

Tutorial 5 – Sensors

Aim

This tutorial shows you how to use the onboard sensors of the sensortag: On board sensors are:

1. Button Sensor
2. Ambient Light Sensor (ODT 3001)
3. Infrared Thermopile Sensor (TMP 007)
4. Pressure Sensor (BMP 280)
5. Reed Sensor (Magnetic Reed switch)
6. Humidity Sensor (HDC 1000)

Sensor Tag Driver Interface

Due to the time required for the sensor to startup, this driver can be used in an asynchronous fashion.

The caller must first activate the sensor by calling `SENSORS_ACTIVATE()` (except for left_button sensor). This will trigger the sensor's startup sequence, but the call will not wait for it to complete so that the CPU can perform other tasks or drop to a low power mode.

Once the sensor is stable, the driver will generate a **sensors_changed event**, when an sensor read operation is complete.

- Turn the sensor off by calling `SENSORS_DEACTIVATE`, but in order to take **subsequent readings** `SENSORS_ACTIVATE` must be called again.

Table 1: Sensor Definitions and driver (used in `value()` and `SENSORS_ACTIVATE()`)

Sensor	Definition	Sensor Driver
Left Button		<code>button_left_sensor</code>
Ambient Light Sensor	<code>OPT_3001</code>	<code>opt_3001_sensor</code>
Infrared Thermopile	<code>TMP_007</code>	<code>tmp_007_sensor</code>
Pressure Sensor	<code>BMP_280</code>	<code>bmp_280_sensor</code>
Reed Switch Sensor		<code>reed_switch_sensor</code>
Humidity Sensor	<code>HDC_1000</code>	<code>hdc_1000_sensor</code>

Left Button Sensor

You must not call `SENSORS_ACTIVATE()`. The left button sensor will detect when the left button is pressed/released and the duration of the press/released state. The caller should then use `left_button_sensor.value(BUTTON_SENSOR_VALUE_STATE)` to read up or down state (pressed = 0, released = 1). Use the `left_button_sensor.value(BUTTON_SENSOR_VALUE_DURATION)` to measure the duration of the state (in seconds).

Do not use the right button, as this will reset the `sensortag`.

See Example: [contiki-examples/button-input](https://contiki-examples.com/button-input).

Ambient Light Sensor (ODT 3001)

The ambient light sensor will measure the ambient light conditions (in lux). The caller should then use `opt_3001_sensor.value(0)` to read the sensor value and latch them.

Sensor value format: 16-bit integer with 2 decimal point precision. E.g. if value is 10099, actual value is 100.99

Infrared Thermopile Sensor (TMP 007)

The infrared thermopile sensors measure the temperature of an object and the surroundings, by using infrared.

The caller should then use: `tmp_007_sensor.value(TMP_007_SENSOR_TYPE_ALL)` to read sensor values and latch them. Once completed successfully, individual readings can be retrieved with calls to

`tmp_007_sensor.value(TMP_007_SENSOR_TYPE_OBJECT)` or
`tmp_007_sensor.value(TMP_007_SENSOR_TYPE_AMBIENT)`.

Once required readings have been taken, the caller can simply keep calling `tmp_007_sensor.value(TMP_007_SENSOR_TYPE_ALL)` to read and latch new values.

Sensor value format: 16-bit integer with 2 decimal point precision. E.g. if value is 10099, actual value is 100.99

Pressure Sensor (BMP 280)

The pressure barometric sensor will measure the current barometric pressure (in Pa). The caller should then use `value()` function to read the sensor value and latch them.

Once completed successfully, individual readings can be retrieved with calls to `bmp_280_sensor.value(BMP_280_SENSOR_TYPE_PRESS)` to measure the pressure value or `bmp_280_sensor.value(BMP_280_SENSOR_TYPE_TEMP)` to measure the BMP280 temperature value.

Sensor value format: 16-bit integer with 2 decimal point precision. E.g. if value is 10099, actual value is 100.99

Reed Switch Sensor

The reed switch sensor will detect when a strong magnetic field is in proximity. The reed switch will close when a strong magnetic field is detected. The caller should then use `reed_relay_sensor.value(0)` to read the current state of the switch (closed (1) or open (0)).

Humidity Sensor (HDC 1000)

The humidity sensor will measure the current humidity (in RH). The caller should then use `value()` function to read the sensor value and latch them.

Once completed successfully, individual readings can be retrieved with calls to `hdc_1000_sensor.value(HDC_1000_SENSOR_TYPE_HUMIDITY)` to measure the humidity value or `hdc_1000_sensor.value(HDC_1000_SENSOR_TYPE_TEMP)` to measure the HDC1000 temperature value.

Sensor value format: 16-bit integer with 2 decimal point precision. E.g. if value is 10099, actual value is 100.99

Example showing reading from Humidity Sensor (HDC 1000)

Initialising Humidity Sensor.

```
SENSORS_ACTIVATE(hdc_1000_sensor)
```

Reading the Humidity Sensor values (when sensing event occurs)

```
//Wait for sensor value
    if(ev == sensors_event && data == &hdc_1000_sensor) {

        //Read sensor values
        humidity_temp_val=hdc_1000_sensor.value(HDC_1000_SENSOR_TYPE_TEMP);
        humidity_val=hdc_1000_sensor.value(HDC_1000_SENSOR_TYPE_HUMIDITY);

        SENSORS_ACTIVATE(hdc_1000_sensor); //Take sensor readings again
```

Process for Humidity reading sensor continuously.

```
#include "contiki.h"

PROCESS(humidity_process, "sensor process");
AUTOSTART_PROCESSES(&humidity_process);

PROCESS_THREAD(humidity_process, ev, data) {

    PROCESS_BEGIN();

    int humidity_temp_val;
    int humidity_val;

    SENSORS_ACTIVATE(hdc_1000_sensor);    //start reading sensor

    while(1) {

        PROCESS_YIELD();

        //Wait for sensor value
        if(ev == sensors_event && data == &hdc_1000_sensor) {

            //Read sensor values
            humidity_temp_val=hdc_1000_sensor.value(HDC_1000_SENSOR_TYPE_TEMP);
            humidity_val=hdc_1000_sensor.value(HDC_1000_SENSOR_TYPE_HUMIDITY);

            SENSORS_ACTIVATE(hdc_1000_sensor); //Take sensor readings again
        }
    }

    PROCESS_END();
}
```

See Example: [contiki-examples/humidity-sensor](#).

Table 2: Sensor Types

Sensor	Type	Description
Infrared Thermopile (TMP007)	TMP_007_SENSOR_TYPE_OBJECT	Object Temperature
	TMP_007_SENSOR_TYPE_AMBIENT	Ambient Temperature (C)
	TMP_007_SENSOR_TYPE_ALL	
Pressure (BMP280)	BMP_280_SENSOR_TYPE_TEMP	Temperature (C)
	BMP_280_SENSOR_TYPE_PRESS	Pressure (Pa)
Humidity (HDC1000)	HDC_1000_SENSOR_TYPE_TEMP	Temperature (C)
	HDC_1000_SENSOR_TYPE_HUMIDITY	Humidity (RH)

Conclusion

This lesson has introduced the contiki sensor library of the sensortag.

Related Documentation

- Contiki Wiki: <https://github.com/contiki-os/contiki/wiki>