

SKU:SEN0385 (<https://www.dfrobot.com/product-2160.html>)



([https://www.dfrobot.com/product-](https://www.dfrobot.com/product-2160.html)

2160.html)

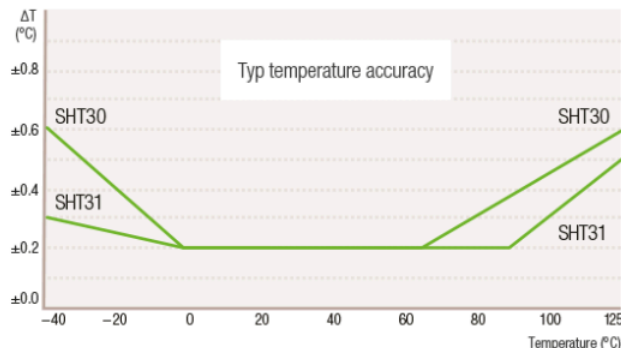
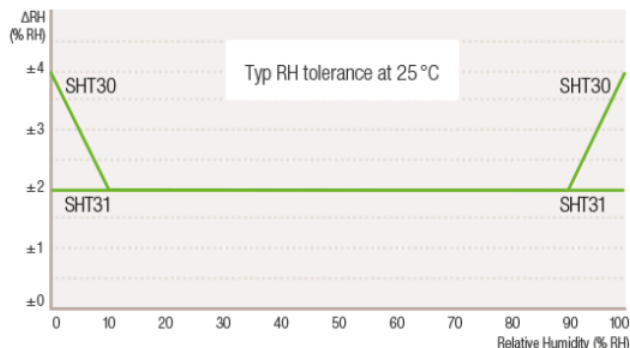
Introduction

This is a SHT31 Temperature & Humidity sensor housed in a weather-proof enclosure. The sensor is designed to be waterproof so you can use it safely in outdoor projects. Please note that although the sensor is waterproof, it is not suggested to submerge it in water. For in-water measurement, DS18B20 (<https://www.dfrobot.com/product-1354.html>) Sensor is highly recommended. Thanks to Sensirion's CMOSens[®] technology, highly integrated capacitive humidity sensing components and band-gap temperature sensing components, the SHT31 offers high reliability and long-term stability with low power consumption, fast response and strong anti-interference ability. The sensor supports I2C communication, and is compatible with 3.3V/5V controllers like Arduino, micro:bit, ESP32.

SHT31 is the standard version of the SHT3x series. It provides humidity accuracy $\pm 2\%RH@0\% RH \sim 100\% RH$ (at 25°C), and temperature accuracy $\pm 0.2^{\circ}C@ 0^{\circ}C \sim 90^{\circ}C$

(typical).

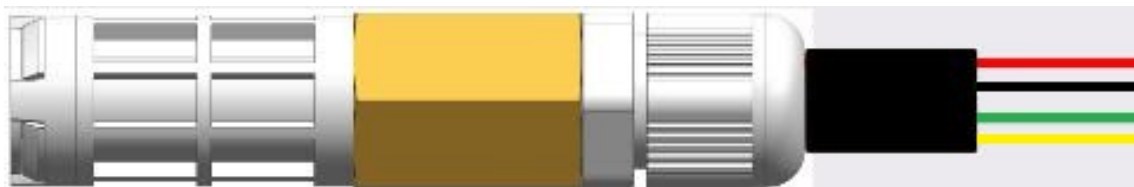
Note: Although the temperature measurement range of the sensor is $-40 \sim 125 \text{ }^{\circ}\text{C}$, we recommend not to place the sensor at a temperature higher than $80 \text{ }^{\circ}\text{C}$ since its shell is made of PE material.



Specification

- Operating Voltage: 3.3~5V
- Operating Current: <math><1.5\text{mA}</math>
- Humidity Detection Range: 0%RH~100%RH
- Humidity Accuracy: $\pm 2\%RH$
- Temperature Detection Range: $-40^{\circ}\text{C} \sim 125^{\circ}\text{C}$
- Temperature Accuracy: $\pm 0.2^{\circ}\text{C}$
- Communication: I2C
- Cable Length: about 1m

PinOut



Color	Name	Description
Red	VCC	+
Black	GND	-

Color	Name	Description
Green	SDA	Data line
Yellow	SCL	Clock line

Tutorial

Note: The I2C address of this sensor is 0x44. Please make sure the I2C address is correct when using it.

Requirements

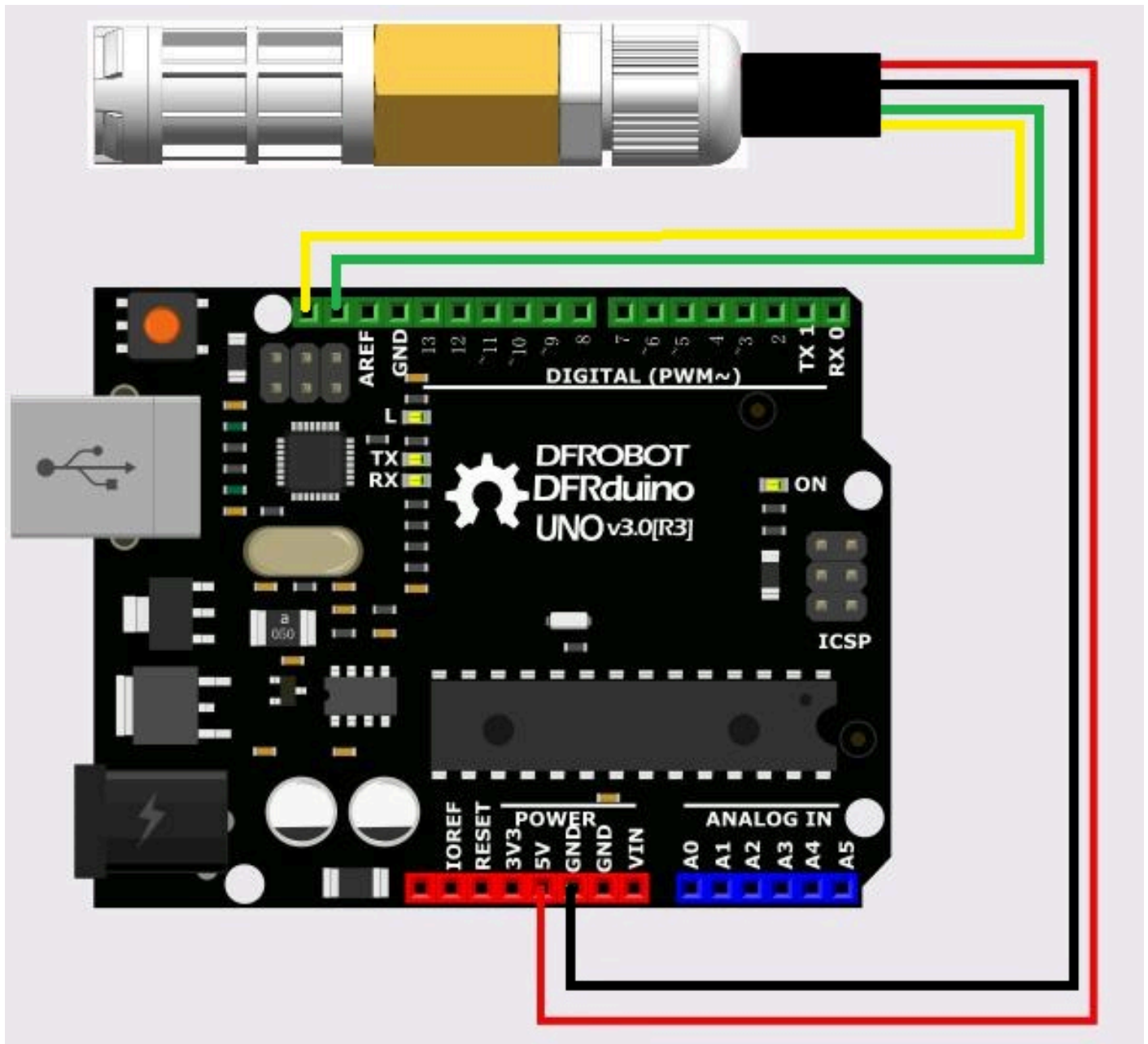
- **Hardware**
 - DFRduino UNO R3 (<https://www.dfrobot.com/product-838.html>) (or similar) x 1
 - SHT31 Temperature & Humidity Sensor (Weather-proof) x1
 - Jumper wires
- **Software**
 - Arduino IDE (<https://www.arduino.cc/en/Main/Software>)
 - Download and install the **SHT3X Library** (https://github.com/DFRobot/DFRobot_SHT3x) (About how to install the library? (<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>))
- API Function List

```
/**
 * @brief Get the measured temperature (in degrees Celsius).
 * @return Return the temperature data of float type.
 */
float getTemperatureC();

/**
 * @brief Get the measured temperature (in degrees Fahrenheit).
 * @return Return the temperature data of float type.
 */
float getTemperatureF();

/**
 * @brief Get measured humidity(%RH).
 * @return Return the humidity data of float type.
 */
float getHumidityRH();
```

Connection Diagram



Sample Code 1-Single Measurement Mode

In single measurement mode, the sensor collects data every time the controller board sends out the data collecting command. The power consumption could be very low in this mode since users can read data according to their needs.

```

/*!
 * @brief Construct the function
 * @param pWire I2C bus pointer object and construction device, can both p
 * @param address Chip I2C address, two optional addresse.
 */
/*!
 * @file singleMeasurement.ino
 * @brief Read ambient temperature (C/F) and relative humidity (%RH) in si
 * @n Experimental phenomenon: the chip defaults in this mode, we need to
 * which means the repeatability of the read needs to be set (the differer
 * then read the temperature and humidity data and print the data in the s
 * @n Single measure mode: read data as needed, power consumption is relat
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence The MIT License (MIT)
 * @author [fengli](li.feng@dfrobot.com)
 * @version V1.0
 * @date 2019-08-21
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_SHT3x
 */

#include <DFRobot_SHT3x.h>

/*!
 * @brief Construct the function
 * @param pWire I2C bus pointer object and construction device, both can p
 * Wire in default.
 * @param address Chip I2C address, two optional addresses 0x44 and 0x45(0
 * @param RST RST Chip reset pin, 4 in default.
 * @n I2C address is determined by the pin addr on the chip.
 * @n When the ADR is connected to VDD, the chip I2C address is 0x45.
 * @n When the ADR is connected to GND, the chip I2C address is 0x44.
 */

DFRobot_SHT3x sht3x(&Wire,/*address=*/0x44,/*RST=*/4);
//DFRobot_SHT3x sht3x;

```

```

void setup() {
  Serial.begin(9600);
  //Initialize the chip
  while (sht3x.begin() != 0) {
    Serial.println("Failed to Initialize the chip, please confirm the wire
    delay(1000);
  }
  /**
   * readSerialNumber Read the serial number of the chip.
   * @return Return 32-digit serial number.
   */
  Serial.print("Chip serial number");
  Serial.println(sht3x.readSerialNumber());

  /**
   * softReset Send command resets via I2C, enter the chip's default mode
   * turn off the heater, and clear the alert of the ALERT pin.
   * @return Read the register status to determine whether the command was
   * and return true indicates success.
   */
  if(!sht3x.softReset()){
    Serial.println("Failed to Initialize the chip....");
  }

  /**
   * heaterEnable(): Turn on the heater inside the chip to enable the sens
   * @return Read the status of the register to determine whether the comm
   * and return true indicates success.
   * @note Heaters should be used in wet environments, and other cases of
   */

  //if(!sht3x.heaterEnable()){
  // Serial.println("Failed to turn on the heater....");
  //}
  Serial.println("-----Read adta in single measurement mode-----
}

void loop() {
  Serial.print("Ambient Temperature(°C/F):");
  /**
   * getTemperatureC Get the measured temperature(°C).

```

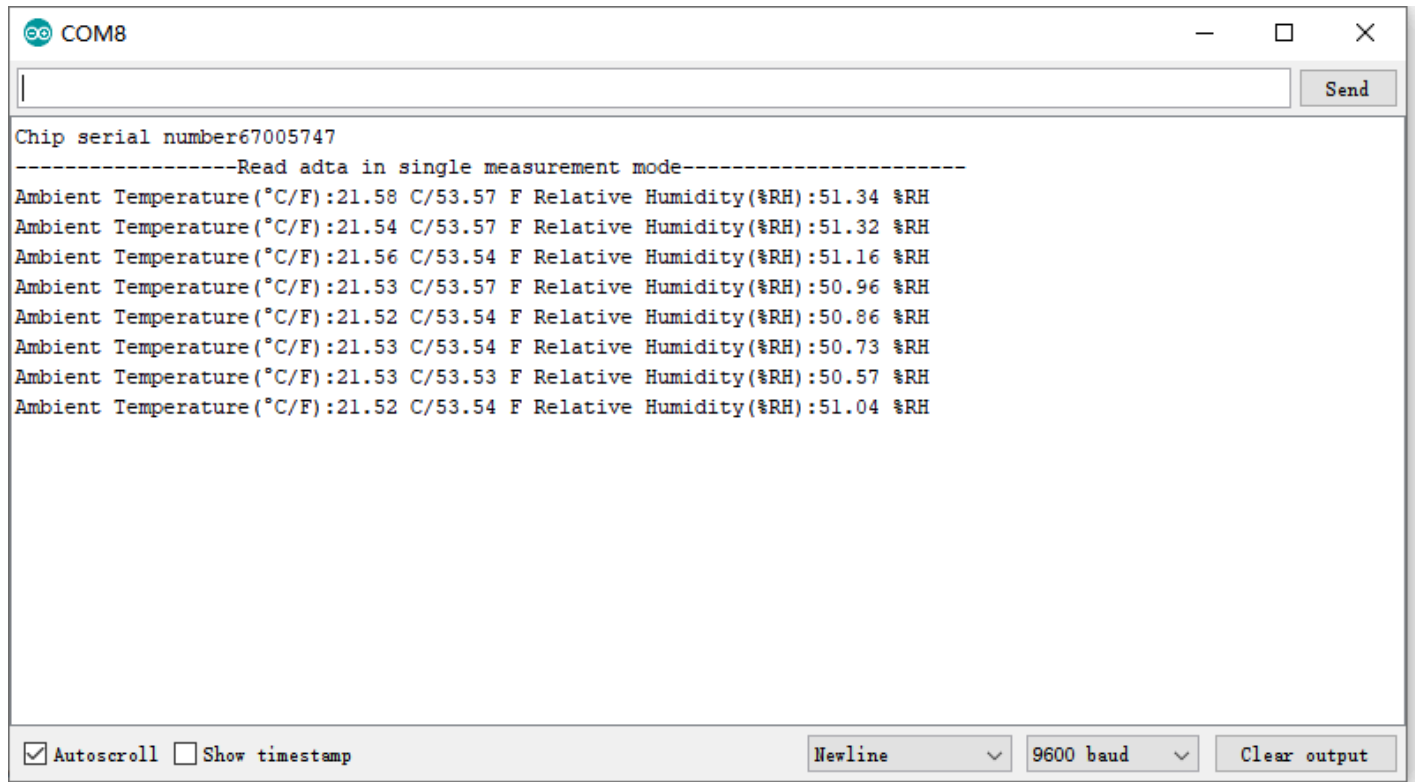
```

    * @return Return float temperature data.
    */
    Serial.print(sht3x.getTemperatureC());
    Serial.print(" C/");
    /**
    * getTemperatureF: Get the measured temperature(°F).
    * @return Return float temperature data.
    */
    Serial.print(sht3x.getTemperatureF());
    Serial.print(" F ");
    Serial.print("Relative Humidity(%RH):");
    /**
    * getHumidityRH: Get the measured humidity (%RH)
    * @return Return float humidity data
    */
    Serial.print(sht3x.getHumidityRH());
    Serial.println(" %RH");

    /**
    * @brief Get temperature and humidity data in single measurement mode.
    * @param repeatability Set repeatability to read temperature and humidity
    * @note Optional parameters:
           eRepeatability_High /**In high repeatability mode, the humidity
           eRepeatability_Medium,/**In medium repeatability mode, the humidity
           eRepeatability_Low, /**In low repeatability mode, the humidity
    * @return Return a structure containing celsius temperature (°C), Fahrenheit
    * @n Return 0 indicates right data return.
    DFRobot_SHT3x::sRHAndTemp_t data = sht3x.readTemperatureAndHumidity(sht3x);
    if(data.ERR == 0){
        Serial.print("Ambient Temperature(°C/F):");
        Serial.print(data.TemperatureC);
        Serial.print(" C/");
        Serial.print(data.TemperatureF);
        Serial.print(" F ");
        Serial.print("Relative Humidity(%RH):");
        Serial.print(data.Humidity);
        Serial.println(" %RH");
    }
    */
    delay(1000);
}

```


Result 1



The screenshot shows a serial terminal window titled "COM8". The output text is as follows:

```
Chip serial number67005747
-----Read adta in single measurement mode-----
Ambient Temperature(°C/F):21.58 C/53.57 F Relative Humidity(%RH):51.34 %RH
Ambient Temperature(°C/F):21.54 C/53.57 F Relative Humidity(%RH):51.32 %RH
Ambient Temperature(°C/F):21.56 C/53.54 F Relative Humidity(%RH):51.16 %RH
Ambient Temperature(°C/F):21.53 C/53.57 F Relative Humidity(%RH):50.96 %RH
Ambient Temperature(°C/F):21.52 C/53.54 F Relative Humidity(%RH):50.86 %RH
Ambient Temperature(°C/F):21.53 C/53.54 F Relative Humidity(%RH):50.73 %RH
Ambient Temperature(°C/F):21.53 C/53.53 F Relative Humidity(%RH):50.57 %RH
Ambient Temperature(°C/F):21.52 C/53.54 F Relative Humidity(%RH):51.04 %RH
```

At the bottom of the window, there are control elements: a checked "Autoscroll" checkbox, an unchecked "Show timestamp" checkbox, a "Newline" dropdown menu, a "9600 baud" dropdown menu, and a "Clear output" button.

Sample Code 2- Period Measurement Mode

In period measurement mode, the sensor collects data at the user-set frequency.

```

/ * !
* @file periodicDataReading.ino
* @brief Read ambient temperature (C/F) and relative humidity (%RH) in cycle
* @n Experimental phenomenon: Before we start, please set the read frequency
* (the difference between the data measured by the chip under the same measurement
* and enter the periodic read mode, and then read the temperature and humidity data
* @n The temperature and humidity data will be printed at the serial port
* @n It will exit the cycle mode and enter 2 measurement mode: Single measurement
* @n Single measurement mode: reflect the difference between the two measurement modes
* @n Cycle measurement mode: the chip periodically monitors temperature and humidity
* @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
* @licence The MIT License (MIT)
* @author [fengli](li.feng@dfrobot.com)
* @version V1.0
* @date 2019-08-20
* @get from https://www.dfrobot.com
* @url https://github.com/DFRobot/DFRobot_SHT3x
*/

#include <DFRobot_SHT3x.h>

/ * !
* @brief Construct the function
* @param pWire I2C bus pointer object and construction device, can both provide
* @param address Chip I2C address, two optional addresses 0x44 and 0x45(0x45 is default)
* @param RST Chip reset pin, 4 in default.
* @n The I2C address is determined by the pin addr on the chip.
* @n When the ADR is connected to VDD, the chip I2C address is 0x45.
* @n When the ADR is connected to GND, the chip I2C address is 0x44.
*/
DFRobot_SHT3x sht3x(&Wire, /*address=*/0x44, /*RST=*/4);

//DFRobot_SHT3x sht3x;

void setup() {

```

```

Serial.begin(9600);
  //Initialize the chip to detect if it can communicate properly.
while (sht3x.begin() != 0) {
  Serial.println("Failed to initialize the chip, please confirm the chip
  delay(1000);
}

/**
 * readSerialNumber Read the serial number of the chip
 * @return Return 32-digit serial number
 */
Serial.print("chip serial number: ");
Serial.println(sht3x.readSerialNumber());
/**
 * softReset Send command resets via I2C, enter the chip's default mode
 * turn off the heater, and clear the alert of the ALERT pin.
 * @return Read the status register to determine whether the command was
 * and return true indicates success.
 */
if(!sht3x.softReset()){
  Serial.println("Failed to reset the chip");
}

/**
 * pinReset Reset through the chip's reset pin, enter the chip's default
 * turn off the heater, and clear the alert of the ALERT pin.
 * @return The status register has a data bit that detects whether the c
 * and return true indicates success.
 * @note When using this API, the reset pin of the chip nRESET should be
 */
//if(!sht3x.pinReset()){
  //Serial.println("Failed to reset the chip");
//}

/**
 * heaterEnable() Turn on the heater inside the chip so that the sensor
 * @return Read the status register to determine whether the command was
 * @NOTE Heaters should be used in wet environment, and other cases of u
 */
//if(!sht3x.heaterEnable()){
  // Serial.println("Failed to turn on the heater");
//}

```

```

/**
 * startPeriodicMode Enter cycle measurement mode and set repeatability
 * @param measureFreq Read the eMeasureFrequency_t data frequency.
 * @note Selectable parameters:
 *         eMeasureFreq_Hz5,    /**the chip collects data in every 2s
 *         eMeasureFreq_1Hz,    /**the chip collects data in every 1s
 *         eMeasureFreq_2Hz,    /**the chip collects data in every 0.5s
 *         eMeasureFreq_4Hz,    /**the chip collects data in every 0.25s
 *         eMeasureFreq_10Hz   /**the chip collects data in every 0.1s
 * @param repeatability Read the repeatability of temperature and humidity
 * @note Optional parameters:
 *         eRepeatability_High /**In high repeatability mode, the humidity
 *         eRepeatability_Medium,/**In medium repeatability mode, the humidity
 *         eRepeatability_Low, /**In low repeatability mode, the humidity
 * @return Read the status of the register to determine whether the command
 */
if(!sht3x.startPeriodicMode(sht3x.eMeasureFreq_1Hz)){
  Serial.println("Failed to enter the periodic mode");
}
Serial.println("-----Read data in cycle measurement mode-----");
}

void loop() {

  Serial.print("Ambient temperature(°C/F):");
  /**
   * getTemperatureC Get the measured temperature (in degrees Celsius).
   * @return Return the float temperature data.
   */
  Serial.print(sht3x.getTemperatureC());
  Serial.print(" C/");
  /**
   * getTemperatureF Get the measured temperature (in degrees Fahrenheit).
   * @return Return the float temperature data.
   */
  Serial.print(sht3x.getTemperatureF());
  Serial.print(" F ");
  Serial.print("Relative humidity(%RH):");
  /**
   * getHumidityRH Get measured humidity(%RH)
   * @return Return the float humidity data
   */

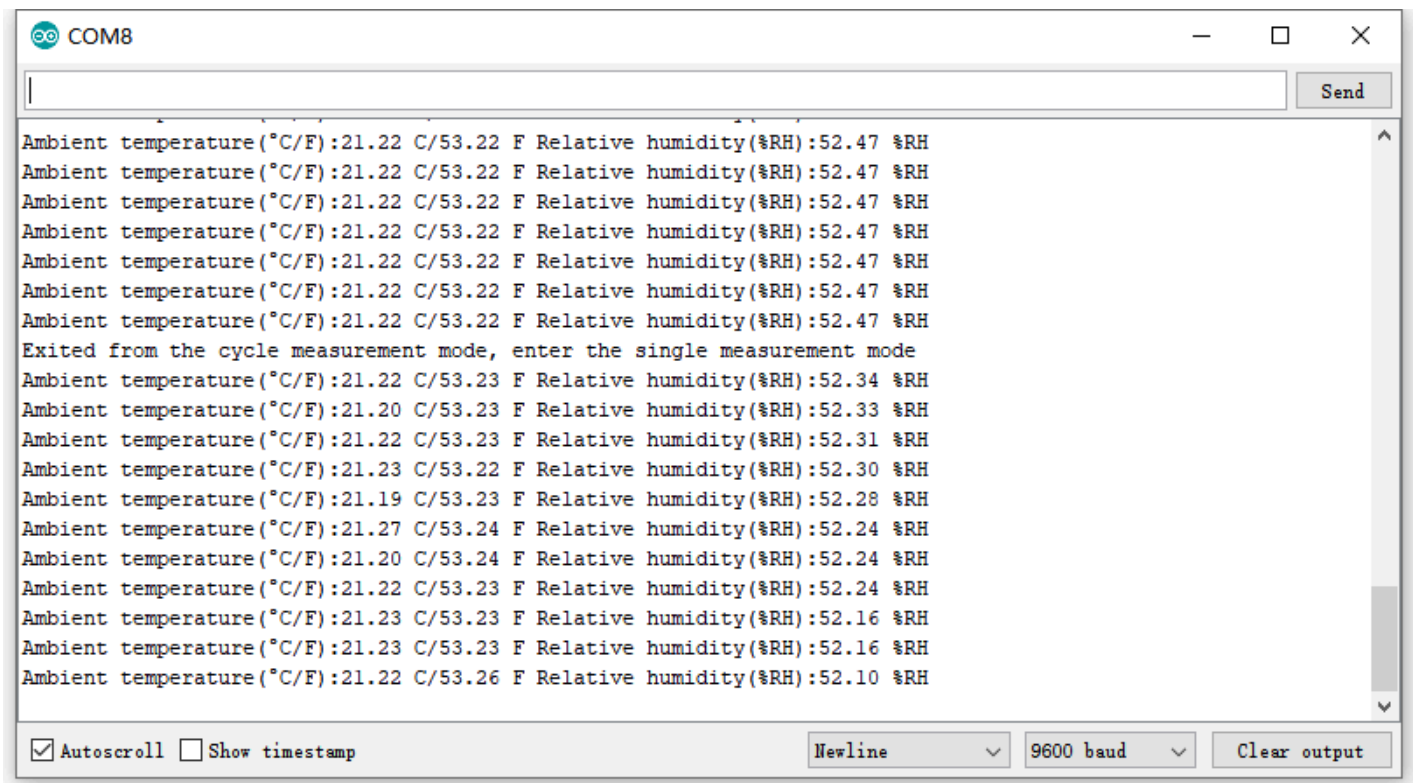
```

```
Serial.print(sht3x.getHumidityRH());
Serial.println(" %RH");
//Please adjust the frequency of reading according to the frequency of t
//The frequency to read data must be greater than the frequency to colle
delay(100);
if(millis() > 10000 && millis() < 10200){
  /**
   * stopPeriodicMode() Exit from the cycle read data
   * @return Read the status of the register to determine whether the co
   * and return true indicates success.
   */
  sht3x.stopPeriodicMode();
  Serial.println("Exited from the cycle measurement mode, enter the sing
}
/**
 * readTemperatureAndHumidity Get temperature and humidity data in cycle
 * @return Return a structure containing celsius temperature (°C), Fahre
 * @n A status of 0 indicates that the right return data.

DFRobot_SHT3x::sRHAndTemp_t data = sht3x.readTemperatureAndHumidity();
if(data.ERR == 0){
  Serial.print("ambient temperature(°C/F):");
  Serial.print(data.TemperatureC);
  Serial.print("C/");
  Serial.print(data.TemperatureF);
  Serial.print("F");
  Serial.print("relative humidity(%RH):");
  Serial.print(data.Humidity);
  Serial.println("%RH");
}
*/
}
```

Result 2

Serial print the temperature and humidity information in period measurement mode for 10s, then exit from this mode and enter single measurement mode, and print the information.



The screenshot shows a serial terminal window titled "COM8". The window contains a series of text lines representing sensor data. The data is formatted as: "Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH". This line is repeated multiple times. After several repetitions, there is a line that says "Exited from the cycle measurement mode, enter the single measurement mode". This is followed by more data lines, with the temperature values varying slightly (e.g., 21.20, 21.23, 21.19, 21.27, 21.20, 21.22, 21.23, 21.23, 21.22). The terminal window has a "Send" button at the top right and a status bar at the bottom with checkboxes for "Autoscroll" (checked) and "Show timestamp" (unchecked), along with dropdown menus for "Newline" and "9600 baud", and a "Clear output" button.

```
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Ambient temperature(°C/F):21.22 C/53.22 F Relative humidity(%RH):52.47 %RH
Exited from the cycle measurement mode, enter the single measurement mode
Ambient temperature(°C/F):21.22 C/53.23 F Relative humidity(%RH):52.34 %RH
Ambient temperature(°C/F):21.20 C/53.23 F Relative humidity(%RH):52.33 %RH
Ambient temperature(°C/F):21.22 C/53.23 F Relative humidity(%RH):52.31 %RH
Ambient temperature(°C/F):21.23 C/53.22 F Relative humidity(%RH):52.30 %RH
Ambient temperature(°C/F):21.19 C/53.23 F Relative humidity(%RH):52.28 %RH
Ambient temperature(°C/F):21.27 C/53.24 F Relative humidity(%RH):52.24 %RH
Ambient temperature(°C/F):21.20 C/53.24 F Relative humidity(%RH):52.24 %RH
Ambient temperature(°C/F):21.22 C/53.23 F Relative humidity(%RH):52.24 %RH
Ambient temperature(°C/F):21.23 C/53.23 F Relative humidity(%RH):52.16 %RH
Ambient temperature(°C/F):21.23 C/53.23 F Relative humidity(%RH):52.16 %RH
Ambient temperature(°C/F):21.22 C/53.26 F Relative humidity(%RH):52.10 %RH
```

FAQ

For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](https://www.dfrobot.com/forum/) (<https://www.dfrobot.com/forum/>).

More Documents

- SHT3x Datasheet
(<https://dfimg.dfrobot.com/nobody/wiki/88b31350da4f54d00989c74c6fa392f7.pdf>)
- SHT3X Handling Instructions
(<https://dfimg.dfrobot.com/nobody/wiki/ed23ca262d11f01046f949078f45a37c.pdf>)
- SEN0385 SIZE
(<https://dfimg.dfrobot.com/5d57611a3416442fa39bffc/wiki/326d6a946be93f58>)

5415938419348d46.zip)