TECH ZONE

IoT Diploma

Take your electronics skills online: ESP32 and ESP8266, Wi-Fi, MQTT, cloud dashboards, security and over-the-air updates.

Level
2, after the Embedded Systems Diploma

Length
40 hours over 8 modules, then the final projects

Boards
ESP32 and ESP8266

Finish
A graduation project



What you will master

Four skill areas that take a device from a single board on the desk to a secure, connected product.

1. Wi-Fi boards and programming

- ESP32 and ESP8266 architecture
- Wi-Fi modes: station (STA) and access point (AP)
- Power saving and deep sleep
- Interrupts and timers

2. Protocols and data transfer

- HTTP and REST APIs
- The MQTT protocol
- WebSockets for real-time data
- JSON and the ArduinoJson library

3. Cloud and dashboards

- Blynk, Adafruit IO and ThingSpeak
- Node-RED
- Alerts by email and message
- Logging data to an SD card and to databases

4. Security and remote updates

- TLS encryption on the ESP32
- Token authentication
- Over-the-air (OTA) firmware updates
- A complete graduation project

Nine modules, step by step

Module	What it covers	Hours
1. Introduction to IoT	What IoT is and where it is used (smart home, agriculture, Industry 4.0). IoT architecture: hardware, communication, cloud, application. ESP8266, ESP32, Arduino and Raspberry Pi compared.	4
2. Getting started with the ESP8266	Its specifications and GPIOs. UART, SPI and I2C. Practical: Blink an LED, GPIO control	5
3. Networking with the ESP8266	Joining Wi-Fi in station mode, the ESP8266 as an access point, LAN communication, the Network Time Protocol (NTP).	6
4. IoT communication protocols	HTTP GET and POST requests, MQTT basics and setting up a broker, HTTP and MQTT compared. Practical: Control an LED and publish sensor data	6
5. Programming the ESP8266 in the Arduino IDE	Setting up the IDE, uploading sketches, debugging with the serial monitor. Practical: LED control with debugging	3
6. Cloud integration with Firebase	Firebase basics, linking the ESP8266 to Firebase, mobile app integration. Practical: Relay control from a Firebase app	6
7. Advanced features	Over-the-air (OTA) updates, Alexa and Google Assistant basics. Practical: An OTA demo and device control by voice assistant	4
8. Working with other microcontrollers	The ESP8266 as a Wi-Fi module for AVR, PIC, ARM and Arduino, over UART and SPI. Turning classic microcontrollers into IoT devices.	6
9. Final projects and case studies	Six core final projects and case studies, on the project pages. Practical: Your graduation project	

40 hours over modules 1 to 8. Module 9 is the final projects.

How an IoT system fits together

Every project in the diploma follows the same five layers, from the sensor to the screen in the student's hand.

Sensors and actuators
DHT11, relays, motors
→
ESP32 or ESP8266
Reads, decides, connects
→
Wi-Fi network
Router or access point
→
Broker or cloud
MQTT, REST, Firebase
→
App and dashboard
Phone, web, alerts

Protocols and data formats

Protocol	What it does	Where
HTTP and REST	Request and response over the web. Simple for reading values and sending commands.	Modules 4 and 6
MQTT	Lightweight publish and subscribe through a broker. Ideal for many devices sending small messages.	Module 4
WebSockets	A lasting two-way connection for real-time data and instant control.	Protocols skill area
JSON	The common format that packages values for APIs and the cloud (ArduinoJson).	Protocols skill area

Cloud platforms and tools

Firebase	Cloud database with mobile app integration. Used for relay control from an app.
Blynk	Ready-made phone dashboards for control and monitoring.
Adafruit IO	Cloud dashboards and data feeds for IoT devices.
ThingSpeak	Data logging and charts for sensor readings.
Node-RED	A visual, flow-based tool for data flows, dashboards and alerts.

The kit

The same kit as Level 1, now built around the ESP32. Every part is sold in the TECH ZONE shop.

Controller and basics



ESP32 board and USB cable
Dual-core, built-in Wi-Fi, ready for TLS and OTA.



Breadboard
Build and change circuits fast, no soldering.



Jumper wires
Connect boards, sensors and modules.



Resistors
Limit current and build simple voltage dividers.



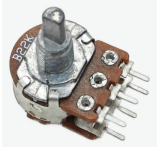
LEDs
Status lights, and brightness with PWM.



Push buttons (5)
Inputs for controls, menus and confirmations.



Buzzer
Alarms, reminders and feedback tones.



Potentiometer
Analog input for values, speed and limits.



Traffic light module
Red, yellow and green LEDs on one board.



Keypad
Codes and choices, for locks and menus.



Joystick
Two-axis control for cars and robots.



Batteries and holder
Portable power for robots and field devices.

What changed from Level 1

Removed	Arduino Uno and cable, Bluetooth module, NRF24L01
Added	ESP32 development board with its USB cable
Unchanged	Every other sensor, the display, motors, relay, driver, chassis and basic parts

Sensors and display



LCD with I2C
A text display on just two wires.



DHT11
Temperature and humidity for weather and climate.



Ultrasonic sensor
Distance, and obstacles ahead.



IR sensor
Nearby objects and passing cars.



IR receiver
Reads commands from a remote control.



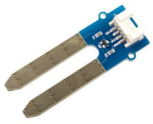
Flame sensor
Flame detection for fire alarms.



Gas sensor
Gas and smoke, for safety alerts.



Water level sensor
Water level in tanks and containers.



Soil moisture sensor
How wet the soil is, for irrigation.

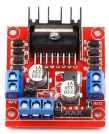
Motion and switching



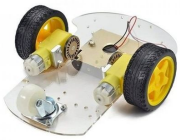
Servo motor
Exact angles for barriers, locks and arms.



2 DC motors and wheels
Drive a robot or a car.



Motor driver
Controls motor speed and direction safely.



2WD chassis
Robot frame with a free caster wheel.



Relay module
Switches bigger loads: lamps, fans, pumps.



DC fan
Cooling, ventilation, smart home projects.

Voltage: ESP32 and ESP8266 pins work at 3.3V and are not 5V tolerant. A module that outputs a 5V signal, such as the echo pin of the ultrasonic sensor, needs a voltage divider or a level shifter.

The projects

Twelve projects. The six core final projects come first, then six suggested ones.

1 **Robot control from a phone over Wi-Fi** — Core

Drive the robot from a phone over Wi-Fi, with no Bluetooth pairing needed.

Parts: ESP32, motor driver, DC motors, 2WD chassis, batteries

2 **Smart home system (Arduino and IoT)** — Core

Control lamps and a fan from a phone, and watch temperature and gas on a dashboard.

Parts: ESP32, Arduino Uno, relay, DHT11, gas sensor, DC fan, LEDs

3 **Lighting control with NTP** — Core

Lights follow a schedule using internet time (NTP), with no clock chip.

Parts: ESP32, relay, LEDs

4 **Smart irrigation system** — Core

Waters the plants when the soil is dry, and warns when the tank runs low.

Parts: ESP32, soil sensor, water level sensor, relay, water pump

5 **Medication reminder** — Core

Shows and sounds a reminder at the set time; a button confirms the dose.

Parts: ESP32, LCD, buzzer, push buttons, LEDs

6 **IoT-enabled microcontroller** — Core

A classic Arduino becomes an IoT device by adding an ESP8266 Wi-Fi module over UART.

Parts: Arduino Uno, ESP8266, LEDs

The projects, continued

7 **IoT weather station** Suggested

Publishes temperature and humidity over MQTT, shows them on a dashboard and logs them to an SD card.

Parts: ESP32, DHT11, LCD, SD card module

8 **Gas leak alert with notifications** Suggested

Detects gas, sounds the alarm, starts the fan and sends an email or a message.

Parts: ESP32, gas sensor, buzzer, DC fan, relay

9 **Fire alarm with cloud alerts** Suggested

Confirms a fire with two sensors and pushes an alert to the phone.

Parts: ESP32, flame sensor, gas sensor, buzzer, LEDs

10 **Smart parking dashboard** Suggested

IR sensors track free spaces, and a live dashboard updates over WebSockets.

Parts: ESP32, IR sensor, servo, LCD, LEDs

11 **Voice-controlled lights** Suggested

Switch the lights with Alexa or Google Assistant.

Parts: ESP32, relay, LEDs

12 **Secure smart safe box** Suggested

A keypad lock that sends an alert after wrong attempts, and takes remote commands only with a valid token.

Parts: ESP32, keypad, servo, LCD, buzzer

Security and remote updates

A connected device is only useful if it is secure and can be kept up to date.

TLS encryption on the ESP32
Encrypts the data between the device and the server so it cannot be read on the way.

Token authentication
Only devices and apps that present a valid token may connect or send commands.

Over-the-air (OTA) updates
Update the firmware over Wi-Fi, with no cable, even on devices already installed.

Power saving and deep sleep
Battery devices last far longer by sleeping between readings and waking on a timer or an interrupt.

Graduation project

Every student finishes with a complete IoT project that combines a connected device, a communication protocol, a cloud dashboard and secure remote updates.

Join the next group

Message us on WhatsApp at **0101 098 4875** to book a place or ask about dates. Details are also at **techzoneegypt.com/learn/university**.