GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Electrical Engineering / Renewable Energy

Subject Code: DI05000201

Subject Name: Embedded & IoT Systems

w. e. f. Academic Year:	2026-27
Semester:	5th
Category of the Course:	PEC - III

Prerequisite:	Basic knowledge of electronic components, digital logic, and elementary programming concepts (C / C++).
Rationale:	The rapid convergence of embedded computing and the Internet of Things (IoT) is transforming every sector of the electrical industry — from smart homes and wearable health monitors to GPS-based asset tracking and real-time industrial dashboards. This subject is designed to equip students with practical, job-ready skills in two industry-relevant platforms: the Raspberry Pi (a Linux-based Single Board Computer ideal for edge processing and dashboards) and the ESP32 (a powerful Wi-Fi/Bluetooth microcontroller ideal for low-cost IoT nodes).

Course Outcomes: After Completion of the Course, Student will be able to:

CO No.	Course Outcomes	RBT Level*
01	Identify the hardware parts, GPIO pins, and operating system of Raspberry Pi and write simple Python programs to control basic output devices.	U, A
02	Connect sensors and output devices to ESP32, write basic programs in Arduino IDE, and send sensor data to a cloud platform using Wi-Fi.	U, A
03	Use Raspberry Pi and ESP32 together to set up and testIoT applications such as home automation, health monitoring, location tracking, and a live data dashboard.	A

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours/Week)			Total Credits L+T+(PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA(M)	PA(I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Raspberry Pi** 1.1 Introduction toRaspberry PiArchitecture: • Difference between Single Board Computers (SBCs) and microcontroller; • Why Raspberry Pi for IoT; • Overview of Raspberry Pi models (3B+, 4B, 5); • Hardware specifications (CPU, RAM, USB, HDMI, Ethernet, GPIO) 1.2 Hardware Architecture and GPIO Pinout: • Block diagram of Raspberry Pi; • 40-pin GPIO layout; • Digital I/O, PWM, I2C, SPI, UART pins; • Safe voltage levels (3.3 V logic); • LED and relay interfacing basics 1.3 Operating System Setup and Linux Basics: • Installing Raspberry Pi OS (Raspbian) using Raspberry Pi Imager; • Headless setup (SSH, VNC); • Essential Linux terminal commands (ls, cd, mkdir, nano, apt, sudo, reboot); • File management andRunning scripts at startup 1.4 Python Programming for Hardware Control: • Python basics — variables, data types, loops, functions, libraries; • RPi.GPIO library for digital I/O; • Reading sensors (DHT11 temperature/humidity); • Controlling output devices (LEDs, buzzers, relays); • Thonny IDE usage 1.5 Communication Interfaces and Data Logging: • I2C protocol, SPI protocol and UART communication; • Logging sensor data to CSV files;	**15**	**33%**
2.	**Introduction to ESP32** 2.1 Introduction to ESP32 Architecture: • ESP32 vs Arduino UNO vs Raspberry Pi; • ESP32 Pinouts and Peripherals; • Power consumption and battery-operation suitability	**15**	**33%**

	2.2 Development Environment and Programming: • Setting up Arduino IDE for ESP32; • Installing ESP32 board package; • Uploading and testing code; • Introduction to MicroPython on ESP32 using Thonny; • Basic programs — digital output, PWM, analog read; • Serial Monitor for debugging 2.3 Sensor Interfacing with ESP32: • Interfacing DHT11 / DHT22 (temperature, humidity); • LDR (light sensor); • Ultrasonic sensor (HC-SR04) for distance; • MQ-2 gas/smoke sensor; • Pulse sensor for heart rate; • GPS module (Neo-6M) for location data; • Displaying readings on OLED/LCD 2.4 Wi-Fi Connectivity and IoT Communication Protocols: • Station mode and Access Point mode; • Connecting ESP32 to Wi-Fi network; • HTTP GET/POST requests to cloud platforms; • Sending sensor data to ThingSpeak / Blynk IoT 2.5 ESP32 Web Server: • Creating a local web server on ESP32; • Controlling GPIOs via browser; • Real-time sensor data display on web page		
3.	**Embedded IoT Applications** 3.1 Home Automation System: • What is home automation and why it is useful; • Block diagram of a simple home automation system (ESP32 as controller, relay as switch, appliance as load); • Introduction to Blynk IoT app — adding buttons and sliders to control devices from mobile; • Practical demonstration of switching two appliances ON/OFF from a mobile app 3.2 Wearable Health Monitoring Device: • What is a wearable device and its use in health monitoring; • Sensors used: MAX30102 (heart rate and SpO2) and DHT11 (temperature); • Understanding the output — what is a normal vs. abnormal reading;	**15**	**34%**

	• Simple threshold alert: if heart rate is too high, light up a red LED 3.3 GPS-Based Location Tracking System: • What is GPS and how it works (satellites, signals, location fix); • GPS module Neo-6M — connecting to ESP32 via UART (Tx, Rx, VCC, GND); • Reading latitude and longitude from Serial Monitor using TinyGPS++ library; • Understanding the output data — what latitude and longitude mean; • Entering coordinates manually on Google Maps to verify the location; • Use cases: school bus tracking, vehicle location, farm equipment monitoring 3.4 Real-Time Data Collection and Dashboard: • Concept of a data dashboard — seeing live sensor values in one place; • Introduction to Node-RED on Raspberry Pi — drag-and-drop visual programming; • Adding an MQTT input node and a gauge/chart widget in Node-RED to display sensor data; • Setting up a simple alert: when temperature crosses a set value light up Red LED		
	Total	**45**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	U Level	A Level	N Level	E Level	C Level
20 %	30 %	50 %	00	00	00

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Reference Books:

1.	Rahul Dubey, "An Introduction to Internet of Things: Connecting Devices, Edge Gateway, and Cloud with Applications", Cengage India Publication

2.	Raj Kamal, "Internet of Things: Architecture and Design Principles", McGraw Hill Education
3.	Hanes et al., "IoT Fundamentals", Cisco Press
4.	Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on Approach)", Paperback, 2015
5.	A. McEwen, H. Cassimally, "Designing the Internet of Things", Wiley, 2013
6.	Donald Norris, "The Internet of Things: Do It Yourself Projects with Arduino, Raspberry Pi and BeagleBone Black", McGraw Hill Publication
7.	Adeel Javed, "Building Arduino Projects for Internet of Things", Apress Publication

(b) Learning Websites &mobile applications:

Sr.	Web-Link / Portal	Description
1.	https://www.raspberrypi.com/documentation/	Official Raspberry Pi Documentation — hardware, OS setup, Python GPIO, projects
2.	https://docs.espressif.com/projects/esp-idf/	Official ESP32 documentation by Espressif Systems — detailed hardware and API reference
3.	https://randomnerdtutorials.com/	Extensive step-by-step tutorials on ESP32, Raspberry Pi, Arduino, and IoT projects
4.	https://nodered.org/docs/	Official Node-RED documentation — flow editor, dashboard, MQTT, integrations
5.	https://thingspeak.com/	ThingSpeak IoT analytics platform — free cloud for sensor data storage and visualization
6.	https://www.kraj.in/raspberry-pi/	Step-by-step tutorials on Raspberry Pi.
7.	https://www.youtube.com/@krajchannel	Explaining Raspberry Pi tutorials in Gujarati.
8.	Android App: Kraj	Embedded & IoT Systemsstudy material in Gujarati.

Suggested Course Practical List: Each week includes one lab sessions (2 hours), designed such that they complement theory topics, means each experiment directly supports the classroom teaching.

Sr. No.	Practical Outcome/Title of experiment	Unit/ CO	Approx. Hours
1	Install Raspberry Pi OS on a microSD card using Pi Imager; perform headless SSH login and explore Linux terminal commands.	CO1	2

2	Write Python programs using RPi.GPIO to blink LEDs and control a relay module (simulating appliance ON/OFF).	CO1	2
3	Interface DHT11 / DHT22 sensor with Raspberry Pi and log temperature & humidity data to a CSV file every 10 seconds.	CO1	2
4	Connect an I2C-based sensor (BMP180 or MPU6050) to Raspberry Pi; display readings on 16x2 LCD using Python.	CO1	2
5	Set up ESP32 development environment in Arduino IDE; upload basic programs — LED blink, PWM fade, analog read.	CO2	2
6	Interface DHT22 and MQ-2 gas sensor with ESP32; display values on OLED display.	CO2	2
7	Connect ESP32 to Wi-Fi (Station mode); publish DHT22 sensor data to ThingSpeak cloud platform and observe live graph.	CO2	2
8	Build an ESP32 web server; control two relay outputs and view live sensor data from a mobile browser.	CO2	2
9	Interface GPS module (Neo-6M) with ESP32 via UART; display latitude, longitude, and fix status on Serial Monitor using TinyGPS++ library.	CO3	2
10	Upload ready-made Blynk home automation code to ESP32; add two button widgets in Blynk app; test switching relay outputs ON/OFF from mobile.	CO3	2
11	Connect MAX30102 to ESP32; upload example library code; read and display heart rate and SpO2 on Serial Monitor and OLED.	CO3	2
12	Install Node-RED on Raspberry Pi; add MQTT input node and two gauge widgets; receive and display temperature and humidity data from sensors.	CO3	2
13	Send DHT22 sensor data from ESP32 to ThingSpeak channel via Wi-Fi; observe live temperature and humidity graphs on ThingSpeak dashboard.	CO3	2
14	Mini Project: Select any one application (home automation / health monitor / GPS tracker / dashboard); set up the hardware using given reference code;	CO3	4
Subject In-charge may add further practicals within the framework of the COs.			

Suggested List of Laboratory / Learning Resources Required:

Sr. No.	Equipment / Resource Name with Specifications
1.	Raspberry Pi 4 Model B (4 GB RAM) — with 32 GB Class 10 microSD card, official USB-C power supply, HDMI cable, cooling case. (One per 2 students recommended)
2.	ESP32 Development Boards (NodeMCU-ESP32S or DOIT ESP32 DevKit V1) with built-in Wi-Fi & Bluetooth — minimum 10 boards per lab batch.
3.	Sensor Kit per bench: DHT22 (temp/humidity), MQ-2 (gas), HC-SR04 (ultrasonic), LDR, BMP180 (pressure), MPU6050 (accelerometer/gyroscope)
4.	Health Monitoring Sensors: MAX30102 Pulse Oximeter module (for wearable application practical)
5.	GPS Module: Neo-6M with antenna — for location tracking practical
6.	Display Modules: 16x2 LCD (I2C), 0.96" OLED I2C modules
7.	Relay Modules: 2-channel and 4-channel 5 V relay modules with optocoupler isolation for appliance control
8.	Wi-Fi Router: Dedicated lab Wi-Fi router (2.4 GHz) for creating isolated IoT lab network
9.	Computers / Laptops: Pre-installed with Raspberry Pi Imager, Arduino IDE (with ESP32 board package), Thonny IDE, PuTTY, VNC Viewer, Node-RED, Mosquitto MQTT Broker

Suggested Activities / Project List:

1. Raspberry Pi

Reports / Seminars / Presentations:

- Comparison of Raspberry Pi models and their suitability for different applications
- Applications of Raspberry Pi in smart home and industrial IoT

Projects / Activities / Demonstrations:

- Build a weather station using Raspberry Pi that logs temperature, humidity, and pressure to a local database every minute
- Design a home energy monitor: read appliance current via ACS712 sensor

2. ESP32

Reports / Seminars / Presentations:

- Technical comparison: ESP32 vs ESP8266 vs Arduino Uno vs Raspberry Pi Pico
- Understanding MQTT protocol — how it powers IoT at scale
- Survey of cloud IoT platforms: Blynk, ThingSpeak, AWS IoT, Azure IoT Hub

Projects / Activities / Demonstrations:

- Design a battery-powered ESP32 soil moisture sensor that wakes every 15 minutes, takes reading, publishes to cloud, and sleeps (deep sleep demo)
- Create a smart energy saving demo: ESP32 controls fan speed with PWM based on temperature sensor reading

3. Raspberry Pi and ESP32 Applications

Reports / Seminars / Presentations:

- How IoT is used in hospitals — patient monitoring systems and smart health bands
- GPS tracking in everyday life — school buses, delivery vehicles, and personal safety apps
- Smart homes in India — available products, their cost, and how they work
- How mobile apps like Blynk are used to control electrical devices wirelessly

Projects / Activities / Demonstrations:

- Home automation demo: control a light bulb and fan using Blynk app on mobile via ESP32 and relay — test ON/OFF switching
- Health monitor demo: display heart rate and SpO2 from MAX30102 on OLED; send readings to ThingSpeak and view the graph
- GPS demo: connect Neo-6M to ESP32, read and print coordinates on Serial Monitor, and verify location on Google Maps
- Dashboard demo: set up a simple Node-RED dashboard on Raspberry Pi to show temperature and humidity gauges from ESP32 via MQTT

****