Code example: Starting I2C communication

The I2C pins are connected to the IMU, Pressure sensor and the expansion ports.
SDA is on pin 21 and SCL pin 22. These are defined in the Mercury_pins.h file so you can just use the name SDA and SCL.
Wire.begin(SDA, SCL);

The code below enables the sensor power, starts I2C and scans for available devices every 10 seconds.
Connect a serial monitor after rebooting the Mercury once compiled and uploaded to view the output.

> **Using Arduino IDE?** Our online programmer includes Mercury_Pins.h by default so the pin names work without issue. If you are using Arduino IDE or another programmer, copy the Mercury_Pins.h tab content and paste it into the top of your program.

```
/*
 * Mercury I2C Scanner
 * Scans the I2C bus and reports all connected devices.
 * Useful for checking sensor addresses after assembly.
 */

#include "Wire.h"
#include "Mercury_Pins.h"

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("Mercury I2C Scanner");
  Serial.println("====================");

  // Power on the sensor rail
  pinMode(VACC, OUTPUT);
  digitalWrite(VACC, HIGH);
  delay(100);  // Allow sensors to stabilise

  // Start I2C on Mercury pins
  Wire.begin(SDA, SCL);

  Serial.println("Sensor power: ON");
  Serial.println("I2C bus ready (SDA=" + String(SDA) + " SCL=" + String(SCL) + ")");
  Serial.println();
}

void loop() {
  int found = 0;

  Serial.println("Scanning I2C bus...");

  for (int row = 0; row < 128; row += 16) {
    // Row label

    for (int col = 0; col < 16; col++) {
      int addr = row + col;
```

```
            // Skip reserved addresses (0x00-0x07 and 0x78-0x7F)
            if (addr < 0x08 || addr > 0x77) {
                continue;
            }

            Wire.beginTransmission(addr);
            int result = Wire.endTransmission();

            if (result == 0) {
                found++;
            } else {
            }
        }
    }

    Serial.println();
    if (found == 0) {
        Serial.println("No devices found. Check wiring and sensor power.");
    } else {
        Serial.println(String(found) + " device(s) found:");
        // Second pass to list found devices with common names
        for (int addr = 0x08; addr <= 0x77; addr++) {
            Wire.beginTransmission(addr);
            if (Wire.endTransmission() == 0) {
                Serial.printf("  0x%02X", addr);
                // Identify common sensors
                switch (addr) {
                    case 0x47: Serial.print(" - BMP581 (Mercury Pressure)"); break;
                    case 0x6B: Serial.print(" - LSM6DSO32 (Mercury IMU)"); break;
                    case 0x77: Serial.print(" - BMP390 (Mercury Pressure)"); break;
                }
                Serial.println();
            }
        }
    }

    Serial.println();
    Serial.println("Next scan in 10 seconds...");
    Serial.println("---");
    delay(10000);
}

#pragma once
/*
 * Mercury (ESP32-C6) Pin Definitions
 * Board-specific GPIO assignments
 */

// ── Status LED (NeoPixel) ──
#define LEDPOWER      3    // NeoPixel power (drive HIGH to enable)
#define LED           2    // NeoPixel data signal

// ── I2C Bus ──
#define SDA           21   // I2C data
#define SCL           22   // I2C clock

// ── Sensor Power ──
#define VACC          20   // Sensor power rail (drive HIGH to enable)
```

```
// —— General Purpose Ports ——
#define GP06        6   // GP06 port
#define GP07        7   // GP07 port

// —— High Current Output ——
#define OUT1        5   // High current output (e.g. pyro / relay)

// —— Battery Bar LEDs ——
#define BL1         4   // Battery LED 1 (lowest)
#define BL2         14  // Battery LED 2
#define BL3         15  // Battery LED 3
#define BL4         18  // Battery LED 4
#define BL5         19  // Battery LED 5 (highest)

// —— Indicators ——
#define DISK        8   // Disk activity LED

// —— Analogue / Detection ——
#define BATIN       0   // Battery voltage (1:1 divider)
#define USBDETECT   1   // USB power detect (HIGH = USB present)
#define BUTTON      9   // BUTTON on the board, boot button but can be used
```