new connection messages.	A new connection message appears in the serial output, indicating that a new node has joined.	√
1.3	Changed Connection Callback	1. Simulate a topology change. 2. Check the serial output for changed connection messages.	A changed connection message appears in the serial output indicating topology change.	√
1.4	Node Time Adjusted Callback	1. Adjust the node time. 2. Check the serial output for time adjustment messages.	A time-adjusted message appears in the serial output.	√
2.1	Send Broadcast Message	1. Call the sendMessage function. 2. Check the serial output and other nodes for received messages.	The message is broadcast to all nodes and appears in their serial outputs.	√
2.2	Receive Message Callback	1. Send a message from another node. 2. Check the receiving node's serial output.	The receiving node prints the received message in the serial output.	√
3.1	Serial Input Handling	1. Send a command via serial input. 2. Check serial output and pin state changes.	The command is correctly received and processed, updating the pin state as the serial output indicates.	√
3.2	Stop Command Handling	1. Send the "stop_leaphy" command via serial input. 2. Check the pin state and serial output.	Pin 14 is set to LOW, and the stop command is broadcast over the mesh network.	√
3.3	Start Command Handling	1. Send the "start_leaphy" command via serial input. 2. Check the pin state and serial output.	Pin 14 is set HIGH, and the start command is broadcast over the mesh network.	√
4.1	Randomised Task Interval	1. Check the taskSendMessage interval. 2. Verify that the interval is correctly set and messages are sent at the specified interval.	The task interval is correctly set to 0.1 seconds, and messages are broadcasted at this interval.	√

Appendix V: Unit test Digital Twin Website

Test Case ID	Description	Steps	Expected Result	Fulfilled
1.1	Verify Page Load	1. Open the website in a web browser.	The page loads without errors and displays the title "Leaphy Digital Twin".	√
1.2	Check Top Container Layout	1. Verify that the top container is displayed with three sections: controls, CSV display, and node ID container.	The layout is as specified, with each section properly aligned and styled.	√
2.1	Start Position Button	1. Click the "Start position" button.	The project initiates to the start position.	√
2.2	Connect Serial Button	1. Click the "Connect Serial" button.	The serial connection is attempted.	√
2.3	Simulate Button	1. Click the "Simulate" button.	Simulation starts.	√
2.4	Download CSV Button	1. Click the "Download CSV" button.	The CSV file is downloaded.	√
3.1	Set Leaphy ID	1. Enter a valid ID in the input field. 2. Click the corresponding "Set Leaphy ID" button.	The ID is set and reflected in the interface.	√
4.1	Send a Message to Leaphy	1. Enter a message in the input field. 2. Click the "Send Message" button.	The message is sent to Leaphy.	√
5.1	Adjust Speed	1. Adjust the speed using the range input. 2. Click the "Set Speed" button.	The speed value is updated and displayed correctly.	√
5.2	Adjust Turn Degree	1. Adjust the turn degree using the range input. 2. Click the "Set Turn Degree" button.	The turn degree value is updated and displayed correctly.	√
5.3	Show Leaphys	1. Enter a number in the input field for Leaphys. 2. Click the "Show Leaphys" button.	The specified number of Leaphys is displayed.	√
5.4	Change Template	1. Enter a template number in the input field. 2. Click the "Change template" button.	The template changes according to the input.	√
5.5	Change Framerate	1. Enter a framerate value in the input field. 2. Click the "Change Framerate" button.	The framerate is updated accordingly.	√
6.1	Display CSV Data	1. Check the CSV display area for data.	The CSV data is displayed correctly in the designated area.	√
7.1	Performance Test	1. Monitor the website's performance under normal use conditions.	The website performs smoothly without lag or crashes.	√
7.2	Responsiveness Test	1. Resize the browser window to various dimensions (desktop, tablet, mobile).	The website layout adjusts correctly for different screen sizes, maintaining usability and readability.	√

Appendix VI: Learning Outcomes DT for MNLE

	The rationale for the row	A - Systems	B – Data
Overall Learning Outcome	*Overall guiding statement for block*	The student is leading in applying systems engineering, modelling and prototyping within an interdisciplinary system development process.	The student is leading in applying data science within an interdisciplinary system development process.
1. Managing and performing a system development process	*1.1 Applying Key Approach*	The student executes phases of the V-model, including system design, detailed design, implementation, and validation, iteratively. They work on interdisciplinary systems such as warehouse automation management, which involves integrating physical systems with digital twins to simulate and optimise performance in real-world scenarios.	The student collects data from sensors from the Leaphy robot. They apply data science techniques, including data preprocessing and analysis, to evaluate system performance. This interdisciplinary approach ensures data-driven decision-making and problem-solving.
	1.2 Realizing Results	The student employs proof of concept and rapid prototyping within a systems engineering framework. They gather requirements, develop design concepts, and create detailed designs. Using digital twins, they simulate system behaviour to validate design decisions before physical implementation, ensuring robust and reliable system development.	The student effectively structures data by applying the Entity-Relationship (ER) and Relational Model. Organising data into logical tables and relationships improves system reliability, supports accurate data analysis, and improves overall system predictability and performance.
2. Critically and holistically assessing the content	*2.1 Holistic (Context)*	The student applies systems thinking and uses digital twins to holistically assess and optimise an interdisciplinary system development process. This involves real-time simulation, monitoring, and system analysis, leading to more informed decision-making throughout the development lifecycle.	The student uses digital twins for real-time inventory tracking, predictive analytics, and system optimisation in a warehouse management context. This holistic assessment improves efficiency and reduces operational costs by leveraging comprehensive data analysis and real-time system monitoring.
	2.2 Critical (Applicability)	The student critically evaluates the applicability of systems engineering, prototyping, and modelling using digital twins. They simulate real-world conditions to test prototypes, optimise workflows, and predict outcomes, ensuring that the system design effectively and efficiently addresses real-world challenges.	The student critically assesses data science techniques and modelling approaches within a warehouse management system. They use digital twins to analyse data patterns, optimise inventory management, and enhance operational efficiency through predictive analytics and real-time simulations, ensuring data-driven and effective system improvements.
3. Attaining scientific robustness	*3.1 Process*	The student collects data from system sources such as sensors, clean it to remove errors and duplicates, integrates it to resolve discrepancies, and transforms it for consistency. They ensure data validity, reliability, and integrity by cross-checking data and maintaining organised storage. This accurate data is essential for effective digital twin modelling and system development.	The student gathers data from sensors, databases, and APIs, cleaning data to remove errors and duplicates, integrating to resolve discrepancies, and transforming for consistency. They ensure data validity, reliability, and integrity through systematic evaluation and organised storage, which is critical for accurate data modelling and decision-making.
	3.2 Outcomes	The student designs and executes experiments by identifying system requirements, creating test plans, and systematically implementing them. They report findings with detailed analysis and follow up by addressing issues and refining the system. This process ensures thorough verification and validation within a systems engineering approach, focusing on meeting objectives and maintaining system integrity through digital twins.	The student uses statistical tools and methods to validate data modelling outcomes. They conduct hypothesis testing, regression analysis, and error checking to ensure the validity of the results. Visualising data and results provides clear, data-driven explanations to support decisions in an interdisciplinary system development process, enhancing the system's effectiveness and reliability.
4. Communicating effectively and efficiently	*4.2 Effective*	The student leads stakeholder interactions relevant to the system development process, ensuring communication and collaboration. They engage stakeholders to align project goals with user needs and expectations, facilitating the successful implementation and integration of digital twins with physical systems.	The student uses data visualisations for specific target groups, effectively presenting data insights in an interdisciplinary system development process. Clear visualisations understand and support informed decision-making for optimising system performance and addressing stakeholder requirements.
5. Self-reflecting, receiving and providing feedback	*5.2 Feedback*	The student provides and receives feedback on relevant learning experiences for the MNLE. This continuous feedback loop helps optimise the learning process in future blocks.	The student proactively acquires and provides feedback on relevant learning experiences for a master's level in Next-Level Engineering. This feedback process helps optimise future learning and development, ensuring the continuous improvement of data science applications in interdisciplinary system development processes.

Appendix VII Learning Outcomes MNLE

	The rationale for the row	A - Systems	B – Data	C - Change	D - Impact
Overall Learning Outcome	*Overall guiding statement for block*	The student is leading the application of systems engineering, modelling, and prototyping within an interdisciplinary system development process.	The student is leading the process of applying data science within an interdisciplinary system development process.	The student plays a leading role in an interdisciplinary system development process by navigating a broader context of socio-technical change.	The student makes a meaningful impact with an applied research project in support of an interdisciplinary system development process.
1. Managing and performing a system development process	*1.1 Applying the Key Approach*	The student can define and apply a systems engineering process approach iteratively while collaborating in an interdisciplinary system development process.	The student can define and apply data science and its modelling approaches, models, and tools while collaborating in an interdisciplinary system development process.	The student can apply complexity theory, its perspectives, models, and methods to a broader context of socio-technical change to leverage change meaningfully.	The student performs and evaluates an individual applied research project concerning an interdisciplinary system development process and its broader context of socio-technical change.
	1.2 Realizing results	The student can define and apply prototyping and modelling approaches to make informed design decisions within a systems engineering process approach.	Using a data modelling approach, the student can improve reliability, predictability, sustainability, and efficiency systems.	The student can apply systems engineering, prototyping, and data science to align an interdisciplinary system development process with the broader context of socio-technical change.	The student realises a meaningful impact on multiple stakeholders, including professionals and the research community.
2. Critically and holistically assessing the content	*2.1 Holistic (context)*	The student can apply systems thinking to assess a particular interdisciplinary system development process holistically.	The student can apply data science to assess a particular data modelling development process holistically.	The student can select, combine and use information from real-life cases to holistically assess socio-technical change in support of a particular interdisciplinary system development process.	The student can holistically assess an applied research project and its context, acknowledging incomplete information and future challenges.
	2.2 Critical (applicability)	The student can critically assess the applicability of prototyping, modelling and systems engineering in a particular interdisciplinary system development process.	The student can critically assess the applicability of data science and its modelling approaches in a particular interdisciplinary system development process.	The student can critically assess the applicability of complexity models in the broader context of socio-technical change to support a particular interdisciplinary system development process.	The student can critically evaluate the approaches' applicability and generalise their applicability to other contexts.
	2.3 Ethical	The student can demonstrate an understanding of ethical principles and considerations in systems engineering by identifying, critically analysing, and addressing ethical dilemmas related to sustainability, uncertainties, and risks.	The student can demonstrate an understanding of ethical principles and considerations in data science by identifying, critically analysing, and addressing ethical dilemmas related to privacy, bias, truthfulness, and fairness.	The student can demonstrate an understanding of ethical principles and considerations relevant to the broader context of socio-technical change by identifying, critically analysing, and addressing ethical dilemmas related to inclusivity and unintended consequences.	The student can demonstrate an understanding of ethical principles and considerations in engineering and applied science by identifying, critically analysing, and addressing ethical dilemmas related to the applied research project and its broader context of socio-technical change.

3. Attaining scientific robustness	*3.1 Process*	The student can methodically connect with a scientific community and its body of knowledge to support a particular interdisciplinary system development process.	The student can methodically process data from various sources and assess the validity, reliability, and integrity of the data used in a data modelling process.	The student can select and apply appropriate quantitative and qualitative research methods to analyse real-world interventions in the broader context of socio-technical change.	The student can report on methods and application strategies employed in the applied research project in a scientifically sound manner.
	3.2 Outcomes	The student can design, execute, report, and follow up on verification and validation experiments using a systems engineering approach.	The student can apply analyses and statistically interpret the outcomes of data modelling processes to ensure their validity and explainability, substantiating decisions in an interdisciplinary system development process.	The student can use scientific research and its outcomes in an interdisciplinary system development process within its broader context of socio-technical change.	The student can methodically demonstrate real-world impact backed up by evidence-based claims.
4. Communicating effectively and efficiently	*4.1 Appropriate*	The student can apply the appropriate terms and concepts associated with systems engineering in an interdisciplinary system development process.	The student can apply the appropriate terms and concepts associated with data science applications in an interdisciplinary system development process.	The student can apply the appropriate terms and concepts associated with complexity theory and systemic design for engineering in the broader context of socio-technical change.	The student can explain the applied research project, its methods, and its results to systems engineers, data scientists, and other relevant stakeholders and communities.
	4.2 Effective	The student can effectively lead interactions with stakeholders relevant to an interdisciplinary system development process.	The student can effectively apply data visualisations to a particular target group in an interdisciplinary system development process.	The student can effectively communicate scientifically, using statistical analysis of project results.	The student can cultivate the commitment of relevant stakeholders to make a lasting impact on the applied research project.
5. Self-reflecting, receiving and providing feedback	*5.1 Self-reflection*	Students can communicate and reflect on their motives, personality, strengths and weaknesses in interdisciplinary system development processes.	The student can communicate and reflect on their motives, personality, strengths, and weaknesses about a data modelling process and follow up on the implications appropriately.	The student can demonstrate how more self-awareness, combined with a higher context awareness, relates to effective collaboration and leadership in an interdisciplinary system development process.	The student can demonstrate how personal motives, strengths, and weaknesses were anticipated to make the most meaningful impact on the applied research project in the broader context of socio-technical change.
	5.2 Feedback	The student can provide and receive feedback on relevant learning experiences for a master's level in Next-Level Engineering, resulting in a practical outlook on further optimising the learning process in the next block(s).	The student can proactively acquire and provide feedback on relevant learning experiences for a master's level in Next-Level Engineering. This results in a practical outlook on further optimising the learning process in the next block(s).	The student can combine feedback from multiple perspectives on relevant learning experiences for a master's level in Next-Level Engineering, resulting in a practical outlook on further optimising the learning process in the next block(s).	The student can engage in a fruitful conversation, providing and receiving feedback on the curriculum's contribution to the development of the relevant competencies, resulting in a practical outlook on how to shape life-long learning as a professional.