precaster

CA111WC

Laser Distance Module

Manual

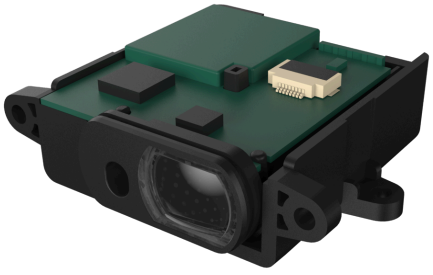


Table of Content

1.INTRODUCTION 2

2.SPECIFICATION 3

3.ELECTRICAL INTERFACE 4

4.I2C MAP ADDRESS 4

5.PROGRAM FLOWCHART 5

6.DIMENSION 12

1. INTRODUCTION

CA111WC is an advanced module based on CA111 as a small compacted and reliable laser distance sensor with a lower operating temperature down to -30°C to 70°C. The phase-shifting reflectorless technology gives good measurement accuracy and response time.

1.1. TARGET AND SCOPE

This specification defines requirements for a small laser distance meter module (LDM) that can be used to mount and be controlled by the master system. The module will be used for two modes. The modes are Mode 1 is Normal Measure Mode with normal accuracy. Mode 2 is Fast Measure Mode with high measurement speed and low accuracy

1.2. FEATURE

i. Special Class 2 laser, can measure from **-30°C to 70°C**.
ii. Digital output with I2C signal and UART
iii. The adjustable speed measures up to 6HZ.
iv. Non-contact measuring low consumption module.

1.3. APPLICATION

i. Robot eyes distance measuring.
ii. Position and monitor of objects.
iii. Security application movement detections.
iv. Level and elevator measurement.

2. Specification

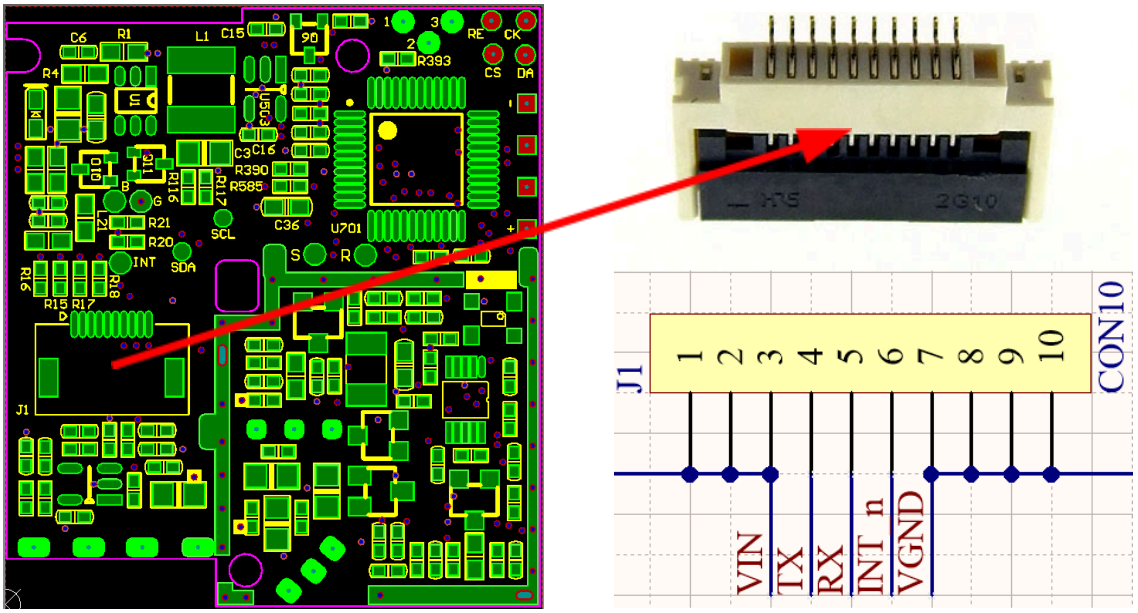
	Normal Measure Mode	Fast Measure Mode
Measure Range	0.05m~40 meters	0.2m~40 meters
Accuracy from 0.05m~30m*	± 2 mm**	N/A
Accuracy from 0.5m~0.5m*	N/A	± 5 mm
Accuracy from 0.2m~0.5m*	N/A	1%
Response Time	1 Hz	6 Hz
Laser Wavelength	650±20nm,	
Laser Power	0.6mW to <1mW	
Laser Spot Size	25mm*30mm @25m	
Power Consumption		
During Measuring	Maximum 130ma	
Laser Off	Maximum 70ma	
When Heater is On	Maximum 330ma	
Supply Voltage	3.0	
Communication protocol	I2C slave, 400 kHz, interrupt register access.	
Spot Vertical Angle	+1 degree	
Spot horizontal Angle	±1.5 degree	
Operation Temperature	-30°C~70°C	
Storage Temperature	-40°C~80°C	

Compliant to	EN60825-1, EN61326-1, FDA, FCC, RoHS, REACH
*Applies for well reflectivity target, in daylight conditions (2000 Lux), 25°C, with a low reflectivity target, could decrease measuring range and increased errors which may defer by target and environment situation, tolerances apply from 0.05m to 10m with a confidence level of 95%.	
**The maximum tolerance may deteriorate by 0.25 mm/m after 10 meters.	

3. Electrical Interface

Connector type Hirose TF31-10S-0.5SH(800) or compatible.

Pin No.	Name	IO Type	Description
1	Vin	Power	Power supply 3.0V
2	Vin	Power	Power supply 3.0V
3	Vin	Power	Power supply 3.0V
4	SCL	IO	I2C clock. Open collector pulled up to 3.0V at master
5	SDA	IO	I2C clock. Open collector pulled up to 3.0V at master
6	INT_n	O	Active low interrupt. Open collector, pulled up to 3.0V at master
7	GND	Power	Power Ground
8	GND	Power	Power Ground
9	GND	Power	Power Ground
10	GND	Power	Power Ground



4. Transfer Mode

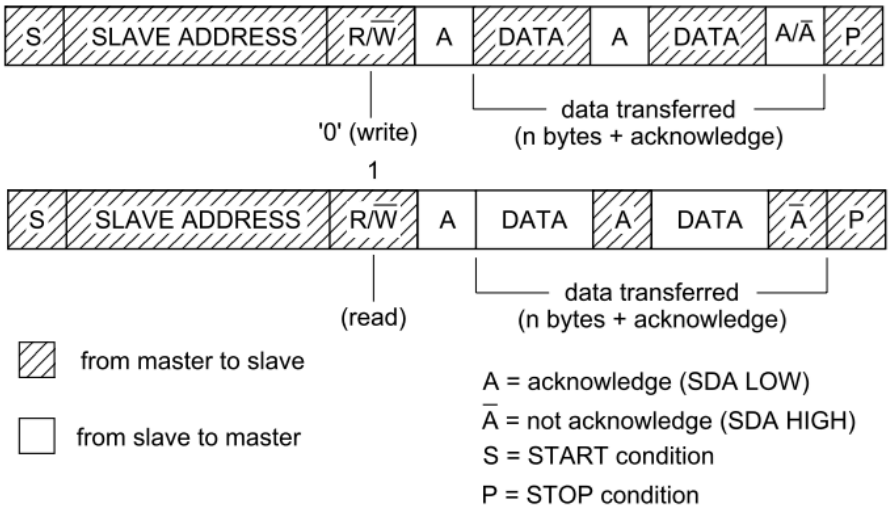
4.1. Data Communication interface:

i. I2C interface

ii. Fase-mode (up to 400kHZ)

iii. Device Addresses 76h

iv. SAD+W= 0xEC; SAD+R=0xED;



4.2. **Transmission Format**:

I2C Terms

Term	Definition
S	Start Condition
SAD	Slave Address
W	Write Bit=0
R	Read Bit=1
ACK	Acknowledge
NACK	Not Acknowledge
Data	Transmitted/Received Data

P	Stop Condition

The Master is writing multiple bytes to the Slave.

Master	S	SAD+W		DATA		P
Slave			ACK		ACK	

Master	S	SAD+W		DATA		DATA		P
Slave			ACK		ACK		ACK	

Master	S	SAD+W		DATA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

The Master is receiving multiple bytes of data from the Slave.

Master	S	SAD+R			ACK		ACK		ACK		NACK	P
Slave			ACK	DATA		DATA		DATA		DATA		

4.3. Command

Writing in Data Byte 1 as the command.

I2C Address (hex)	Command (hex)	Function
0xEC	0x6C	Laser turn on
0xEC	0x6D	Single measurement (high accuracy and low speed)
0xEC	0x63	Continuously measurement (high accuracy and low speed)
0xEC	0x4D	Fast single measurement (low accuracy and high speed)
0xEC	0x43	Fast continuously measurement (low accuracy and high speed)
0xEC	0x72	Laser status return
0xEC	0x73	Laser turn off or stop continuously measurement
0xEC	0x77	Modify the slave address
0xEC	0x54	Measure temperature
0xEC	0x55	Turn on temperature alert
0xEC	0x56	Turn off temperature alert
0xEC	0x57	Temperature alert status
0xEC	0x58	Modified alert temperature (lower limit)
0xEC	0x80	Heater on control
0xEC	0x81	Heater off control
0xEC	0x82	Heater status return
0xED	Read 3 bytes	Result output

CA11WC-20240124

Master	S	SAD+W		Command		P
Slave			ACK		ACK	

CA11WC-20240124

58 (hex)：Modified alert temperature

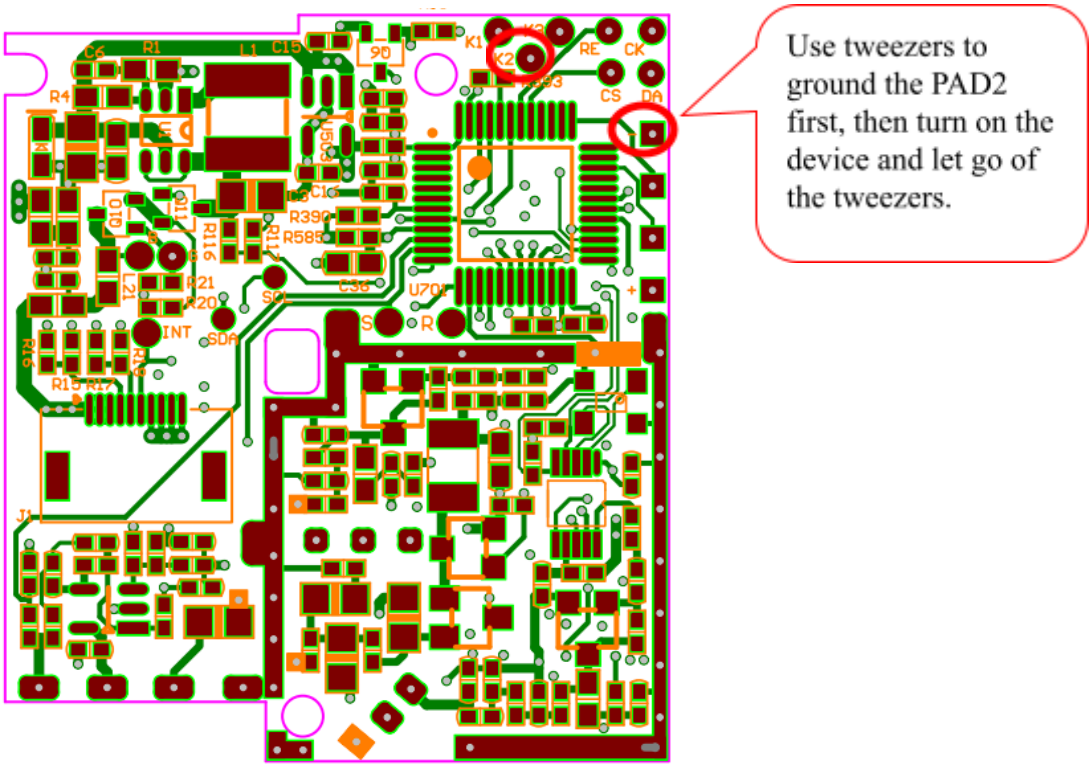
Master	S	SAD + W		58 (Byte 1)		Data (Byte 2)		Data (Byte 3)		P
Slave			ACK		ACK		ACK		ACK	

Byte 2 and Byte 3 are for length (2 bytes Integer).
Byte 2 is Most Significant Byte (MSB)
Byte 3 is Least Significant Byte (LSB)
Range: 0000h~7FFFh = 0 ~ 32767 (Unit: 0.01°C) = 0 ~ 327.67 (Unit: °C)
Range: 8000h~FFFFh = -32768 ~ -1 (Unit: 0.01°C) = -327.68 ~ -0.01 (Unit: °C)

77 (hex)：Modify the slave address (Must restart the device)

Master	S	SAD + W		77		addresses		P
Slave			ACK		ACK		ACK	

Reset slave address：



4.4. Data Format, length is 4 Bytes

Master	S	SAD+R			ACK		ACK		ACK		NACK	P
Slave			ACK	DATA Byte 1		DATA Byte2		DATA Byte 3		DATA Byte 4		

Byte1 : Most Significant Byte (MSB)
Byte2 : Least Significant Byte (LSB)
Byte3 : Error Code
Byte4 : Signal Strength

4.4.1 Distance measurement.

Byte 1 and Byte 2 are for length (2 bytes Integer).
Range : 0000h ~ FFFFh = 0~65535(mm)
Byte 1 is Most Significant Byte (MSB)
Byte 2 is Least Significant Byte (LSB)

Example : Length Data is 6573h

Byte 1	Byte 2	Byte 3	Byte 4
65h	73h	00h	00h

MSB = 65h
LSB = 73h
6573h = 25971(Unit : mm)

4.4.2 Temperature measurement.

Byte 1 and Byte 2 are for temperature (2 bytes Integer).
Range : 0000h~7FFFh = 0 ~ 32767 (Unit : 0.01°C) = 0 ~ 327.67 (Unit : °C)
Range : 8000h~FFFFh = -32768 ~ -1 (Unit : 0.01°C) = -327.68 ~ -0.01 (Unit : °C)
Byte 1 is Most Significant Byte (MSB)
Byte 2 is Least Significant Byte (LSB)

Example : Temperature Data is F211h

Byte 1	Byte 2	Byte 3	Byte 4
F2h	11h	54h	00h

MSB = F2h
LSB = 11h

F211h = -3567 (Unit : 0.01°C) = -35.67 (Unit : °C)

4.4.3 Byte 3 is Error Code

In normal situation, the byte 3 data is 00h.

When the following situation occurred, will have error codes:

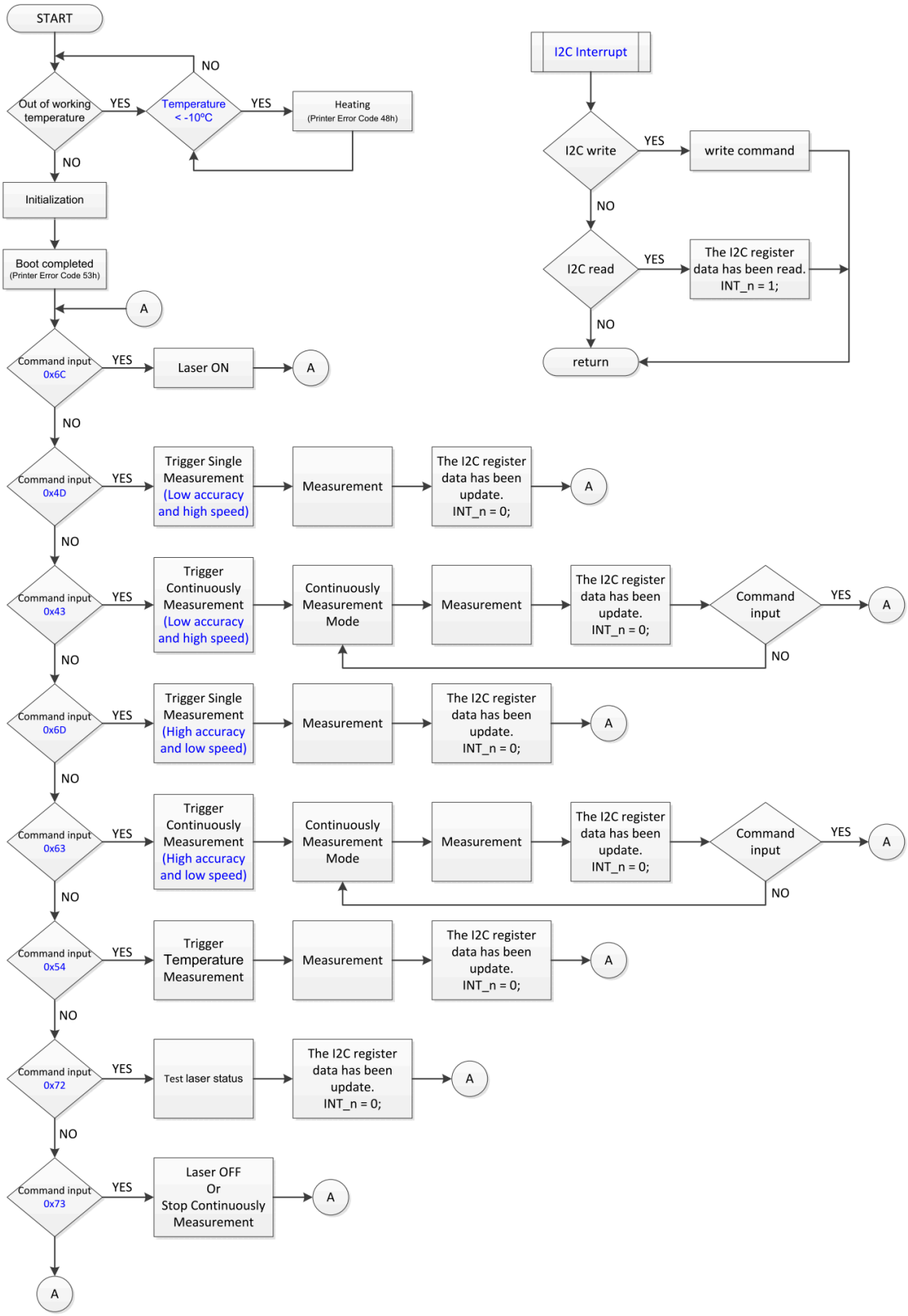
00h: Normal
01h: Out of range
02h: Signal is too weak
05h: Out of operating voltage
06h: Out of working temperature
09h: Out of laser power
0Ah: Laser turn on
0Bh: Laser turn off
0Ch: Weak signal process increase the laser power
53h: Boot completed
54h: Measure temperature
55h: Turn on temperature alert
56h: Turn off temperature alert
80h: Heater on control
81h: Heater off control

4.4.4 Byte 4 is Signal Strength

The signal level range is 0~255

Represented as 00h ~ FFh

5. Program Flowchart



5. Reference Code

The following code is provided as a platform-independent reference for CA111W-C I²C integration. The customer must implement the platform-specific I²C read, I²C write, and measurement-ready functions according to the host controller and software environment. The CA111W-C returns four bytes for measurement results: Distance MSB, Distance LSB, Status Code, and Signal Strength.

```
#include <stdint.h>
#include <stddef.h>
/*
 * CA111W-C Reference Code
 *
 * Default 7-bit I2C address: 0x76
 *
 * The values 0xEC and 0xED shown in the communication diagrams
 * are the corresponding 8-bit write and read address bytes.
 */

#define CA111WC_DEFAULT_I2C_ADDR       0x76U

/* Common commands */
#define CA111WC_CMD_LASER_ON           0x6CU
#define CA111WC_CMD_SINGLE_NORMAL      0x6DU
#define CA111WC_CMD_CONTINUOUS_NORMAL  0x63U
#define CA111WC_CMD_SINGLE_FAST        0x4DU
#define CA111WC_CMD_CONTINUOUS_FAST    0x43U
#define CA111WC_CMD_LASER_STATUS       0x72U
#define CA111WC_CMD_STOP_LASER_OFF     0x73U
#define CA111WC_CMD_MEASURE_TEMP       0x54U
```

```
/* CA111W-C specific commands */
#define CA111WC_CMD_CHANGE_ADDRESS     0x77U
#define CA111WC_CMD_TEMP_ALERT_ON      0x55U
#define CA111WC_CMD_TEMP_ALERT_OFF     0x56U
#define CA111WC_CMD_TEMP_ALERT_STATUS  0x57U
#define CA111WC_CMD_SET_TEMP_LIMIT     0x58U
#define CA111WC_CMD_HEATER_ON          0x80U
#define CA111WC_CMD_HEATER_OFF         0x81U
#define CA111WC_CMD_HEATER_STATUS      0x82U

/* Status and error codes */
#define CA111WC_STATUS_NORMAL          0x00U
#define CA111WC_STATUS_OUT_OF_RANGE    0x01U
#define CA111WC_STATUS_WEAK_SIGNAL     0x02U
#define CA111WC_STATUS_BAD_VOLTAGE     0x05U
#define CA111WC_STATUS_BAD_TEMP        0x06U
#define CA111WC_STATUS_LASER_POWER     0x09U
#define CA111WC_STATUS_LASER_ON        0x0AU
#define CA111WC_STATUS_LASER_OFF       0x0BU
#define CA111WC_STATUS_SIGNAL_PROCESS  0x0CU
#define CA111WC_STATUS_BOOT_COMPLETE   0x53U
#define CA111WC_STATUS_TEMP_RESULT     0x54U
#define CA111WC_STATUS_TEMP_ALERT_ON   0x55U
#define CA111WC_STATUS_TEMP_ALERT_OFF  0x56U
#define CA111WC_STATUS_HEATER_ON       0x80U
#define CA111WC_STATUS_HEATER_OFF      0x81U

typedef struct {
    uint8_t i2c_address;
```

```
} ca111wc_device_t;

typedef struct {
    uint16_t distance_mm;
    uint8_t status;
    uint8_t signal_strength;
} ca111wc_result_t;

typedef struct {
    int16_t temperature_x100;
    uint8_t status;
    uint8_t signal_strength;
} ca111wc_temperature_t;

/*
 * Implement these functions using the host platform's I2C driver.
 *
 * The address parameter is the 7-bit I2C address.
 */
int platform_i2c_write(uint8_t address,
                       const uint8_t *data,
                       uint16_t length);

int platform_i2c_read(uint8_t address,
                      uint8_t *data,
                      uint16_t length);

/*
 * Wait until the measurement result is ready.
```

```
 *
 * The preferred method is to monitor the active-low INT_n pin.
 * A platform-specific delay may be used for initial evaluation.
 */
void platform_wait_measurement_ready(void);

/*
 * Send one command byte.
 */
int ca111wc_send_command(const ca111wc_device_t *device,
                         uint8_t command)
{
    if (device == NULL) {
        return -1;
    }

    return platform_i2c_write(
        device->i2c_address,
        &command,
        1U
    );
}

/*
 * Read the standard 4-byte CA111W-C response:
 *
 * Byte 1: Distance MSB
 * Byte 2: Distance LSB
 * Byte 3: Status or error code
```

```
 * Byte 4: Signal strength
 */
int ca111wc_read_result(const ca111wc_device_t *device,
                        ca111wc_result_t *result)
{
    uint8_t data[4];

    if ((device == NULL) || (result == NULL)) {
        return -1;
    }

    if (platform_i2c_read(
            device->i2c_address,
            data,
            sizeof(data)) != 0) {
        return -2;
    }

    result->distance_mm =
        ((uint16_t)data[0] << 8) | (uint16_t)data[1];

    result->status = data[2];
    result->signal_strength = data[3];

    return 0;
}

/*
 * Perform one normal-speed measurement.
```

```
 */
int ca111wc_measure_once(const ca111wc_device_t *device,
                         ca111wc_result_t *result)
{
    int status;

    status = ca111wc_send_command(
        device,
        CA111WC_CMD_SINGLE_NORMAL
    );

    if (status != 0) {
        return -1;
    }

    platform_wait_measurement_ready();

    return ca111wc_read_result(device, result);
}

/*
 * Perform one fast measurement.
 */
int ca111wc_measure_fast_once(const ca111wc_device_t *device,
                              ca111wc_result_t *result)
{
    int status;

    status = ca111wc_send_command(
```

```
        device,
        CA111WC_CMD_SINGLE_FAST
    );

    if (status != 0) {
        return -1;
    }

    platform_wait_measurement_ready();

    return ca111wc_read_result(device, result);
}

/*
 * Read the 4-byte temperature response.
 *
 * Byte 1: Temperature MSB
 * Byte 2: Temperature LSB
 * Byte 3: Status code, normally 0x54
 * Byte 4: Signal strength or reserved status data
 *
 * Temperature unit: 0.01 degrees Celsius
 *
 * Example:
 *   temperature_x100 = -3567
 *   temperature = -35.67 degrees Celsius
 */
int ca111wc_read_temperature(
    const ca111wc_device_t *device,
```

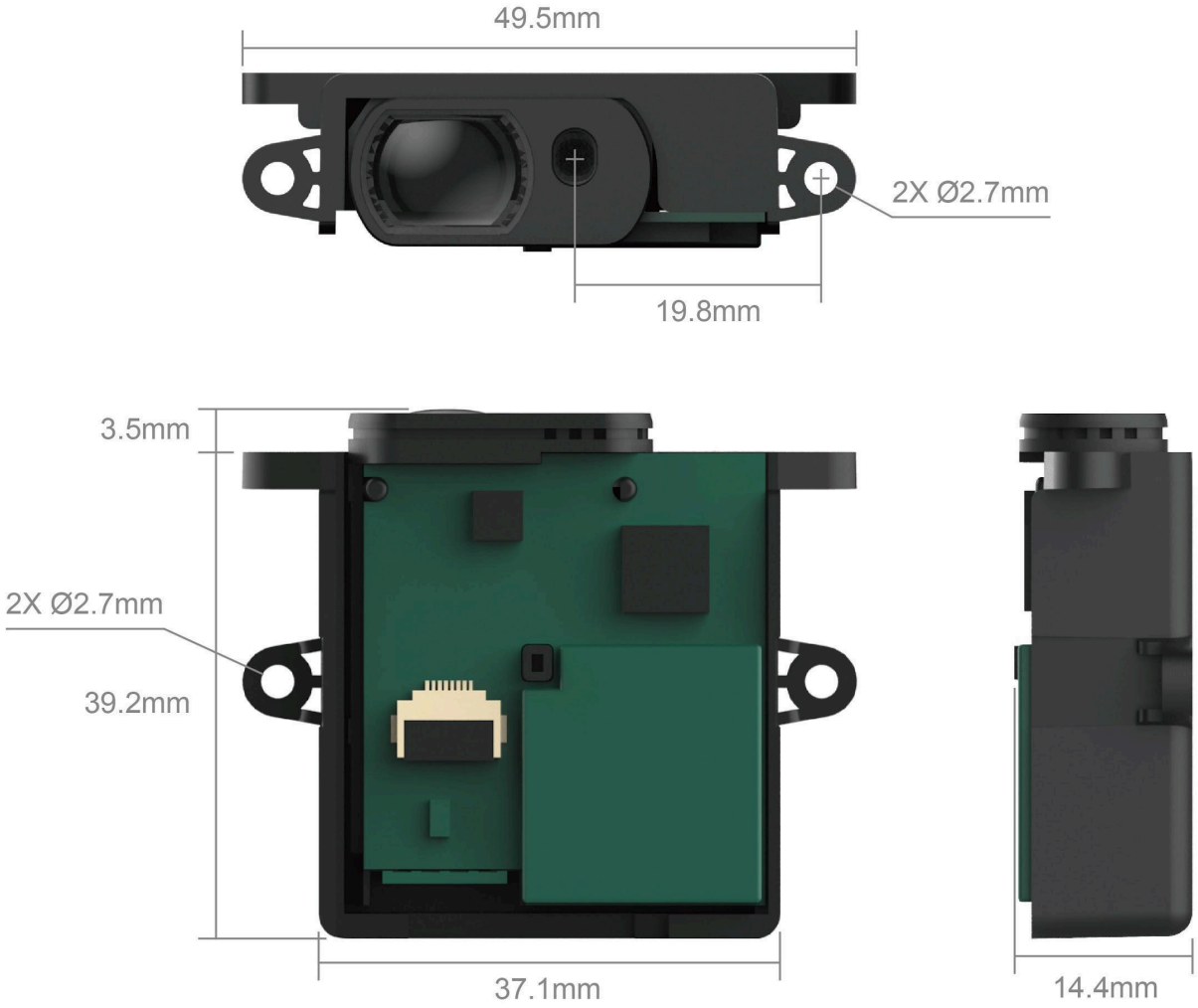
```
    ca111wc_temperature_t *temperature)
{
    uint8_t data[4];
    uint16_t raw_value;

    if ((device == NULL) || (temperature == NULL)) {
        return -1;
    }

    if (ca111wc_send_command(
            device,
            CA111WC_CMD_MEASURE_TEMP) != 0) {
        return -2;
    }

    platform_wait_measurement_ready();

    if (platform_i2c_read(
            device->i2c_address,
            data,
            sizeof(data)) != 0) {
        return -3;
    }

    raw_value =
        ((uint16_t)data[0] << 8) | (uint16_t)data[1];

    temperature->temperature_x100 = (int16_t)raw_value;
    temperature->status = data[2];
```

```
    temperature->signal_strength = data[3];

    return 0;
}

/*
 * Set the lower temperature alert limit.
 *
 * Temperature unit: 0.01 degrees Celsius
 *
 * Examples:
 *    2500 = 25.00 degrees Celsius
 *   -3000 = -30.00 degrees Celsius
 */
int ca111wc_set_temperature_limit(
    const ca111wc_device_t *device,
    int16_t temperature_x100)
{
    uint16_t raw_value;
    uint8_t data[3];

    if (device == NULL) {
        return -1;
    }

    raw_value = (uint16_t)temperature_x100;

    data[0] = CA111WC_CMD_SET_TEMP_LIMIT;
    data[1] = (uint8_t)(raw_value >> 8);
```

```
    data[2] = (uint8_t)(raw_value & 0xFFU);

    return platform_i2c_write(
        device->i2c_address,
        data,
        sizeof(data)
    );
}

/*
 * Change the module's 7-bit I2C address.
 *
 * The module must be restarted after the address is changed.
 *
 * Confirm the permitted address range with the product
 * specification before using this function.
 */
int ca111wc_change_i2c_address(
    const ca111wc_device_t *device,
    uint8_t new_address)
{
    uint8_t data[2];

    if (device == NULL) {
        return -1;
    }

    data[0] = CA111WC_CMD_CHANGE_ADDRESS;
    data[1] = new_address;
```

```
    return platform_i2c_write(
        device->i2c_address,
        data,
        sizeof(data)
    );
}
```

6. Dimension



Precaster Enterprises Co., Ltd.
All Rights Reserved
http://www.precaster.com.tw
contact@precaster.com.tw

Head Office
No.574, Wuquan South Road, South Dist.,Taichung, 402, Taiwan
Tel:+886-42262 1000
Fax:+886-42261 6565

Factory

Rm. 276, 2F., No. 170, Xuefu Rd., South Dist., Taichung, 402 Taiwan