

Progress report: Sensors

Boyan & Mihai

2025-12-02

The framework debacle

Name	Comfort level
Arduino	High: comfortable with C++ and Arduino.
CMSIS	Very low: can read examples and tweak small bits, but not comfortable writing code from scratch.
LibOpenCM3	Very low: only superficial familiarity with the API and build setup.
Standard Peripheral Library	No.
STM32Cube / HAL	Low-medium: can follow, but not fully understand, time-pressured in terms of learning (the manual is 2000+ pages).

The framework debacle

Name	Comfort level
Arduino	High: comfortable with C++ and Arduino.
CMSIS	Very low: can read examples and tweak small bits, but not comfortable writing code from scratch.
LibOpenCM3	Very low: only superficial familiarity with the API and build setup.
Standard Peripheral Library	No.
STM32Cube / HAL	Low-medium: can follow, but not fully understand, time-pressured in terms of learning (the manual is 2000+ pages).

Opinions?

Our experience with the HAL

- Initial experiments with STM32CubeMX generated code were promising.
- We started hating the IDE very quickly.
- Tried to move to VSCode + Makefile, but the HAL documentation is not great (especially for our capabilities).
- Moved to PlatformIO for easier dependency management.
- Realized we don't have enough time to learn about:
 - ARM Cortex-M architecture
 - STM32 peripherals
 - HAL API

In that order, so we...

... switched to Arduino

- Benefits:
 - Familiar environment.
 - Lots of examples online.
 - Simple API for basic peripherals.
- Downsides:
 - Less control over hardware.
 - Performance overhead.
 - Limited access to advanced features (debatable, can still use the HAL if needed).



Intermezzo – The Sensor interface

Looks like this:

```
#include <Arduino.h>

class Sensor {
public:
    virtual bool begin() = 0;
    virtual auto data() = 0; // return type is sensor-specific struct
};
```

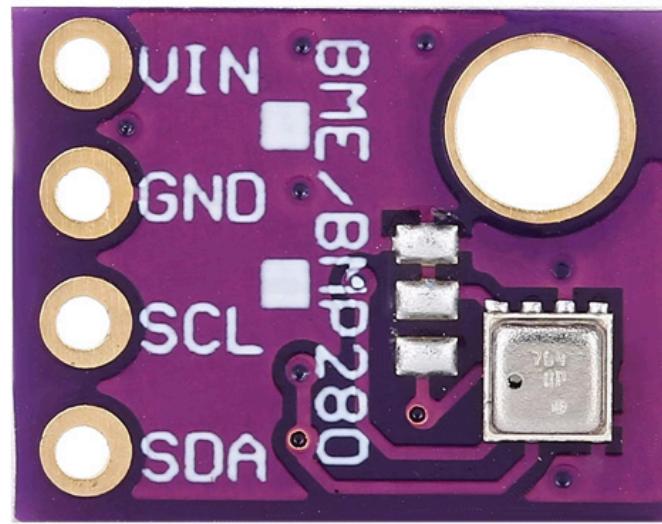
We call the `begin()` method in `setup()` and `data()` in `loop()`.

Actual Progress

:)

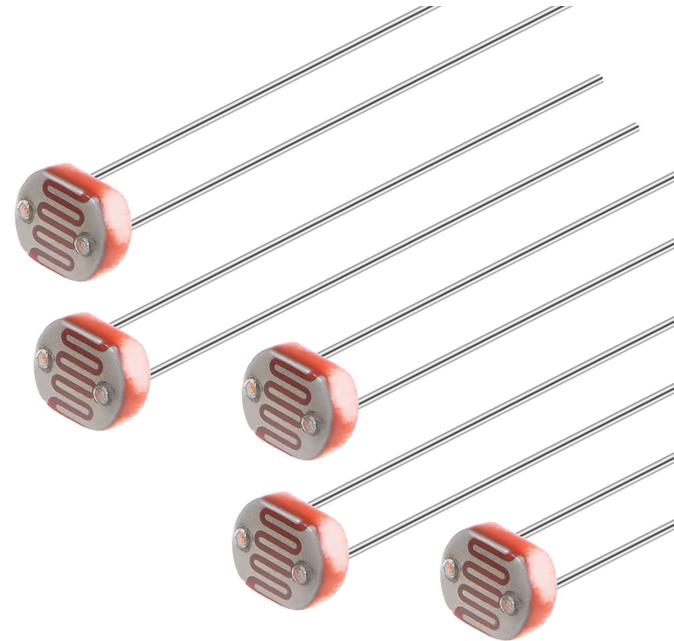
BME280 sensor

- [Adafruit BME280 library](#) used
- Using I2C interface (SDA: PB9,
SCL: PB8)



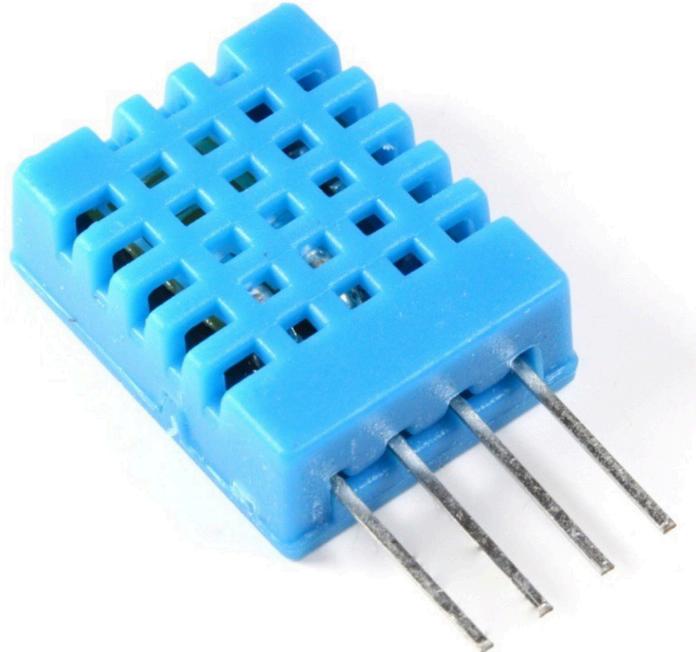
Photoresistor

- Simple voltage divider circuit used (10 k Ω resistor to GND, photoresistor to VCC)
- Analog reading from pin PA0



DHT11 sensor

- Just reading the digital pin (PC14)



We're ready for LoRa!