

Pressure Sensor

Pressure sensors measure fluid/gas pressure for robotic gripper force feedback and control, depth sensing, altitude measurement, or system pressure monitoring. The system supports dual pressure sensor configuration for dual-pad gripper control with load distribution feedback.

Hardware Specifications

Parameter	Value
Sensor Count	2 (independent)
Interface	Analog ADC or I2C
Measurement Range	Variable (typically 0-300 kPa)
Update Rate	Configurable

Data Structure

```
typedef struct {  
    float pressure_kpa;           // Pressure in kilopascals (kPa)  
    float temperature_c;         // Temperature in Celsius (°C)  
    float voltage;               // Sensor output voltage  
    bool is_calibrated;          // Calibration status flag  
} pressure_sensor_data_t;
```

Initialization

Initialize Pressure Sensors

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

Poll Pressure Sensor

```
result_t ps_result = poll_pressure_sensor(&pressure_data[0]);

if (ps_result == RESULT_OK) {
    float pressure_kpa = pressure_data[0].pressure_kpa;
    float temperature_c = pressure_data[0].temperature_c;
}
```

Data Access Functions

```
// Get pressure in kilopascals
result_t pressure_sensor_get_pressure_kpa(
    const pressure_sensor_data_t *data,
    float *pressure_kpa
);

// Get temperature in Celsius
result_t pressure_sensor_get_temperature_c(
    const pressure_sensor_data_t *data,
    float *temperature_c
);

// Get raw sensor voltage
result_t pressure_sensor_get_voltage(
    const pressure_sensor_data_t *data,
    float *voltage
);

// Verify sensor validity
result_t pressure_sensor_is_valid(
    const pressure_sensor_data_t *data,
    bool *is_valid
);
```

Pressure Unit Conversions

Function-Based Conversions

```
// Convert pressure from bar to psi
result_t bar_to_psi(float bar, float *psi);

// Convert pressure from psi to bar
result_t psi_to_bar(float psi, float *bar);
```

Conversion Table

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

Examples

```
// 100 kPa = 1 bar
float kpa = 100.0f;
float bar = kpa * 0.01f; // Result: 1.0 bar

// 50 psi to bar
float psi = 50.0f;
float bar = psi / 14.504f; // Result: 3.45 bar

// Altitude from pressure (simplified)
// Altitude ≈ 44330 × (1 - (P/P0)^(1/5.255))
float altitude_m = 44330.0f * (1.0f - pow(pressure_kpa/101.325f, 1.0f/5.255f));
```

Protobuf Message Format

```
message SensorBoardPressureInfo {
    uint32 sensor_index;          // 0 or 1
    float pressure_kpa;
    float temperature_c;
    SensorState state;
    PressureErrorCode error_code;
}
```

Dual Sensor Management

Configuration Example

```
// Initialize both sensors
for (size_t i = 0; i < 2; i++) {
    pressure_sensor_init(&pressure_data[i]);
}

// Poll both in sequence
poll_pressure_sensor(&pressure_data[0]);
poll_pressure_sensor(&pressure_data[1]);

// Access by index
float pressure_0_kpa = pressure_data[0].pressure_kpa;
float pressure_1_kpa = pressure_data[1].pressure_kpa;
```

Temperature Compensation

Pressure readings often need temperature compensation for accuracy:

```
// Simplified temperature compensation
float compensated_pressure = pressure_data[0].pressure_kpa *
    (reference_temperature + 273.15f) /
    (pressure_data[0].temperature_c + 273.15f);
```

Applications

Robotic Gripper Control (Primary Use Case)

```
// Gripper force feedback for adaptive grip strength
// Pressure reading controls servo/motor PWM to regulate grip force

#define GRIPPER_MIN_PRESSURE_KPA 20.0f // Minimum safe grip
#define GRIPPER_MAX_PRESSURE_KPA 150.0f // Maximum allowed grip
#define GRIPPER_TARGET_PRESSURE_KPA 80.0f // Desired grip force

// PID controller for gripper force regulation
typedef struct {
    float kp, ki, kd; // PID coefficients
    float integral_error;
    float previous_error;
} gripper_pid_t;

// Adjust servo PWM based on pressure feedback
void adjust_gripper_force(float current_pressure_kpa, gripper_pid_t *pid) {
    float error = GRIPPER_TARGET_PRESSURE_KPA - current_pressure_kpa;

    pid->integral_error += error;
    float derivative_error = error - pid->previous_error;

    float pid_output = (pid->kp * error) +
        (pid->ki * pid->integral_error) +
        (pid->kd * derivative_error);

    // Clamp servo PWM to valid range
    uint16_t servo_pwm = (uint16_t)(GRIPPER_NEUTRAL_PWM + pid_output);
    servo_pwm = (servo_pwm < GRIPPER_MIN_PWM) ? GRIPPER_MIN_PWM : servo_pwm;
    servo_pwm = (servo_pwm > GRIPPER_MAX_PWM) ? GRIPPER_MAX_PWM : servo_pwm;

    set_gripper_pwm(servo_pwm);
    pid->previous_error = error;
}
```

Gripper Control Features:

- Grip force feedback for object handling
- Object presence detection (pressure spike threshold)

- Adaptive compliance for varying object sizes/materials
- Dual sensors support load sharing across gripper pads

Implemented but Not Primary

Depth Sensing (Water)

```
// Pressure to depth in water
// P = ρ × g × h
// where ρ = 1025 kg/m³ (seawater), g = 9.81 m/s²
float depth_meters = (pressure_kpa - atmospheric_pressure_kpa) / 10.0f;
```

Altitude Sensing (Air)

```
// Barometric formula (simplified)
float altitude_m = 44330.0f * (1.0f - pow(pressure_kpa/101.325f, 1.0f/5.255f));
```

System Pressure Monitoring

```
if (pressure_kpa > PRESSURE_WARNING_THRESHOLD) {
    // High pressure detected - safety alert
}
```

Common Pressure Sensor Ranges

Application	Range	Typical Sensor
Altitude (aviation)	10-110 kPa	BMP280/BMP390
Depth (diving)	0-300+ kPa	Custom depth sensor
System pressure	0-500+ kPa	Industrial pressure transducer

Integration Notes

- **Primary Application:** Robotic gripper force feedback and control
- Supports up to 2 independent pressure sensors (dual gripper pads)
- Temperature measurement for compensation algorithms
- Hardware-specific ADC or I2C implementation
- Pressure-voltage conversion implemented internally
- Real-time depth/altitude sensing capability
- PID control loop integration for adaptive grip force

- Each sensor maintains independent calibration
 - Temperature tracking for accuracy improvements
 - Slip detection via pressure variance analysis
-

Revision #3

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

Updated 2026-04-14 17:17:30 UTC by Shishir Nambiar