

Pressure Sensor

The pressure sensors are analog force sensing resistor (FSR) pads read via ADC, intended primarily for robotic gripper force feedback: grip force sensing, object presence detection, and load distribution across two gripper pads. The system supports a dual sensor configuration.

Hardware Specifications

Parameter	Value
Sensor Count	2 (independent)
Interface	Analog ADC (compile gated, see Hardware Status below)
Board pins	PD15 (FORCE_ANALOG_DATA_1), PF3 (FORCE_ANALOG_DATA_2)
Output unit	kPa via linear conversion, plus raw voltage and temperature field

Hardware Status

The ADC path is compile gated by `PRESSURE_USE_ADC` because no ADC is enabled in CubeMX yet. Without the flag, `poll_pressure_sensor()` returns `RESULT_ERR_UNIMPLEMENTED` and both sensors report `IDLE / DISCONNECTED` (the firmware still links). To enable:

1. In CubeMX enable an ADC and the channel(s) for the force pins. On the STM32H753, PD15 has NO ADC function and PF3 = ADC3_INP5, so `FORCE_ANALOG_DATA_1` must be moved to an ADC-capable pin.
2. Build with `-D PRESSURE_USE_ADC`.
3. Bind each unit with `pressure_sensor_init_hw()`.

Conversion Model

```
voltage      = raw / adc_max × reference_voltage
pressure_kpa = voltage × scale_kpa_per_volt + offset_kpa
```

Defaults: `scale_kpa_per_volt = 1.0`, `offset_kpa = 0.0` (passthrough until calibrated).

Data Structure

```
typedef struct {
    float pressure_kpa;
    float temperature_c;
    float voltage;
    bool is_calibrated;
    bool read_ok;           /* true if the last poll read succeeded */

    /* ADC binding (set by pressure_sensor_init_hw) */
    void *adc_handle;       /* ADC_HandleTypeDef* (void* keeps header HAL-free) */
    uint32_t adc_channel;   /* ADC_CHANNEL_x */
    uint32_t adc_max;       /* full-scale count (e.g. 65535 for 16-bit) */
    float reference_voltage; /* ADC Vref+ in volts */
    float scale_kpa_per_volt; /* linear gain (default 1.0) */
    float offset_kpa;        /* linear offset (default 0.0) */
} pressure_sensor_data_t;
```

Initialization

Initialize Pressure Sensors (as in main.c)

```
pressure_sensor_data_t pressure_data[2];
for (size_t i = 0; i < 2; i++) {
    pressure_sensor_init(&pressure_data[i]);
}

/* Once an ADC exists, bind it per unit: */
pressure_sensor_init_hw(&pressure_data[1], &hadc3, ADC_CHANNEL_5,
                       65535U, 3.3f);
```

Poll Pressure Sensor

```
result_t pr_result = poll_pressure_sensor(&pressure_data[i]);
```

Data Access Functions

```
float kpa, temp_c, voltage;
bool valid;

pressure_sensor_get_pressure_kpa(&sensor, &kpa);
pressure_sensor_get_temperature_c(&sensor, &temp_c);
pressure_sensor_get_voltage(&sensor, &voltage);
pressure_sensor_is_valid(&sensor, &valid);
```

Calibration

```
/* kPa = V × scale + offset; marks the sensor calibrated */
pressure_sensor_set_calibration(&sensor, scale_kpa_per_volt, offset_kpa);
```

Pressure Unit Conversions

From	To	Multiply By
bar	kPa	100
psi	kPa	6.895
atm	kPa	101.325
kPa	bar	0.01
kPa	psi	0.145
kPa	atm	0.00987

Function-based conversions (bar_to_psi, psi_to_bar) are declared in the utility library but currently commented out; see Sensor Board Utility Library.

Protobuf Message Format

```
message SensorBoardPressureInfo {
  uint32 sensor_index;          /* 0 or 1 */
  float pressure_kpa;
```

```
float temperature_c;
float voltage;
bool is_calibrated;
SensorState state;
PressureErrorCode error_code;    /* NO_ERROR, COMMUNICATION_FAILURE, INVALID_DATA */
}
```

Applications

Robotic Gripper Control (Primary Use Case)

- Grip force feedback for object handling
- Object presence detection (pressure spike threshold)
- Adaptive compliance for varying object sizes and materials
- Dual sensors support load sharing across gripper pads

Possible Secondary Uses (not implemented)

- Depth sensing (water), altitude sensing (air), system pressure monitoring, if a suitable transducer replaces the FSR pads

Integration Notes

- Two independent units polled every main loop iteration, each transmitted in its own envelope with its sensor_index (logged under the name "Force0"/"Force1")
- Each sensor maintains independent calibration and error reporting
- The temperature_c field exists for future compensation algorithms; no temperature source is wired up yet
- Until the ADC is enabled the sensors are harmless placeholders: IDLE / DISCONNECTED, zeroed values

Revision #4

Created 2026-04-14 15:54:05 UTC by Shishir Nambiar

Updated 2026-07-08 14:29:35 UTC by Shishir Nambiar