

# Testing

Test organization, validation functions, Unity testing framework usage, manual hardware testing checklists and debugging/troubleshooting guide.

## Test Organization

```
test/sensor_board/  
├─ test_gps_sensor/  
|   ├─ NMEA parsing verification  
|   ├─ Coordinate validation  
|   └─ Fix quality enumeration tests  
|  
├─ test_imu_sensor/  
|   ├─ 3-axis data structure tests  
|   ├─ Acceleration magnitude calculation  
|   └─ Timestamp tracking validation  
|  
├─ test_load_sensor/  
|   ├─ Force calculation (tare/scale)  
|   ├─ Mass conversion (g/kg/lbf)  
|   ├─ Dual sensor independence  
|   └─ Calibration parameter storage  
|  
├─ test_ph_sensor/  
|   ├─ ADC to voltage conversion  
|   ├─ pH value calculation  
|   ├─ Sample averaging (40-sample buffer)  
|   ├─ Calibration offset/slope  
|   └─ Edge cases (pH 0, 14)  
|  
├─ test_pressure_sensor/  
|   ├─ Pressure reading validation  
|   ├─ Temperature compensation  
|   ├─ Unit conversions (bar/psi/kPa)  
|   └─ Dual sensor independence
```

```
|
└─ test_sensor_basics/
    └─ Conversion functions
    └─ Validation functions
    └─ Range checking
    └─ Boundary conditions
```

# Validation Functions

## GPS Validation

```
result_t validate_gps_latitude(double latitude);
// Valid range: -90.0 to +90.0 degrees

result_t validate_gps_longitude(double longitude);
// Valid range: -180.0 to +180.0 degrees

result_t validate_gps_hdop(float hdop);
// Typical range: 0.0 to 50.0

result_t validate_gps_satellite_count(int32_t satellites);
// Typical range: 0 to 30 satellites
```

## pH Validation

```
result_t validate_ph_value(float ph_value);
// Valid range: 0.0 to 14.0
```

## IMU Validation

```
result_t validate_accelerometer_value(float accel_value);
// Typical range: -50.0 to +50.0 m/s2

result_t validate_imu_data(float accel_x, float accel_y, float accel_z);
```

## Unit Conversion Tests(extra stuff)

```
result_t celsius_to_fahrenheit(float celsius, float *fahrenheit);
result_t fahrenheit_to_celsius(float fahrenheit, float *celsius);
result_t bar_to_psi(float bar, float *psi);
result_t psi_to_bar(float psi, float *bar);
```

# Test Framework

## Testing Technology

- **Framework:** Unity (open-source testing framework)
- **Build System:** CMake / PlatformIO
- **Test Type:** Hosted unit tests (run on PC)
- **Mocking:** Mock ADC/UART/I2C interfaces

## Building Tests

```
// Build all tests
pio test -e sensor_board

// Run specific test suite
pio test -e sensor_board -f test_ph_sensor
```

# Debugging & Troubleshooting

| Issue            | Cause                 | Solution                     |
|------------------|-----------------------|------------------------------|
| Sensor IDLE      | Not connected         | Check UART/I2C/SPI wiring    |
| Sensor ERROR     | Communication failure | Verify baud rates, addresses |
| Invalid data     | Out of range          | Check calibration parameters |
| Low heap warning | Memory leak           | Review packet encoder        |

Revision #5

Created 2026-04-14 15:58:45 UTC by Shishir Nambiar

Updated 2026-04-14 17:19:59 UTC by Shishir Nambiar