AN INTERNATIONAL AWARD-WINNING INSTITUTION FOR SUSTAINABILITY

Assignment 1:

Comparative Analysis of Microcontrollers, Microprocessors, and Embedded Systems in Mechatronics

Embedded Systems Design

MCTE 4342

SECTION 1

SEM 2 SESSION 2023/2024

NAME: WAN MUHAMMAD RAFIQ BIN WAN MOHD RUSHDAN

NO MATRIC: 2011341

LECTURER: ASSOC. PROF. DR. ZULKIFLI BIN ZAINAL ABIDIN

Introduction

1. Embedded system

- Embedded systems are information processing systems that are embedded into an enclosing product.
- Needed for the design of Cyber-physical systems (CPS) and IoT systems.
- Characteristics of embedded systems; single functioned, tightly-constrained, reactive and real-time
- **Impacts** of the embedded system:
  1) Transportation and mobility
     - Avionics: autopilots, flight control systems, and emission detection systems
     - Automotive: Anti-Breaking system, anti-theft protection electronic stability program
  2) Maritime engineering: bookkeeping system
  3) Civil engineering: Structural health monitoring
  4) Power engineering: Stable power with less energy loss
  5) Agricultural engineering: regulation for traceability for disease

- **Challenges** in creating embedded system are security, confidentiality, safety, reliability, repairability and the availability of system.
- To overcome the challenges, embedded systems must use resources efficiently. Example of **resources**:
  1) Energy: less energy use with high efficiency
  2) Run time: optimize in execution times across all levels, from algorithms to hardware implementations.
  3) Code size: Larger memory density in in single chip
  4) Cost: High volume in mass market that implement the required functionality with minimum resources.

2. Microcontroller

- A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.
- Another terms for microcontroller: embedded controller or microcontroller unit (MCU)
- Function: control small features of a larger component without complex front-end operating systems (OS).
- Operation of Microcontroller:
    - Interpreting received data from I/O peripherals from central processor.
    - Temporary information stored in data memory.
    - Processors access the memory and uses instructions in its program memory
    - I/O peripherals used to communicate the appropriate action.

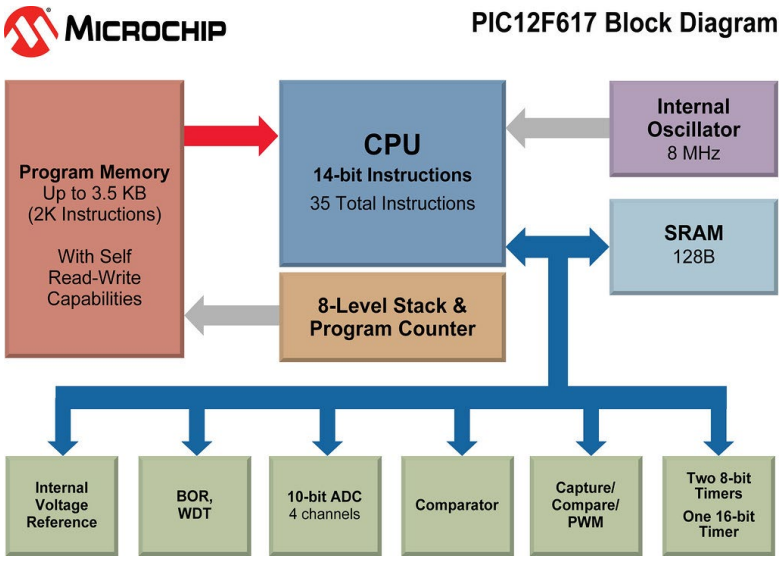


Figure 0-1: Simple representation of 8 bit Microcontroller from Microchip

- **Architecture/Elements** of a microcontroller
    - The processor (CPU): processes and responds to various instructions that direct the microcontroller's function. Including; basic arithmetic, logic and I/O operations, data transfer.
    - Memory: store the data that processor receive and uses to respond with the programmed instructors. Two types of memories are: Program memory and Data Memory

- I/O Peripherals: The input and output devices are the interface for the processor to the outside world.
- Analog to Digital Converter (ADC)
- Digital to Analog Converter (DAC)
- System bus: connective wire that links all the components of microcontroller.
- Serial port: allows the microcontroller to connect to external components.

- **Application**: home and enterprise, building automation, manufacturing, robotics, automotive, lighting, smart energy, industrial automation, communications and internet of things (IoT) deployments. Another specific microcontroller application are digital signal processor where microcontroller can use its ADC and DAC to convert the incoming noisy analog signal into an even outgoing digital signal.
  - Electromechanical systems: video games systems, office machines and security systems.
  - Sophisticated microcontrollers: aircraft, ocean-going vessel and artificial heart to control more complex algorithms and functions in critical situations.

Advantages	**Limitation**
Designed and developed to developed to obtain specific task= Multitasking	Used in small digital equipment mainly
Consume heat and electricity	Cannot access multitasking devices
Compact system	Only handle single operation and task at single time
Available in 4 bit until 128 bit	Did not possess any zero flags
Can be scaled up or down to meet the requirements of different applications, from simple embedded systems to high-performance computing clusters	Security vulnerabilities such as hardware exploits and side channel attack
Execute instructions at incredibly high speeds, enabling fast data processing and computation.	physical limits to the miniaturization
	Compatibility issues when upgrades or changes with existing software or hardware

3. Microprocessor

- Microprocessor is a programmable device that takes in input performs some arithmetic and logical operations over it and produces the desired output. Simple words can describe microprocessor as digital device on a chip that can fetch instruction from memory, decode and execute them and give results.
- Elements in executing instruction in machine language:
    - Arithmetic and Logic Unit: performs the arithmetic and logical operations.
    - Timing and Control Unit:
        - used to generate timing and control signals which are necessary for the execution of instructions.
        - control data flow between CPU and peripherals (including memory).
        - provide status, control and timing signals which are required for the operation of memory and I/O devices.
        - control the entire operations of the microprocessor and peripherals connected to it.
    - Registers: temporary storage and manipulation of data and instructions by the microprocessor. Example type of register are instruction register, general purpose register, temporary register and 16-bit stack pointer and counter.
    - Flag: flip-flop which indicates some conditions which arises after the execution of an arithmetic or logical instruction.
    - Data and Address Bus: data bus is 8-bit wide and therefore; 8 bits of data can be transmitted in parallel from or to the microprocessor while address bus serve dual purpose. They are used for the least significant 8 bits of the memory address or I/O address during the first cycle.

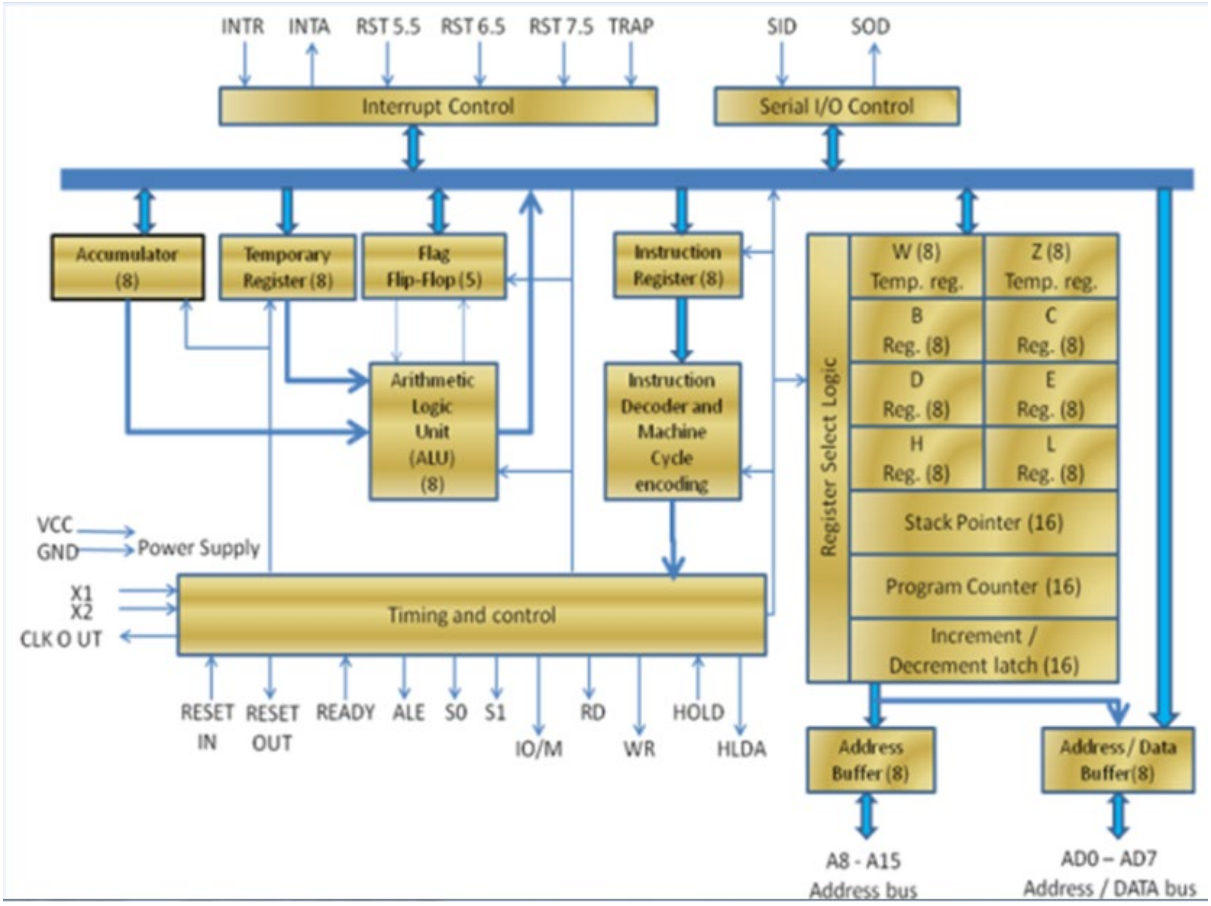


Figure 0-2: Intel 8085 microprocessor.

- Application:
    - Consumer Electronics: smartphones, digital cameras, and home appliances
    - Automotive: GPS, and ECU of the car
    - Medical: health monitor such as pacemakers, insulin pumps
    - Industrial Control Systems: SCADA ((Supervisory Control and Data Acquisition) and PLCs (programmable logic control).

Advantage	**Limitation**
Suitable for rapid data processing and computation	Requires specialised knowledge in programming, debugging and optimization
Can be programmed to perfom a wide range of task	Susceptible to security vulnerabilities

Compact integrated circuit, small form factor with substantial computing power	Physical limits to miniaturization
Advance technology create more efficient and low energy consumption microprocessor	Compatibility issues when upgrading
Can be scaled up down to different requirement or application	Create more heat in extensive use
Integrate various components and reducing external components	Doesn't support floating-point operations

4. Comparison between Microcontrollers and Microprocessors

Microcontrollers	Features	Microprocessors
Both process and control the specific task	**Function**	Process the general task only
No- in built memory	**Memory**	In built ROM and RAM memories
Programmeable digital and analog I/O pins	**Peripherals**	Need external peripherals with I?O pins
Low because the design time is short	**Cost**	High because the design time is long.
Low	**Processing power**	High
More	**Efficiency**	Less
Harvard (program and data stored in different memory)	**Architecture**	Von Nuemann (program and data store

The choice of the embedded system design depends on criteria

- Performance vs. Cost: The choice between a microcontroller and a microprocessor often involves a trade-off between performance and cost. For applications where real-time control and low power consumption are critical, a microcontroller may be preferred. Conversely, applications requiring high computational power and flexibility may benefit from a microprocessor.

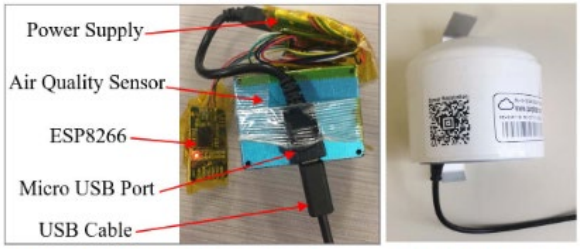
- System Complexity: The selection of a microcontroller or microprocessor influences the complexity of the embedded system design. Microcontrollers offer integrated peripherals and simplicity in design, whereas microprocessors require additional external components and may involve more complex hardware and software integration.

- Application Requirements: The specific requirements of the mechatronic system, such as processing speed, real-time control, power consumption, and cost constraints, play a crucial role in determining whether a microcontroller or microprocessor is more suitable for the application.

Case Study

Case Study 1

Title: *Microcontroller Based IoT System Firmware Security: Case Studies*.



(a) PurpleAir Sensor Without Case (b) PurpleAir Sensor

Fig. 1. PurpleAir Sensor

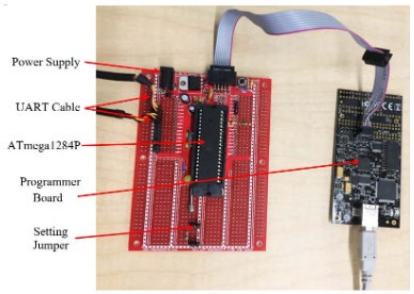


Fig. 8. A secure firmware upgrade system

Problem Statement:

- Security vulnerabilities in Microcontroller (MCU) based IoT firmware, particularly focusing on flaws in contemporary firmware upgrade models. The researchers aim to address the security concerns and exploits that have become significant challenges in the IoT industry, with a specific focus on firmware security.
- How lack of authentication and encryption processes in the firmware upgrade mechanism.

Objective Case Study:

- Analyze the firmware upgrade mechanism of the PurpleAir air quality sensor device to identify weaknesses such as the lack of authentication and encryption processes.
- Exploit the identified flaws in the firmware upgrade models, showcasing potential attacks through hardware and remote methods.
- Develop a prototype of a secure firmware upgrade system on an ATmega1284P chip to counter the discovered attacks and enhance firmware security in MCU-based IoT systems.
- Highlight the importance of implementing secure firmware upgrade mechanisms to protect IoT devices from unauthorized access, data breaches, and malicious activities.
- Provide guidelines and insights into potential pitfalls that manufacturers may encounter during the implementation of secure firmware upgrade systems in MCU-based IoT devices.

Methods:

1. **Selection of Target Device**: The researchers choose the PurpleAir air quality sensor device as the target for the case study, focusing on its firmware upgrade mechanism and security vulnerabilities.
2. **Firmware Analysis**: A detailed analysis of the architecture of the firmware in the PurpleAir device is conducted to understand how the firmware upgrade process works and to identify any weaknesses in the system.
3. **Identification of Security Flaws:** The researchers identify security flaws in the firmware upgrade mechanism, such as the absence of authentication and encryption processes, which can be exploited by attackers.
4. **Attack Scenarios**: Two types of attacks are considered: a hardware attack, where the attacker physically accesses the device to manipulate the firmware, and a remote attack, where the attacker exploits the firmware upgrade mechanism remotely.

5. **Prototype Development**: A secure firmware upgrade system is designed and implemented on an ATmega1284P chip to demonstrate a defense mechanism against the identified attacks and enhance firmware security in MCU-based IoT systems.
6. **Evaluation and Validation**: The implemented prototype is evaluated to assess its effectiveness in countering the attacks and improving the security of firmware upgrades in IoT devices.
7. **Documentation and Analysis**: The findings from the case studies, including the identified security vulnerabilities, attack scenarios, defense mechanisms, and potential pitfalls, are documented and analyzed to provide insights and recommendations for improving firmware security in MCU-based IoT systems.

Discussion on Analysis and Finding:

- The study shows that the microcontroller (ESP 8266) on PurpleAir sensor vulnerable to physical attack and remote attack. Below are list of consequences:
  i) Physical attack: Connect direct to the computer

1. Flashing a malicious firmware: cause misinform air quality sensor data.

2. Stealing Wi-Fi credentials: lack of authentication and encryption enable attackers to access the flash contents.

ii) Remote attack: via cloud server

1. Trivially fabricate malicious firmware
2. Obtain new firmware by sending a query request to the cloud server

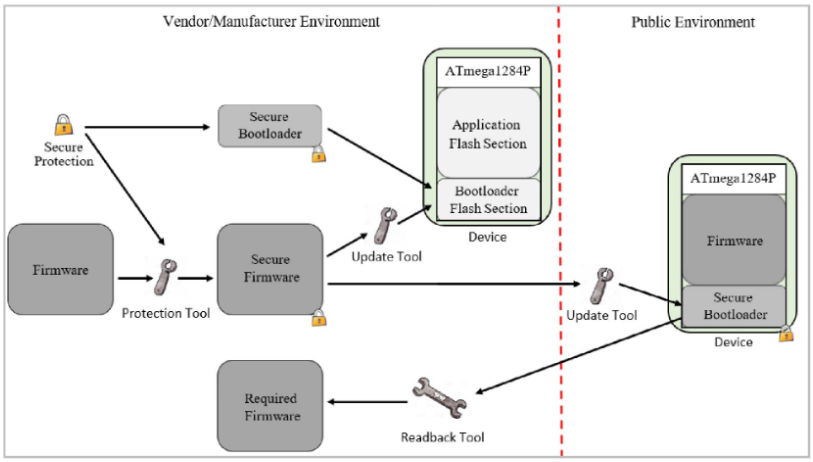


Fig. 7. An overview of the secure firmware upgrade System

- The upgraded system, which includes a secure firmware upgrade mechanism implemented on an ATmega1284P chip, offers several advantages, and may have some limitations. Here are some potential advantages and limitations of the upgraded system based on the information provided in the document:
    - **Advantages**:
        - **Enhanced Security:** The secure firmware upgrade system incorporates encryption, authentication, and integrity verification processes, which enhance the security of firmware upgrades in MCU-based IoT systems.
        - **Protection Against Attacks**: By following the requirements proposed by the MITRE Cyber Academy, the system can defend against hardware and remote attacks that exploit vulnerabilities in firmware upgrade mechanisms.
        - **Secure Bootloader:** The inclusion of a secure bootloader in the system helps prevent unauthorized firmware modifications and ensures the integrity of the upgrade process.
        - **Manufacturer Readback Tool:** The system includes a manufacturer readback tool, which can be used to verify the authenticity of firmware updates and prevent unauthorized modifications.
        - **Guidelines for Implementation:** The system provides guidelines for designing and implementing secure firmware upgrades, helping IoT vendors avoid common pitfalls and improve the overall security of their devices.

- **Limitations**:
  - **Hardware Compatibility:** The secure firmware upgrade system may be limited by the hardware capabilities of the ATmega1284P chip, which could restrict the implementation of certain security features.
  - **Complexity:** Implementing a secure firmware upgrade system with encryption and authentication processes may add complexity to the firmware upgrade process, potentially increasing development time and costs.
  - **Resource Constraints:** The system may face resource constraints on the MCU, such as limited memory or processing power, which could impact the performance of the firmware upgrade process.
  - **Maintenance and Updates:** Ensuring the ongoing security of the firmware upgrade system may require regular maintenance and updates to address new security threats and vulnerabilities.
  - **User Experience:** Introducing additional security measures in the firmware upgrade process could potentially impact the user experience, especially if the upgrade process becomes more cumbersome or time-consuming.

Case Study 2

Title: A Lightweight Security Checking Module to Protect Microprocessors against Hardware Trojan Horses.

Problem Statement:

- The problem addressed in the study is the threat posed by Hardware Trojan Horses (HTHs) to microprocessors.
- HTHs can allow malicious users to execute unauthorized software or gain unauthorized privileges, making them a serious security concern for both academic and industrial systems.

Objective of Case Study:

- To propose a security checking module that can protect microprocessor-based systems against Hardware Trojan Horses (HTHs).

- To develop a system-level solution that can detect and prevent HTHs from forcing the system to run malicious programs by altering the execution flow.
- To integrate this security checker between the microprocessor and the instruction memory, allowing it to monitor the fetching activity in real-time and verify that the correct instructions are being loaded from authorized memory locations.

Methods:

1) **Development of a security checking module**: The researchers designed and implemented a security checking module to be inserted between the microprocessor and the instruction memory. This module is responsible for monitoring the fetching activity to detect the activation of Hardware Trojan Horses (HTHs) that may alter the execution flow by launching malicious programs.
2) **Integration within a case study system:** The proposed security checking module was integrated into a case study system based on a RISC-V microprocessor implemented on an FPGA. The system ran a set of software benchmarks to evaluate the effectiveness of the security module in detecting HTH activations.
3) **Performance evaluation:** The researchers measured the performance overhead of the security checking module in terms of Look-Up Table (LUT) usage, Flip-Flop (FF) overhead, power consumption increase, and working frequency reduction. The goal was to assess the impact of the security module on the system's efficiency and functionality.
4) **Detection capability assessment:** The researchers conducted experiments to evaluate the detection capability of the security checking module, aiming to detect 100% of possible HTH activations with no false alarms. This assessment was crucial in determining the effectiveness of the proposed solution in safeguarding microprocessors against hardware-based attacks.

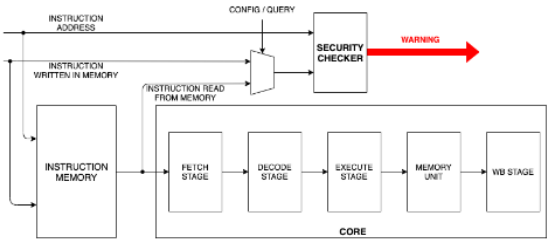


Figure 1: The proposed protection architecture

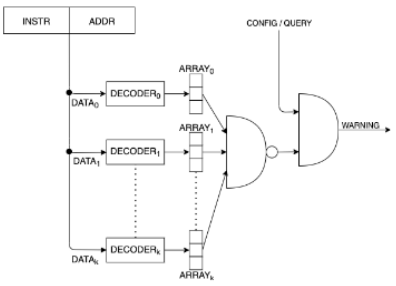


Figure 2: The structure of the proposed Security Checker

Discussion and Analysis

The study compared the proposed security checking module with a previous solution 2 and highlighted the following differences:

1. **Focus on different aspects:** The previous solution 2 focused on checking the **legality of executed instructions, control signals, clock cycles**, and **privilege modes** of the microprocessor. In contrast, the proposed solution aimed to detect the activation of Hardware Trojan Horses (HTHs) that alter the execution flow by launching malicious programs. The new module targeted HTHs that could change the system's functionality by forcing the CPU to execute unwanted software, providing a more comprehensive security approach.

2. **Detection capability:** The experimental results showed that the proposed security checking module outperformed the previous solution in terms of accuracy, detection capability, and false alarm rate. The new module demonstrated a **higher detection rate** of HTH activations without triggering any false alarms, indicating its superior performance in safeguarding microprocessors against hardware-based attacks.

3. **Overhead comparison**: The study reported a lower overhead for the proposed security checking module compared to the previous solution 2. The new module resulted in a LUT overhead of 0.5% and a FF overhead of 0.3%, with a **minimal increase in power consumption and no reduction in the working frequency**. This indicates that the proposed solution is more efficient in terms of resource utilization and system performance.

4. **Experimental validation:** The researchers conducted experiments using benchmark programs to evaluate the effectiveness of the proposed security checking module in real-world scenarios. The results demonstrated that the new module was able to detect 100% of possible HTH activations with no false alarms, showcasing its reliability and robustness in protecting microprocessor-based systems against hardware-based threats.

Overall, the comparison between the proposed security checking module and the previous solution 2 highlighted the advancements in detection capability, accuracy, and efficiency

achieved by the new module in enhancing the security of microprocessors against Hardware Trojan Horses.

Summary of the Case Study

The method for secure the vulnerability of the microcontroller and microprocessor are different and have different approach. In short, the microcontroller needed use external peripherals to protect the system while the microprocessor used the software to create a system that can detect potential threats to the system. Furthermore, the researcher for microprocessors covers the performance in terms of power consumption and reliability of the methods while the research for the microprocessors only validates the method in terms of chips that have been used in the research. Hence, it can be concluded that the microprocessors security is better than the microprocessors.

Conclusion

The main areas of distinction between microprocessors and microcontrollers are memory capacity, adaptability, real-time control capabilities, processing power, and peripheral integration. Microprocessors frequently depend on external peripherals; however microcontrollers combine peripherals like ADCs and communication interfaces on a single chip. Microcontrollers have limited processing capability; microprocessors can do complex jobs with greater processing power. Microprocessors might not provide deterministic guarantees, but microcontrollers with their deterministic behavior are perfect for real-time control applications. Furthermore, microprocessors frequently offer greater memory capabilities than microcontrollers, which usually have less memory. Microprocessors are more flexible than microcontrollers since they can run different software stacks, while microcontrollers are more specialized. Despite these distinctions, the programmability and digital logic components of microcontrollers and microprocessors allow them to be widely utilized in embedded systems for managing devices and operations.

The decision between microcontrollers and microprocessors has a big impact on mechatronic embedded systems development. Microcontrollers are suited for applications where deterministic behavior is essential, including accurate motion control or sensor feedback, because they are effective for real-time control tasks with low processing requirements. Their integrated peripherals lower costs and simplify hardware design, especially in areas with limited resources. Microprocessors, on the other hand, provide more flexibility, making it possible to build intricate control algorithms designed for particular mechatronics applications. To assure the development of effective and efficient embedded systems in mechatronics, engineers must weigh a variety of criteria when choosing between microcontrollers and microprocessors, including processing capability, real-time control requirements, power consumption, cost, and flexibility.

Reference:

Marwedel, P. (2021). Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things (p. 433). Springer Nature.

GfG. (2018, September 27). *Introduction of Microprocessor*. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/introduction-of-microprocessor/

Lutkevich, B. (2019). *microcontroller (MCU)*. IoT Agenda; TechTarget. https://www.techtarget.com/iotagenda/definition/microcontroller#:~:text=A%20microcontroller%20is%20a%20compact,peripherals%20on%20a%20single%20chip.

Gao, C., Luo, L., Zhang, Y., Pearson, B., & Fu, X. (2019). *Microcontroller Based IoT System Firmware Security: Case Studies*. https://doi.org/10.1109/icii.2019.00045

Microprocessor Architecture - javatpoint. (2021). Www.javatpoint.com. https://www.javatpoint.com/microprocessor-architecture

Dally E, C. (2023, March 31). *Application of Microprocessor*. Linkedin.com. https://www.linkedin.com/pulse/applications-microprocessor-christina-dally-e-1c/

Palumbo, A., Cassano, L., Reviriego, P., Bianchi, G., & Ottavi, M. (2021). A Lightweight Security Checking Module to Protect Microprocessors against Hardware Trojan Horses. *Virtual Community of Pathological Anatomy (University of Castilla La Mancha)*. https://doi.org/10.1109/dft52944.2021.9568291