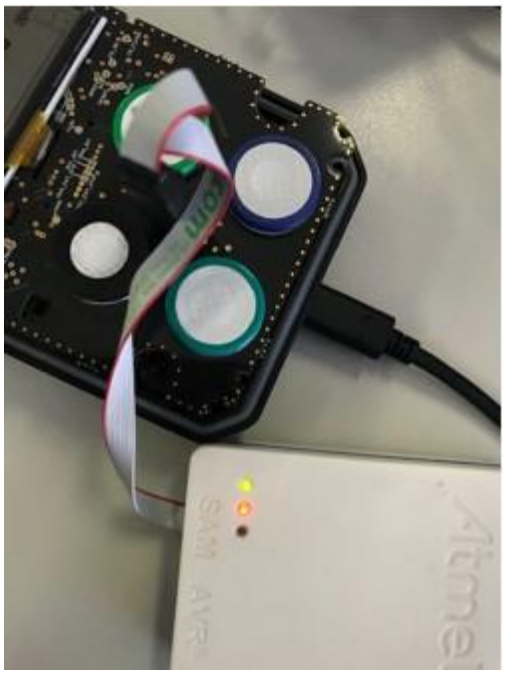
Programming the AuraPT

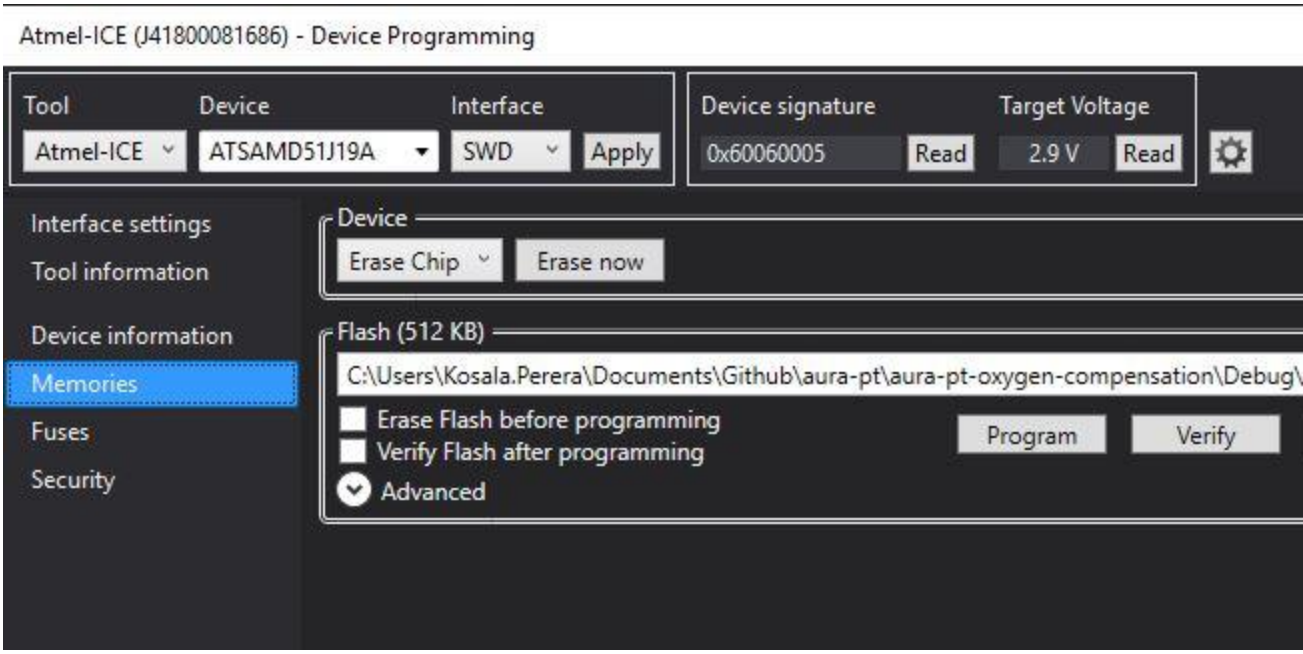
1. We need Atmel Studio to program the PT and you can download it from the following link. https://www.microchip.com/mplab/microchip-studio

2. Remove the plastics of the PT and locate the programming port. Refer below.

3. Insert program probe of the Atmel ICE to the port and the green led should light up as shown below (you may need to hold the cable to ensure there is a proper contact between the Atmel ICE and the PT during the upload).



4. Open Atmel Studio, got to Tools → Device Programming. Under tools, select Atmel-ICE. Under device, select ATSAMD51J19A. Under interface, select SWD.
   Click apply and then click read. Refer below.



5. Select Memories. Click Erase Now. Under Flash, select the location where you saved the binary file (attached with this mail. Password 1234). Click program.
6. If programmed successfully, PT will restart.

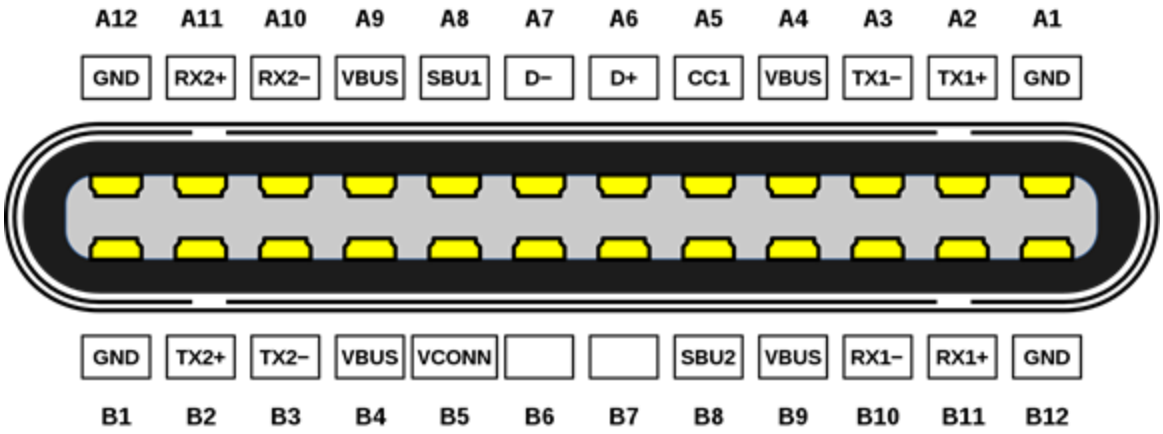
Receiving data

1. I have created a simple serial output and please refer the wiring diagram below. USB type C pinout.

AURA PT 2/4 USB TYPE C PINOUT

A1	GND		B12	GND
A2	TX		B11	NC
A3	RX		B10	NC
A4	5V		B9	5V
A5	NC		B8	NC
A6	D+		B7	D-
A7	D-		B6	D+
A8	NC		B5	NC
A9	5V		B4	5V
A10	NC		B3	RX
A11	NC		B2	TX
A12	GND		B1	GND

Note : Recommend to use USB3.1 Type C cable



2. The following structure is sent out from the PT. This will be sent in a form of an array.

```
typedef struct Gas_Data{
    uint8_t battery_level = 0;
    Gas_Type _gas_1 = Unknown;
    Gas_Type _gas_2 = Unknown;
    Gas_Type _gas_3 = Unknown;
    Gas_Type _gas_4 = Unknown;
    float _gas_1_val = 0;
    float _gas_2_val = 0;
    float _gas_3_val = 0;
    float _gas_4_val = 0;
    uint32_t _crc = 0;

    Gas_Data(){

    }
};
```

```
typedef union gas_data {
  Gas_Data t;
  uint8_t b[28];

  gas_data(){

  }
}gas_data;
```

3. I have included a CRC check for data verification as well.

```
Serial.readBytes(gas_details.b, 28);
crc.update(gas_details.t._gas_1_val);
_crc = crc.finalize();
```