IEEE

ASU STUDENT BRANCH

Embedded Systems Workshops

Module 4

Learning Automotive Communication Protocols and AUTOSAR Phase

Table of Content

Number	*Data*	*Title*	*Outlines*
Basics Communication Protocols			
Session 01	20 Nov 2023 (Monday)	UART	– Serial Communication Characteristics – UART Basics – Building UART Driver – Interfacing USB TTL Module
Session 02	27 Nov 2023 (Monday)	SPI	– SPI Basics – Building SPI Driver
Session 03	4 Dec 2023 (Monday)	I2C	– I2C Basics – Building I2C Driver – EEPROM Interfacing
Automotive Communications Protocols			
Session 04	11 Dec 2023 (Monday)	CAN	– Introduction to Controller Area Network (CAN) protocol. – History and evolution of CAN in automotive systems. – CAN architecture: nodes, messages, and bus. – Understanding CAN message formats.

			– CAN communication modes: simplex, half-duplex, full duplex. – Error handling and fault tolerance in CAN. – Real-world applications and use cases of CAN in automotive industry.
Session 05	18 Dec 2023 (Monday)	CAN FD	– Overview of CAN FD and its benefits over traditional CAN. – Comparison between CAN FD and classic CAN. – CAN FD frame format and increased data rates. – Implementation challenges and solutions. – Practical examples and case studies demonstrating CAN FD applications. – Hands-on exercises and demonstrations.
Session 06	29 Jan 2024 (Monday)	LIN Communication	– Introduction to Local Interconnect Network (LIN) protocol. – LIN network architecture: master-slave configuration. – LIN frame structure and message encoding. – LIN scheduling: slots and frames. – LIN communication modes and baud rates. – Application of LIN in automotive electronics. – LIN diagnostic features and tools.
Session 07	5 Feb 2024 (Monday)	CAN Lab	– Practical lab session focusing on CAN protocol. – Hands-on exercises involving CAN communication setup. – Message transmission and reception using CAN. – Error detection and handling in a controlled lab environment. – Troubleshooting common issues encountered in CAN communication. – Interfacing microcontrollers with CAN modules. – Real-time demonstrations and troubleshooting techniques.
Session 08	12 Feb 2024 (Monday)	Unified Diagnostic Services (UDS)	– Overview of Unified Diagnostic Services (UDS) protocol. – UDS architecture and communication model. – Diagnostic services supported by UDS. – Diagnostic Trouble Codes (DTCs) and their interpretation. – UDS security mechanisms and access control. – Implementing UDS in automotive diagnostics.

			– Case studies showcasing UDS applications.
Session 09	19 Feb 2024 (Monday)	Ethernet 01	– Introduction to Ethernet communication in automotive systems. – Ethernet network architecture and topology. – Ethernet frame format and data transmission. – Ethernet switches and routers in automotive networks. – Automotive Ethernet standards (e.g., Broad-Reach). – Real-time capabilities and determinism in Ethernet-based systems. – Automotive applications of Ethernet networks.
Session 10	26 Feb 2024 (Monday)	Ethernet 02	– Advanced topics in automotive Ethernet communication. – Time-Sensitive Networking (TSN) for Ethernet-based real-time communication. – Automotive Ethernet security: protocols and best practices. – Ethernet-based in-vehicle infotainment (IVI) systems. – Ethernet-based advanced driver-assistance systems (ADAS). – Future trends and developments in automotive Ethernet technology. – Hands-on demonstrations and practical exercises.
AUTOSAR Architecture			
Session 11	4 Mar 2024 (Monday)	– AUTOSAR Overview – AUTOSAR SW layers and Stacks – BSW Integration	
Session 12	1 Apr 2024 (Monday)	AUTOSAR MEM Stack	
Session 13	8 Apr 2024 (Monday)	– Diagnostics in Automotive domain – AUTOSAR DIAG modules	
Session 14	15 Apr 2024 (Monday)	– AUTOSAR COM Stack Configuration and integration – Communication Testing Process	

Session 15	22 Apr 2024 (Monday)	– AUTOSAR Network Management – AUTOSAR Mode Management – AUTOSAR Watchdog	

Prerequisites Knowledge:

- Strong grasp of programming languages, with a focus on C/C++ proficiency.
- In-depth understanding of digital electronics and microcontroller concepts, including but not limited to logic gates, circuits, and basic electronic components.

Hardware Requirements:

- Microcontroller development boards such as ATmega32, STM32, etc.
- Essential hardware components including USB cables and connectors, breadboards, jumper wires, LEDs, resistors, sensors, and other electronic elements.

Software Requirements:

- Proficiency in using Integrated Development Environments (IDEs) like Platform IO, Keil uVision or STM32CubeIDE, suitable for the specific microcontroller board.
- Knowledge of compilers and toolchains compatible with the microcontroller architecture, ensuring efficient code compilation.
- Familiarity with versatile code editors such as Visual Studio Code, Sublime Text, or Atom, enhancing code readability and debugging capabilities.
- Expertise in version control systems, such as Git, and utilizing platforms like GitHub or GitLab for seamless collaboration, version tracking, and project management.

Reference Materials:

- Ability to navigate and comprehend technical data sheets and documentation specific to the chosen microcontroller, enabling informed decision-making during the development process.
- Resourcefulness in utilizing online platforms, tutorials, and educational materials focused on microcontroller programming and driver development, fostering continuous learning and skill enhancement.