Programmable System-on-Chip architecture for ARM Cortex-M0 devices (PSOC):

Course Objectives	● Equip students with a comprehensive understanding of PSoC architecture and ARM Cortex-M0 core, enabling them to proficiently design and develop embedded systems applications. ● Provide hands-on experience in configuring digital and analog components. ● Implement serial and wireless communication protocols. ● Integrate advanced features such as capacitive sensing and real-time operating systems, and create custom components. ● Demonstrate the ability to apply these skills to develop complex embedded systems solutions for various real-world applications through practical projects and a capstone assignment.
Learning Outcomes	● Proficiency in the basics of PSoC architecture and its digital and analog components. ● Delve into the ARM Cortex-M0 core, exploring its features and instruction set. ● Set up the development environment using PSoC Creator IDE and delve into digital and analog design, learning to configure blocks and peripherals. ● Advance into serial and wireless communication, capacitive sensing, RTOS implementation, and custom component creations.

Course Duration: 45 Hours

Course Content:

Unit 1: Introduction to PSoC and ARM Cortex-M0

Overview of PSoC Architecture: Introduction to PSoC (Programmable System-on-Chip), Comparison with traditional microcontrollers and FPGAs, PSoC architecture: Digital and analog blocks, UDBs (Universal Digital Blocks), PLDs, and configurable interconnect, Introduction to ARM Cortex-M0: Overview of ARM Cortex-M0 core: Features, advantages, and applications, ARM Cortex-M0 instruction set and architecture basics. Development Environment Setup: Introduction to PSoC Creator IDE, Installation and setup of software tools, Overview of development workflow: Design, build, program, and Debug.

Unit 2: Basic Concepts and Peripherals

Digital Design with PSoC: Configuring digital blocks: Timers, counters, PWMs, interfacing digital peripherals: GPIO, LEDs, switches, Basic digital project: Blinking LED with PWM control, Analog Design with PSoC: Configuring analog blocks: ADCs, DACs, OpAmps, interfacing analog peripherals: Sensors, potentiometers, Basic analog project: Reading sensor data and displaying on a serial monitor.

Unit 3: ARM Cortex-M0 Cortex-M0 Programming

Programming Basics: Introduction to C, programming for ARM Cortex-M0, Writing and debugging simple programs, Using CMSIS (Cortex Microcontroller Software Interface Standard) libraries, Interrupts and Low Power Modes: Configuring and handling interrupts. Implementing low power modes for energy-efficient applications, Practical example: Button press interrupt and power-saving mode.

Unit 4: Communication Protocols

Serial Communication: UART: Configuration and communication, I2C: Configuration and communication with sensors, SPI: Configuration and communication with external devices, Practical project: Interfacing an external EEPROM using I2C, Wireless Communication: Introduction to wireless communication protocols: BLE, Wi-Fi, Configuring and using BLE in PSoC, Practical project: Creating a BLE-enabled sensor application.

Unit 5: Advanced Topics and Projects

Capacitive Sensing: Introduction to capacitive sensing, configuring capacitive touch sensors in PSoC, Practical project: Implementing a capacitive touch button, RTOS and Multitasking: Introduction to Real- Time Operating Systems (RTOS), Implementing FreeRTOS on ARM Cortex-M0, Practical project: Creating a multitasking application with FreeRTOS. Custom Components and APIs: Creating custom components in PSoC Creator, Writing and using custom APIs. Practical project: Designing a custom component for specific application, Capstone Project: Students propose, design, implement, and test a complex project of their choice, Example projects: Smart home automation system, portable medical monitoring device, or a mini robotics controller, Presentation and demonstration of the final project.

Test Projects:

Use Cases

1. **Digital Thermometer:**
   Project: Develop a digital thermometer using a temperature sensor.
   Details: Interface an analog temperature sensor with PSoC, convert the analog signal to digital using ADC, and display the temperature on an LCD or serial monitor.
2. **Capacitive Touch Interface**:
   Project: Implement a capacitive touch button or slider.
   Details: Configure the capacitive touch block in PSoC to create a touch-sensitive interface for user input.
3. **PWM Motor Speed Controller:**
   Project: Control the speed of a DC motor using PWM.
   Details: Use a potentiometer to adjust the PWM duty cycle and vary the motor speed.
4. **UART Communication:**
   Project: Create a UART communication link between PSoC and a PC.
   Details: Send sensor data from PSoC to a PC terminal via UART and display the data.
5. **Wireless Sensor Node:**
   Project: Develop a wireless sensor node using BLE.
   Details: Collect data from a sensor, transmit the data over BLE, and display it on a mobile app.
6. **Real-Time Clock (RTC):**
   Project: Implement a real-time clock using an external RTC module.
   Details: Display the current date and time on an LCD and keep time even when the device is powered off.
7. **Battery Management System:**
   Project: Design a simple battery management system.
   Details: Monitor battery voltage and current, display the battery status, and implement low battery warning.
8. **Gesture-Based Control:**
   Project: Develop a gesture-based control system using an accelerometer.
   Details: Use accelerometer data to recognize gestures and control an application (e.g., lighting control).

9. **Smart Home Controller:**
   Project: Create a smart home controller.
   Details: Interface with sensors and actuators to control home appliances based on sensor inputs and user commands.
10. **Environmental Monitoring System:**
   Project: Develop an environmental monitoring system.
   Details: Use sensors to measure temperature, humidity, and air quality, and display the data on an LCD or send it to a cloud server.
11. **Voice-Activated Device:**
   Project: Implement a simple voice-activated device.
   Details: Use a microphone and audio processing to recognize specific voice commands and control an application.
12. **Robotics Controller:**
   Project: Design a controller for a simple robotic arm.
   Details: Use PSoC to control the motors and read sensors to perform basic robotic tasks.
13. **Health Monitoring System:**
   Project: Develop a health monitoring system.
   Details: Interface with heart rate and temperature sensors to monitor vital signs and display the data.
14. **Data Logger:**
   Project: Create a data logger for environmental data.
   Details: Collect data from multiple sensors, store the data in memory, and periodically upload it to a PC or cloud server.
15. **Home Security System:**
   Project: Implement a home security system.
   Details: Use motion sensors, door/window sensors, and a camera to monitor and secure a home.
16. **Energy Meter:**
   Project: Design an energy meter to measure power consumption.
   Details: Measure voltage and current to calculate power and energy consumption, and display the results.
17. **Remote-Controlled Vehicle:**
   Project: Develop a remote-controlled vehicle.
   Details: Use a wireless communication module (e.g., BLE or RF) to control a small vehicle with PSoC as the controller.
18. **Smart Irrigation System:**
   Project: Create a smart irrigation system.
   Details: Use soil moisture sensors to control irrigation valves and optimize water usage for plants.
19. **Wearable Fitness Tracker:**
   Project: Develop a wearable fitness tracker.

Details: Use accelerometers and heart rate sensors to track physical activity and health metrics, and display the data on a wearable display or mobile app.

20. **Automated Plant Watering System:**

    Project: Implement an automated plant watering system.

    Details: Use soil moisture sensors and a water pump to automatically water plants when the soil is dry.