DAY 1

Introduction to Electronics & Components

The first day introduced us to the fascinating world of electronics, beginning with the most basic yet essential components: resistors, capacitors, diodes, and transistors. We learned how each component has a unique role in controlling voltage, current, and signal flow within a circuit.

In the practical session, we identified real-world components and practiced placing them into breadboards. This exercise helped us move from abstract theory to tangible understanding. Holding the components in hand and seeing them work in a simple circuit gave us confidence. We realized how these tiny parts are the core of every electronic system, from smartphones to satellites. Our journey into electronics began with a comprehensive introduction to the essential building blocks of electronic systems. We studied the theory behind basic components such as **resistors, capacitors, diodes, and transistors**. Each of these has a unique function — resistors limit current, capacitors store and release energy, diodes control current direction, and transistors act as amplifiers or switches.

Understanding these components was crucial because they appear in almost every circuit. During the practical session, we physically identified each part and placed them on a **breadboard**, a tool for prototyping without soldering. It was the first time many of us held real components and saw how they connected to form a functioning circuit. This hands-on experience helped bridge the gap between theory and real-world application. It marked the beginning of thinking like an engineer — not just knowing what components do, but how and where to use them.

DAY 2

Circuit Laws (Ohm's & Kirchhoff's Laws)

On Day 2, we dove into the rules that govern electric circuits: Ohm's Law and Kirchhoff's Laws. Ohm's Law taught us the relationship between voltage, current, and resistance, while Kirchhoff's Current and Voltage Laws helped us analyze how electricity behaves at junctions and loops in a circuit.

Using software like Proteus or Multisim, we simulated these laws in action. Afterward, we built physical circuits to verify our results. It was satisfying to see that theory and practice matched. These principles now feel like tools we can use to solve any circuit problem with logic and confidence. On the second day, we explored the fundamental **laws of electrical circuits**: **Ohm's Law** and **Kirchhoff's Laws**. Ohm's Law taught us how voltage, current, and resistance are mathematically related, while Kirchhoff's Laws helped us analyze more complex circuits by applying rules for current and voltage in loops and junctions.

We didn't just learn equations — we applied them. Using software like **Proteus** or **Multisim**, we simulated circuits and calculated expected values, then built real versions to verify those calculations. It was fascinating to see how mathematical predictions matched physical results. This day built the foundation for troubleshooting and designing circuits, making us more confident in our analytical skills.

DAY 3

Semiconductor Basics

Semiconductors were the focus of Day 3 — particularly diodes, BJTs (Bipolar Junction Transistors), and MOSFETs (Metal Oxide Semiconductor Field Effect Transistors). We explored how semiconductors are neither pure conductors nor insulators, but materials whose conductivity can be controlled — which makes them ideal for creating switches and amplifiers.

In the lab, we used multimeters to test these components. Learning how to identify good versus faulty parts through testing was a valuable skill. It felt like detective work — searching for clues in a circuit. Understanding semiconductors opened our eyes to how modern electronics, like processors and memory chips, actually work. Semiconductors are at the heart of modern electronics, and Day 3 focused on understanding them. We studied **diodes**, **BJTs (Bipolar Junction Transistors)**, and **MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors)** — devices that control the flow of electricity in sophisticated ways.

We learned how a diode only allows current in one direction, how BJTs can amplify signals, and how MOSFETs act like electronic switches in computers and power systems. In the lab, we used **multimeters** to test and identify faulty or functional semiconductors. Handling these components gave us insight into how logic circuits, amplifiers, and even CPUs work at a microscopic level. It was the beginning of appreciating the invisible complexity behind everyday devices.

DAY 4

Digital Electronics (Logic Gates)

Today was all about logic — quite literally! We studied the basic logic gates that make up digital electronics: AND, OR, NOT, NAND, and NOR. These gates are the core of computers and all digital systems, performing decision-making tasks based on binary inputs (0s and 1s).

Using TinkerCAD, we simulated logic circuits virtually, then implemented them on breadboards. Seeing LEDs light up based on our logic combinations was like solving puzzles with wires. We now understand how even the most advanced computers operate based on these simple logic principles, repeated at an enormous. Digital electronics form the basis of all computing systems, and this day introduced us to **logic gates** — the building blocks of digital circuits. We explored **AND, OR, NOT, NAND,** and **NOR** gates, learning how they process binary inputs to produce specific outputs.

Using **TinkerCAD** simulations, we designed and tested simple logic circuits virtually. Later, we implemented these gates on breadboards using integrated circuits. It was amazing to realize that complex decisions in computers start with such simple elements. This day gave us a new way of thinking — in binary logic — and showed us how everything from calculators to robots makes decisions.

DAY 5

Soldering & PCB Basics

The final day of Week 1 introduced us to the physical creation of lasting electronic systems — soldering and PCB (Printed Circuit Board) design. We learned how PCBs are designed using software like KiCad or Eagle, following a workflow that includes schematic design, layout, and preparing files for manufacturing.

We then practiced soldering components onto a board. It required patience, precision, and a steady hand. This hands-on session gave us a deep appreciation for hardware engineers and technicians. By the end of the day, we had turned theory into something physical — a working circuit on a board. It was an empowering end to a foundational week. Soldering is a vital skill for every electronics enthusiast, and Day 5 introduced us to it alongside the basics of **PCB (Printed Circuit Board)** design. In the theory session, we explored the workflow of PCB design — from schematics to layout — using tools like **KiCad** or **Eagle**.

During the practical, we learned how to safely handle a soldering iron and how to join components permanently to a board. This was one of the most hands-on days, and while mistakes were made (like too much solder or cold joints), we learned how to correct them. Soldering taught us patience, precision, and the importance of physical craftsmanship in electronics. By the end, we had created small, functional circuits — not just in theory, but in solid, permanent form.

DAY 6

Arduino/Raspberry Pi Introduction

Week 2 began with a transition from basic electronics to embedded systems. We were introduced to microcontrollers and single-board computers, specifically Arduino and Raspberry Pi. These platforms allow us to combine software and hardware to build smart, automated systems.

Our first project was simple but exciting: the "Blink LED" program. We wrote code to make an LED blink on and off. Uploading that first sketch and seeing the LED respond was incredibly satisfying. It was a powerful moment—realizing that lines of code could control physical components. This marked our first step into programming the real world. Week 2 marked a major shift from passive components and basic circuits to smart systems using **microcontrollers** and **embedded computing platforms**. On Day 6, we were introduced to two powerful tools: **Arduino** and **Raspberry Pi**.

We began by understanding the differences between them. Arduino is ideal for hardware-level control with real-time inputs and outputs, while Raspberry Pi runs a full operating system, making it more like a mini-computer. For the practical task, we used the Arduino Uno to create our first embedded program: the classic **"Blink LED"** sketch. It was a simple program, but for many of us, it was a magical moment to see code interact with the physical world. We wrote the code in the Arduino IDE, uploaded it, and watched the LED blink — a symbol of our journey into embedded systems.

DAY 7

Sensor Interfacing

On Day 7, we focused on sensors — the senses of an electronic system. We learned to interface Light Dependent Resistors (LDRs), ultrasonic sensors, and Infrared (IR) sensors with Arduino boards. These components allow devices to detect light, distance, and motion.

Through hands-on projects, we built circuits that react to their environment. For example, the LDR circuit could turn on a light when it gets dark. This taught us how electronics can gather data and react in real time, just like humans. It was the beginning of understanding how to create interactive systems like smart lights or security systems. On Day 7, we learned how electronic systems gather information from the environment using **sensors**. We interfaced different types of sensors with the Arduino — including **LDRs (Light Dependent Resistors)**, **ultrasonic sensors**, and **IR (Infrared) sensors**.

We wrote Arduino code to read values from these sensors and display the output on serial monitors or use it to control devices like LEDs or buzzers. For instance, the ultrasonic sensor measured distance, triggering an alert if something got too close — a basic but functional model of an obstacle detector.

This session showed us how systems can "sense" their surroundings. It was our first step into building **autonomous** devices that react to light, distance, and motion. It also helped us understand how to convert physical parameters (like light or distance) into electrical signals that can be processed by microcontrollers. In essence, we gave our circuits "eyes" and "ears."

DAY 8

Motor Control

Motors bring movement to electronics, and Day 8 was about controlling them. Using the L293D motor driver, we explored how to manage both DC and servo motors. The concept of direction, speed control, and angle positioning was introduced.

We wrote Arduino code to rotate motors, reverse direction, and control speed using PWM (Pulse Width Modulation). Seeing a servo precisely move to a position based on our commands felt like unlocking another level of control. This day was key for anyone dreaming of building robots, drones, or automated machines. We learned the difference between continuous rotation and precise angular control, and how servo motors are used in robotic arms and automation systems. The practical session involved writing Arduino programs to control motor speed and direction using Pulse Width Modulation (PWM). We built a basic motor control setup where a robot wheel could move forward, reverse, or stop based on programmed logic.

This was a major turning point. For the first time, we saw mechanical action controlled by electronics and code. We realized that motors are what bring electronics to life — allowing our devices not just to blink or beep, but to **move and interact physically** with the world.

We learned the difference between continuous rotation and precise angular control, and how servo motors are used in robotic arms and automation systems. The practical session involved writing Arduino programs to control motor speed and direction using Pulse Width Modulation (PWM). We built a basic motor control setup where a robot wheel could move forward, reverse, or stop based on programmed logic.This was a major turning point. For the first time, we saw mechanical action controlled by electronics and code. We realized that motors are what bring electronics to life — allowing our devices not just to blink or beep, but to **move and interact physically** with the world.

DAY 9

Wireless Communication (IoT)

Wireless communication was the theme of Day 9, and it brought us closer to the concept of the Internet of Things (IoT). Using ESP8266 Wi-Fi modules and Bluetooth communication, we connected our devices to each other and to the internet.

We tested sending data from sensors to mobile devices or cloud platforms. This allowed us to build simple systems like remote-controlled LEDs or temperature monitors that send updates online. It was exciting to think about how everyday devices in smart homes or cities work—and now, we had the knowledge to create something similar. On Day 9, we took our projects to the next level with **wireless communication**, a key part of the **Internet of Things (IoT)**. We worked with **ESP8266 Wi-Fi modules** and Bluetooth setups to send and receive data wirelessly.

In our demo, we used the ESP8266 to send sensor data to a mobile phone or a web page, opening up the idea of remote monitoring. We learned about the concepts of IP addresses, Wi-Fi networks, and simple web-based interfaces. Some teams built systems where a smartphone could turn an LED on or off using Bluetooth commands — small, but powerful steps toward building smart devices.

This was one of the most exciting days, showing how electronics are no longer limited by wires. We learned how modern devices like smart bulbs, security systems, or fitness trackers communicate with each other and the cloud. This is the future — and we were building it.

DAY 10

Mini Project

Introduction

Day 10 served as the pivotal midpoint of the three-week program, challenging us to synthesize every skill acquired thus far—circuit theory, embedded coding, sensor integration, motor control, and even rudimentary IoT connectivity. Unlike previous guided labs, the **Mini Project** transformed the classroom into a small R&D incubator where creativity, teamwork, and problem-solving reigned.

Project Selection & Planning

In the morning briefing, mentors offered two broad themes:

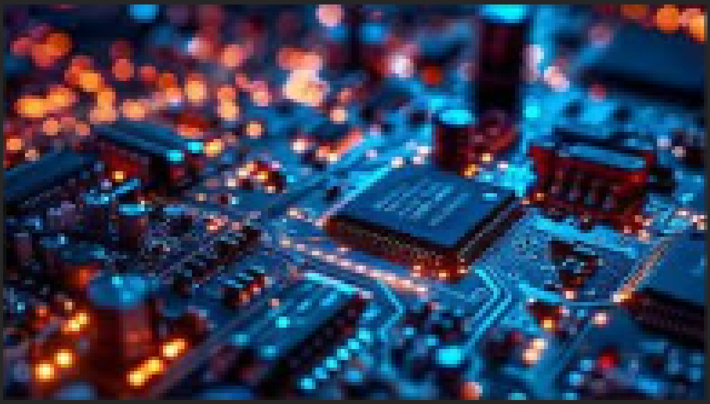
1. **Obstacle-Avoidance Robot**
2. **Smart-Home Demonstrator**

Teams of four to five students conducted a fast-paced design sprint—brainstorming concepts, defining functional requirements, drafting Gantt charts, and assigning roles (hardware lead, firmware lead, documentation, and tester). Our group opted for the robot, aiming to navigate a 2 × 2 m maze autonomously within two minutes.

We mapped out a **system-level block diagram:**

- Sensor suite – dual HC-SR04 ultrasonic modules (front & side)
- Control core – Arduino Nano (ATmega328P @ 16 MHz)
- Actuation – two DC gear motors driven by an L293D H-bridge
- Power – 2-cell 7.4 V Li-ion pack + 5 V buck regulator

- **Feedback** – RGB LED status indicator, piezo buzzer for error tones



Hardware Implementation

By midday, breadboards and jumper wires sprawled across benches. We laser-cut an acrylic chassis, mounting the motors and a swivel caster for balance. Emphasis was placed on **wiring hygiene**—twisted-pair signal lines and decoupling capacitors soldered directly across motor leads to quell EMI. A 3-A polyfuse safeguarded against stall currents.

Firmware Architecture

Coding followed a **finite-state-machine (FSM)** paradigm:

- **IDLE** – Await start button; calibrate sensors.
- **FORWARD** – Advance while distance > 20 cm.
- **AVOID_LEFT / AVOID_RIGHT** – Execute coordinated turn based on side-sensor differential.
- **FINISHED** – Sound victory melody upon maze completion.

Each loop iteration spanned ~25 ms, ensuring real-time responsiveness. We leveraged the **NewPing** Arduino library for non-blocking ultrasonic reads, freeing CPU cycles for motor PWM updates. A watchdog timer (WDT) reset the MCU if the main loop exceeded 250 ms—a safeguard learned from Day 8's motor stalls.

Testing & Iteration

Initial trials exposed flaws: sonar reflections off glossy floor tiles produced false positives, causing erratic reversals. We mitigated this by averaging three consecutive readings and implementing a median filter.

DAY 11

Power Electronics

We started Week 3 by diving into **Power Electronics**, a crucial field for modern energy systems. This day focused on devices like **rectifiers**, which convert AC to DC, and **Switch Mode Power Supplies (SMPS)**, which are used in everything from phone chargers to laptops.

Through a mix of theory and demonstrations, we saw how power is managed efficiently in electronics. We examined real circuits and observed how filtering and regulation work to maintain stable voltage outputs. It was eye-opening to understand what goes on inside the chargers and power adapters we use every day — and how much engineering ensures safety and efficiency. On Day 11, we ventured into the field of **Power Electronics**, a discipline that deals with converting and controlling electrical power efficiently. We explored circuits like **rectifiers** (which convert AC to DC) and **Switch-Mode Power Supplies (SMPS)**, which are used in everything from phone chargers to industrial equipment.

Through theoretical sessions and demonstrations, we understood how these systems are crucial in reducing energy loss and ensuring safe, reliable power delivery. We saw the difference between linear regulators and switching regulators, and how heat, efficiency, and voltage control are managed in real devices.

The live demos gave us a clearer picture of how power electronics are everywhere — inside TVs, computers, electric vehicles, and solar inverters. It was both technical and eye-opening, showing how even the act of powering our devices involves engineere systems. This session deepened our appreciation of energy managemen

DAY 12

IoT & Industry

This day took us into the future. We learned about the **Internet of Things (IoT)** and Industry 4.0 — the ongoing automation and data exchange in modern manufacturing and smart environments. The case study focused on **cloud data logging**, where we used platforms like **ThingSpeak** and **Blynk** to send sensor data to the internet.

We connected sensors to microcontrollers and used Wi-Fi modules to upload real-time data like temperature or light levels to the cloud, where it could be monitored from anywhere. This is the same concept behind smart agriculture, home automation, and health monitoring systems.

Learning about IoT made us realize how electronics now stretch beyond physical circuits — into data, networks, and smart infrastructure. Industry 4.0, with concepts like predictive maintenance, remote monitoring, and AI-driven control, gave us a glimpse of what jobs and systems of the future will look like. It was both technical and philosophical — about connectivity, efficiency, and global integration.

Introduction

Industry4.0 represents the fusion of cyber-physical systems, big data, and cloud analytics. Day 12 positioned us at that intersection, showing how humble microcontrollers become gateways to global information networks.

Theory: From MQTT to Digital Twins

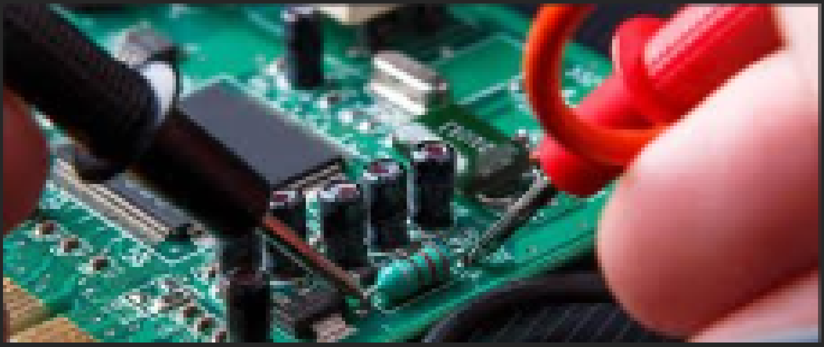
We mapped the IoT stack: sensors → microcontroller → communication layer → cloud → application dashboard. Emphasis was placed on the **MQTT protocol**, prized for its low overhead (CONNECT, PUBLISH, SUBSCRIBE cycles) and "last-will" messages that enhance reliability. We contrasted it with HTTP REST calls, highlighting latency and bandwidth trade-offs.**Case Study & Cloud Data Logging**

Working in teams, we configured NodeMCU boards (ESP8266) to read temperature and humidity via a DHT22 sensor, publishing data every 15seconds to **ThingSpeak**. On the dashboard we built real-time graphs and set threshold alerts that emailed us if humidity exceeded 70%. A parallel demo used **Blynk** to create a smartphone interface with virtual LEDs and sliders, proving bidirectional control.

We also discussed **digital twins**—virtual replicas of physical assets. Using a simulated conveyor motor, we streamed RPM data and plotted predictive maintenance trends, illustrating how factories minimize downtime.

Reflection

The exercise made clear that embedded engineers must straddle hardware, firmware, and cloud services. Security, too, loomed large: we enabled TLS and unique API keys, appreciating that every connected device can be an attack vector. By day's end, "Industry 4.0" was no longer a buzzword but a tangible workflow we had implemented ourselves.



DAY 13

PCB Fabrication & Troubleshooting

Introduction

After earlier dabbling in PCB design, Day 13 immersed us in fabrication realities and the detective work of troubleshooting. Seeing manufacturing up close revealed why design-for-manufacture (DFM) rules exist.

Fab-Lab Visit

A guided tour (video for remote students) walked us through UV photo-plotting, chemical etching, CNC drilling, and automated optical inspection (AOI). We noted tolerance limits: 0.2 mm minimum trace width, 0.4 mm drill size, 6 mil solder-mask clearance. Watching a pick-and-place machine deposit 18,000 components per hour was mesmerizing.

Advanced Soldering & Rework

Back in the lab we graduated from through-hole to 0805 SMD parts, learning the "drag-solder" technique with flux and solder wick. Deliberate faults were introduced—lifted pads, solder bridges, missing decoupling capacitors—and we practiced systematic troubleshooting:

1. **Visual inspection** under 10× magnification.
2. **Continuity tests** to locate open circuits.
3. **Power-rail sanity check** before applying full voltage.
4. **Signal tracing** with an oscilloscope.

Reflection

Troubleshooting cultivated patience and logical reasoning. It reminded us that even perfect schematics can fail if layout, assembly, or component quality is poor. We left respecting the craftsmanship and rigor that turn a schematic into a reliable product.

improvements. Judges scored innovation, technical depth, and presentation clarity.

Assessment & Feedback

Instructors highlighted excellence but also common pitfalls: undocumented code, lack of EMI shielding, and absent watchdog timers. Constructive critiques framed these as growth opportunities rather than failures.

Certification & Reflection

Receiving certificates capped three intense weeks, yet the true reward was intangible: newfound confidence, an engineer's mindset, and camaraderie forged through late-night debugging sessions. As we packed up breadboards and laptops, the blinking LEDs on our projects felt symbolic—a series of green lights pointing toward future explorations in electronics and embedded systems.

DAY 14

Guest Lecture

Speaker Highlights

Our guest, Dr. Elena Petrescu—lead firmware engineer at a multinational automotive supplier—shared her career arc from student projects to managing over-the-air (OTA) updates for electric-vehicle ECUs. Key takeaways:

- **Lifelong learning:** new MCU families and toolchains emerge yearly.
- **Cross-disciplinary teams:** firmware, hardware, mechanical, and data-science units collaborate daily.
- **Regulatory landscape:** ISO 26262 (functional safety) and cybersecurity standards are redefining design cycles.

Interactive Segment

We presented our mini-projects for critique. Dr. Petrescu probed our design choices, emphasizing robustness: "What happens if your ultrasonic sensor misfires? Fail-safe states matter." Her constructive feedback reframed our thinking from "Does it work?" to "Will it *always* work under real-world stress?"

Career Guidance

She recommended contributing to open-source RTOS projects, mastering version control (Git), and cultivating soft skills—communication often differentiates good engineers from great ones. The session transformed abstract career dreams into actionable plans.

DAY 15

Project Presentation

The final day was a celebration of everything we had learned and built. Each team presented their **mini project**, explaining the design, functionality, challenges, and improvements they had made. From obstacle-avoiding robots to smart homes, each project reflected hard work, creativity, and teamwork.

It wasn't just about showing a working circuit — it was about explaining **why and how** it worked, and how we approached solving real problems. We received feedback from instructors and peers, which made the experience feel professional and meaningful.

After the presentations, we received **certificates of completion** — but more importantly, we walked away with knowledge, confidence, and inspiration. This wasn't just a 3-week program. It was a launchpad — into electronics, engineering, and innovation.

The Showcase

Day 15 was equal parts science fair and celebration. Teams unveiled projects:

- **Obstacle-Avoidance Rover:** ultrasonic sensing, PID motor control, real-time telemetry via Bluetooth.
- **Smart Home Tabletop Model:** ESP32 gateway, DHT22 climate data, relay-driven lights, and a Blynk dashboard.
- **Energy-Harvesting Weather Station:** solar-powered, LoRa uplink to a campus gateway, Grafana visualization.

Each presentation covered problem statement, system architecture (block diagram + PCB art), firmware flowcharts, test results, and future