

ANNEXURE: 1 MODULE WISE COURSE CONTENT AND OUTCOME

| TABLE: MODULE WISE COURSE CONTENT AND OUTCOME |  |   |   |                |
|---|--|---|---|----------------|
| S.N   | MODULE NAME                                    | MODULE CONTENT  | MODULE LEARNING OUTCOME   | DURATION (HRS) |
| 1   | <b>FUNDAMENTALS OF SENSORS AND INTERFACING</b> | <ul style="list-style-type: none"> <li>• <b>Introduction to Sensors:</b></li> <li>• Overview of sensors: analog, digital, and smart sensors</li> <li>• Key parameters: sensitivity, accuracy, resolution, and response time</li> <li>• <b>Interfacing Basics:</b></li> <li>• Analog interfacing: ADC and signal conditioning</li> <li>• Digital interfacing: I2C, SPI, UART, and GPIO</li> <li>• Power considerations for sensors</li> <li>• <b>Arduino Basics:</b></li> <li>• Pin configurations, basic coding in Arduino IDE</li> <li>• Reading sensor data using analog and digital pins</li> <li>• Raspberry Pi Basics:</li> <li>• GPIO setup and Python programming</li> <li>• Using Raspberry Pi OS to access sensors</li> <li>• <b>Hands-on:</b></li> <li>• Read analog sensor data using Arduino and</li> </ul> | <ul style="list-style-type: none"> <li>• Understand the basic types of sensors (analog, digital, and smart) and their key parameters (e.g., sensitivity, accuracy, and resolution).</li> <li>• Gain proficiency in interfacing sensors with Arduino and Raspberry Pi using both analog and digital communication methods (e.g., ADC, I2C, SPI, GPIO).</li> <li>• Develop basic skills in programming and using the Arduino IDE for reading sensor data and visualizing it via the Serial Monitor.</li> <li>• Learn to set up Raspberry Pi for sensor interfacing, including GPIO setup and using Python to access sensor data.</li> </ul> | 9 Hrs          |

|   |   |  |  |       |
|---|---|--|--|-------|
|   |   | <p>visualize it using the Serial Monitor.</p> <ul style="list-style-type: none"> <li>• Interface a digital sensor with Raspberry Pi using I2C protocol.</li> </ul>   |  |       |
| 2 | <b>ANALOG SENSORS AND SIGNAL CONDITIONING</b> | <ul style="list-style-type: none"> <li>• <b>Analog Sensor Interfacing:</b></li> <li>• Resistive sensors: thermistors, strain gauges</li> <li>• Voltage and current output sensors</li> <li>• <b>Signal conditioning:</b> amplification, filtering, impedance matching</li> <li>• Using Arduino for Analog Sensors:</li> <li>• Using built-in ADC for signal acquisition</li> <li>• Plotting data with serial plotter</li> <li>• <b>Using Raspberry Pi for Analog Sensors:</b></li> <li>• Using external ADC modules (e.g., MCP3008)</li> <li>• Visualizing data with Python libraries (e.g., Matplotlib)</li> <li>• <b>Hands-on:</b></li> <li>• Interface a thermistor with Arduino and display temperature readings.</li> <li>• Interface an analog gas sensor with Raspberry Pi using an MCP3008 ADC.</li> </ul> | <ul style="list-style-type: none"> <li>• Apply principles of analog sensor interfacing, including the use of resistive sensors (e.g., thermistors) and voltage/current output sensors.</li> <li>• Implement signal conditioning techniques like amplification, filtering, and impedance matching to enhance sensor data accuracy.</li> <li>• Develop skills in using Arduino's built-in ADC for signal acquisition and visualization.</li> <li>• Learn how to interface analog sensors with Raspberry Pi using external ADC modules (e.g., MCP3008) and visualize data with Python.</li> </ul> | 9 Hrs |

|   |  |   |  |       |
|---|--|---|--|-------|
| 3 | <b>DIGITAL SENSORS AND COMMUNICATION PROTOCOLS</b> | <ul style="list-style-type: none"> <li>• <b>Digital Sensors:</b></li> <li>• MEMS sensors (e.g., accelerometers, gyroscopes)</li> <li>• Smart sensors with onboard data processing</li> <li>• <b>Communication Protocols:</b></li> <li>• I2C, SPI, and UART basics</li> <li>• Addressing and data transfer techniques</li> <li>• <b>Using Arduino for Digital Sensors:</b></li> <li>• Accessing sensor data via I2C and SPI libraries</li> <li>• Debugging communication issues</li> <li>• <b>Using Raspberry Pi for Digital Sensors:</b></li> <li>• Accessing GPIO, I2C, SPI, and UART interfaces via Python</li> <li>• Using smbus and spidev Python libraries for digital sensors</li> <li>• <b>Hands-on:</b></li> <li>• Interface an accelerometer with Arduino and display real-time data.</li> <li>• Use a humidity sensor with Raspberry Pi via I2C.</li> </ul> | <ul style="list-style-type: none"> <li>• Master the use of digital sensors (e.g., MEMS sensors) and communication protocols (I2C, SPI, UART).</li> <li>• Develop the ability to interface with digital sensors via I2C and SPI using both Arduino and Raspberry Pi, including handling addressing and data transfer.</li> <li>• Learn to troubleshoot communication issues in digital sensor interfacing.</li> <li>• Gain hands-on experience accessing sensor data from digital sensors and displaying it in real time using Arduino and Raspberry Pi.</li> </ul> | 9 Hrs |
| 4 | <b>ADVANCED SENSOR INTERFACI</b>                   | <ul style="list-style-type: none"> <li>• <b>Wireless Sensor Interfacing:</b></li> </ul>   | <ul style="list-style-type: none"> <li>• Understand wireless sensor interfacing using modules like</li> </ul>  | 9 Hrs |

|   |  |  |  |       |
|---|--|--|--|-------|
|   | <b>NG<br/>TECHNIQUES</b>                   | <ul style="list-style-type: none"> <li>• Bluetooth and Wi-Fi modules (e.g., ESP8266, ESP32)</li> <li>• Setting up wireless data transmission to a server or mobile app</li> <li>• <b>Sensor Fusion:</b></li> <li>• Combining accelerometer and gyroscope data for IMU calculations</li> <li>• Filtering algorithms: Kalman and complementary filters</li> <li>• <b>Advanced Raspberry Pi Applications:</b></li> <li>• Real-time sensor monitoring with MQTT and Node-RED</li> <li>• Database integration for logging sensor data</li> <li>• <b>Hands-on:</b></li> <li>• Create a wireless temperature and humidity monitoring system using ESP32 with Arduino IDE.</li> <li>• Implement sensor fusion with accelerometer and gyroscope on Raspberry Pi.</li> </ul> | <p>ESP8266 and ESP32, and design systems for wireless data transmission.</p> <ul style="list-style-type: none"> <li>↳ Learn sensor fusion techniques by combining accelerometer and gyroscope data, and implement filtering algorithms (Kalman and complementary filters) for data refinement.</li> </ul> <ul style="list-style-type: none"> <li>• Explore advanced applications on Raspberry Pi, including real-time monitoring using MQTT and Node-RED, as well as integrating sensor data into a database or cloud server for remote access and storage.</li> </ul> |       |
| 5 | <b>APPLICATIONS AND PRACTICAL PROJECTS</b> | <ul style="list-style-type: none"> <li>• <b>Applications:</b></li> <li>• Environmental monitoring (e.g., air quality, temperature, and light sensors)</li> <li>• Smart homes (e.g., motion</li> </ul>  | <ul style="list-style-type: none"> <li>↳ Apply sensor interfacing skills to real-world applications, including environmental monitoring, smart home systems, and robotics.</li> </ul>  | 9 Hrs |

|  |  |   |   |  |
|--|--|---|---|--|
|  |  | <p>detection, door open/close sensors)</p> <ul style="list-style-type: none"> <li>• Robotics (e.g., proximity sensors, LIDAR for navigation)</li> <li>• <b>Practical Projects:</b></li> <li>• Design a smart home system using Arduino to monitor temperature, motion, and light intensity.</li> <li>• Build a weather monitoring station with Raspberry Pi, logging temperature, pressure, and humidity data to a cloud server.</li> <li>• Create a robotic arm using position and force sensors, controlled by Raspberry Pi.</li> <li>• <b>Final Assessment:</b></li> <li>• Integrate multiple sensors with Arduino or Raspberry Pi for a real-world project.</li> <li>• Present performance analysis, design challenges, and solutions.</li> </ul> | <ul style="list-style-type: none"> <li>• Design and implement complete sensor-based systems with Arduino and Raspberry Pi, integrating multiple sensors for specific projects (e.g., smart home, weather station, or robotic arm).</li> <li>• Gain practical experience in presenting project designs, performance analysis, and addressing challenges in real-world sensor interfacing scenarios.</li> </ul> |  |
|--|--|---|---|--|

## **ANNEXURE : 2 Industry Use Cases/Final Projects**

### **1.Smart Home Temperature and Humidity Monitoring**

Scenario:

Monitor and regulate the air quality inside a home to ensure optimal comfort.

Task:

Interface temperature and humidity sensors with Arduino or Raspberry Pi to collect and visualize real-time data, and automate HVAC systems based on readings.

### **2.Industrial Equipment Monitoring**

Scenario:

Prevent machine downtime in an industrial setting by monitoring operating conditions.

Task:

Use analog and digital sensors with Raspberry Pi to monitor temperature, vibration, and pressure of equipment, sending alerts when thresholds are exceeded.

### **3.Environmental Pollution Monitoring in Urban Areas**

Scenario:

Track pollution levels in urban areas to comply with environmental standards.

Task:

Interface gas sensors (CO<sub>2</sub>, CO, NO<sub>2</sub>) with Raspberry Pi for real-time pollution monitoring, and log data to a cloud server for further analysis.

### **4.Agricultural Irrigation Automation**

Scenario:

Automate irrigation systems based on weather conditions to optimize water usage.

Task:

Use temperature, humidity, and soil moisture sensors with Arduino to monitor environmental conditions and control irrigation systems accordingly.

### **5.Smart Lighting Systems for Offices**

Scenario:

Save energy by automatically adjusting lighting based on occupancy and ambient light.

Task:

Interface motion and light sensors with Raspberry Pi to control the lighting system in an office building, reducing energy consumption.

### 1.Real-Time Temperature Monitoring in Industrial Furnaces

Scenario:

Monitor furnace temperatures to ensure safe and optimal operation.

Task:

Interface thermocouples or RTDs with Arduino, using signal conditioning techniques to acquire accurate temperature data for real-time monitoring and control.

### 2.Monitoring of Fuel Systems in Vehicles

Scenario:

Monitor fuel pressure and temperature for efficient vehicle operation.

Task:

Interface resistive pressure sensors and thermistors with Raspberry Pi to monitor fuel system parameters, ensuring optimal vehicle performance.

### 3.Strain Measurement for Structural Integrity

Scenario:

Ensure the safety of structural components in buildings or bridges.

Task:

Use strain gauges with Arduino to measure stress and strain on critical structural components, providing data for predictive maintenance.

### 4.Solar Energy Production Monitoring

Scenario:

Track and optimize solar panel performance for maximum energy output.

Task:

Interface current and voltage sensors with Raspberry Pi to monitor solar panel efficiency, logging data and alerting users to underperformance.

### 5.Precision Agricultural Equipment Calibration

Scenario:

Calibrate and monitor agricultural machinery for accurate planting and fertilizing.

Task:

Use analog sensors for soil moisture and temperature, interfaced with Arduino, to ensure precision in automated agricultural equipment.

### 1. Smart Wearables for Healthcare Monitoring

Scenario:

Track vital health parameters of individuals for continuous monitoring.

Task:

Interface MEMS sensors (accelerometer, gyroscope) with Arduino for activity tracking and health monitoring in a wearable device.

### 2. Wireless Industrial Asset Tracking

Scenario:

Automatically track the location and condition of industrial assets.

Task:

Use digital sensors (e.g., RFID) with Raspberry Pi to monitor and track assets in real-time, reducing loss and improving inventory management.

### 3. Automated Security Systems for Critical Infrastructure

Scenario:

Enhance security by automating motion detection and alarm systems.

Task:

Interface motion sensors with Raspberry Pi to trigger security alarms and monitor unauthorized movement in sensitive areas.

### 4. IoT-based Smart Factory Monitoring

Scenario:

Track real-time data from various machines and equipment on a production line.

Task:

Use accelerometers, humidity, and temperature sensors with Raspberry Pi, transmitting data via I2C/SPI to a central control system for analysis and decision-making.

### 5. Smart Traffic Management System

Scenario:

Improve traffic flow and reduce congestion in urban centers.

Task:

Use proximity and environmental sensors interfaced with Raspberry Pi to monitor traffic patterns and adjust traffic signals accordingly.



### 1. Remote Weather Station for Disaster Management

Scenario:

Monitor weather conditions remotely to predict and manage natural disasters.

Task:

Interface wireless weather sensors (e.g., barometer, anemometer) with Raspberry Pi and transmit data over Wi-Fi to a central server for analysis.

### 2. Wireless Environmental Monitoring for Remote Locations

Scenario:

Monitor environmental parameters (e.g., temperature, humidity) in remote locations.

Task:

Use ESP32 or similar wireless modules with Arduino to transmit data from remote locations to a cloud-based system for continuous monitoring.

### 3. Industrial Machine Health Monitoring Using Wireless Sensors

Scenario:

Monitor the health of industrial machinery remotely to predict maintenance needs.

Task:

Interface vibration and temperature sensors with ESP32, using wireless communication to send data to a cloud-based monitoring system.

### 4. Smart City Waste Management System

Scenario:

Optimize waste collection in urban areas by monitoring garbage bin levels.

Task:

Use ultrasonic distance sensors with Raspberry Pi to monitor the fill level of waste bins and send alerts when bins need to be emptied.

### 5. Sensor Fusion for Autonomous Navigation in Robotics

Scenario:

Enable robots to navigate autonomously in complex environments.

Task:

Use sensor fusion techniques combining accelerometer, gyroscope, and LIDAR data on Raspberry Pi to enable real-time autonomous navigation of robots in dynamic settings.

| <b>TABLE: LIST OF FINAL PROJECTS (20 PROJECTS THAT COMPREHENSIVELY COVER ALL THE LEARNING OUTCOME)</b> |   |
|--|---|
| <b>SL.NO</b>   | <b>FINAL PROJECT</b>  |
| 1  | Smart Home Temperature and Humidity Monitoring              |
| 2  | Industrial Equipment Monitoring                             |
| 3  | Environmental Pollution Monitoring in Urban Areas           |
| 4  | Agricultural Irrigation Automation                          |
| 5  | <b>Smart Lighting Systems for Offices</b>                   |
| 6  | Real-Time Temperature Monitoring in Industrial Furnaces     |
| 7  | Sensor Fusion for Autonomous Navigation in Robotics         |
| 8  | Monitoring of Fuel Systems in Vehicles                      |
| 9  | Strain Measurement for Structural Integrity                 |
| 10   | Solar Energy Production Monitoring                          |
| 11   | Precision Agricultural Equipment Calibration                |
| 12   | Smart Wearables for Healthcare Monitoring                   |
| 13   | Wireless Industrial Asset Tracking                          |
| 14   | Automated Security Systems for Critical Infrastructure      |
| 15   | IoT-based Smart Factory Monitoring                          |
| 16   | Smart Traffic Management System                             |
| 17   | Remote Weather Station for Disaster Management              |
| 18   | Wireless Environmental Monitoring for Remote Locations      |
| 19   | Industrial Machine Health Monitoring Using Wireless Sensors |
| 20   | Smart City Waste Management System                          |

**ANNEXURE :3 Assessment Rubrics**

| <b>TABLE 5: COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 70)</b> |   |  |   |                    |
|---|---|--|---|--------------------|
| <b>ASSESSMENT CRITERIA</b>                                  | <b>DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE</b>                                |  |   | <b>TOTAL MARKS</b> |
|   | <b>FAIR</b>   | <b>GOOD</b>  | <b>EXCELLENT</b>  |                    |
| Knowledge and understanding                                 | Recalls key definitions and concepts.   | Explains concepts in a clear and concise manner.   | Applies concepts to solve problems and answer questions in a comprehensive and insightful manner.   | 20                 |
| Application and Analysis                                    | Attempts to apply knowledge to solve problems, even if the solution is not entirely accurate. | Applies knowledge to solve problems correctly, demonstrating a clear understanding of the concepts involved.     | Critically analyzes problems, identifies relevant concepts, and applies knowledge to develop creative and effective solutions.            | 20                 |
| Evaluation and Synthesis                                    | Identifies relevant information from various sources.   | Analyzes and critiques information from various sources, identifying strengths and weaknesses.                   | Synthesizes information from various sources to form well-founded arguments and evidence-based conclusions.                               | 15                 |
| Communication Skills  | Presents information in a clear and organized manner, but may lack detail or clarity.         | Presents information in a clear, concise, and well-organized manner, using appropriate language and terminology. | Presents information in a clear, concise, and well-organized manner, using sophisticated language and terminology to engage the audience. | 15                 |