Programming and Debugging the Portenta H7: USB-C, ST-LINK, and Connector Access Methods

Method 1: Using USB-C cable (Arduino/Mbed bootloader or STM32 system bootloader)

- The Portenta H7 supports programming over its USB-C port.
- To enter the Arduino/Mbed bootloader, you only need to double-tap the RESET button. This works out of the box and allows flashing via Arduino IDE or arduino-cli.
- To enter the STM32 system bootloader (DFU/USB mode), the MCU requires BOOT0 = 1 + RESET.
- On Portenta H7, the BOOT0 pin is not connected to any MKR header.
- Instead, it is exposed only on the high-density connector J2, pin 3 (BOOT0 / Boot_source).
- Therefore, to access the STM32 system bootloader, you need either the Portenta Breakout Board (which gives you a Bootsource DIP switch) or to solder a wire directly to the J2-3 pad.

Method 2: Using ST-LINK (SWD debugging/programming)

- The Portenta H7 supports standard ARM SWD (Serial Wire Debug) with an ST-LINK.
- Required signals:
    - SWDIO (PA13) → J2-13
    - SWCLK (PA14) → J2-15
    - NRST (Reset) → J1-12
    - Plus 3.3V reference (e.g., J2-18) and GND (e.g., J2-20).

- These pins are not exposed on the MKR headers; they exist only on the J1/J2 high-density connectors.
- To use them, you must either have the Portenta Breakout Board (which brings them out cleanly) or solder wires directly to the J1/J2 pads.

Soldering VS Breakout Board

1. **Soldering Method**
    - ➢ **How to do it:**

      Identify the required signals on the J1/J2 HD connector pads and solder very

thin enamel wires (≈30–34 AWG) directly to those pads (e.g., J2-13 for SWDIO, J2-15 for SWCLK). Route the wires out and attach them to a 2.54 mm pin header or Dupont jumper ends for easy connection to your ST-Link.

- **Risks/Precautions:**

  The 0.4 mm pitch pads are fragile — too much heat or pulling can lift them off permanently. Always use flux, a fine soldering tip, and minimal solder, then secure the wires with glue/epoxy/tape immediately to avoid stress. If a pad is damaged, that signal may be lost forever.

- **Pros/Cons:**

  It's cheap, requires no extra hardware, and works if you only need a few signals, but it's high-risk, demands soldering skill, isn't reusable, and only exposes the pins you solder.

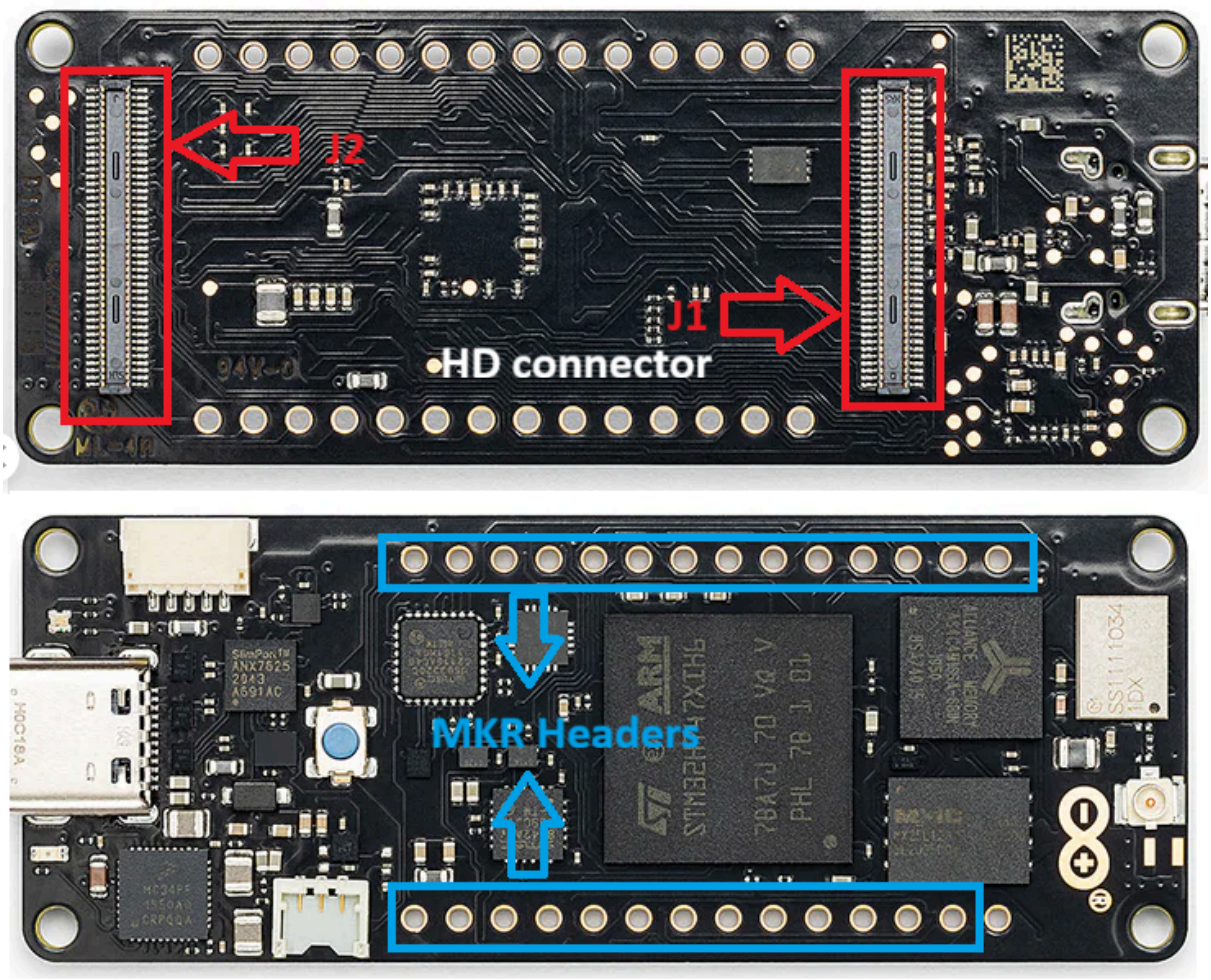
2. **Breakout Board Method**

   - **How to do it:**

     Use the official Arduino Portenta Breakout Board (ASX00031) or a DF40 connector breakout adapter, which plugs into J1/J2 and maps all 160 pins to 2.54 mm headers. SWD, BOOT0, NRST, power rails, and many peripherals become accessible without fine soldering, and you can connect your ST-Link directly with jumper wires.

   - **Pros/Cons:**

     This method is safe and reliable, avoids pad damage, and provides full access to all signals along with extras like a BOOT0 switch, power button, USB host, and microSD slot. It's reusable and convenient for multiple projects, but costs more (~RM 270) and makes the setup bulkier.



In summary: If the PCB lab has the capability to solder fine-pitch pads (0.4 mm) using professional tools such as a microscope, fine soldering stations, and flux handling equipment, then the soldering method is a viable option. Otherwise, the breakout board method is the safer and more reliable choice.