SKILL LAB DAY 1 REPORT

Organized by: Electronics and Instrumentation Engineering (EIE)

Speaker: Mr. P M Prabhu Avinya Networks

Date: 06/04/2026

1. Introduction

The Department of Electronics and Instrumentation Engineering (EIE) organized a Skill Lab session on IoT and Arduino-based embedded systems. The session was led by Mr. P M Prabhu from Avinya Networks, who delivered an in-depth introduction to the Arduino Uno and its applications.

2. Components of Arduino UNO:

Arduino Uno is a beginner-friendly microcontroller board based on the ATmega328P, widely used for electronics projects and coding.

2.1 Key Features of Arduino Uno

- Microcontroller: ATmega328P
- Digital I/O Pins: 14 (6 support PWM output)
- Analog Input Pins: 6
- Clock Speed: 16 MHz ceramic resonator
- Memory: 32 KB Flash, 2 KB SRAM, 1 KB EEPROM
- Connectivity: USB port, ICSP header, power jack
- Power Options: USB cable, AC-to-DC adapter, or 9V battery via barrel plug connector
- Reset Button: Allows restarting programs easily

Arduino UNO is a microcontroller-based development board that forms the core of most experiments in this lab. Its key components are described below.

2.2 Reset Push Button

- Reloads (restarts) the current program.
- Located at the top-right corner of the board.

2.3 USB Connector

- Has 4 wires internally.

- 3rd and 4th wires: +D and −D (data lines).
- 1st and 2nd wires: +5V and GND (power lines).
- The large steel-like block next to the reset button.
- The microcontroller understands only binary (0 and 1).

2.4 Microcontroller

- Located at the center of the board (black chip).
- Operates at 0–1V (logic LOW) and 1–5V (logic HIGH).

2.5 CH340 Chip

- Located beside the USB connector.
- Converts USB data format to TTL (Transistor-Transistor Logic).

2.6 LDO – Low-Dropout Regulator

- Located below the USB connector, with 3 steel buttons.
- Protects the board from short-circuit conditions.
- It is a DC linear voltage regulator.
- Advantages: no switching noise, small device size, simplicity of design.
- Drawback: must dissipate power to control output voltage.

2.7 Crystal Oscillators

- 3 oscillators total (identifiable by metal caps).
- Has mechanical and electrical axes.
- Made of piezo crystal material — two axes at 14° to each other.
- Has a natural frequency at which it vibrates when force is applied.
- 8 MHz oscillator — provides clock pulses.
- 16 MHz oscillator (2 times faster) — main clock.
- RC Oscillator — 128 KHz, used for delay.

2.8 Analog Pins

- 6 analog pins, each of 10-bit resolution.
- Output range: 0 to 1023 (analog to digital conversion).
- LED is located at pin 13.

2.9 Arduino Boards

The popular Arduino boards are, Arduino Uno (ATmega328P), Arduino Mega (ATmega2560), Arduino Nano (ATmega328P)

Tables 1 shows the Comparison of Arduino Boards

Feature	Arduino Uno (ATmega328P)	Arduino Mega (ATmega2560)	Arduino Nano (ATmega328P)
Digital I/O Pins	14	54	14
Analog Inputs	6	16	8

Feature	Arduino Uno (ATmega328P)	Arduino Mega (ATmega2560)	Arduino Nano (ATmega328P)
Flash Memory	32 KB	256 KB	32 KB
Size	Standard board	Larger board	Compact, breadboard-friendly
Best Use	Beginners, small projects	Complex projects, robotics	Portable, space-limited projects

3. Applications

- Robotics: Controlling motors, sensors, and actuators.
- IoT Projects: Connecting sensors to cloud platforms.
- Automation: Simple process control and monitoring.
- Education: Teaching embedded systems, electronics, and programming.

Conclusion

The introductory session on Arduino Uno successfully familiarized participants with the fundamentals of microcontroller-based design. By exploring its architecture, features, and versatile applications, students gained a clear understanding of how Arduino Uno serves as a practical platform for learning electronics, programming, and automation. The session not only highlighted its relevance in robotics, IoT, and instrumentation projects but also encouraged students to experiment, innovate, and apply their knowledge to real-world problem-solving.

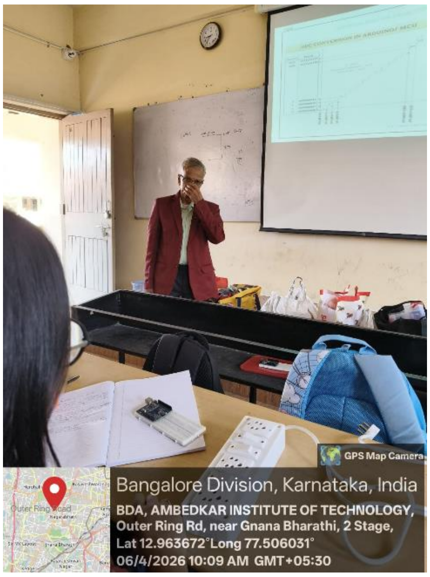

Fig 1: Day 1 Skill Lab

SKILL LAB DAY 2 REPORT

Organized by: Electronics and Instrumentation Engineering (EIE)

Speaker: Mr. P M Prabhu, Avinya Networks

Date: 07/04/2026

"Hands-on Exploration of Arduino Uno: Programming, Sensors, and Real-World Applications"

Focus Areas

- **Programming Basics:** Writing and uploading simple sketches using the Arduino IDE.
- **Interfacing Sensors:** Connecting input devices (temperature, light, motion sensors) and observing outputs.
- **ADC & Signal Concepts**
- **Actuators & Control:** Using LEDs, motors, and buzzers to demonstrate automation.
- **Mini Projects:** Simple tasks like traffic light simulation, temperature monitoring, or motion detection.
- **Industry Relevance:** Linking Arduino applications to robotics, IoT, and smart instrumentation.

Second day of the program stated with ADC & Signal Concepts

1 Analog Pins – Digital Pins

- Arduino has 6 digital pins and 14 analog pins.
- All 6 digital pins can be used as output or input.
- Accuracy: 1/1024 = 0.001 = 0.1%.

2 ADC Conversion in Arduino/MCU – 10 bit

- Successive approximation is the type of A-to-D conversion used.
- Each step = 5V / 1024 = 4.887 mV.
- Example: For 2V input → 2000 mV / 4.887 mV = 409th step.

3 PWM – Analog Write

- analogWrite(0) → Output = 0V
- analogWrite(256) → Output = 5V (PWM is 8-bit, so 256 = full).

Duty Cycle Reference:

0%	0
25%	64
50%	127
75%	191
100%	255

The following experiments are conducted on day 2 of skill lab

Experiment 1 – Digital Output: LED & Buzzer (Blink)

Objective: Produce a digital output to pin no. 2 using LED and Buzzer.

Pin Configuration:

- LED pin: PIN 2
- Buzzer pin: PIN 4

Important Notes:

- Long leg of LED = Anode (+).
- Short leg / flat side = Cathode (−).
- The Arduino board has only 2 buzzers (active buzzer type).
- Positive leg of LED → pin no. 9.
- GND pin of Arduino → GND of Arduino board.
- VCC pin → 5V of Arduino board.
- Changing leg of LED → connect to clockwise.

Sample Code:

```
#define TILT_PIN 2
#define BUZZER_PIN 4
void setup() {
  pinMode(TILT_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}
void loop() {
  bool tiltState = digitalRead(TILT_PIN);
  if (tiltState == HIGH) {
```

```
    digitalWrite(BUZZER_PIN, HIGH);
  } else {
    digitalWrite(BUZZER_PIN, LOW);
  }
  delay(200);
}
```

Experiment 2 – Automatic Street Light (Remote Areas) using LDR

Objective: When we close the sensor (i.e., no light), the LED lights up and the buzzer rings.

Component: LDR – Light Dependent Resistor

- Made of cadmium sulphate.
- Pin: LDR_PIN = 7, LED_PIN = 8, BUZZER_PIN = 9.

Sample Code:

```
#define LDR_PIN   7
#define LED_PIN   8
#define BUZZER_PIN 9
void setup() {
  Serial.begin(9600);
  pinMode(LDR_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}
void loop() {
  int sensorValue = digitalRead(LDR_PIN);
  if (sensorValue == LOW) {
    Serial.println("Light");
    digitalWrite(LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
  } else {
    Serial.println("No light");
    digitalWrite(LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
  }
  delay(1000);
}
```

Relay:

- Has 6 terminals — Primary side (VCC, GND, IN) and Secondary side (Common, NO, NC).
- When HIGH → turns OFF automatically.
- When LOW → turns ON automatically.

Conclusion of Day 2: Day 2 of the Skill Lab provided students with a practical foundation in programming and interfacing using the Arduino Uno. Through hands-on exercises with sensors, relay, students gained confidence in applying theoretical concepts to real-world scenarios. The session not only strengthened their technical skills but also highlighted the relevance of microcontroller-based design. By fostering creativity and problem-solving.

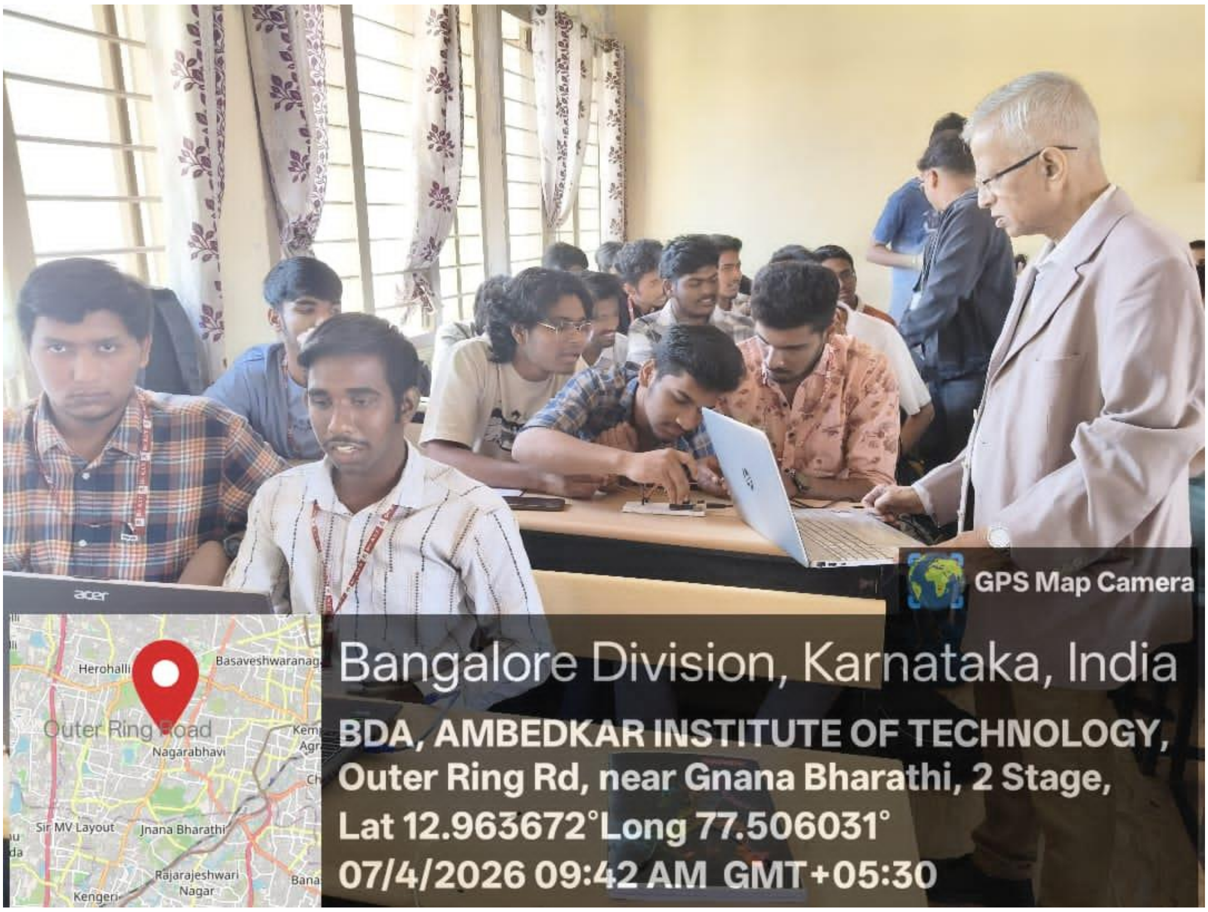

Fig 2: Day 2 Skill Lab

SKILL LAB DAY 3 REPORT

Organized by: Electronics and Instrumentation Engineering (EIE)

Speaker: Mr. P M Prabhu, Avinya Networks

Date: 08/04/2026

Day 3 of skill lab continued with practical experiment on Arduino board

The following sensors and output devices were used to demonstrate the working.

Sensors: tile sensor, LDR, Temperature and humidity sensors

Output devices: LED, Relay, Buzzer

Experiment 3 – Tile Sensor with LDR, Relay, LED & Buzzer

Objective: To design and implement a light-sensitive automation system using an LDR sensor interfaced with Arduino Uno, where variations in ambient light intensity trigger outputs such as LED illumination, buzzer alerts, and relay activation, thereby demonstrating the principles of sensor-based control and environmental monitoring.

Application: Level of machinery to be the same.

Pin Configuration:

- LDR_SENSOR: Pin 7
- LED_PIN: Pin 8
- BUZZER_PIN: Pin 9
- RELAY_PIN: Pin 6

Sample Code:

```
#define LDR_SENSOR 7
#define LED_PIN 8
#define BUZZER_PIN 9
#define RELAY_PIN 6
void setup() {
  Serial.begin(9600);
```

```
  pinMode(LDR_SENSOR, INPUT);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RELAY_PIN, OUTPUT);
}
void loop() {
  int sensorValue = digitalRead(LDR_SENSOR);
  if (sensorValue == LOW) {
    Serial.println("Light");
    digitalWrite(LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(RELAY_PIN, LOW);
  } else {
    Serial.println("No light");
    digitalWrite(LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(RELAY_PIN, HIGH);
  }
  delay(1000);
}
```

Experiment 4 – Analog Pin A2: Servo Motor Simulation

Objective: To measure and monitor environmental parameters such as temperature and humidity using the DHT11 sensor interfaced with Arduino Uno, thereby demonstrating real-time data acquisition and its applications in smart homes, weather monitoring, and industrial automation.

Analog pin A2 is specially designed and accepts DAA. The joystick is moved in the X-axis when the pointer is moved in that direction.

Key Concepts:

- VCC = 5V, GND for Joystick and ADC.

- VRX = A0, VRY = A1.
- Analog has A5 (4 low accepting) RDAA.

Servo Motor:

- A servo motor is a closed-loop system.
- The joystick output is on either side of the potentiometer.
- 0 to 5V: when we mean the joystick output is bk analog.
- 0 kind: 5V when we mean the output varies bk.
- When we move along the X-axis, the output changes.
- When the joystick is in the centre, the output voltage is half of VCC = 2.5V.
- DC has no position control; it runs and runs.
- A servo motor has a built-in position feedback (closed-loop system).
- 1 ms = −90°, 1.5 ms = 0°, 2 ms = +90°.
- Servo motor can only be controlled by 20 ms signal; only be controlled by 1 ms.

Sample Code:

```
#include <Servo.h>
Servo servoY;
void setup() {
  servoY.attach(8);
  Serial.begin(9600);
}
void loop() {
  int joystickX = A0;
  if (servoX is connected) {
    servoY.attach(8);
    Serial.println("Y: " + servoYPos);
  }
  int posX = analogRead(joystickX);
  int servoXPos = map(posX, 0, 1023, 0, 180);
  int servoYPos = map(posX, 0, 1023, 0, 180);
  servoY.write(servoYPos);
  Serial.println("X: " + String(servoXPos));
```

```
 Serial.println(servoYPos);
}
```

Conclusion of Day 3: The Skill Lab experiments on *Tile Sensor with LDR, Relay, LED & Buzzer* and *DHT11 Temperature & Humidity Sensor* provided students with valuable hands-on exposure to sensor interfacing and automation using Arduino Uno. The first experiment demonstrated how light intensity can be effectively monitored and used to trigger automated responses, while the second highlighted the importance of environmental data acquisition for real-time monitoring. Together, these exercises strengthened students' understanding of microcontroller-based design, encouraged practical problem-solving, The session concluded with students gaining both technical confidence and inspiration to pursue innovative projects in electronics and automation.

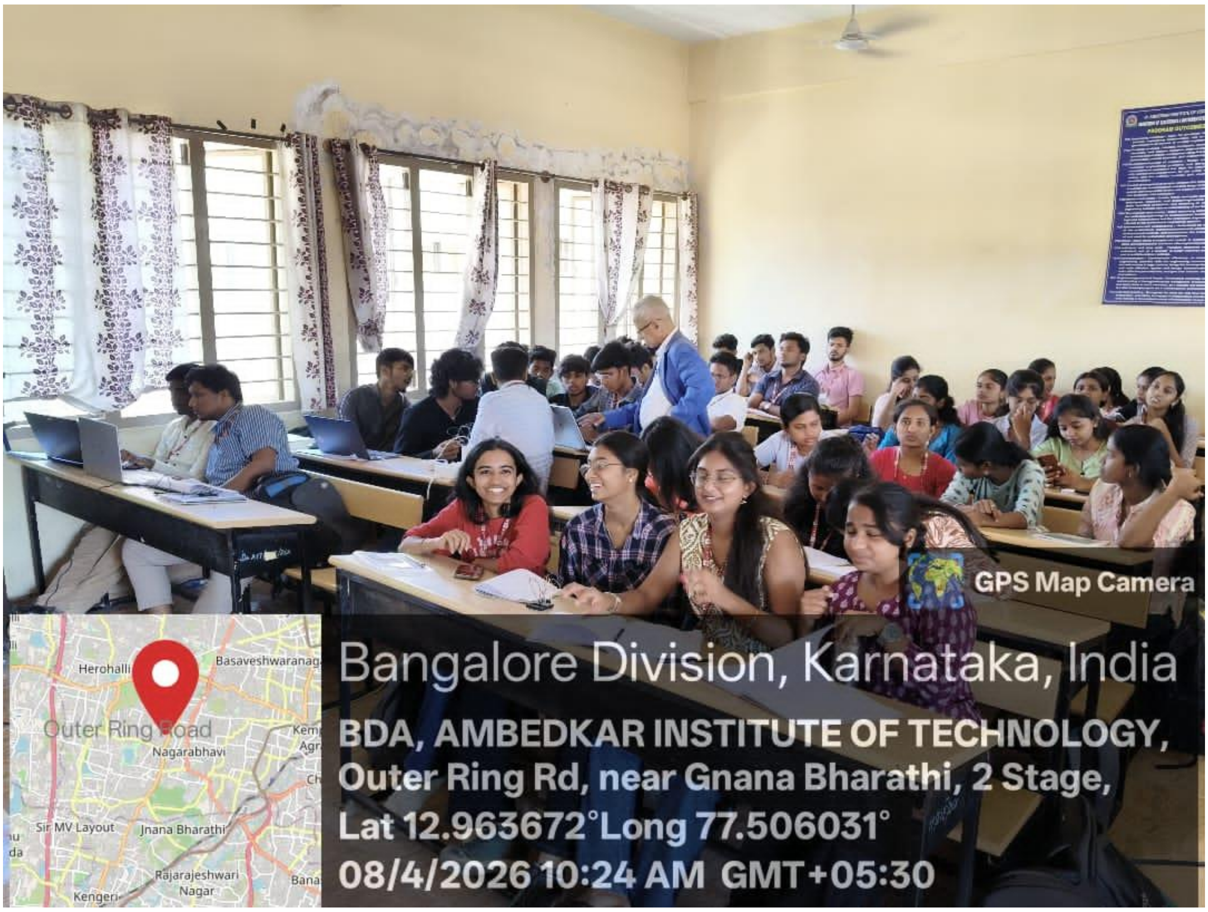

Fig 3: Day 3 Skill Lab

SKILL LAB DAY 4 REPORT

Organized by: Electronics and Instrumentation Engineering (EIE)

Speaker: Mr. P M Prabhu, Avinya Networks

Date: 09/04/2026

Day 4 of skill lab continued with ESP8266 Wi-Fi Module Architecture

Arduino UNO cannot connect to Wi-Fi natively. We connect the ESP8266-01 Wi-Fi module to enable wireless connectivity. The ESP8266 Wi-Fi Module is a low-cost, highly integrated SoC that provides full TCP/IP stack and microcontroller capability, making it ideal for IoT and wireless communication projects. Its architecture combines a Tensilica L106 32-bit processor, integrated Wi-Fi transceiver, memory, and multiple interfaces for sensors and peripherals.

ESP8266 Architecture (32-bit):

- Built-in USB-to-Serial converter: CH340.
- Holds buttons to put into Transmitter Logic (LQFP).
- Has SPI: connected to CH340 inside, provides full RST.
- Flash memory buttons to reset the buttons and minus with ISP antenna.
- Pin GPIO2: overlaid to the override of 21 (5 byte), makes external antenna.

Key Specs:

- Input voltage: 1V–12V.
- I2C: 1 I2C (8-bit) frameover.
- Wi-Fi (802.11 b/g/n) Frameover.
- Operating voltage: 2.5V (down to about clean 5).
- Power consumption: 15kA–400mA.
- 8 MB external flash.
- 128 KB internal RAM.
- SD to 160 MHz 16 cycle frequency.
- 2.4 GHz antenna.
- No Bluetooth.

- 1 SPI, 1 I2C, 1 UART.
- Wi-Fi (802.11 b/g/n) frameover.

Communication Protocols Covered:

SPI – Serial Peripheral Interface (Motorola):

- SCLK, MOSI, MISO, SS (SS1, SS2, SS3...).
- SPI has one master but one or more slaves.
- One bit of data is transferred in every clock cycle.
- For chip select, generally Pin 10 is used on Arduino UNO.
- SPI uses system clock for synchronising data transmission — Pin 13 of Arduino.
- SPI MOSI is output when master communicates with slave — Pin 11.
- SPI MISO is input when slave communicates with master — Pin 12.

I2C – Inter-Integrated Circuit:

- Combines the best features of SPI and UART.
- 128 devices can be addressed (7-bit addressing) using just 2 wires.
- Pins on I2C: SDA & SCL.
- SDA = Serial Data, SCL = Serial Clock.
- Master ↔ Slave communication.
- Increases the accuracy using I2C.

Feature	Details
CPU	Tensilica L106, 32-bit RISC, up to 160 MHz
Wi-Fi Standard	IEEE 802.11 b/g/n (2.4 GHz)
Flash Memory	External SPI flash (up to 16 MB)
RAM	160 KB (instruction + data)
Interfaces	UART, SPI, I²C, I²S, PWM, ADC
Operating Voltage	2.5 V – 3.6 V
Power Consumption	20 µA (deep sleep), ~80 mA (active Wi-Fi)
Package	QFN32, compact PCB footprint

Conclusion of Day 4: Day 4 of the Skill Lab offered participants a comprehensive understanding of the ESP8266 Wi-Fi Module and its architecture. The session highlighted the module's integrated processor, memory system, and wireless communication capabilities, emphasizing its role as a cornerstone in IoT development. By exploring its interfaces, power management features, and practical applications, students gained valuable insights into how ESP8266 enables real-time connectivity and smart automation.

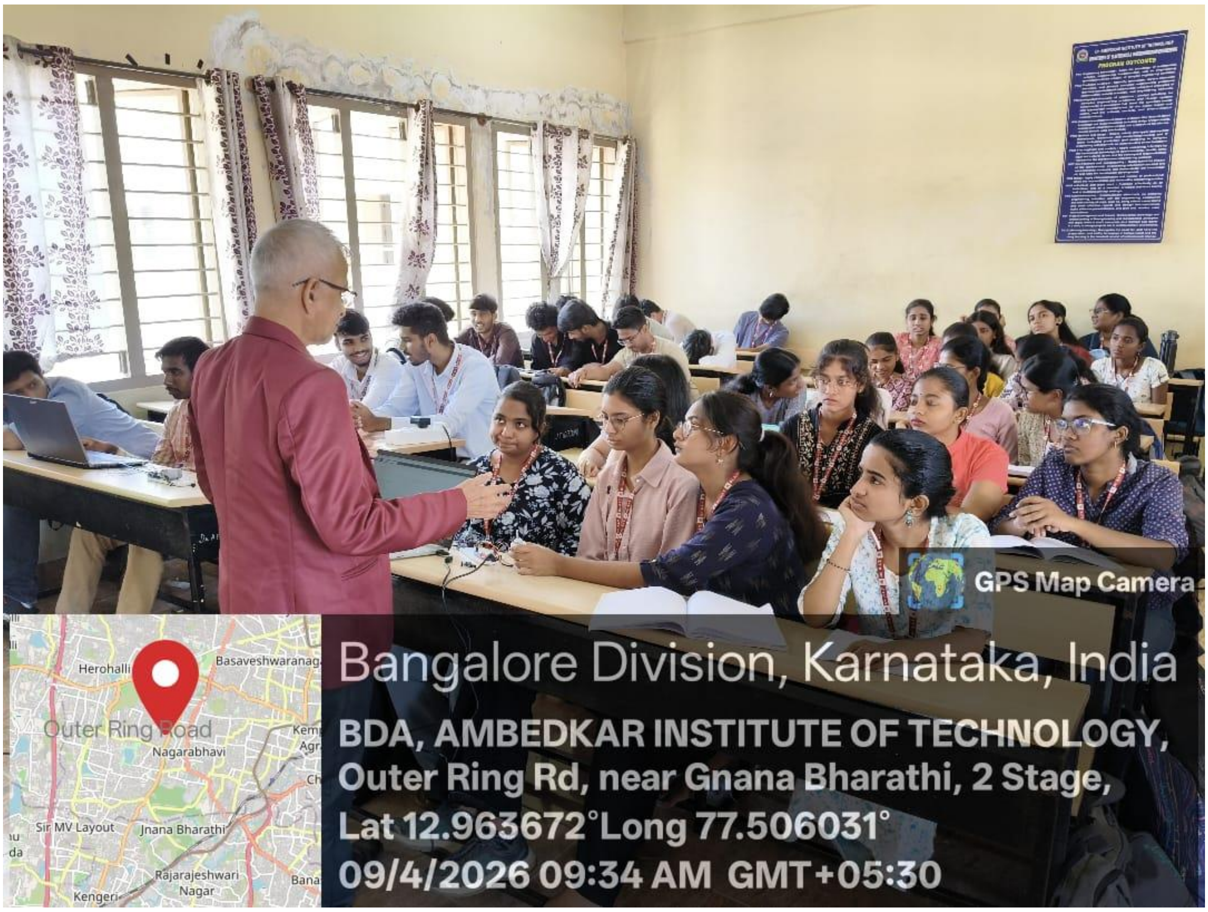

Fig 4: Day 4 Skill Lab

SKILL LAB DAY 5 REPORT

Organized by: Electronics and Instrumentation Engineering (EIE)

Speaker: Mr. P M Prabhu, Avinya Networks

Date: 10/04/2026

Day 5 of skill lab continued with practical sessions

Experiment 5 : Tilt Sensor with LCD Display:

- Uses LiquidCrystal I2C lcd(0x27, 16, 2).
- Pin: TILT_PIN = INPUT, LED_PIN = OUTPUT, BUZZER_PIN = OUTPUT.
- When tilt detected → LCD prints 'WARNING! Tilt Detected!', LED HIGH, Buzzer HIGH.
- Else → 'Normal Position', LED LOW, Buzzer LOW.

Inductive Proximity Sensor:

- At its heart is an electronic oscillator consisting of an inductive coil made of numerous turns of very fine wire.
- Has a capacitor for storing electrical charge and an energy source to provide electrical excitation.
- Metal cans containing juice are counted by the inductive proximity sensor as they pass on the conveyor belt.

DHT Sensor (DHT11, DHT22):

- Consists of a capacitive sensing element.
- Uses NTC Thermistor: Resistance increases as Temperature increases.
- Temperature has 1 threshold; Humidity has 2 thresholds.

Sample code:

```
#include <LiquidCrystal.h>

// Initialize the LCD (RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int tiltPin = 7;   // Tilt sensor connected to digital pin 7
int tiltState = 0;

void setup() {
  lcd.begin(16, 2);         // Set up the LCD's number of columns and rows
```

```
  pinMode(tiltPin, INPUT);   // Set tilt sensor pin as input
  lcd.print("Tilt Sensor Demo");
  delay(2000);
  lcd.clear();
}

void loop() {
  tiltState = digitalRead(tiltPin);  // Read tilt sensor state

  lcd.setCursor(0, 0);
  lcd.print("Tilt Status:     ");

  if (tiltState == HIGH) {
    lcd.setCursor(0, 1);
    lcd.print("Position: TILTED ");
  } else {
    lcd.setCursor(0, 1);
    lcd.print("Position: NORMAL ");
  }
  delay(500);  // Small delay for stability
}
```

Experiment 6: Arduino UNO Interfacing with DC Motor

Uses a Bump Sensor to control a brushed DC motor via H-Bridge.

Pin Configuration:

- LED: Pin 8
- Buzzer: Pin 9
- Bump Sensor: Pin 7 (OUT), GND–GND, VCC–5V
- ENA_PIN: Pin 3, IN1_PIN: Pin 4, IN2_PIN: Pin 5

DC Motor Theory:

- Brushed DC motor: electrical current through coils arranged in a fixed magnetic field.
- Current generates magnetic field in the coil → coil assembly rotates.

H-Bridge:

- Drives a DC motor from low-power control signals.
- Changes the direction of rotation.
- We need to change the polarities of voltage.

LOW	HIGH	Right
HIGH	LOW	Left
LOW	LOW	Stop
HIGH	HIGH	Stop

Note:

- On Analog we can read Digital, but on Digital we cannot read Analog.

Experiment 7 – Flame Sensor using Arduino UNO

Connections:

- Sensor: DC–7 pin, GND, VCC–5V.
- LED (+): Pin 8, (−): GND.
- Buzzer (+): Pin 9, (−): GND.

Key Components of Flame Sensor:

- YG1006 NPN Phototransistor.
- LM393 Comparator IC.
- Variable resistor.
- Power LED, Output LED.
- Range: 760–1100 nm wavelength.
- Long terminal: Emitter; Short terminal: Collector.

Using Display:

- SDA: A4, SCL: A5, VCC: 5V, GND: GND.

Experiment 8 – Door Sensor & Relay with LED Bulb

Arduino UNO interfacing with Door Sensor and Relay to control an LED bulb.

Pin Configuration:

- LED: (+)–Pin 8, (−)–GND
- Buzzer: (+)–Pin 9, (−)–GND
- Door Sensor: VCC–5V, GND, DO–Pin 7
- Bulb: IN–Pin 6, GND–GND, VCC–5V

Conclusion of Day 5: The day 5 of skill lab students successfully explored and implemented multiple sensor-based and actuator-based systems using the Arduino Uno microcontroller platform. Each experiment helped in understanding both fundamental electronics and practical embedded system design.

- The **flame sensor with Arduino UNO** demonstrated how fire detection systems work by sensing infrared light emitted by flames. This experiment highlighted the importance of safety systems and real-time monitoring.
- The **door sensor with relay and LED bulb** provided insight into automation and security applications. By integrating a relay module, we learned how low-power signals from the Arduino can control high-power devices like an LED bulb, simulating real-world smart home systems.
- The **DC motor interfacing with Arduino UNO** helped us understand motor control, including direction and speed basics. This experiment is essential for robotics and automation applications where movement control is required.
- The **tilt sensor with LCD display** showcased how orientation or motion changes can be detected and displayed in real time. It emphasized the use of sensors for condition monitoring and user feedback through displays.
- Overall, this lab enhanced students understanding of interfacing different sensors and actuators with Arduino, improved the coding and circuit design skills, and demonstrated how such systems are applied in real-world scenarios like safety systems, home automation, and robotics.

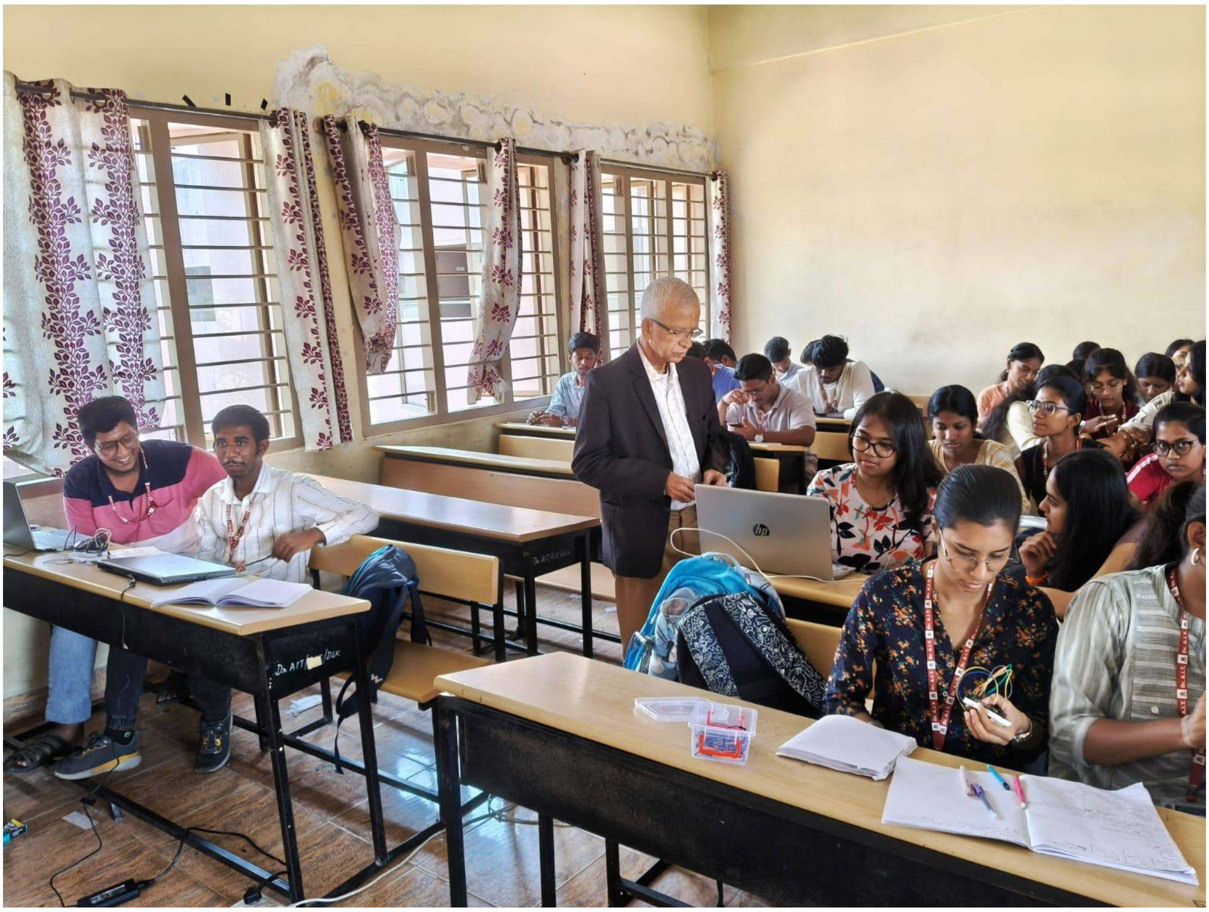

Fig 5: Day 5 Skill Lab

SKILL LAB DAY 6 REPORT

Organized by: Electronics and Instrumentation Engineering (EIE)

Speaker: Mr. P M Prabhu, Avinya Networks

Date: 11/04/2026

The final day of skill lab that is 6th day the experts covered ESP32 Architecture

The ESP32 is a powerful, low-cost microcontroller developed by Expressive Systems, designed for IoT (Internet of Things) and embedded applications. Its architecture combines high performance, wireless connectivity, and multiple peripherals on a single chip. The ESP32 is a more advanced microcontroller compared to the ESP8266, offering dual-core processing, Bluetooth, and extensive peripherals

1 Key Specifications

- Wi-Fi: 2.4 GHz up to 150 Mbits/s.
- Architecture: 32-bit.
- RAM: 512 KB.
- Built-in buttons: RESET & BOOT.
- Built-in LEDs: Blue LED connected to GPIO2; Red LED indicating power.
- USB to UART Bridge: CP2102.
- Comes with 36 pins — 18 on each side.

2 Peripherals

- Capacitive Touch
- ADC (Analog to Digital Converter)
- DAC (Digital to Analog Converter)
- I2C
- UART (2 interfaces)
- CAN 2.0 (Controller Area Network)
- SPI (Three SPI, one I2C interface)
- I2S (Integrated Inter-IC Sound)
- RMII (Reduced Media Independent Interface)
- PWM (25 PWM pins — used for power regulation and speed control)
- 2 UART interfaces
- 2 8-bit DAC to generate true analog
- 9 GPIOs

3 Important Notes

- Pins GPIO34, GPIO35, GPIO36(VP) and GPIO39(VN) cannot be configured as outputs.
- The ADC2 pins cannot be used when Wi-Fi is enabled.
- Accuracy: 0.02% — To increase accuracy, use I2C.

- A camera is present in the case of ESP32. BOOT and RESET/EN buttons put the board in flashing mode or reset it.
- PWM used for power regulation and speed control.

4. Summary of Experiments

Exp No.	Title	Key Components	Key Learning
Exp 1	Digital Output – LED & Buzzer	LED, Buzzer, Arduino	digitalRead/Write, pinMode
Exp 2	Automatic Street Light	LDR, LED, Buzzer	Light sensing, conditional output
Exp 3	Tile Sensor with Relay	LDR, Relay, LED, Buzzer	Relay control logic
Exp 4	Servo Motor Simulation	Joystick, Servo	Analog read, servo mapping
Exp 5	Ultrasonic + LCD Display	HC-SR04, TM1637	Distance measurement, display
Exp 6	DHT11 Temp & Humidity	DHT11, LED, Buzzer	Threshold-based alerting
Exp 7	ESP8266 Wi-Fi Module	ESP8266, SPI, I2C	Communication protocols
Exp 8	Tilt & Inductive Sensor	Tilt, Inductive, LCD	Proximity & tilt detection

Followed the technical session at the end of the last day feedback was taken from the students about their learning experience. The day concluded with vote of thanks.

Conclusion of Day 6: The ESP32 architecture provides a perfect balance of **processing power, connectivity, and versatility**. Its dual-core design, built-in Wi-Fi and Bluetooth, and rich peripheral set make it highly suitable for applications such as IoT systems, smart homes, robotics, and industrial automation.

Key Takeaways a& Learning

- Mastered the physical architecture of Arduino UNO and its core components.
- Understood ADC resolution (10-bit) and PWM (8-bit) signal generation.
- Implemented LDR, DHT11, Ultrasonic, Tilt, Inductive, Flame, and Door sensors.
- Learned communication protocols: SPI, I2C, UART, and their Arduino pin mappings.
- Interfaced display modules: TM1637 (4-digit) and I2C LCD (16×2).
- Controlled actuators: LED, Buzzer, Relay, Servo Motor, and DC Motor.
- Studied ESP8266 and ESP32 Wi-Fi module architectures and their GPIO limitations.
- Applied H-Bridge logic for DC motor direction control.
- Understood the role of LDO regulators and crystal oscillators on the Arduino board.
- Practiced complete Arduino programs integrating sensors, actuators, and displays.

The study of the Arduino Uno and ESP32 platforms provided a strong foundation in embedded systems and IoT development. Through various experiments involving sensors (flame, tilt, and door) and actuators (relay, LED bulb, and DC motor), we gained practical knowledge of interfacing hardware components with microcontrollers.

The Arduino Uno helped in understanding basic concepts such as digital and analog input/output, simple control logic, and circuit design. On the other hand, the ESP32 expanded these concepts by introducing advanced features like dual-core processing, built-in Wi-Fi, and Bluetooth, enabling the development of smarter and connected systems.

Overall, this lab enhanced our skills in programming, hardware interfacing, and system integration. It also demonstrated how these technologies are applied in real-world scenarios such as safety systems, home automation, and robotics, thereby preparing us for more advanced projects in embedded systems and IoT.

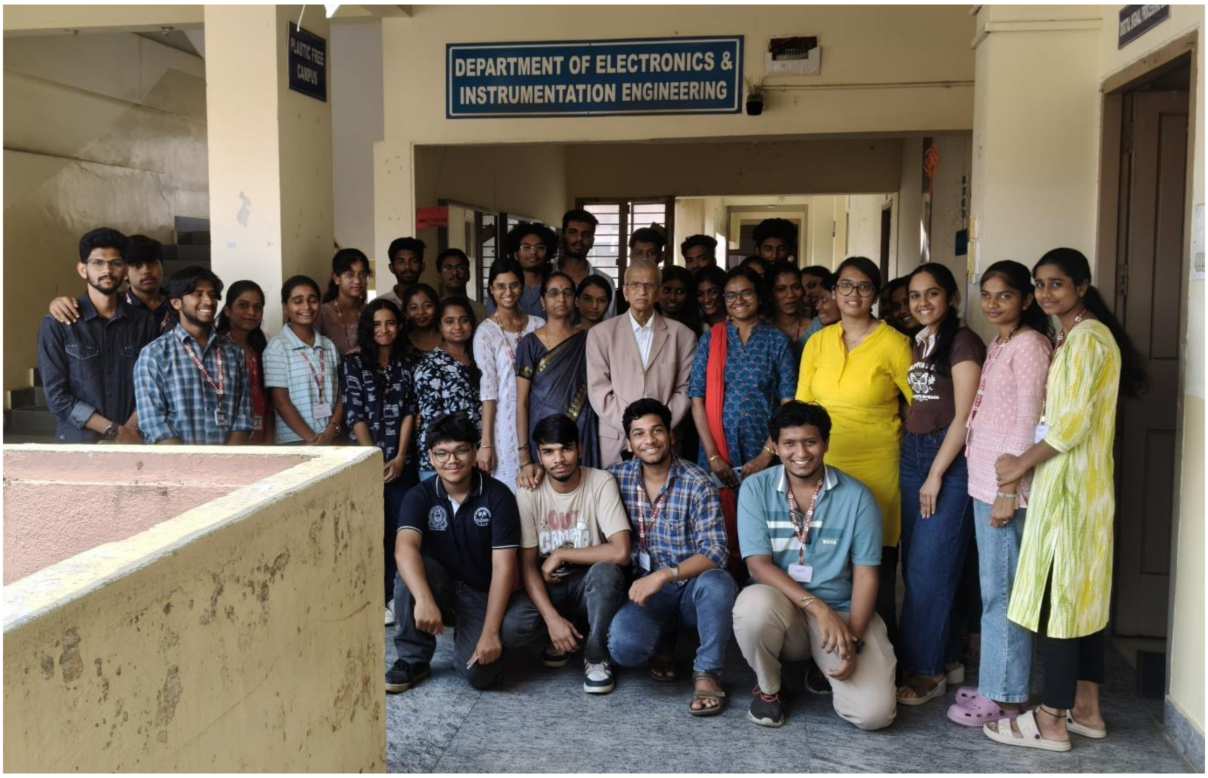

Fig 6: Day 6 Skill Lab