

Load Cell

Load cells measure force/weight to detect object presence, evaluate structural loading, or monitor mechanical stress. The system supports a dual load cell configuration, each read through its own HX711 24-bit ADC using GPIO bit-banging. This is the most complete sensor driver on the board: it talks to real hardware with no compile gate.

Hardware Specifications

Parameter	Value
Sensor Count	2 (independent)
ADC	HX711 24-bit, one per load cell
Interface	GPIO bit-bang (DOUT input, SCK output)
Gain	Channel A, gain 128 (LOAD_CELL_GAIN_PULSES = 1)
Measurement	Force (Newtons) / Mass (grams) after calibration; raw counts always available
Ready timeout	200 ms (LOAD_CELL_READY_TIMEOUT_MS)
Supply	HX711 VCC 2.7 to 5.5 V, GND common with the STM32

Wiring and Pin Map (from firmware.ioc)

Unit	HX711 DOUT (data, low = ready)	HX711 SCK (clock)
0	PA5 (WEIGHT_INPUT_1)	PC7 (WEIGHT_CLOCK_1)
1	PA6 (WEIGHT_INPUT_2)	PB5 (WEIGHT_CLOCK_2)

Data Structure

```
typedef struct {
    int32_t raw_counts;           /* 24-bit two's complement reading */
    float force_newtons;
    float mass_grams;
    float scale_newtons_per_count; /* default 1.0 (passthrough) */
    int32_t tare_offset_counts;
```

```

bool is_calibrated;           /* true once load_cell_set_scale() called */
bool read_ok;                 /* true if the last poll succeeded */

/* HX711 hardware binding (set by load_cell_sensor_init_hw) */
GPIO_TypeDef *dout_port;
uint16_t dout_pin;
GPIO_TypeDef *sck_port;
uint16_t sck_pin;
uint8_t gain_pulses;
} load_cell_data_t;

```

Initialization

Initialize Load Cells (as in main.c)

```

load_cell_data_t load_cell_data[2];

/* Unit 0: DOUT = PA5, SCK = PC7. Unit 1: DOUT = PA6, SCK = PB5. */
load_cell_sensor_init_hw(&load_cell_data[0], GPIOA, GPIO_PIN_5, GPIOC, GPIO_PIN_7);
load_cell_sensor_init_hw(&load_cell_data[1], GPIOA, GPIO_PIN_6, GPIOB, GPIO_PIN_5);
/* init_hw powers up the HX711 and auto-tares */

/* Alternative: load_cell_sensor_init(&data) zero-initialises WITHOUT a
 * hardware binding; poll() then returns RESULT_ERR_UNIMPLEMENTED */

```

Poll Load Cell Sensor

```

result_t lc_result = poll_load_cell_sensor(&load_cell_data[i]);

```

Data Access Functions

```

float force, scale;
float mass;
int32_t counts, tare;
bool valid;

```

```
load_cell_get_force_newtons(&cell, &force);
load_cell_get_mass_grams(&cell, &mass);
load_cell_get_raw_counts(&cell, &counts);
load_cell_get_calibration(&cell, &scale, &tare);
load_cell_sensor_is_valid(&cell, &valid);
```

Calibration Procedure

Two-Step Calibration

Step 1: Tare (Zero Load)

```
/* With nothing on the cell: average N raw reads and store as zero offset.
 * init_hw already does this automatically at startup. */
load_cell_tare(&cell, 10);
```

Step 2: Span (Known Weight)

```
/* Place a known mass, read raw counts, then:
 *   scale = known_force_newtons / (raw_counts - tare_offset_counts) */
load_cell_set_scale(&cell, newtons_per_count); /* sets is_calibrated = true */
```

Measurement Formulas

```
force_newtons = (raw_counts - tare_offset_counts) × scale_newtons_per_count
mass_grams    = force_newtons / 9.81 × 1000
```

Note: before calibration the default scale is 1.0 (passthrough): raw_counts is trustworthy but force_newtons and mass_grams are not physical units yet.

Protobuf Message Format

```
message SensorBoardLoadCellInfo {
  uint32 sensor_index;          /* 0 or 1 */
  float force_newtons;
```

```
float mass_grams;
int32 raw_counts;
float scale_newtons_per_count;
int32 tare_offset_counts;
bool is_calibrated;
SensorState state;
LoadCellErrorCode error_code;  /* NO_ERROR, COMMUNICATION_FAILURE, INVALID_DATA */
}
```

Unit Conversions

From	To	Factor
Newtons	kilograms-force (kgf)	÷ 9.81
Newtons	pounds-force (lbf)	÷ 4.448
grams	kilograms	÷ 1000

Integration Notes

- Two independent units polled every main loop iteration; each is transmitted in its own envelope with its sensor_index
- Each sensor maintains separate calibration (tare + scale) and independent error reporting
- HX711 gain/channel is selected by extra SCK pulses after the 24 data bits (1 = channel A gain 128, 2 = channel B gain 32, 3 = channel A gain 64)
- The blocking wait for data-ready can take up to 200 ms per cell per poll; keep this in mind when reducing the loop interval
- A failed read maps to state ERROR with LOAD_CELL_COMMUNICATION_FAILURE; a read with implausible data maps to LOAD_CELL_INVALID_DATA