Erasmus

Day 1 Danca Tommaso

1. Resistors

Function: Limit current flow, divide voltage.

Key Concept: Ohm's Law V=IRV = IRV=IR

Types: Fixed, variable (potentiometers)

Color Code: Learn how to read resistor values.

2. Capacitors

Function: Store and release electrical energy.

Key Concepts: Capacitance (Farads), Charging/Discharging curves.

Types: Ceramic, Electrolytic, Tantalum

Applications: Filtering, timing circuits.

3. Diodes

Function: Allow current in one direction only.

Types: Standard, Zener, LED

Key Concepts: Forward voltage drop, reverse breakdown.

Applications: Rectifiers, voltage regulation.

4. Transistors

Function: Act as switches or amplifiers.

Types: Bipolar Junction Transistor (BJT), Field Effect Transistor (FET)

Key Concepts: Base/Collector/Emitter (BJT), Gate/Source/Drain (FET)

Applications: Amplifiers, digital switch

Erasmus

Day 2 Danca Tommaso

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

Theory: Circuit analysis techniques make use of fundamental laws like Ohm's Law and Kirchhoff's Laws to understand and predict the behavior of electrical circuits.

- **Ohm's Law**: Describes the relationship between voltage (V), current (I), and resistance (R).
    - Formula: $V = I \times R$
- **Kirchhoff's Laws**:
    - **Kirchhoff's Current Law (KCL)**: The total current entering a junction equals the total current leaving the junction.
    - **Kirchhoff's Voltage Law (KVL)**: The total voltage around any closed loop in a circuit is zero.

Applications:

- Solving for unknown values in electric circuits
- Analyzing series and parallel circuits
- Designing and troubleshooting circuit systems

Erasmus

Day 3 Danca Tommaso

Semiconductor Basics

Theory: Semiconductors are materials with electrical conductivity between conductors and insulators. They form the foundation of modern electronics. Key semiconductor devices include:

- **Diodes**: Allow current to flow in one direction only, used for rectification.
- **BJTs (Bipolar Junction Transistors)**: Current-controlled devices used for amplification and switching.
- **MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors)**: Voltage-controlled devices used in digital and analog circuits for switching and amplification.

Applications:

- Power regulation and conversion
- Signal amplification
- Switching and digital logic circuits
- Microprocessors and memory devices

Erasmus

Day 4 Danca Tommaso

Digital Electronics (Logic Gates) – Theory

Logic gates are the core components of digital circuits. They take binary inputs (0s and 1s) and produce a binary output based on logical functions.

Basic Logic Gates:

- **AND Gate**
  - Outputs 1 only if *all* inputs are 1.
  - **Expression**: A • B (A AND B)
- **OR Gate**
  - Outputs 1 if *any* input is 1.
  - **Expression**: A + B (A OR B)
- **NOT Gate**
  - Inverts the input (i.e., `0` becomes `1`, `1` becomes `0`).
  - **Expression**: ¬A or A'
- **NAND Gate**
  - Outputs `0` only if *all* inputs are `1`.
  - It's an AND gate followed by a NOT.
  - **Expression**: ¬(A • B)
- **NOR Gate**
  - Outputs `1` only if *all* inputs are `0`.
  - It's an OR gate followed by a NOT.
  - **Expression**: ¬(A + B)

Erasmus

Day 5 Danca Tommaso

Soldering & PCB Basics

Theory: PCB Design Workflow

PCB (Printed Circuit Board) design is a step-by-step process used to create a physical board that connects and supports electronic components using conductive tracks.

PCB Design Workflow:

1. **Schematic Design**
    a. Draw the circuit using electronic symbols in a schematic editor.
    b. Connect components logically (resistors, ICs, capacitors, etc.).
    c. Use design software like **Eagle, KiCad, Altium Designer**, or **EasyEDA**.
2. **Component Selection**
    a. Choose real-world components for each part of the schematic.
    b. Define part numbers, packages, values (e.g., 1kΩ resistor).
3. **Electrical Rule Check (ERC)**
    a. Run checks for errors like unconnected pins or incorrect connections.
4. **PCB Layout**
    a. Convert schematic to PCB format.
    b. Place components on a board outline.
    c. Route the copper traces to connect components.
5. **Design Rule Check (DRC)**
    a. Ensures minimum spacing, track width, and clearance rules are followed.
6. **Gerber File Generation**
    a. Export files that define copper layers, silkscreen, solder mask, etc.
    b. Gerber files are sent to PCB manufacturers.
7. **PCB Fabrication**
    a. Manufacturer produces the physical board based on Gerber files.
8. **Soldering & Assembly**
    a. Solder electronic components onto the fabricated PCB.
    b. Can be done manually (for DIY projects) or by machines (for mass production).

Erasmus

Day 6 Danca Tommaso

Arduino is a microcontroller platform. That means it's designed to run one program at a time and interact directly with hardware — turning things on or off, reading sensors, or controlling motors. It's excellent for real-time tasks where speed and simplicity matter.

You write code in the Arduino IDE using a simplified version of C/C++.

Once uploaded, the program runs automatically whenever the board is powered on.

Ideal for tasks like: blinking lights, reading temperature sensors, controlling servos, or reacting to button presses.

Raspberry Pi

The Raspberry Pi is a full mini-computer. It runs an operating system (usually a Linux-based one like Raspberry Pi OS), and it's capable of multitasking, running applications, accessing the internet, and interfacing with hardware too.

You can use Python, C/C++, Java, or even Scratch.

You can connect a monitor, keyboard, and mouse — or use it remotely over SSH.

Ideal for complex tasks like automation servers, media centers, or smart systems that require data logging, networking, or cloud access.

Practical: Blink LED Program

This is often the very first thing you do on both platforms — it's the "Hello World" of hardware. You make an LED blink to confirm that your board, code, and connections all work.

Required Components

1 LED (any color)

1 resistor (220Ω to 330Ω range to limit current)

Breadboard

Jumper wires

Basic Wiring:

The long leg (anode) of the LED connects to the control/output pin.

The short leg (cathode) goes to one end of the resistor.

The other end of the resistor goes to GND (ground).

Erasmus

Day 7 Danca Tommaso

Sensor Interfacing

This section explains how to connect and use three basic sensors with an Arduino: LDR (light sensor), Ultrasonic sensor, and IR (infrared) obstacle sensor. These sensors help detect light, distance, and the presence of objects.

1. LDR – Light Dependent Resistor

The LDR's resistance changes based on light levels. In bright light, resistance is low. In darkness, resistance increases. You connect it using a voltage divider with a fixed resistor (usually 10k ohms), and read the voltage in an analog pin.

Use case: Turn on an LED when it's dark.

Wiring:

- One side of LDR goes to 5V.
- The other side connects to analog pin A0 and to one end of a 10k resistor.
- The other end of the resistor goes to ground.
- An LED is connected to pin 13 through a current-limiting resistor.

Code summary: Read analog value from A0. If the value is low (dark), turn on the LED. Otherwise, keep it off.

2. Ultrasonic Sensor – HC-SR04

This sensor measures distance using sound waves. It sends a short burst from the trigger pin and waits for the echo to return. The time between sending and receiving tells how far the object is.

Use case: Measure distance and display it in centimeters.

Wiring:

- VCC connects to 5V.
- GND to ground.
- Trig to digital pin 9.
- Echo to digital pin 10.

Code summary: Send a short pulse on the trig pin. Measure how long it takes for the echo pin to go high. Use the time to calculate distance in centimeters and print it.

3. IR Obstacle Sensor

This sensor emits infrared light and checks for reflection. When an object is close, the light reflects back and is detected. It gives a digital output: LOW when something is detected, HIGH otherwise.

Use case: Detect obstacle and light up an LED.

Wiring:

- VCC to 5V.
- GND to ground.
- OUT to digital pin 7.
- LED connected to pin 13 through a resistor.

Code summary: Read the sensor output. If it's LOW (obstacle detected), turn on the LED. Otherwise, turn it off.

Summary

LDR gives analog input based on light. Ultrasonic sensor gives distance using sound. IR sensor detects the presence of nearby objects. These simple sensors are commonly used in automation, robotics, and environmental sensing.

Erasmus

Day 8 Danca Tommaso

Motor Control

This session focuses on controlling DC motors and servo motors using the L293D motor driver with Arduino. Motors are essential for movement in robotics and automation. Since Arduino can't power motors directly due to current limitations, a driver like the L293D is used.

DC Motor Control with L293D

L293D is an H-bridge motor driver IC that can control the direction and speed of DC motors. It allows control of two motors independently.

Wiring for one DC motor:

- Motor terminals connect to output pins (OUT1 and OUT2) of the L293D.
- IN1 and IN2 go to Arduino digital pins (e.g., pin 2 and pin 3).
- EN1 (enable pin) goes to Arduino PWM pin (e.g., pin 9) or directly to 5V.
- VCC1 to 5V (logic power), VCC2 to external power supply (e.g., 6V–12V).
- GND to Arduino GND and motor power ground.

Use case: Control motor direction and speed.

Basic control logic:

- IN1 HIGH and IN2 LOW: motor moves forward.
- IN1 LOW and IN2 HIGH: motor moves backward.
- Both LOW or both HIGH: motor stops.
- PWM signal on EN1 controls speed.

Code logic:

- Set IN1 and IN2 to control direction.
- Write analog value to EN1 for speed (0 to 255).

Servo Motor Control with Arduino

Servo motors rotate to a specific angle between 0 and 180 degrees. They are used in precise control applications like robotic arms and steering.

Wiring:

- Signal wire of servo to Arduino digital pin (e.g., pin 9).
- VCC to 5V (or external power if needed).
- GND to Arduino GND.

Servo motor is controlled using PWM signals sent from Arduino. The Servo library is used to simplify control.

Use case: Rotate servo to a specific angle.

Code logic:

- Attach servo to a pin.
- Use command to set angle, like servo.write(90) for 90 degrees.

Summary

DC motors provide continuous rotation and are great for movement. Servo motors allow precise angular control. The L293D driver enables safe and effective control of DC motors by managing voltage and current. Servo motors are powered directly from Arduino or an external supply and controlled using PWM signals. Both motor types are key components in robotics and automation projects.

Erasmus

Day 9 Danca Tommaso

Wireless Communication (IoT)

This session introduces wireless communication in embedded systems using Wi-Fi and Bluetooth. You'll learn how to connect devices without wires and send data over a network. The ESP8266 module is commonly used for Wi-Fi-based IoT applications, while Bluetooth is ideal for short-range wireless communication.

ESP8266 Wi-Fi Module

The ESP8266 is a low-cost Wi-Fi module with built-in TCP/IP stack and microcontroller capabilities. It can be used with Arduino to connect to the internet and exchange data with web servers or mobile apps.

Common ESP8266 modules include ESP-01 and NodeMCU. NodeMCU has USB and GPIO pins built in, making it easier to program.

Use case: Send sensor data to the cloud or control an LED from a web page.

Wiring (for ESP-01 with Arduino):

- VCC to 3.3V
- GND to GND
- CH_PD to 3.3V
- TX to Arduino RX (through voltage divider)
- RX to Arduino TX (use 3.3V logic level or divider)
- EN to 3.3V

In practice, a USB-to-serial adapter or a board like NodeMCU is easier to use.

Basic steps:

- Connect ESP8266 to Wi-Fi using SSID and password.
- Open a connection to a web server or host a simple web page.
- Control GPIO pins or send sensor data over HTTP.

Code logic:

- Use ESP8266WiFi library.
- Connect to Wi-Fi using WiFi.begin().
- Host a web server or make HTTP requests.

Example: Turn on an LED via a browser command.

Bluetooth Module (HC-05/HC-06)

Bluetooth modules allow communication between Arduino and a smartphone or other Bluetooth device over short range.

Wiring for HC-05:

- VCC to 5V
- GND to GND
- TX to Arduino RX
- RX to Arduino TX (through a voltage divider to reduce 5V to 3.3V)

Use case: Control an LED from a phone using a Bluetooth terminal app.

Basic operation:

- Pair the Bluetooth module with the phone.
- Send characters via serial.
- Arduino reads the characters and performs actions.

Code logic:

- Use Serial.read() to receive commands.
- If '1' is received, turn on LED. If '0', turn it off.

Summary

Wireless communication adds remote control and monitoring to your projects. ESP8266 enables internet-based control using Wi-Fi, useful for IoT applications. Bluetooth allows easy, low-power, close-range communication. Both technologies are widely used in smart home systems, wearables, and remote sensors.

Erasmus

Day 10 Danca Tommaso

Day 10 marks an important moment in the learning process, where theory meets practical application. Participants work in teams to build a real-world embedded system, applying all the concepts they've learned over the previous sessions. The mini project serves not only as a technical challenge but also as an opportunity to develop teamwork, problem-solving, and creativity.

There are two project options: an **obstacle-avoidance robot** or a **smart home model**. Both involve using Arduino, sensors, and actuators to create an interactive system.

The **obstacle-avoidance robot** is a mobile vehicle that uses an ultrasonic sensor to detect objects in front of it. When it senses an obstacle too close, it stops and changes direction automatically. This robot relies on an Arduino board, a motor driver (L293D), and two DC motors for movement. The logic is simple but powerful: measure the distance constantly, and if the path is blocked, turn to avoid the obstacle. This kind of robot introduces students to basic autonomous systems and real-time decision-making.

The **smart home model** simulates automation in a household environment. It might use a light-dependent resistor (LDR) to turn on lights when it's dark, a DHT11 sensor to control a fan based on temperature, and an IR sensor to detect door activity. Each component responds automatically to changes in the environment, demonstrating how embedded systems can make homes more efficient, safe, and comfortable.

Both projects challenge participants to design circuits, write code, and integrate multiple components into a working system. It's a great opportunity to face real-world problems like power management, timing, sensor accuracy, and user-friendly behavior.

In conclusion, the Day 10 mini project is not just a test of technical skills—it's a chance to innovate and collaborate. Whether building a robot that thinks or a home that reacts, students take a big step toward creating smart systems, just like those used in modern industry and daily life.

You've hit the Free plan limit for GPT.
Responses will use another model until your limit resets tomorrow after 2:52 AM.

Erasmus

Day 11 Danca Tommaso

Power Electronics

This session introduces the foundational concepts of power electronics, focusing on how electrical power is converted, controlled, and supplied to electronic circuits. Two key topics are covered: rectifiers and SMPS (Switched-Mode Power Supply) circuits.

Rectifiers

Rectifiers convert AC (alternating current) to DC (direct current), which is essential since most electronic devices run on DC.

Types of Rectifiers:

Half-Wave Rectifier

Uses a single diode

Allows only one half (positive or negative) of the AC waveform through

Simple but inefficient

Output: Pulsating DC with gaps

Full-Wave Rectifier (Center-Tap)

Uses two diodes and a center-tap transformer

Converts both halves of AC to DC

Output: Smoother than half-wave

Bridge Rectifier

Uses four diodes arranged in a bridge

No need for center-tap transformer

Converts both halves of AC into DC

More efficient and commonly used

Demo Summary:
Show different rectifier circuits and their waveforms using an oscilloscope or simulation. Highlight how filtering (using capacitors) smooths the output DC voltage.

SMPS – Switched-Mode Power Supply

SMPS circuits efficiently convert one DC voltage level to another. They work by rapidly switching on and off and using inductors, capacitors, and transformers to regulate the output.

Key Features:

High efficiency (up to 90% or more)

Compact size

Generates less heat than linear regulators

Common in chargers, power adapters, and embedded systems

Basic Types of SMPS:

Buck Converter – Steps down voltage

Boost Converter – Steps up voltage

Buck-Boost Converter – Can both increase and decrease voltage

Flyback Converter – Used when isolation is needed (with a transformer)

Demo Summary:
Present a simple buck or boost converter in action. Show how varying the duty cycle of the switching signal changes the output voltage.

Conclusion

Power electronics ensures that circuits receive the correct voltage and current. Rectifiers make AC usable, while SMPS circuits make power conversion efficient. Understanding these components is key to building safe and reliable electronic systems.

Erasmus

Day 12 Danca Tommaso

IoT & Industry 4.0

This session connects embedded systems with real-world applications through the Internet of Things (IoT) and introduces the concept of Industry 4.0 — the digital transformation of manufacturing using smart technologies.

IoT Overview

IoT involves connecting devices to the internet so they can collect, share, and act on data. Sensors, microcontrollers, and wireless modules (like ESP8266) form the base of IoT systems.

Industry 4.0

Industry 4.0 refers to the fourth industrial revolution, driven by:

- Automation and real-time monitoring
- Smart sensors and systems
- Cloud computing and data analysis
- Predictive maintenance and remote control

Case Study: Cloud Data Logging

Practical demonstration of sending sensor data (like temperature or light) from an ESP8266 to a cloud platform such as ThingSpeak or Blynk.

1. Using ThingSpeak

ThingSpeak is a free cloud platform that allows you to log, visualize, and analyze sensor data in real-time.

Steps:

- Create a ThingSpeak account

- Create a new channel and note the API key
- Connect ESP8266 to Wi-Fi
- Read sensor data (e.g., temperature or LDR)
- Send data to ThingSpeak using HTTP requests

Code logic:

- Use WiFi and HTTP client libraries
- Format data into URL with API key and value
- Send GET request to ThingSpeak server
- View real-time graph on the dashboard

2. Using Blynk (Legacy or Blynk IoT)

Blynk is a mobile app platform for IoT projects, ideal for remote control and monitoring.

Steps:

- Create Blynk account and project
- Add widgets (e.g., gauge or chart) linked to virtual pins
- Get authentication token
- Program ESP8266 with Blynk library
- Send or receive data using `Blynk.virtualWrite()` and sensor readings

Use case:

- View temperature on your phone
- Turn on a fan or LED from the app
- Monitor room conditions remotely

Summary

This session ties together all previous learning with real-world IoT applications. You send data from sensors to the cloud and visualize it, just like in industrial monitoring systems. It shows how embedded devices can form the backbone of smart factories and modern connected homes, aligning with Industry 4.0 goals.

Erasmus

Day 13 Danca Tommaso

PCB Fabrication & Troubleshooting – Summary

- **PCB Fabrication:**
  - Understanding how a circuit becomes a real, permanent board
  - Design made in software (e.g., KiCad, Eagle)
  - Steps: layout → print → etch → drill
  - PCBs are more reliable than breadboards and used in all final products
- **Lab Visit (if possible):**
  - Observe professional PCB manufacturing equipment and processes
  - Learn how real PCBs are made, tested, and packaged
- **Soldering Practice:**
  - Hands-on session using soldering irons and basic components
  - Learn safe and clean soldering techniques
  - Focus on avoiding cold joints, shorts, and damage to components
- **Troubleshooting:**
  - Use multimeters to check connections and voltages
  - Find common hardware errors (wrong polarity, loose wires, broken paths)
  - Apply logic and testing methods to fix non-working circuits

Goal: Gain real-world hardware skills—making, assembling, and fixing your own electronic boards.

Erasmus

Day 14 Danca Tommaso

Guest Lecture – Summary

- **Topic:** Career Trends in Electronics, IoT & Embedded Systems
- **Speaker:** Industry expert (engineer, entrepreneur, or tech lead)

Key Points Covered:

- **Industry Overview:**
  - Growth of IoT, automation, robotics, and smart devices
  - High demand for embedded systems engineers across sectors (automotive, healthcare, smart homes, etc.)
- **Career Paths:**
  - Roles: Embedded Developer, IoT Engineer, PCB Designer, Firmware Developer
  - Options in startups, MNCs, R&D, and freelancing
- **Skills in Demand:**
  - Programming (C/C++, Python, embedded Linux)
  - Hardware knowledge (microcontrollers, sensors, PCB design)
  - Soft skills: problem-solving, teamwork, continuous learning
- **Real-world Insights:**
  - Project examples and challenges faced in the industry
  - Importance of hands-on experience and project portfolios

- **Q&A Session:**
    - Students interact directly with the expert
    - Career guidance, resume tips, and learning resources shared

Goal:
To inspire students, connect classroom learning with real-world opportunities, and help them plan their future in tech.

Erasmus

Day 15 Danca Tommaso

Project Presentation & Certification – Summary

- **Project Showcase:**
    - Each student or team presents their mini project (e.g., obstacle-avoidance robot, smart home system)
    - Explains the concept, components used, challenges faced, and final outcome
    - Live demonstrations to show project functionality
- **Feedback Session:**
    - Instructors and peers give constructive feedback
    - Focus on technical implementation, creativity, and problem-solving
- **Certification Distribution:**
    - Certificates awarded to all participants for completing the training
    - Special mentions or awards for best projects (if applicable)
- **Wrap-Up:**
    - Recap of the 15-day journey
    - Encouragement to continue learning and exploring electronics/IoT
    - Group photo, networking, and closing remarks

Goal:
Celebrate the students' efforts, recognize their growth, and motivate them to keep building and innovating.