

# Configuration

Compile-time parameters, runtime configuration, sensor calibration setup, network addressing, UDP ports, performance tuning options

## UDP Configuration

```
// In src/sensor_board/main.c
#define SENSOR_UDP_DEST_PORT 7
uint8_t dest_ip[4] = {192, 168, 0, 255}; // Broadcast address
```

## Main Loop Timing

```
// In src/sensor_board/main.c
#define MAIN_TASK_DELAY_MS 5000 // 5 seconds between updates
```

## Heap Management

```
// Critical heap threshold
#define CRITICAL_HEAP_THRESHOLD 8192 // 8 KB minimum
```

## Task Configuration

```
const osThreadAttr_t mainTask_attributes = {
    .name = "mainTask",
    .stack_size = 1024 * 8, // 8 KB stack
    .priority = (osPriority_t)osPriorityNormal,
};
```

# Runtime Configuration

## Changing Sensor Poll Interval

```
// Current: 5 seconds
#define MAIN_TASK_DELAY_MS 5000

// Change to 1 second
```

```
#define MAIN_TASK_DELAY_MS 1000

// Change to 10 seconds
#define MAIN_TASK_DELAY_MS 10000
```

## Enabling/Disabling Sensors

```
// Current: skip all sensor polling during development
const bool skip_sensor_polling = true;

// To ENABLE actual polling:
const bool skip_sensor_polling = false;

// To SKIP (for testing without hardware):
const bool skip_sensor_polling = true;
```

## Adjusting Heap Threshold

```
// Current: 8 KB critical
if (free_heap < 8192) {
    LOGE(TAG, "CRITICAL: Low heap detected! Free: %lu bytes", free_heap);
}
```

# Sensor Calibration Configuration

## pH Sensor Calibration

```
// Set reference voltage (typically 3.3V or 5.0V)
ph_sensor_init(&ph_sensor, 3.3f); // 3.3V reference

// Update calibration (after two-point calibration)
ph_sensor.calibration.offset = 0.0f; // Adjust after pH 7.0 test
ph_sensor.calibration.slope = 3.5f; // Adjust after second pH test
```

## Load Cell Calibration

```
// After tare (no load):  
load_cell_data[0].tare_offset_counts = adc_reading_no_load;  
  
// After known weight measurement:  
// scale = (force_newtons) / (adc_reading - tare_offset)  
load_cell_data[0].scale_newtons_per_count = scale_factor;  
load_cell_data[0].is_calibrated = true;
```

# Network Configuration

## Changing Broadcast Address

```
//To be implemented
```

## Changing UDP Port

```
//To be implemented
```

---

Revision #5

Created 2026-04-14 16:05:24 UTC by Shishir Nambiar

Updated 2026-04-14 17:20:27 UTC by Shishir Nambiar