Coincent
Learning Drives Excellence

Coincent 3 Year Program Curriculum
Embedded System and IoT Domain

Partnered by

APSIS Solutions
INNOVATING THROUGHOUT

Empowering Learners,
Accelerating Careers.

coincent.ai

ABOUT COINCENT

Coincent offers a 3-Year Program that is a well-structured, career-focused initiative designed to equip students with practical skills, real-world experience, and strong placement support. The program is tailored to ensure progressive learning and career readiness across three year phases.

Why It's Unique

- Only one batch per year with limited seats (150 students) per Domain to maintain quality.
- Prepares students step-by-step to become job-ready by graduation.

DETAILED ABOUT COINCENT 3 YEAR PROGRAM OF EMBEDDED SYSTEM & IoT

"Embedded System and IOT Program at Coincent – Learn by Doing"

IoT (Internet of Things):

IoT refers to a network of connected devices (like sensors, appliances, or vehicles) that collect and exchange data over the internet. These devices can

monitor, control, and automate tasks in real time (e.g., smart homes, wearables, industrial sensors).

Embedded System:

An embedded system is a dedicated computer system designed to perform a specific task within a larger device. It includes both hardware and software, and examples include microwave controllers, car engine systems, or smartwatches.

- IoT connects smart devices to the internet.

- Embedded systems are the internal "brains" that control specific functions in those devices.

3-Year Program Structure Breakdown

TABLE OF CONTENTS

Year 1 :- Industrial Training

Module 1 – Fundamentals of IoT

Learning Objectives:

- Understand what IoT is and why it matters
- Identify real-world applications

Learn the structure of IoT architecture

Topics Covered:

- What is IoT? Definitions & Evolution
- Applications: Smart Homes, Healthcare, Agriculture, Industry 4.0
- IoT Architecture:
  - Perception layer (sensors, actuators)
  - Network layer (communication)
  - Application layer (interface, cloud)
- Embedded systems: basics, components
- Microcontroller vs Microprocessor

Tools: Slides, videos, real-world case studies

Key Benefits and Outcomes:

Key Benefits:

Clear Understanding of IoT Concepts – Learn what IoT is, how it evolved, and why it's transforming industries.

Exposure to Real-World Applications – Discover how IoT is used in smart homes, healthcare, agriculture, and more.

Grasp of IoT Architecture – Understand the core layers (Perception, Network, Application) and how they work together.

Embedded Systems Knowledge – Gain foundational knowledge of embedded devices, sensors, and controllers.

Outcomes:

Ability to explain IoT basics and structure.
Awareness of key IoT applications and industry relevance.
Understanding of the hardware-software interaction in embedded systems. Preparedness for deeper IoT and hardware integration modules.

Module 2: Arduino and TinkerCAD Simulation

Learning Objectives:

- Learn to simulate circuits and code using TinkerCAD
- Understand Arduino UNO board basics

Topics Covered:

- Arduino UNO Pinout, onboard components
- TinkerCAD setup and usage
- Connecting LED and resistor
- Blinking LED program (digitalWrite, delay)

Hands-On: Blinking LED simulation

Assignment: LED blink with varied delays and multiple LEDs

Key Benefits and Outcomes:

Key Benefits:

- Practical Circuit Simulation – Learn to build and test circuits virtually using TinkerCAD
- Basic Arduino Knowledge – Understand Arduino UNO components and pin configuration.
- Intro to Programming Hardware – Gain hands-on coding experience with digital outputs like LED blinking.
- Visual Learning – Simulate and debug real-world hardware setups without physical components.

Outcomes:

- Ability to use TinkerCAD for circuit design and simulation.
- Understand and work with Arduino board basics.
- Write and test simple Arduino programs (e.g., blinking LEDs).
- Ready to move on to more advanced IoT hardware programming.

Module 3: Digital I/O and Serial Communication

Learning Objectives:

Interface with basic I/O components

Use Serial Monitor to send and receive data

Topics Covered:

- Digital Inputs: Pushbuttons
- Digital Outputs: LEDs, Buzzer
- Serial communication with PC
- Serial.begin(), Serial.print(), Serial.read()

Project Idea: LED ON/OFF using serial commands

Key Benefits and Outcomes:

Key Benefits:

- Hands-on I/O Control – Learn how to interface pushbuttons, LEDs, and buzzers with Arduino.
- Serial Communication Skills – Understand how devices communicate with a computer using the Serial Monitor.

- Basic Hardware Programming – Practice reading inputs and controlling outputs using Arduino code.

Outcomes:

- Ability to connect and control basic input/output components.
- Use Serial.begin(), Serial.print(), and Serial.read() for data transfer.
- Build simple interactive hardware systems with Arduino.

Module 4: Analog Input and PWM

Learning Objectives:

- Work with analog sensors
- Learn PWM control for brightness and speed

Topics Covered:

- Analog inputs: Potentiometer, LDR
- PWM control: Fading LED
- analogRead(), analogWrite()
- LED forward voltage and safety

Mini Project: RGB LED color control using potentiometer

Key Benefits and Outcomes:

Key Benefits:

Sensor Integration – Learn to read data from analog sensors like potentiometers and LDRs.

PWM Control Skills – Understand how to control brightness or motor speed using Pulse Width Modulation.

Practical Coding Practice – Use analogRead() and analogWrite() to work with real-time sensor data and outputs.

Outcomes:

- Ability to interface and read analog input values.
- Control LED brightness or similar devices using PWM.
- Apply safe voltage practices for hardware components.

Module 5: Sensors and Data Acquisition

Learning Objectives:

- Interface environmental and motion sensors
- Capture real-time sensor data

Topics Covered:

- PIR sensor (Motion detection)
- DHT11 (Temp & humidity)
- Ultrasonic sensor (Distance measurement)
- IR sensor, Gas sensor, Soil moisture sensor

Hands-On Projects:

- Automatic light using PIR
- Water level indicator using Ultrasonic

Key Benefits and Outcomes:

Benefits:

- Gain hands-on experience with motion and environmental sensors used in real-world IoT systems.
- Learn to collect and process real-time data for automation and monitoring applications.

- Build practical projects to reinforce sensor integration and hardware programming skills.

Outcomes:

- Interface sensors like PIR, DHT11, ultrasonic, IR, gas, and soil moisture
- Capture temperature, humidity, motion, and distance data
- Develop automation projects (e.g., automatic lights, water level indicator)
- Practice sensor data handling in Arduino
- Strengthen hardware-software integration skills
-

Module 6: Actuators and Motor Control

Learning Objectives:

- Understand and control actuators
- Use motor driver circuits

Topics Covered:

- DC Motors – forward, reverse
- L293D motor driver
- Servo Motor – angle control
- Motor commands via serial or buttons

Mini Project: Line follower robot (Simulated)

Benefits:

- Interface sensors like PIR, DHT11, ultrasonic, IR, gas, and soil moisture
- Capture temperature, humidity, motion, and distance data
- Develop automation projects (e.g., automatic lights, water level indicator)
- Practice sensor data handling in Arduino
- Strengthen hardware-software integration skills

Outputs:

- Control DC motors in forward and reverse directions
- Use L293D motor driver for safe and efficient motor operation
- Interface and control servo motors with precise angle positioning
- Send motor commands using serial monitor or pushbuttons
- Gain practical skills in actuator control and automation systems

Module 7: Robotics Integration

Learning Objectives:

- Build basic robotic systems
- Combine sensors with actuators

Topics Covered:

- Robot design principles
- Remote-controlled robot using pushbuttons
- Obstacle avoidance logic using ultrasonic
- Speed control using PWM

Projects:

- Obstacle Avoidance Robot
- Smart Vacuum Cleaner Prototype

Benefits and Outcomes:

Benefits:
1. Learn to design and build basic robotic systems by integrating sensors and actuators.

2. Understand core robotics principles like movement, obstacle detection, and control logic.

3. Develop real-world skills in building interactive, sensor-driven robots using Arduino.

Outputs:

- Apply robot design principles for structure and function
- Build a remote-controlled robot using pushbuttons
- Implement obstacle avoidance using ultrasonic sensors
- Control motor speed with PWM for smoother movement
- Integrate sensors and actuators for intelligent robot behavior

Module 8: Introduction to IoT Platforms & Protocols

Learning Objectives:

- Understand IoT system component
- Learn about boards and protocols

Topics Covered:

- ESP8266 vs ESP32 vs Arduino UNO
- Wi-Fi, Bluetooth, LoRa
- HTTP vs MQTT protocol
- Cloud computing for IoT

Case Study: Smart irrigation system architecture

Benefits:

- Gain a solid understanding of IoT hardware and communication protocols essential for smart device connectivity.
- Learn to compare popular IoT boards and choose suitable technologies for various applications.
- Understand how data is transmitted and managed through cloud platforms using protocols like HTTP and MQTT

Outputs:

- Compare IoT boards: ESP8266, ESP32, and Arduino UNO
- Learn wireless communication protocols like Wi-Fi, Bluetooth, and LoRa
- Understand cloud integration for real-time data access and storage
- Explore data transmission using HTTP and MQTT protocols
- Build a foundation for developing scalable and connected IoT systems
-

Module 9: IoT Cloud – ThingSpeak

Learning Objectives:

- Send sensor data to the cloud
- Visualize real-time data online

Topics Covered:

- Creating ThingSpeak channel
- HTTP API keys
- Sending DHT11 data to cloud
- Reading from ThingSpeak using Arduino

Project: Remote temperature logging system

Key Benefits and Output:

Benefits:

1. Learn to connect IoT devices to the cloud and send real-time sensor data using HTTP APIs.
2. Gain hands-on experience with ThingSpeak to visualize and monitor data like temperature and humidity.
3. Understand how to read and display live data from the cloud using Arduino.

Outputs:

- Create and configure ThingSpeak channels
- Use API keys for secure data communication
- Send DHT11 sensor data to the cloud
- Visualize data in real time on dashboards

- Read cloud data back into Arduino for smart applications

Module 10: Home Automation Projects

Learning Objectives:

- Build a smart home interface
- Use sensors and relays to control appliances

Topics Covered:

- Relay module wiring and usage
- IR and Bluetooth control
- Cloud-triggered relay control
- Adafruit IO dashboards

Project: IoT-based smart home controller

Key Benefits and Output:

Benefits:
1. Learn to build a smart home system by integrating sensors, relays, and IoT dashboards.
2. Gain skills to control appliances using IR, Bluetooth, and cloud-based triggers.
3. Understand how to create user interfaces with Adafruit IO for real-time remote control.

Outputs

- Wire and control appliances using relay modules
- Implement IR and Bluetooth-based device control
- Use cloud services to trigger actions remotely
- Design interactive dashboards with Adafruit IO
- Build a functional smart home prototype using IoT concepts

Module 11: App Development with MIT App Inventor

Learning Objectives:

- Create mobile apps to control IoT devices
- Use Bluetooth and Wi-Fi communication

Topics Covered:

- Introduction to MIT App Inventor
- UI design for control panels
- App-to-Arduino communication (via Bluetooth/Wi-Fi)
- Text-to-Speech, Voice control

Projects:

- IoT Home Light App
- Speech Controlled Fan

Key Benefits and Output:

Benefits:

1. Learn to create custom mobile apps to control IoT devices using Bluetooth and Wi-Fi.

2. Gain hands-on experience with MIT App Inventor for building interactive, voice-enabled interfaces.

3. Understand how to connect smartphones to Arduino for real-time control and automation.

Outputs:

- Design user-friendly mobile control panels
- Enable Bluetooth/Wi-Fi communication with IoT hardware
- Implement features like voice commands and text-to-speech
- Build functional apps to monitor and control devices
- Strengthen app-hardware integration skills for smart systems

Module 12: MQTT Messaging Protocol

Learning Objectives:

- Learn lightweight IoT communication
- Use Mosquitto and Adafruit MQTT

Topics Covered:

- MQTT Broker, Publisher, Subscriber
- Installing Mosquitto locally
- Using Adafruit MQTT with Arduino
- Authentication with MQTT

Project: Sensor updates via MQTT on dashboard

Key Benefits and Output:

Benefits:

1. Gain practical knowledge of lightweight and efficient IoT communication using the MQTT protocol.
   2. Learn to set up local and cloud-based MQTT brokers for real-time device messaging.
2.

3. Understand secure data exchange between IoT devices using publishers, subscribers, and authentication.a

Outputs:

Set up and use Mosquitto MQTT broker locally
Implement publisher-subscriber communication models
Connect Arduino to Adafruit MQTT for cloud interaction
Enable secure communication with MQTT authentication
Build scalable, real-time IoT systems with efficient messaging

Module 13: Python for IoT

Learning Objectives:

- Learn Python basics and IoT scripting
- Control devices and log data with Python

Topics Covered:

- Python basics: variables, loops, conditions
- PySerial for Arduino communication
- Using MQTT and HTTP in Python
- DHT/Ultrasonic logging with Python

Project: Python dashboard for sensor data

Key Benefits and Output:

Benefits:

1. Build a strong foundation in Python programming tailored for IoT applications.
2. Learn to control devices, read sensor data, and log it using Python scripts.
3. Gain hands-on experience with PySerial, MQTT, and HTTP for real-time communication and automation.

Outcomes:

- Understand Python fundamentals: variables, loops, conditions
- Communicate with Arduino using PySerial
- Send and receive data using MQTT and HTTP protocols
- Log data from DHT11 and ultrasonic sensors using Python
- Automate IoT tasks with simple, script-based control systems

Module 14: Networking & Protocols

Learning Objectives:

- Learn how devices communicate on a network

Topics Covered:

- TCP/IP model and protocol stack
- UART, I2C, SPI communication
- Basics of sockets in Python
- Building a chatroom with sockets

Key Benefits and Output:

Benefits:

1.Understand how IoT devices communicate through wired and wireless protocols over networks.
2. Gain hands-on experience with communication methods like UART, I2C, and SPI.
3. Learn to use Python sockets to build basic network applications like a chatroom.

Outputs:

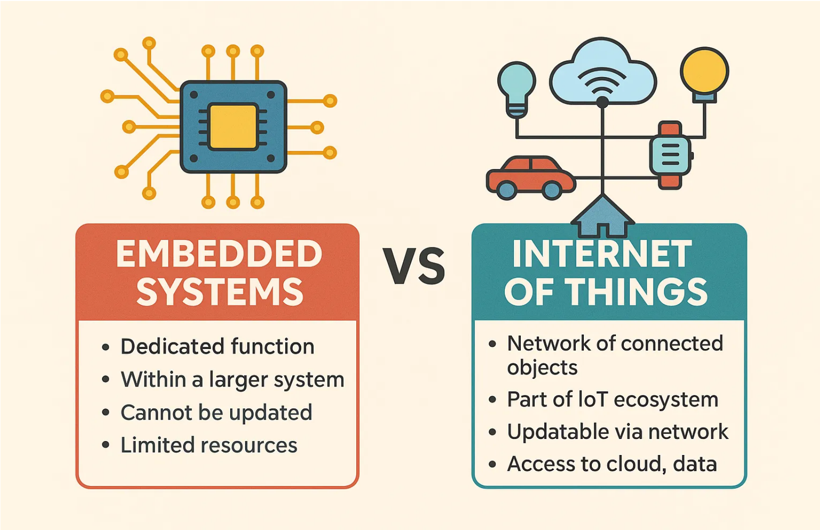
- Grasp the TCP/IP model and protocol stack
- Work with UART, I2C, and SPI for device-to-device communication
- Understand sockets and their role in networking
- Build a simple chatroom using Python sockets

Module 15: Capstone Projects

Objective: Apply all learned concepts to real-world projects

Sample Projects:

- COVID-19 Live Dashboard (API + Python)
- Smart Irrigation System
- Cloud-connected Weather Station
- Motion-activated Security System
- Voice-controlled Home Automation (App + Arduino)



Year 2 – Application & Project Phase

– Year 2 is full of hands-on-experience on 6 live projects –

1. Automatic Room Temperature Controller
   This project uses sensors like DHT11 or DHT22 to monitor room temperature and humidity in real-time. Based on sensor data, it controls fans or air conditioners using a microcontroller like Arduino or ESP32. The system maintains a desired temperature range automatically. It enhances energy efficiency and comfort in smart homes. Data can be displayed on an LCD or IoT dashboard.

   Tools used: **Arduino**, **Relay Module**, **DHT Sensor**, **ESP32**.

2. Voice Control Home Automation
   This system enables controlling home appliances like lights and fans using voice commands via Google Assistant or Alexa. It integrates IoT platforms like Blynk or IFTTT with

ESP32/NodeMCU and Wi-Fi. Voice input is processed through a smartphone or smart speaker. The system offers convenience, especially for the elderly or disabled. It supports remote and hands-free operation.

Tools: **ESP32**, **Google Assistant**, **IFTTT**, **Relay Module**.

3. Obstacle Avoidance Robot using Ultrasonic Sensor
   The robot detects obstacles using ultrasonic sensors and navigates by changing direction to avoid collisions. It uses an Arduino or Raspberry Pi to control motors based on sensor input. The system enhances autonomous mobility for robotics applications. It's ideal for environments where manual control is difficult. Widely used in research and automation.

   Tools: **Ultrasonic Sensor**, **Arduino UNO**, **Motor Driver**, **Wheels/Chassis**.

4. IoT based Smart Factory System
   This project monitors factory parameters like temperature, vibration, or gas levels using sensors connected to ESP32. The data is transmitted to cloud platforms like ThingSpeak or Blynk for visualization and alerts. It improves production efficiency and safety through real-time monitoring. Automation can be triggered based on thresholds.
   Tools: **ESP32**, **Industrial Sensors**, **Blynk**, **ThingSpeak**, **MQTT**.

5. IoT based Weather Data System using ESP32
   This system collects environmental data like temperature, humidity, and air pressure using sensors like DHT11 or BMP180. The ESP32 sends this data to cloud dashboards for live

monitoring and logging. Users can access the weather data remotely via web or mobile apps. It's useful for agriculture, smart cities, and education.

Tools: **ESP32**, **DHT11/BMP180**, **ThingSpeak**, **Blynk**.

6. Smart Building using BLYNK
   This IoT solution automates and monitors building systems like lighting, security, and temperature using Blynk and ESP32. Data from sensors is sent to the Blynk app for real-time visualization and control. The system supports scheduling, alerts, and remote operation. It enhances energy efficiency, safety, and user convenience.

   Tools: **ESP32**, **Blynk App**, **Sensors (PIR, DHT)**, **Relay Module**.

RISE OF IoT and EMBEDDED SYSTEMS

The impact of IoT and embedded systems on society and education from 2015 to 2030. The Rise of IoT and Embedded Systems: 2015-2030

The period from 2015 to 2030 has witnessed a profound transformation. These technologies have seamlessly integrated into our daily lives, reshaping industries, improving efficiency, and creating an increasingly interconnected world.

Impact on Society:

- **Smart Homes and Cities:** From 2015 onwards, smart home devices like intelligent thermostats, lighting systems, and security cameras became increasingly common, offering greater convenience and energy efficiency. By 2030, the concept has expanded to "smart cities," where IoT sensors and embedded systems optimize traffic management and enery distribution, leading to more sustainable and livable urban environments.

- **Healthcare Revolution:** IoT has revolutionized healthcare by enabling remote patient monitoring, wearable health trackers, and smart medical devices. This has led to more proactive healthcare, personalized treatment plans, and improved outcomes, especially for chronic disease management and elder care.

- **Transportation and Logistics:** IoT and embedded systems have been instrumental in the development of autonomous vehicles, intelligent transportation systems, and efficient logistics. Real-time tracking, route optimization, and smart fleet management have made transportation safer, more efficient, and more sustainable.

- **Environmental Monitoring:** IoT-driven embedded systems play a crucial role in environmental conservation by providing real-time monitoring of air quality, water usage, and pollution levels. These systems help track and reduce carbon footprints, contributing to global sustainability goals.

- **Economic Growth:** The economic value unlocked by IoT has been substantial. Estimates suggest that by 2030, IoT could enable trillions of dollars in global value, with a significant portion coming from B2B applications, though B2C applications like smart home solutions have also seen faster-than-expected adoption.

Impact on Education:

- **Enhanced Learning Environments:** IoT devices have transformed traditional classrooms into smart learning environments. Interactive whiteboards, smart sensors, and tablets provide students with engaging and visual learning experiences, offering uninterrupted access to theoretical and practical materials.

- **Automated Administration:** IoT solutions have streamlined administrative tasks for teachers and school administrators. Smart attendance trackers, automated grading systems, and efficient resource management (e.g., tracking textbooks) free up educators' time, allowing them to focus more on teaching and student interaction.

- **Improved Safety and Security:** IoT-enabled security cameras, smart ID cards, and access control systems enhance safety and security within educational institutions. These systems can monitor school grounds, detect potential threats, and alert authorities in emergencies.

- **Practical and Experiential Learning:** Embedded systems and IoT kits, like those based on Raspberry Pi and Arduino, have become valuable tools for hands-on learning in STEM fields. These tools allow students to design, build, and program their own IoT systems, fostering critical thinking and problem-solving skills.While the rise of IoT and embedded systems brings immense benefits, it also presents challenges related to data security, privacy, interoperability, and the need for robust infrastructure.

Year 3 – Placement & Internship Phase:

1. Guaranteed Internship Phase
    - In Year 3, Coincent guarantees an internship with partner companies. The internship includes a formal Internship Offer Letter and a Completion Certificate upon successful completion.
    - This is part of their "Industrial Training + Internship" model — It covers live classes, mentorship, and project work, but the internship phase itself is completely complimentary
2. Structured Placement Preparation
    - Coincent supports students in portfolio-building with multiple completed projects (typically around 8) and Microsoft-aligned certifications .
    - Coincent provides mock interviews, resume reviews, and training for HR and technical rounds — all aimed at preparing you for real-world hiring.
3. Final Take
    - Coincent's 3rd year transforms theory into practical experience through a guaranteed internship, builds a robust credentials portfolio, and equips you with placement-ready skills via mock interviews and resume prep.
    - If you're in your 4th year, this phase sets you on a clear trajectory from "training" to "hired."

Step Into Top Tech Roles

The leading and high-demand roles in the IoT and Embedded Systems Field along with a brief description of each:

1. IoT Developer:

- Develops software and firmware for IoT devices and applications, focusing on connectivity and communication protocols.
- May work with various programming languages (like C, Python) and hardware platforms.
- Involved in creating APIs and integrating IoT data with databases and cloud platforms.

2. Embedded Systems Engineer:

- Designs and develops the core hardware and software components of embedded systems.
- May specialize in areas like firmware development, hardware design, or system integration.
- Works on optimizing performance, power consumption, and real-time functionality.

3. IoT Architect:

- Designs and oversees the overall architecture of IoT solutions, focusing on scalability, security, and performance.

- Collaborates with various teams (product management, software engineering) to translate requirements into architecture and design.
- Recommends and integrates emerging technologies to solve complex business problems.

4. Firmware Engineer:

- Develops and maintains the low-level software (firmware) that runs on embedded devices.
- May be responsible for porting or rewriting firmware for new hardware.
- Works with programming languages like C and C++.

5. Embedded Hardware Engineer:

- Designs and develops the physical components of embedded systems, including circuit boards, sensors, and microcontrollers.
- Requires strong knowledge of electronics, PCB design, and signal processing.
- May be involved in testing and debugging hardware.

Other notable roles:

- **IoT Security Engineer:** Focuses on securing IoT devices and networks.
- **IoT Network Engineer:** Designs and manages the network infrastructure for IoT deployments.

- **Embedded Linux Engineer:** Specializes in developing software for embedded systems running Linux.
- **System Integrator:** Responsible for integrating hardware and software components of IoT systems.
- **Test Engineer:** Develops and executes test plans to ensure the quality and reliability of IoT and embedded systems.

These are just some of the many exciting career paths in the rapidly growing fields of IoT and embedded systems.