**Naming Conventions for Each Layer**

1. **Hardware Layer: (UPPER CASE)**

**Components**

1. **Resistors**: R<Number>\_<Description>
   * Example: R1\_PULLUP, R2\_SENSE
2. **Capacitors**: C<Number>\_<Description>
   * Example: C1\_FILTER, C2\_DECOUPLING
3. **Inductors**: L<Number>\_<Description>
   * Example: L1\_POWER, L2\_SIGNAL
4. **Diodes**: D<Number>\_<Description>
   * Example: D1\_PROTECTION, D2\_LED
5. **Transistors**: Q<Number>\_<Type>
   * Example: Q1\_NMOS, Q2\_PMOS
6. **Integrated Circuits (ICs)**: U<Number>\_<Function>
   * Example: U1\_MICROCONTROLLER, U2\_OPAMP
7. **Connectors**: J<Number>\_<Type>
   * Example: J1\_POWER, J2\_SIGNAL
8. **Switches**: SW<Number>\_<Type>
   * Example: SW1\_TACTILE, SW2\_SLIDE
9. **LEDs**: LED<Number>\_<Color>
   * Example: LED1\_RED, LED2\_GREEN

**Signals**

1. **Power Signals**: PWR\_<Voltage>\_<Description>
   * Example: PWR\_5V\_MAIN, PWR\_3V3\_LOGIC
2. **Ground Signals**: GND\_<Description>
   * Example: GND\_MAIN, GND\_ISOLATED
3. **Control Signals**: CTRL\_<Description>
   * Example: CTRL\_ENABLE, CTRL\_RESET
4. **Data Signals**: DATA\_<Bus>\_<Description>
   * Example: DATA\_SPI\_MOSI, DATA\_I2C\_SDA
5. **Analog Signals**: ANALOG\_<Description>
   * Example: ANALOG\_TEMP, ANALOG\_VOLTAGE

**PCB Design Elements**

1. **Nets**: <Type>\_<Description>
   * Example: VCC\_5V, GND\_MAIN, SIG\_PWM
2. **Layers**: <LayerName>\_<Number>
   * Example: TOP\_LAYER, BOTTOM\_LAYER, INTERNAL\_LAYER\_1
3. **Footprints**: <ComponentType>\_<Package>
   * Example: R\_0805, U\_SOIC\_8, C\_0603
4. **Driver Layer: ( Software snake\_case )**
   * **Prefixes:** Use hardware-specific prefixes to clarify which device a driver pertains to, e.g., gpio\_, uart\_, adc\_.
   * **Function Naming:** Use action-based verbs that clearly state what the function does, like init, read, write, enable, disable. For example, uart\_init(), adc\_read().
   * **Suffixes for Handlers and Callbacks:** Use \_handler for interrupt handlers and \_callback for callback functions, indicating their roles clearly, e.g., timer\_interrupt\_handler().
5. **Hardware Abstraction Layer (HAL):**
   * **Interface Naming:** Use consistent and abstracted names that do not directly reflect the underlying hardware, aiming for portability and abstraction, e.g., sensor\_get\_temperature() instead of adc\_read\_channel\_1().
   * **File Naming:** Name files according to their functionality or associated hardware module, such as sensor\_hal.c or motor\_controller\_hal.h.
6. **Service Layer (e.g., microROS):**
   * **Module Naming:** Name modules according to the services they provide, using clear and descriptive names that communicate the functionality, such as navigation\_service, communication\_manager.
   * **Service Functions:** Name functions to reflect the high-level actions they perform, e.g., start\_navigation(), send\_message().

MicroROS Example:

1. **Nodes**: node\_<Function>
   * Example: node\_motor\_control, node\_sensor\_fusion
2. **Topics**: topic\_<DataType>\_<Description>
   * Example: topic\_cmd\_vel, topic\_sensor\_data
3. **Services**: service\_<Action>
   * Example: service\_start\_motor, service\_stop\_motor
4. **Application Layer:**
   * **Application Specific Naming:** Names should reflect the application's domain and the functionality, e.g., robot\_arm\_move\_to\_position(), temperature\_monitoring\_task().
   * **Control Functions:** Use verbs that describe control actions, such as configure, execute, process, e.g., configure\_system\_settings(), process\_user\_input().