Block 4. Hardware-Software Interfacing

Module 1. Language

1.1 C++

C++ is a high-performance, general-purpose programming language that extends C by adding object-oriented, generic, and functional programming features. It is widely used for system programming, game development, real-time applications, embedded systems, and high-performance computing. C++ offers fine control over memory and hardware, making it suitable for applications requiring efficiency and speed.

Task: Sign up for this course to understand the basics of C++ and follow along. While this course is introductory, it will give you an understanding of the language you will be using. Submit a screenshot of completion and a screenshot of one of the projects within the course. If you are proficient in C++, submit a screenshot of a previous project as a replacement.

1.2 Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is widely used in web development, data science, artificial intelligence, automation, and scientific computing due to its extensive libraries and ease of use.

Task: Watch this video to learn the basics of Python and follow along. Submit a picture of your work as you follow the video. If you are proficient in Python, submit a screenshot of a previous project instead.

Module 2. Development

2.1 Arduino

An Arduino is a versatile microcontroller platform used in electronics development for prototyping, automation, and signal processing through easy-to-use hardware and software. In neurotech applications, it facilitates the development of brain-computer interface (BCI) prototypes, biosignal acquisition systems, and neural stimulation devices, enabling rapid testing and real-time data processing for innovative neuroelectronic solutions.

Task: Download Arduino and then watch these videos to understand the user interface of the software. Watch the following video to understand the basic components of Arduino. This will give you a basic idea of the hardware you will be working with.

2.2 Arduino (Integration of C++ skills)

C++ is essential for Arduino because the framework is built on it, and its core functions rely on C++. It enables object-oriented programming, efficient memory management, and high performance, which are crucial for microcontrollers with limited resources. Many Arduino libraries use C++ classes, so understanding the language helps with customization and extending functionality. While basic projects require minimal C++ knowledge, mastering it allows for more advanced and efficient programming.

Task: Watch this video and follow along to learn how to code C++ for Arduino.

Module 3. Communication Between Hardware and Software

3.1 Bluetooth and WiFi

Arduino communication systems allow interaction between microcontrollers and external devices using serial communication, I2C, SPI, or other protocols. Python, using Spyder or another IDE, provides a platform for processing, analyzing, and visualizing data collected from Arduino-connected sensors.

In neurotech, this hardware-software interfacing is essential for applications like brain-computer interfaces (BCIs), biofeedback systems, and neural signal processing. Arduino can collect biosignals (e.g., EEG, EMG), transmit data to Python, and use machine learning or signal processing techniques for real-time feedback or control.

Task: Watch the following videos to understand how WiFi and Bluetooth work and how to implement it.

- Lesson 1
- Lesson 2

3.2 ESP32

The ESP32 is a powerful microcontroller with Wi-Fi and Bluetooth connectivity, ideal for electronics development, IoT applications, and real-time signal processing. In neurotech, it enables wireless brain-computer interfaces (BCIs), remote biosignal monitoring, and low-power neural data transmission, making it a key tool for developing portable and connected neuroelectronic systems.

Task: Watch the following videos and learn how an ESP32 can upgrade an Arduino.

1. Lesson 1
2. Lesson 2