COURSE TABLE OF CONTENTS

I2C Sensor Networks

OEM: Koenig Original • Duration: 2 Days (14 hrs)

COURSE MODULES & TOPICS

Module 1: I2C Fundamentals

- Compare I2C, SPI and UART serial buses
- Controller and target device roles
- Electrical characteristics: SDA, SCL, open-drain, pull-up resistors
- 3.3 V and 5 V logic considerations
- Protocol: start/stop, addressing, ACK/NACK, repeated start, clock stretching
- Bus speeds: standard, fast, fast-plus
- Lab: Breadboard I2C bus and observe frames on logic analyser

Module 2: Raspberry Pi I2C Setup and Tooling

- Raspberry Pi I2C pins and on-board pull-ups
- Enable I2C with raspi-config and device-tree overlays
- Set bus speed
- i2c-tools: i2cdetect, i2cget, i2cset, i2cdump, /dev/i2c-1
- Headless operation: static IP and SSH
- Lab: Scan bus, identify device, read chip ID register

Module 3: Sensors, Datasheets and Register-Level Access

- Read datasheets: address pins, register map, configuration, timing
- Common I2C sensors: BME280, BH1750, MPU-6050, ADS1115, DS3231, SSD1306
- Convert raw bytes: calibration, endianness, signed values, bit masks
- Python drivers with smbus2: byte/block reads/writes, error handling
- Lab: Read sensor data in Python, build reusable driver class

Module 4: ESP32 Sensor Nodes (Arduino Framework)

- ESP32 I2C peripheral: default pins, remapping
- Wire library: scan bus, read sensors in C++
- Non-blocking sampling
- Basic Wi-Fi connection to lab network
- Lab: Build ESP32 node reading two sensors, report over serial

Module 5: Multi-Device Bus Design

- Address conflicts and address-select pins
- TCA9548A I2C multiplexer usage
- Bus capacitance, cable length, pull-up sizing, buffers, extenders
- Power: decoupling, 3.3 V supply budget
- Lab: Connect two same-address devices behind multiplexer, read both

Module 6: Telemetry with MQTT

- MQTT topic design, JSON payloads, QoS, retained messages
- Publish from Python (paho-mqtt)
- Publish from ESP32 to Mosquitto broker

© Koenig Solutions

- Lab: Publish and subscribe to readings from Raspberry Pi and ESP32

Module 7: Fault Finding and Hardening

- Symptoms: missing device, wrong readings, NACKs, SDA held low
- Bus recovery: clocking pulses, retries, watchdogs
- Logic analyser protocol decoder, troubleshooting checklist
- Lab: Diagnose faults-wrong address, missing pull-up, stuck SDA

Module 8: Capstone: Environmental Monitoring Network

- Raspberry Pi hub with BME280, BH1750, OLED display
- Remote ESP32 node reporting over MQTT
- Demonstration and review
- Lab: Build and demonstrate multi-node sensor network