Developing Your Flow Sensing Solution with FLS110

Introduction

The FLS110 is an analogue mass flow sensor. Combined with a standard microcontroller and firmware provided by Flusso it enables digital flow or differential pressure (DP) sensing solutions with exceptional performance-cost ratio and versatility for integration into your product. So, what's the quickest way to find out if FLS110 can meet your requirements and how would you move forward from there? This technical note provides an overview, which is summarised in the chart below. The steps are covered in more detail in sections with the same number. More detailed information is available in the documents referred to, which are available through our customer portal (https://www.flussoltd.com/customer-portal).

1. Get to know the FLS110

This technical note is a good starting point for getting to know what FLS110 digital flow sensing solutions and their development cycle look like.

The *FLS110 Miniature Gas Flow Sensor Datasheet* provides the usual technical information and performance figures for a practical reference design – one which could be very close to what you need for your own product.

2. Rapid hands-on evaluation

The **FLS110 Evaluation Kit** lets you try out the FLS110 with a ready-made flow fixture and a Windows 10 GUI. You just need a flow or pressure source. The sensor module in the kit is the one whose performance is specified in the FLS110 datasheet. It is programmed with standard FLS110 firmware. For more information, please refer to:

- *FLS110 Evaluation Kit User Guide*
- *FLS110 Evaluation Kit Fixture Catalogue.*

3. Design FLS110 into your product flow path

Next, we recommend incorporating the FLS110 into *your* **flow path design** and characterising the system. Using the Evaluation Kit sensor module minimises development effort at this early stage. You can use the GUI again or automate your tests with our FLS110 Python Test Scripts. Find all the details in:

- *Mechanical and Fluidic Integration of the FLS110*
- *FLS110 System Characterisation and Calibration*
- *FLS110 Python Test Scripts User Guide*

4. Your hardware design and firmware build

Having validated FLS110 performance for your application, it's time to make your own **hardware design** and, if you wish, make your own, customised **firmware build**. For guidance on these inter-related tasks, please refer to:

- *FLS110 Hardware Design Guide*
- *FLS110 Firmware Integration Guide*

5. Product testing

Next, repeat test and characterisation of your flow (or DP) sensing subsystem on your fully integrated product design – with your own hardware and firmware.

6. Product assembly and test

The FLS110 is compatible with standard surface mount component handling and PCB assembly processes. The firmware provides full support for system calibration and testing. And you might find the Python Test Scripts useful once again, when developing your production test software.

Contents

References

These documents are available through our customer portal (https://www.flussoltd.com/customer-portal).

[1] FLS110 Miniature Gas Flow Sensor Datasheet (FL-000038-DS)
[2] FLS110 Evaluation Kit User Guide (FL-000956-UG)
[3] FLS110 Evaluation Kit Fixture Catalogue (FL-000736-TN)
[4] Mechanical and Fluidic Integration of the FLS110 (FL-000479-TN)
[5] FLS110 System Characterisation and Calibration (FL-000561-TN)
[6] FLS110 Python Test Script User Guide (FL-000985-UG)
[7] FLS110 Hardware Design Guide (FL-000607-TN)
[8] FLS110 Firmware Integration Guide (FL-000939-TN)

1 Get to know the FLS110

The FLS110 is an analogue mass flow sensor based on the principle of a hot wire anemometer. To make a digital flow (or differential pressure) sensing solution for your application, the FLS110 is combined with a standard microcontroller running firmware provided by Flusso. The firmware is rich with features for system level characterisation, calibration, and modes of operation. It can be configured and even customised to your exact needs for sensing accuracy. FLS110 offers great versatility for product integration.

System-level thinking and approach to integration is central to FLS110 flow (or DP) sensing solutions. The firmware reports **system flow** directly and accurately to your application. You are not forced to use an expensive calibrated DP sensor and figure out flow for yourself (but you *can* configure the firmware to report DP readings if you want). Because Flusso provides the sensing algorithms already implemented in firmware you can achieve class-leading cost-performance without a high burden of development effort.

In the following sections we look at the elements of an FLS110-based solution in some more detail.

1.1 The FLS110 component and flow path interface

The FLS110 is a solid-state, silicon-MEMS sensor die in an open-cavity plastic package with two flow ports. Figure 1 shows the structure, with a cross-section through the package lid to reveal the die and its two sensing elements.

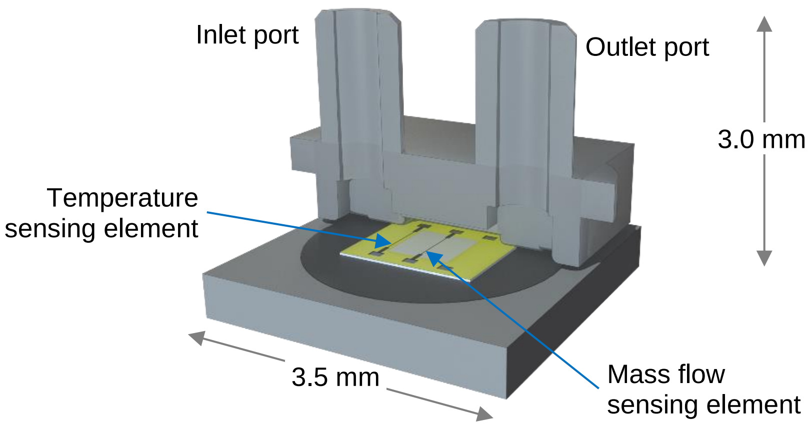

Figure 1: Inside the FLS110

The FLS110 can be connected into your system flow path in either a through-flow or bypass arrangement. Both are illustrated in Figure 2. There are a number of ways to implement bypass or non-through-flow configurations. We cover that topic more in section 3.

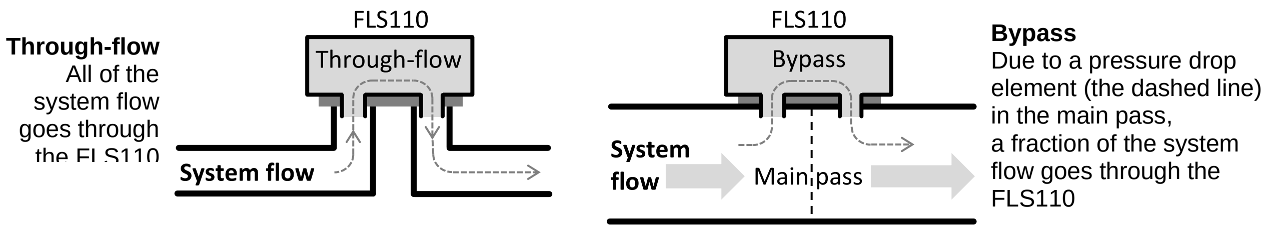

Figure 2: Through-flow and bypass system configurations

Sensing performance is specified in the *FLS110 Miniature Gas Flow Sensor Datasheet*, for up to 200 sccm in a through-flow configuration and 500 Pa differential pressure, but it can be safely operated above those limits. Most applications will use a bypass (or non-through-flow) arrangement in order to measure much higher system flow rates. There are fluidic fixtures in the *FLS110 Evaluation Kit Fixture Catalogue* that are designed for system flows as high as 500 slm, while staying within the FLS110's specified through-flow operating range of 200 sccm / 500 Pa differential pressure. You can design for even higher system flow rates than that. Also, you can operate the FLS110 at greater than 200 sccm through-flow or 500 Pa DP. In fact, we have implemented a drone airspeed sensing solution with a Pitot tube that operates up to 5 kPa.

1.2 Hardware–firmware architecture

Figure 3 illustrates a system with the FLS110 in a bypass configuration and flow sensing firmware running on a dedicated microcontroller. Application software on the host processor controls the flow sensing firmware and obtains readings using register write and read transactions over a serial interface (typically I²C-bus®).

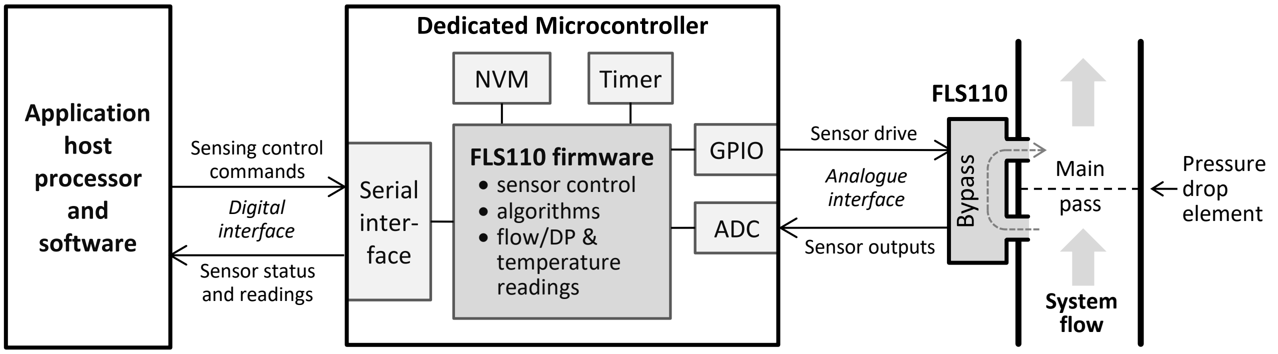

Figure 3: FLS110 flow/DP sensing solution with a dedicated microcontroller

The FLS110 Evaluation Kit sensor module is an implementation of this architecture (see section 2).

Using the software development kit (SDK) provided by Flusso, your application code can also be compiled and linked with the FLS110 flow sensing firmware to run on a single microcontroller. In this architecture the flow sensing firmware presents an application programming interface (API) with a set of function calls that correspond to the I²C-bus registers. This is illustrated in Figure 4.

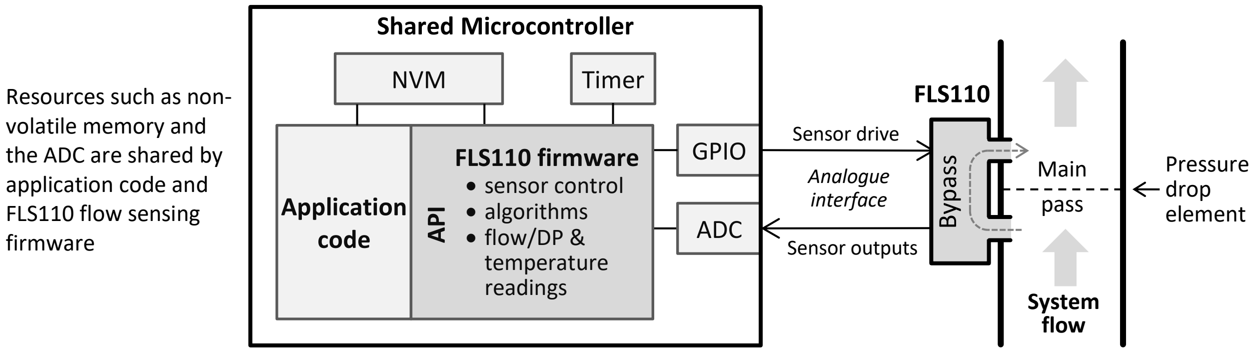

Figure 4: FLS110 flow/DP sensing solution with a shared microcontroller

1.3 Flow sensing firmware functionality

Functionality of the FLS110 firmware is summarised under two headings in the following sections. The I²C registers associated with these functions are listed in the FLS110 datasheet. Corresponding API calls are described in *FLS110 Firmware Integration Guide*.

1.3.1 Control and status

I²C registers and API calls are provided to

- Set the **measurement basis** to be mass flow or volumetric flow / differential pressure
- Set the number of measurements over which a **moving average** is calculated for each reading
- Set the firmware **operating mode**:
  - Idle – a low power state in which no measurements are taken
  - Continuous – measurements are taken, and readings are updated continuously or
  - Single Shot – a single reading is taken, and the firmware returns to Idle mode
- Read the **status** of readings and other firmware functions – either “in progress” or “ready”

1.3.2 Calculation of readings

The FLS110 firmware drives the analogue sensing elements of the device, digitises their outputs, and applies algorithms to generate readings which are made available to the host application via I^2C-bus registers or API function calls.

Flow temperature

The digitised output of the **temperature sensor** is converted to a flow temperature measurement (T_{flow}) in °C. This involves an offset, which is determined by having the FLS110 at a known temperature and passing the value to the firmware by writing it to I^2C register or by an API call. This must be done for every end-product unit, typically during your production test. See

Mass flow

The digitised output of the **mass flow sensing element** is processed by the firmware in two stages to create a reading:

1. **Heat power transfer** (*h*) to the gas in flow is calculated. An offset is applied to ensure that *h* is zero when there is no flow through the system. The offset is determined by an instruction to the FLS110 firmware during your production test or perhaps at system start-up, when there is no system flow.
2. **Mass flow** ($\dot{m}$) is then calculated as a cubic function of *h*:

$$\dot{m} = C_3 h^3 + C_2 h^2 + C_1 h$$

There is no constant term in the cubic because *h* is zero when there is no flow. The coefficients C_1, C_2 and C_3 are obtained by a process of individual product unit calibration or by general characterisation of your design (see section 3.2). Because the procedure is carried out at system-level, the readings provided by the FLS110 firmware are mass flow in the system, not just through the FLS110 itself. The coefficients are stored in microcontroller non-volatile memory (NVM).

Volumetric flow or differential pressure

The firmware can also calculate **system volumetric flow** (*Q*) or **differential pressure** (Δp) measurements, taking into account the ratio of flow density when the measurement is made to flow density when the zero-flow/DP offset was determined. Density of a particular gas is proportional to pressure and inversely proportional to temperature, so:

$$Q \;\; or \; \Delta p = (C_3 h^3 + C_2 h^2 + C_1 h) \cdot \frac{T_{flow}}{T_0} \cdot \frac{p_0}{p_{flow}}$$

T_0 was the flow temperature (in kelvin) when the zero-point offset was determined (stored in NVM).

p_0 was the flow pressure (in pascals) when the zero-point offset was determined (stored in NVM).

T_{flow} is the flow temperature, measured by the firmware using the FLS110 integrated temperature sensor.

p_{flow} is the flow pressure, provided to FLS110 firmware by your host application (via I^2C or API call) or, alternatively, hardcoded in your FLS110 firmware build as a default for normal operating conditions.

Averaging

A rolling average can be applied to measurements of flow temperature (T_{flow}) and *h* to produce readings with less noise. A window size of up to 128 measurements can be set by an I^2C register write operation or API call to the FLS110 firmware. Note: The same averaging window is applied to both T_{flow} and *h*.

2 Rapid hands-on evaluation

The FLS110 Evaluation Kit allows you to try out the FLS110-STM32 reference design with no development effort. You can order a kit at https://flussoltd.com/products/fls110-evaluation-kit. The hardware, shown in Figure 5, consists of a sensor module pre-assembled into a fluidic fixture, a USB-I^{2}C adapter and a ribbon cable to connect them. Sensor modules are available with or without a pressure sensor – your application requires sensing on a volumetric flow / DP basis, and you wish to compensate for flow pressure dynamically, you might find this convenient for your initial testing. When requesting your kit, choose a fixture from the *FLS110 Evaluation Kit Fixture Catalogue* that best matches your requirements – they are available with and without a port for the pressure sensor.

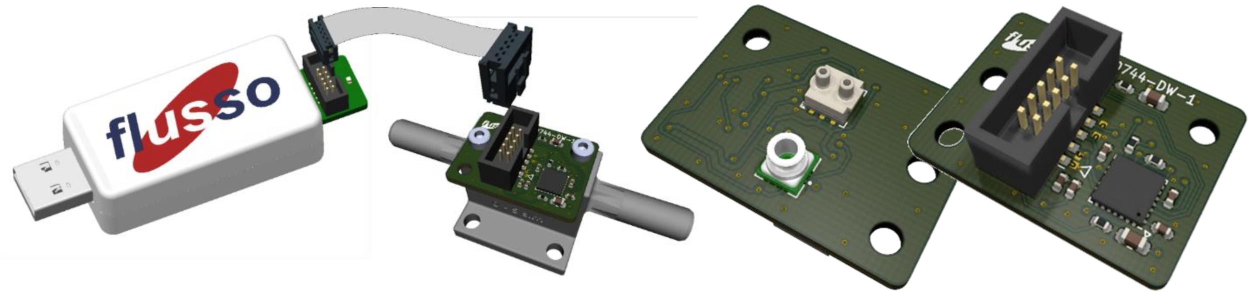

Figure 5: FLS110 Evaluation Kit hardware (sensor module with optional pressure sensor)

The Evaluation Kit has an easy-to-use GUI (Figure 6) that lets you

- Choose various options and measure flow or differential pressure
- Plot the readings on-screen and log them to a .csv file for later analysis
- Try out offset and system calibration procedures (see section 3.2 for more about this)

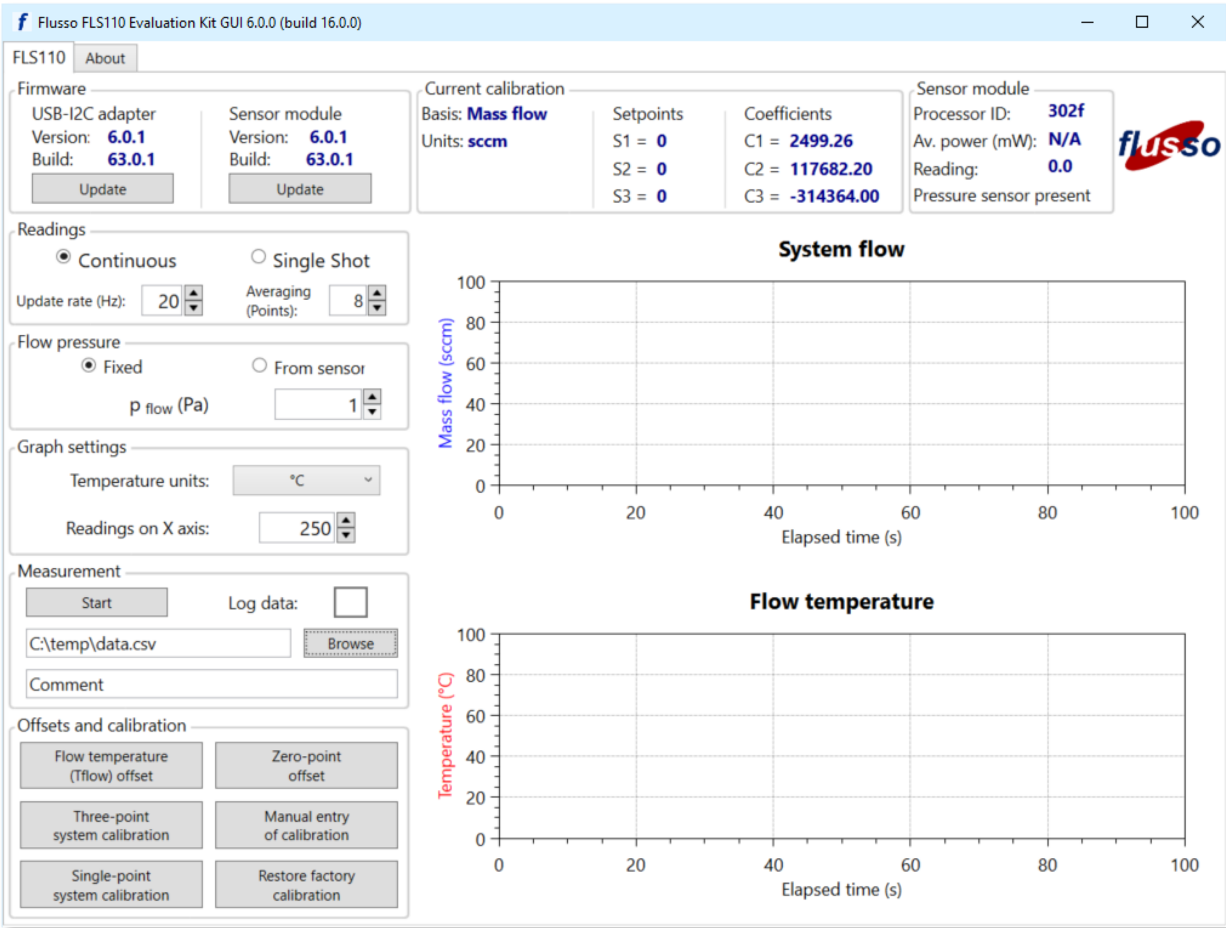

Figure 6: FLS110 Evaluation Kit GUI main window

3 Design FLS110 into your flow path

After initial evaluation we recommend tackling mechanical/fluidic integration of the FLS110 into your product flow path and characterising your system with the FLS110 in-situ. If you can fit the Evaluation Kit sensor module into your prototype setup then no electronics design or firmware adaptation will be necessary at this stage, and you can use the GUI for limited manual testing. If you need to adapt the sensor module PCB to fit, we can provide you with the standard layout to give you a head start. If you wish to automate testing at this stage, e.g., to coordinate operation of the FLS110 firmware with control of your test equipment, then you can leverage our Python test scripts. See the *FLS110 Python Test Scripts User Guide* for more information.

3.1 Mechanical and fluidic integration

Key considerations for mechanical integration include

- Sealing the FLS110 reliably to your flow path
- Avoiding excessive temperatures, strain, shock, or vibration
- Keeping the FLS110 and the microcontroller that drives it close together, for signal integrity

There are a number of ways of integrating the FLS110 fluidically. The best choice will depend on your requirements for flow range and maximum acceptable pressure drop:

1. **Through-flow (direct flow measurement).** All of the system flow goes through the FLS11 (see Figure 2)0. This configuration is appropriate if the dynamic range of your system flow is within (or nearly within) the FLS110's nominal operating range of up to 200 sccm or 500 Pa.
2. **Bypass flow, using a viscous pressure drop element.** A fraction of the system flow goes through the FLS110 via a bypass route. This configuration is good for applications with a system flow range of interest that far exceeds the FLS110 nominal rating of 200 sccm.
3. **Bypass flow, using a Venturi.** Venturis are insensitive to turbulence in the flow. They are not as sensitive as viscous pressure drop elements at low flow but have lower pressure drop at high flows. They are often the best choice for applications with turbulent flow (high Reynolds number).

The FLS110 datasheet provides information you need to design the pressure drop element in your main pass to give bypass flow in the FLS110 nominal through-flow range. Please also refer to *Mechanical and Fluidic Integration of the FLS110* for more guidance.

3.2 System calibration and characterisation

This is a brief summary, see technical note *FLS110 System Characterisation and Calibration* for details.

The coefficients C_1, C_2 and C_3 needed for calculation of flow or DP readings (as we explained in section 1.3.2) are determined by a system-level calibration procedure, with the FLS110 in-situ, using three non-zero system flow (or differential pressure) setpoints. At each setpoint the FLS110 firmware is instructed to take a flow or DP measurement. The coefficients C_1, C_2 and C_3 are then calculated by a cubic curve fit.

Three-point calibration can be executed manually using the FLS110 Evaluation Kit GUI or automated with the FLS110 Python Test Scripts. The FLS110 firmware I^2C registers or API function calls that support the procedure are listed in the *FLS110 Miniature Gas Flow Sensor Datasheet* and described in more detail in the technical note *FLS110 Firmware Integration Guide*.

"Default" coefficients for your system design can be determined by doing three-point calibration on a number of end-product prototypes and taking average values of C_1, C_2 and C_3. We refer to this process as "characterisation". For many applications, the linearity and repeatability achieved with default coefficients will be good enough across all your production units and only temperature and zero-flow/DP offset procedures will be needed for individual ones. However, if you need to mitigate unit-to-unit variation in your product (due to manufacturing tolerance spread) and achieve better accuracy, calibration with non-zero flows (or differential pressures) can be done during production test of every unit. There are two options:

- Single-point calibration combines default coefficients with a single non-zero flow/DP measurement and produces coefficients that enable a particular product unit to achieve better accuracy. This method is particularly effective if you are only interested in measurement of a restricted flow range.
- Three-point calibration generates unit-specific coefficients that achieve best accuracy.

4 Electronic hardware design and firmware build

The *FLS110 Miniature Gas Flow Sensor Datasheet* specifies through-flow performance for a reference design that we call FLS110-STM32. Understanding this design is a good way to get started on your own design. Here we provide an overview, full information can be found in *FLS110 Hardware Design Guide* and *FLS110 Firmware Integration Guide*.

With our standard firmware build, FLS110-STM32 implements a **dedicated microcontroller** architecture (as shown in Figure 3). It has a STMicro™ STM32L031G6U6 (arm® Cortex®-M0+ core) microcontroller with I²C interface to the host. The schematic and a module implementation are shown in Figure 7, below.

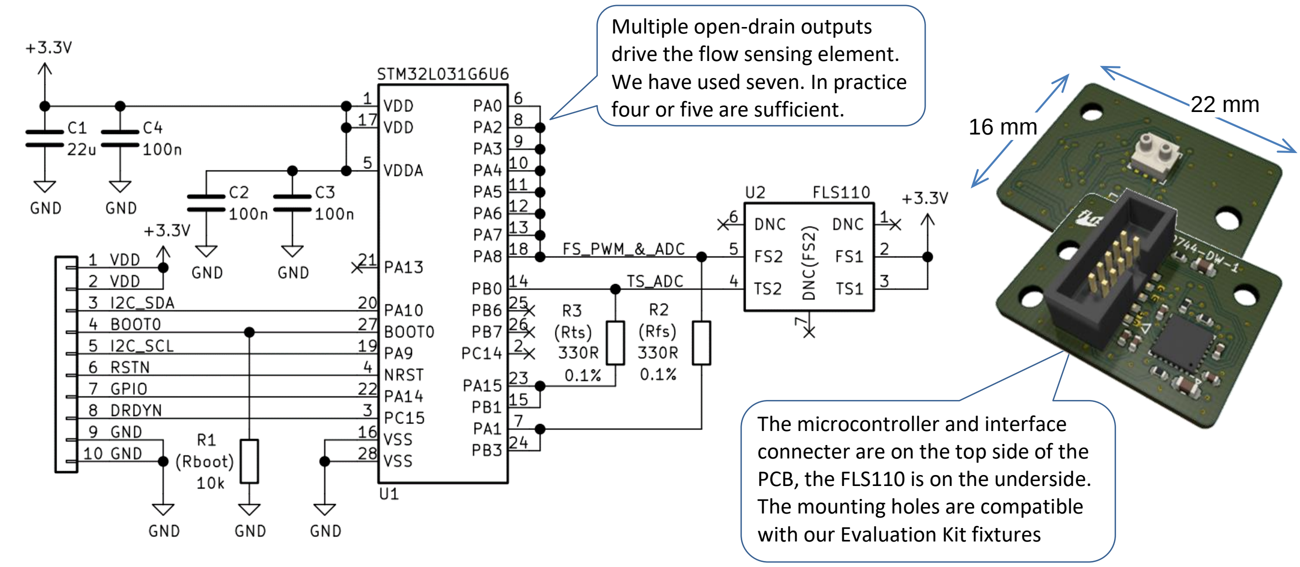

Figure 7: FLS110-STM32 reference design schematic and module implementation

The first step for your own design is choosing either a dedicated or shared microcontroller architecture. A **dedicated microcontroller** solution based on the FLS110-STM32 reference design would have a number of advantages, for example:

1. The hardware design and firmware are already available and can be adapted as necessary to your specific needs – refer to the *FLS110 Hardware Design Guide* and *Firmware Integration Guide*.
2. The host processor does not need to be close to the FLS110 because the I²C-bus interface can operate reliably over a long distance.
3. Your overall development time and effort will be minimised.

A **shared microcontroller** architecture (see Figure 4) requires more development effort but could be more cost-effective in your circumstances if

1. There is already a supported microcontroller (e.g., STM32 family) in your product design that
   i. Has the resources needed to operate the FLS110 as well as run your application code (or you can upgrade it to the next family member at little additional cost).
   ii. Can be close enough to the FLS110 to preserve signal integrity at the analogue interface.

Or

2. Your application requirements can actually be met with the spare resources of the STM32L031G6U6 microcontroller in the FLS110-STM32 reference design, which makes your development process one of adaptation, rather than starting from a “clean sheet”.

FLS110 firmware is provided as an SDK project with a linkable library and source code modules that you can customise. The *FLS110 Firmware Integration Guide* explains how you can customise the standard flow sensing firmware and build it for your hardware design in an industry standard development environment.

5 Product testing

The next step is thorough testing of your fully integrated product design – effectively repetition and extension of the testing you did in step 3, on a larger number of prototype units. So, it's most likely you will automate your testing and the calibration procedures mentioned in section 3.2 (and fully described in *FLS110 System Characterisation and Calibration*). Your test software might again be based on our Python test scripts,– augmented and modified for your particular test setup and test sequences. Please refer to *FLS110 Python Test Script User Guide* for more information.

6 Product manufacture

The FLS110 is designed for use in cost-sensitive, high-volume products, so has been made fully compatible with standard processes and equipment for surface mount device handling and assembly. For example, it is supplied in tape and reel for automated pick and place and is IR reflow solderable. The optional fluidic sealing solution we have developed is also "SMT-friendly" (see *Mechanical and Fluidic Integration of the FLS110* for more details).

Of course, like any sensor with an opening in the package, some care is needed during handling and PCB assembly to avoid ingress of dust particles or other contaminants. The *FLS110 Miniature Gas Flow Sensor Datasheet* provides more information about this.

You use the procedures described *FLS110 System Characterisation and Calibration* in your production test to determine offsets do the system-level calibration (if any) necessary to achieve your required flow sensing accuracy. You will be using your own test software, perhaps based again on the FLS110 Python Test Scripts we provide.