

## Introduction

---

The TMF8701 is a time-of-flight (ToF) distance ranging sensor that uses the time difference between light pulses to measure distance. It provides single zone detection of an object irrespective of the color, reflectivity and texture of the object. The TMF8701 offers high dynamic range and operates in either a proximity mode (1 –10cm) or a ranging mode (10–60cm) for detection sensing. And the device can make highly accurate distance measurements within  $\pm 5\%$  and is capable of operation in dark environments and in the presence of sunlight. A built-in histogram is featured to delivers dynamic cover glass calibration and crosstalk compensation and background light noise is minimized through on-chip sunlight rejection filters. The TMF8701 outputs data through a I2C fast-mode communications interface and an integrated micro controller, which is featured with all algorithms included on-chip with no need for external optics.

## Features

---

- 21° FOI, detect the object closest to the center
- 10–600mm distance sensing
- Enables dark and sunlight environment distance measurement within  $\pm 5\%$
- A built-in histogram for compensating for dirt and smudges on cover glass
- 940nm VCSEL Class 1 Eye Safety
- Low power consumption, 940 $\mu$ A power consumption at 10Hz(proximity mode); 26mA power consumption at 60Hz (distance and proximity mode); 0.26 $\mu$ A power-down current consumption (EN=0)

## Applications

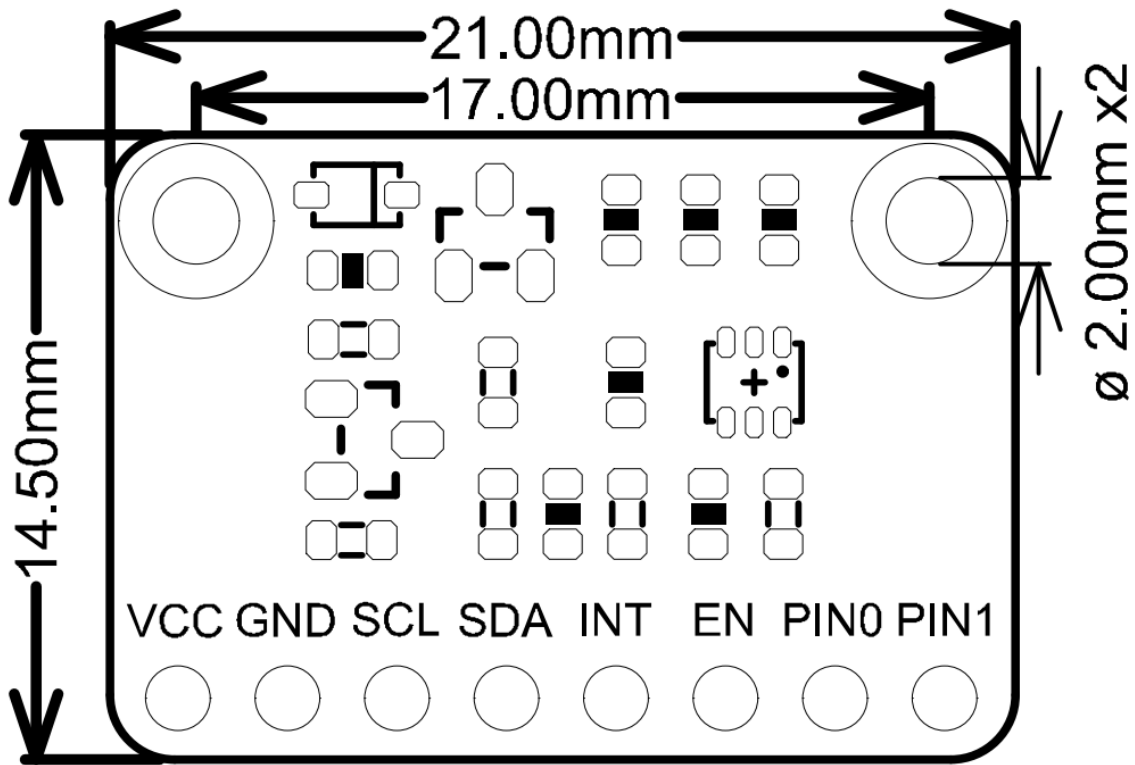
---

- 3D face recognition
- Proximity detection
- Presence detection
- Object detection
- Distance measurement
- Collision avoidance

## Specification

- Operating Voltage: 2.7~3.3V
- Operating Current: <1.5mA
- Proximity Detection Range: 1-10cm
- Distance Sensing Range: 10-60cm
- Operating Temperature: -30 ~70°C
- Communication Interface: Breakout 2.54mm-8Pin I2C
- Dimension: 21×14.5mm/0.83×0.57inch
- Mounting Hole Size: 2.0mm
- Mounting Hole Pitch: 17mm

## Board Overview



Name	Function
VCC	+
GND	-
SCL	Clock line
SDA	Data line
INT	Alert Interrupt
EN	Reset

Pin	Reset
PIN0	Interrupt output pin0
PIN1	Interrupt output pin1

# Tutorial

---

## Requirements

- **Hardware**
  - DFRduino UNO R3 (<https://www.dfrobot.com/product-838.html>) (or similar) x 1
  - TMF8701 Sensor x1
  - Jumper wires
- **Software**
  - Arduino IDE (<https://www.arduino.cc/en/Main/Software>)
  - Download and install the **TMF8x01 Library** ([https://github.com/DFRobot/DFRobot\\_TMF8x01](https://github.com/DFRobot/DFRobot_TMF8x01)) (About how to install the library? (<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>))
- **API Functions**

```

int begin();

/**
 * @brief sleep sensor by software, the sensor enter sleep mode(bootloader). Need to
 */
void sleep();
/**
 * @brief wakeup device from sleep mode, it will running app0
 * @return enter app0 return true, or return false.
 */
bool wakeup();

/**
 * @brief get a unique number of sensor .Each sensor has a unique identifier.
 * @return return 4bytes unique number:
 * @n the byte0 of return: serial_number_0
 * @n the byte1 of return: serial_number_1
 * @n the byte2 of return: identification_number_1
 * @n the byte2 of return: identification_number_0
 */
uint32_t getUniqueID();
/**
 * @brief get sensor's model.
 * @return return a String:
 * @n TMF8801: the sensor is TMF8801
 * @n TMF8701: the sensor is TMF8701
 * @n unknown : unknown device
 */
String getSensorModel();
/**
 * @brief get software version of patch.
 * @return return string of device software version,format:
 * @n major.minor.patch numbers.chip id version
 */
String getSoftwareVersion();
/**
 * @brief Get 14 bytes of calibration data.
 * @param data Cache for storing calibration data
 * @param len The bytes of calibration data,its value can only be 14 bytes
 * @return Vail data return true, or return false.
 */
bool getCalibrationData(uint8_t *data, uint8_t len = SENSOR_MTF8x01_CALIBRATION_S
/**
 * @brief set 14 bytes of calibration data.
 * @param data Pointer to calibration data.
 * @param len The bytes of calibration data,its value can only be 14 bytes

```

```

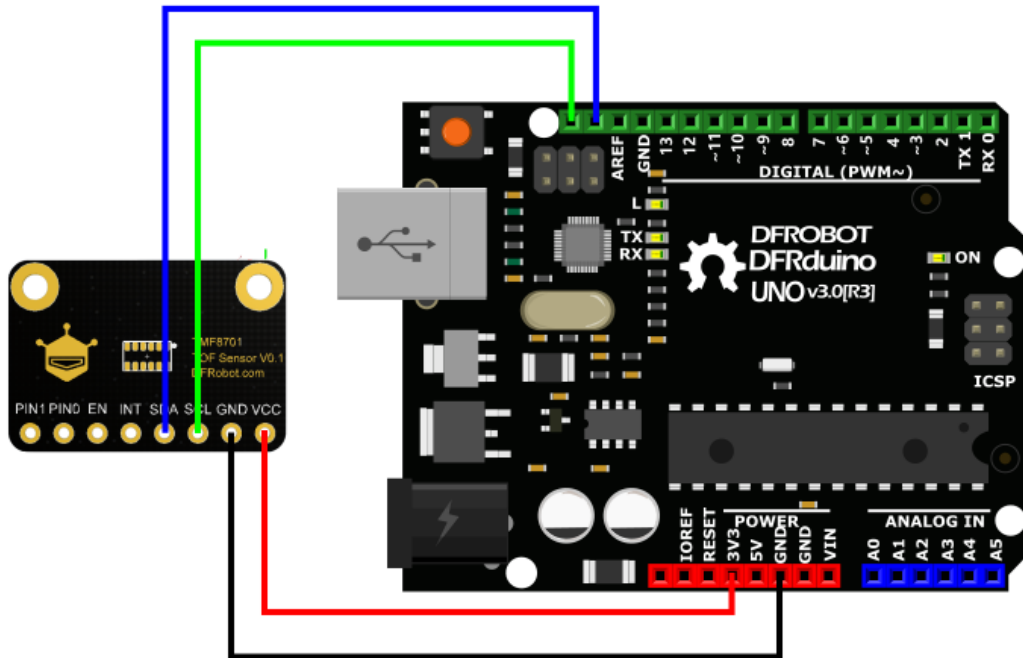
* @return set sucess return true, or return false.
*/
bool setCalibrationData(uint8_t *data, uint8_t len = SENSOR_MTF8x01_CALIBRATION_S

/**
* @brief disable measurement config. Need to call startMeasurement before using th
*/
void stopMeasurement();
/**
* @brief Waiting for data ready.
* @return if data is valid, return true, or return false.
*/
bool isDataReady();
/**
* @brief get distance, unit mm. Before using this function, you need to call isDat
* @return return distance value, unit mm.
*/
uint16_t getDistance_mm();
/**
* @brief enable INT pin. If you call this function,which will report a interrupt
* @n signal to host by INT pin when measure data is ready.
*/
void enableIntPin();
/**
* @brief disable INT pin.
*/
void disableIntPin();
/**
* @brief power on sensor when power down sensor by EN pin.
* @return sucess return True, or return False
*/
bool powerOn();
/**
* @brief power down sensor by EN pin.
* @return sucess return True, or return False
*/
bool powerDown();

/**
* @brief get I2C address.
* @return return 7 bits I2C address
*/
uint8_t getI2CAddress();
/**
* @brief Config the pin of sensor.
* @param pin: The pin of sensor, example PIN0 and PIN1,which is an enumerated vari
* @n ePIN0: The PIN0 of sensor config.
* @n ePIN1: The PIN1 of sensor.
* @n eGPIOTotal: both of PIN0 and PIN1.
* @param config: The config of pin, which is an enumerated variable of ePinContro
*/
void pinConfig(ePin_t pin, ePinControl_t config);

```

## Connection Diagram



## Sample Code 1 - Distance Detection

```

/#!/
* @file getDistance.ino
* @brief Get measurement data by PROXIMITY and DISTANCE hybrid mode.
* @n note: TMF8801 only suport one mode, PROXIMITY and DISTANCE hybrid mode.
* *
* Ranging mode configuration table:
* -----
* | Type      | suport ranging mode | ranging ranges | Accuracy
* |-----|-----|-----|-----|
* | TMF8801   | PROXIMITY and DISTANCE | 20~240cm      | 20~100mm: +/-15mm
* |           | hybrid mode(only one) |                | 100~200mm: +/-10mm
* |           |                       |                | >=200: +/-%5
* |-----|-----|-----|-----|
* |           | PROXIMITY mode       | 0~10cm        | >=200: +/-%5
* | TMF8701   | DISTANCE mode        | 10~60cm       | 100~200mm: +/-10mm
* |           |-----|-----|-----|
* |           | PROXIMITY and DISTANCE | 0~60cm        |
* |           | hybrid mode          |                |
* |-----|-----|-----|-----|
* *
* @n hardware conneted table:
* -----
* | TMF8x01   | MCU                  |
* |-----|-----|
* | I2C       | I2C Interface       |
* |-----|-----|
* | EN        | not connected, floating |
* |-----|-----|
* | INT       | not connected, floating |
* |-----|-----|
* | PIN0      | not connected, floating |
* |-----|-----|
* | PIN1      | not connected, floating |
* |-----|-----|
*
* @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
* @licence The MIT License (MIT)
* @author [Arya](xue.peng@dfrobot.com)
* @version V1.0
* @data 2021-03-26
* @get from https://www.dfrobot.com
* @url https://github.com/DFRobot/DFRobot\_TMF8x01
*/

#include "DFRobot_TMF8x01.h"

```

```

#define EN          -1                //EN pin of of TMF8x01 module is floating,
#define INT        -1                //INT pin of of TMF8x01 module is floating

//DFRobot_TMF8801 tof(/*enPin =*/EN,/*intPin=*/INT);
DFRobot_TMF8701 tof(/*enPin =*/EN,/*intPin=*/INT);

uint8_t caliDataBuf[14] = {0x41,0x57,0x01,0xFD,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
void setup() {

    Serial.begin(115200);                //Se
    while(!Serial){                    //Wa
    }

    Serial.print("Initialization ranging sensor TMF8x01.....");
    while(tof.begin() != 0){                //Initia
        Serial.println("failed.");
        delay(1000);
    }
    Serial.println("done.");

    Serial.print("Software Version: ");
    Serial.println(tof.getSoftwareVersion());
    Serial.print("Unique ID: ");
    Serial.println(tof.getUniqueID(),HEX);
    Serial.print("Model: ");
    Serial.println(tof.getSensorModel());

    tof.setCalibrationData(caliDataBuf, sizeof(caliDataBuf));                //Set ca

/**
 * @brief Config measurement params to enable measurement. Need to call stopMeasure
 * @param cailbMode: Is an enumerated variable of eCalibModeConfig_t, which is to c
 * @n     eModeNoCalib :           Measuring without any calibration data.
 * @n     eModeCalib   :           Measuring with calibration data.
 * @n     eModeCalibAndAlgoState : Measuring with calibration and algorithm state.
 * @param disMode : the ranging mode of TMF8701 sensor.(this mode only TMF8701 supp
 * @n     ePROXIMITY: Raing in PROXIMITY mode,ranging range 0~10cm
 * @n     eDISTANCE: Raing in distance mode,ranging range 10~60cm
 * @n     eCOMBINE:  Raing in PROXIMITY and DISTANCE hybrid mode,ranging range 0~60
 */
    //tof.startMeasurement(/*cailbMode =*/tof.eModeCalib);                //Enable m
    tof.startMeasurement(/*cailbMode =*/tof.eModeCalib, /*disMode =*/tof.eCOMBINE);
}

void loop() {
    if (tof.isDataReady()) {                //Is che
        Serial.print("Distance = ");
        Serial.print(tof.getDistance_mm());                //Print
        Serial.println(" mm");
    }
}

```



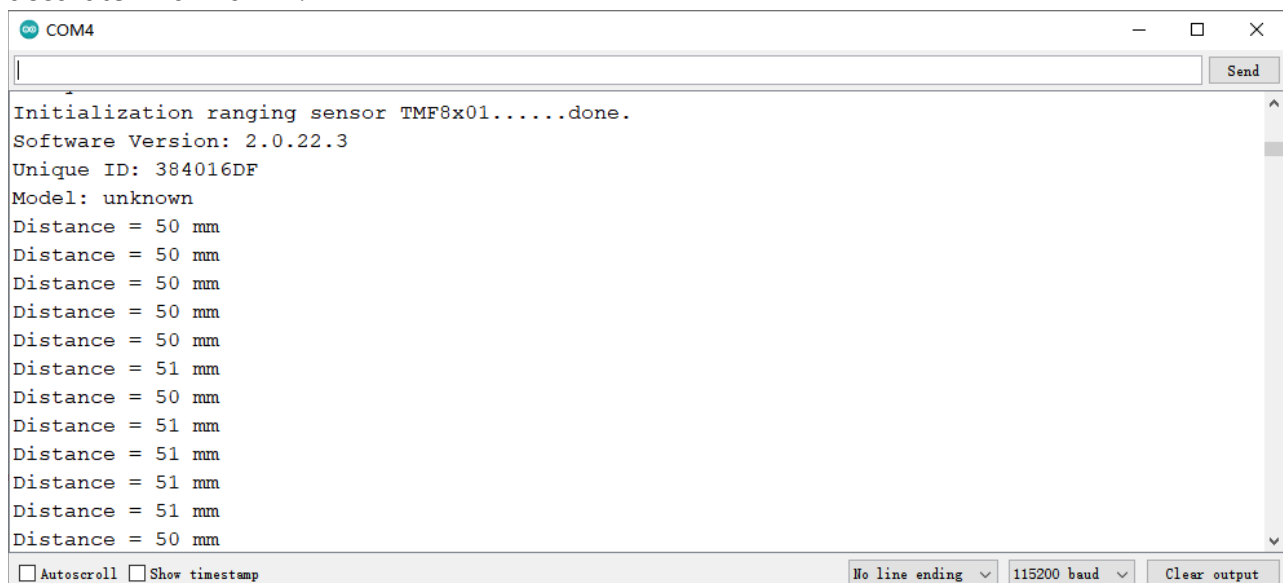
## Expected Results

Detect distance in proximity mode and ranging mode.

Effective ranging distance:

- Proximity mode: 0-10cm
- Ranging mode: 10-60cm
- Proximity and ranging mode: 0-60cm

Note: The sensor outputs 0mm when out of range. The measured data may not be accurate in 0~10mm.



The screenshot shows a serial terminal window titled 'COM4'. The output text is as follows:

```
Initialization ranging sensor TMF8x01.....done.  
Software Version: 2.0.22.3  
Unique ID: 384016DF  
Model: unknown  
Distance = 50 mm  
Distance = 50 mm  
Distance = 50 mm  
Distance = 50 mm  
Distance = 50 mm  
Distance = 51 mm  
Distance = 50 mm  
Distance = 51 mm  
Distance = 51 mm  
Distance = 51 mm  
Distance = 51 mm  
Distance = 51 mm  
Distance = 50 mm
```

At the bottom of the window, there are control buttons:  Autoscroll,  Show timestamp, a dropdown menu for 'No line ending', a dropdown menu for '115200 baud', and a 'Clear output' button.

## Sample Code 2 - Interrupt Output

You can connect the INT of TMF8701 to the MCU's external interrupt pin. When there is data output from the sensor, the INT will generate a LOW level and MCU can determine the coming data by detect that low level.

```

/#!
* @file interrupt.ino
* @brief If you enable INT pin, MCU will capture a interrupt signal when the measu
* @n You can attach the INT pin of TMF8x01 to MCU external interrupt pin.
* @n When there is data from sensor, pin INI generates a Low level, so MCU can det
* *
* Ranging mode configuration table:
* -----
* | Type      | suport ranging mode | ranging ranges | Accuracy
* |-----|-----|-----|-----|
* | TMF8801   | PROXIMITY and DISTANCE | 20~240cm      | 20~100mm: +/-15mm
* |           | hybrid mode(only one) |                | 100~200mm: +/-10mm
* |           |                       |                | >=200: +/-%5
* |-----|-----|-----|-----|
* |           | PROXIMITY mode       | 0~10cm        |
* |           |-----|-----|-----|
* |           |                       |                | >=200: +/-%5
* | TMF8701   | DISTANCE mode        | 10~60cm       | 100~200mm: +/-10mm
* |           |-----|-----|-----|
* |           | PROXIMITY and DISTANCE | 0~60cm        |
* |           | hybrid mode          |                |
* |-----|-----|-----|-----|
* *
* @n hardware conneted table:
* -----
* | TMF8x01   | MCU |
* |-----|-----|
* | I2C       | I2C Interface |
* |-----|-----|
* | EN        | not connected, floating |
* |-----|-----|
* | INT       | to the external interrupt pin of MCU |
* |-----|-----|
* | PIN0      | not connected, floating |
* |-----|-----|
* | PIN1      | not connected, floating |
* |-----|-----|
*
* @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
* @licence The MIT License (MIT)
* @author [Arya](xue.peng@dfrobot.com)
* @version V1.0
* @data 2021-03-26
* @get from https://www.dfrobot.com
* @url https://github.com/DFRobot/DFRobot\_TMF8x01
*/

```

```

#include "DFRobot_TMF8x01.h"

#define EN      -1                                //EN pin o
#define INT     2                                //connecte

//DFRobot_TMF8801 tof(/*enPin =*/EN,/*intPin=*/INT);
DFRobot_TMF8701 tof(/*enPin =*/EN,/*intPin=*/INT);

bool irqFlag = false;
void notifyFun(){
    irqFlag = true;
}

void setup() {
    Serial.begin(115200);
    while(!Serial){
    }

    Serial.print("Initialization ranging sensor TMF8x01.....");
    while(tof.begin() != 0){
        Serial.println("failed.");
        delay(1000);
    }
    Serial.println("done.");

    Serial.print("Software Version: "); Serial.println(tof.getSoftwareVersion());
    Serial.print("Unique ID: ");       Serial.println(tof.getUniqueID(),HEX);
    Serial.print("Model: ");           Serial.println(tof.getSensorModel());

    tof.enableIntPin();

#ifdef ESP32 || defined(ESP8266)
/* mPython Interrupt Pin vs Interrupt NO
 * -----
 * |           | DigitalPin |           can be used as external inte
 * |   ESP32   |-----|
 * |   ESP8266 | Interrupt No | use digitalPinToInterrupt(pin numbers)
 * |-----|
 */
attachInterrupt(/*Interrupt NO=*/digitalPinToInterrupt(INT),notifyFun,FALLING);
#else
/* Main-board of AVR series    Interrupt Pin vs Interrupt NO
 * -----
 * |           | DigitalPin | 2 | 3 |
 * |   Uno, Nano, Mini, other 328-based |-----|
 * |           | Interrupt No | 0 | 1 |
 * |-----|
 * |           | Pin        | 2 | 3 | 21 | 20 |
 * |           |-----|
 * |           | Interrupt No | 0 | 1 | 2 | 3 |
 * |-----|
 * |           | Pin        | 3 | 2 | 0 | 1 |

```

```

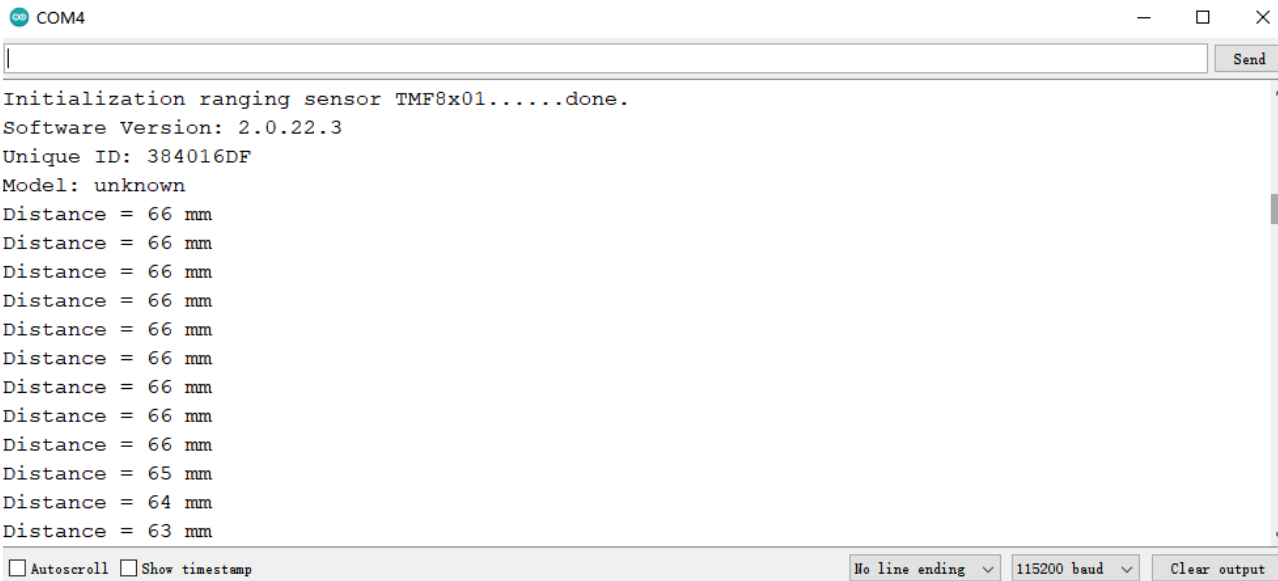
* | Leonardo, other 32u4-based |-----
* | | Interrupt No | 0 | 1 | 2 | 3 |
* |-----
*/
/* microbit Interrupt Pin vs Interrupt NO
* -----
* | | DigitalPin | P0~P20 ca
* | microbit |-----
* |(when used as external interrupt, do not need to | Interrupt No | Interrupt
* | set it to input mode via pinMode) | | the Inter
* |-----
*/
attachInterrupt(/*Interrupt NO=*/0,notifyFun,FALLING);
#endif

/**
 * @brief Config measurement params to enable measurement. Need to call stopMeasure
 * @param cailbMode: Is an enumerated variable of eCalibModeConfig_t, which is to c
 * @n eModeNoCalib : Measuring without any calibration data.
 * @n eModeCalib : Measuring with calibration data.
 * @n eModeCalibAndAlgoState : Measuring with calibration and algorithm state.
 * @param disMode : the ranging mode of TMF8701 sensor.(this mode only TMF8701 supp
 * @n ePROXIMITY: Raing in PROXIMITY mode,ranging range 0~10cm
 * @n eDISTANCE: Raing in distance mode,ranging range 10~60cm
 * @n eCOMBINE: Raing in PROXIMITY and DISTANCE hybrid mode,ranging range 0~60
 */
tof.startMeasurement(/*cailbMode =*/tof.eModeCalib);
//tof.startMeasurement(/*cailbMode =*/tof.eModeCalib, /*disMode =*/tof.ePROXIMITY
}

void loop() {
  if(irqFlag){
    irqFlag = false;
    if (tof.isDataReady()) {
      Serial.print("Distance = ");
      Serial.print(tof.getDistance_mm());
      Serial.println(" mm");
    }
  }
}
}

```

## Expected Result



```
COM4
Initialization ranging sensor TMF8x01.....done.
Software Version: 2.0.22.3
Unique ID: 384016DF
Model: unknown
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 66 mm
Distance = 65 mm
Distance = 64 mm
Distance = 63 mm
```

Autoscroll  Show timestamp No line ending 115200 baud Clear output

## Sample Code 3 - Sleep Mode

The sensor sleeps for 2s when finishing 20 measurements. In sleep mode, it stops ranging. We can activate it by wakeup function.

Note: it consumes 37.9mA current in ranging mode, 1.2mA in sleep mode.

```

/!*
 * @file sleep.ino
 * @brief The sensor sleeps for 2s when finishing 20 measurements. In sleep mode, i
 * @n note: it consumes 37.9mA current in ranging mode, 1.2mA in sleep mode.
 * *
 * Ranging mode configuration table:
 * TMF8X01_MODE_PROXIMITY: PROXIMITY mode
 * TMF8X01_MODE_DISTANCE: DISTANCE mode
 * TMF8X01_MODE_COMBINE: PROXIMITY and DISTANCE hybrid mode
 * default mode: TMF8X01_MODE_COMBINE
 * -----
 * | Type      | suport ranging mode | ranging ranges | Accuracy
 * |-----|-----|-----|-----
 * | TMF8801   | PROXIMITY and DISTANCE | 20~240cm      | 20~100mm: +/-15mm
 * |           | hybrid mode(only one) |                | 100~200mm: +/-10mm
 * |           |                        |                | >=200: +/-%5
 * |-----|-----|-----|-----
 * |           | PROXIMITY mode       | 0~10cm        | >=200: +/-%5
 * | TMF8701   | DISTANCE mode        | 10~60cm       | 100~200mm: +/-10mm
 * |           |-----|-----|-----
 * |           | PROXIMITY and DISTANCE | 0~60cm        |
 * |           | hybrid mode          |                |
 * |-----|-----|-----|-----
 * *
 * @n hardware conneted table:
 * -----
 * | TMF8x01   | MCU |
 * |-----|-----|
 * | I2C       | I2C Interface |
 * |-----|-----|
 * | EN        | not connected, floating |
 * |-----|-----|
 * | INT       | not connected, floating |
 * |-----|-----|
 * | PIN0      | not connected, floating |
 * |-----|-----|
 * | PIN1      | not connected, floating |
 * |-----|-----|
 *
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence The MIT License (MIT)
 * @author [Arya](xue.peng@dfrobot.com)
 * @version V1.0
 * @data 2021-03-26
 * @get from https://www.dfrobot.com

```

```

* @url https://github.com/DFRobot/DFRobot_TMF8x01
*/

#include "DFRobot_TMF8x01.h"

#define EN          -1                //EN pin of of TMF8x01 module is floating,
#define INT         -1                //INT pin of of TMF8x01 module is floating

DFRobot_TMF8701 tmf8x01(/*enPin =*/EN,/*intPin=*/INT);
//DFRobot_TMF8801 tmf8x01(/*enPin =*/EN,/*intPin=*/INT);

#define NUM_OF_MEASUREMENT    20                //20
#define SLEEP_TIME            1000              //sl
uint8_t count = 0;                             //Me

void setup() {

    Serial.begin(115200);                        //Se
    while(!Serial){                              //Wa
    }

    Serial.print("Initialization ranging sensor TMF8x01.....");
    while(tmf8x01.begin() != 0){                 //In
        Serial.println("failed.");
        delay(1000);
    }
    Serial.println("done.");

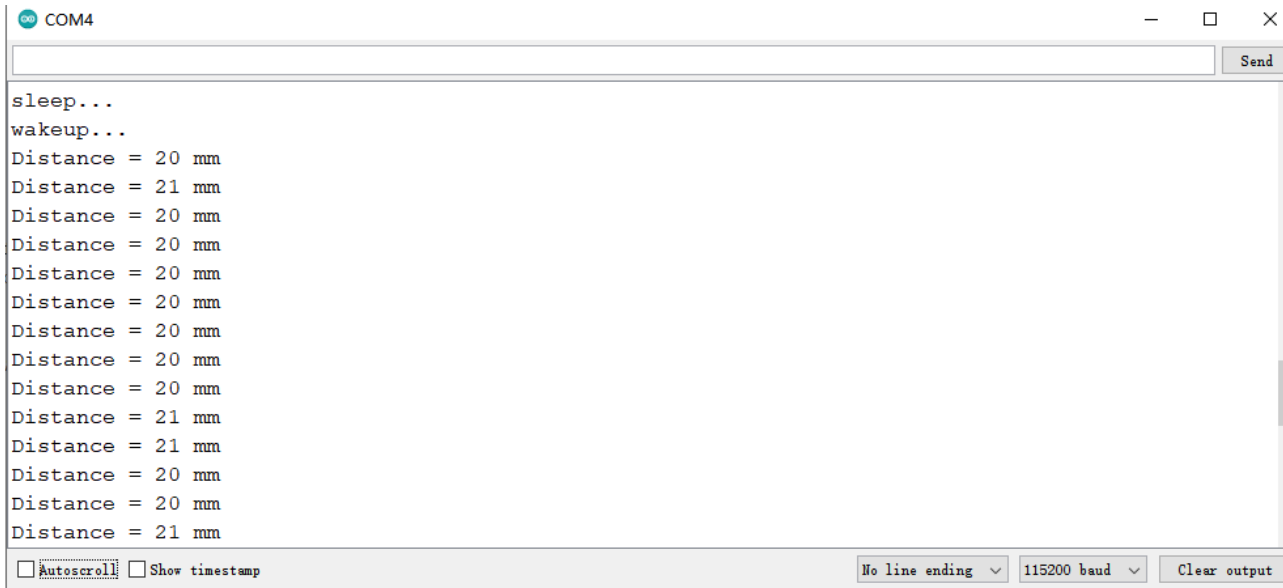
    Serial.print("Sensor Version info: ");
    Serial.println(tmf8x01.getVersion());        //Pr

    tmf8x01.startMeasurement(/*cailbMode =*/tmf8x01.eModeCalib); //En
    //tmf8x01.startMeasurement(/*cailbMode =*/tmf8x01.eModeNoCalib); //En
    //tmf8x01.startMeasurement(/*cailbMode =*/tmf8x01.eModeCalibAndAlgoState); //En
}

void loop() {
    if (tmf8x01.isDataReady()) {                 //Is
        Serial.print("Distance = ");
        Serial.print(tmf8x01.getDistance_mm()); //Pr
        Serial.println(" mm");
        count++;
    }
    if(count > NUM_OF_MEASUREMENT){
        count = 0;
        tmf8x01.sleep();                         //se
        Serial.println("sleep...");
        delay(SLEEP_TIME);
        Serial.println("wakeup...");
        tmf8x01.wakeup();                        //wak
    }
}

```

## Expected Result



The screenshot shows a serial terminal window titled 'COM4'. The output text is as follows:

```
sleep...
wakeup...
Distance = 20 mm
Distance = 21 mm
Distance = 20 mm
Distance = 20 mm
Distance = 20 mm
Distance = 20 mm
Distance = 20 mm
Distance = 20 mm
Distance = 20 mm
Distance = 20 mm
Distance = 21 mm
Distance = 21 mm
Distance = 20 mm
Distance = 20 mm
Distance = 21 mm
```

At the bottom of the window, there are several controls: a checkbox for 'Autoscroll' (unchecked), a checkbox for 'Show timestamp' (unchecked), a dropdown menu for 'No line ending', a dropdown menu for '115200 baud', and a 'Clear output' button.

## Sample 4 - Calibration Mode

The demo shows us how to obtain 14byte calibration and set calibration range.

Experiment condition: dark environment, no objects within 40cm around the sensor.



```

/#!
* @file calibration.ino
* @brief This demo tells how to get 14 bytes calibration data.
* @n If you want to obtain reliable calibration data, you need to calibrate under
* @ 1. no target within 40cm of the sensor;
* @ 2. in dark conditions.
*
* @n hardware conneted table:
* -----
* | TMF8x01 | MCU |
* |-----|
* | I2C | I2C Interface |
* |-----|
* | EN | not connected, floating |
* |-----|
* | INT | not connected, floating |
* |-----|
* | PIN0 | not connected, floating |
* |-----|
* | PIN1 | not connected, floating |
* |-----|
*
* @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
* @licence The MIT License (MIT)
* @author [Arya](xue.peng@dfrobot.com)
* @version V1.0
* @data 2021-03-26
* @get from https://www.dfrobot.com
* @url https://github.com/DFRobot/DFRobot\_TMF8x01
*/

#include "DFRobot_TMF8x01.h"

#define EN -1 //EN pin of of TMF8x01 module is floating,
#define INT -1 //INT pin of of TMF8x01 module is floating
//DFRobot_TMF8801 tof(/*enPin =*/EN,/*intPin=*/INT);
DFRobot_TMF8701 tof(/*enPin =*/EN,/*intPin=*/INT);

void setup() {
  Serial.begin(115200);
  while(!Serial){
  }

  Serial.print("Initialization ranging sensor TMF8x01.....");
  while(tof.begin() != 0){
    Serial.println("failed.");
  }
}

```

```

        delay(1000);
    }
    Serial.println("done.");

    /* If you want to obtain reliable calibration data, you need to calibrate under the
       1. no target within 40cm of the sensor;
       2. in dark conditions.*/

    // while(!Serial.available());
    // while(!Serial.available()){
    //     // Serial.read();
    // }

    uint8_t caliDataBuf[14] = {0};
    Serial.print("Calibration ");
    while(tof.getCalibrationData(caliDataBuf, sizeof(caliDataBuf)) != true){
        Serial.print(".");
        delay(1000);
    }

    Serial.println("Calibration complete.");
    Serial.print("caliDataBuf[0-13] = {0x");
    for(int i = 0; i < sizeof(caliDataBuf); i++){
        if(caliDataBuf[i] < 16) Serial.print("0");
        Serial.print(caliDataBuf[i],HEX);
        if(i < 13) Serial.print(", 0x");
    }
    Serial.println("}");

    tof.setCalibrationData(caliDataBuf, sizeof(caliDataBuf)); //Set c

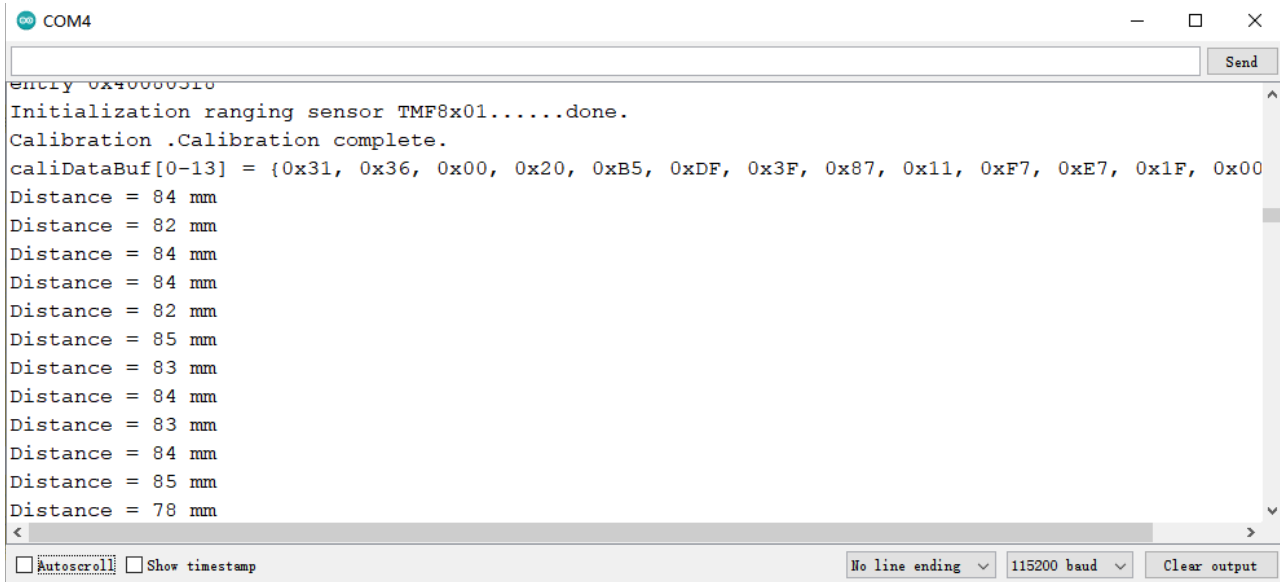
/**
 * @brief Config measurement params to enable measurement. Need to call stopMeasurement
 * @param cailbMode: Is an enumerated variable of eCalibModeConfig_t, which is to c
 * @n     eModeNoCalib :           Measuring without any calibration data.
 * @n     eModeCalib    :           Measuring with calibration data.
 * @n     eModeCalibAndAlgoState : Measuring with calibration and algorithm state.
 * @param disMode : the ranging mode of TMF8701 sensor.(this mode only TMF8701 support)
 * @n     ePROXIMITY: Raing in PROXIMITY mode,ranging range 0~10cm
 * @n     eDISTANCE: Raing in distance mode,ranging range 10~60cm
 * @n     eCOMBINE:  Raing in PROXIMITY and DISTANCE hybrid mode,ranging range 0~60cm
 */
    if(!tof.startMeasurement(/*cailbMode =*/tof.eModeCalib)){ //Enable
        //if(!tof.startMeasurement(/*cailbMode =*/tof.eModeCalib, /*disMode =*/tof.eCOMBINE))
        Serial.println("Enable measurement failed.\nPlease check the calibration data");
        return;
    }
}

void loop() {
    if (tof.isDataReady()) { //Is check
        Serial.print("Distance = ");

```

```
Serial.print(tof.getDistance_mm()); //Print  
Serial.println(" mm");  
}  
}
```

## Expected Result



```
COM4  
Entry 0x40000310  
Initialization ranging sensor TMF8x01.....done.  
Calibration .Calibration complete.  
caliDataBuf[0-13] = {0x31, 0x36, 0x00, 0x20, 0xB5, 0xDF, 0x3F, 0x87, 0x11, 0xF7, 0xE7, 0x1F, 0x00}  
Distance = 84 mm  
Distance = 82 mm  
Distance = 84 mm  
Distance = 84 mm  
Distance = 82 mm  
Distance = 85 mm  
Distance = 83 mm  
Distance = 84 mm  
Distance = 83 mm  
Distance = 84 mm  
Distance = 85 mm  
Distance = 78 mm  
Autoscroll Show timestamp No line ending 115200 baud Clear output
```

## Sample Code 5 - Calibration Mode

The demo `setRangingMode.ino` only supports TMF8701.

Set measurement mode to adjustment ranging distance in TMF8701.

```

/#!
* @file setRangingMode.ino(This demo is only suport TMF8701 sensor)
* @brief Set Ranging mode to adjustment measurement distance in TMF8701
* @n note: TMF8801 only suport one mode, PROXIMITY and DISTANCE hybrid mode.
* *
* -----
* | Type      | suport ranging mode      | ranging ranges | Accuracy
* |-----|-----|-----|-----|
* | TMF8801   | PROXIMITY and DISTANCE   |                | 20~100mm: +/-15mm
* |           | hybrid mode(only one)   | 20~240cm      | 100~200mm: +/-10mm
* |           |                           |                | >=200: +/-%5
* |-----|-----|-----|-----|
* |           | PROXIMITY mode          | 0~10cm        |
* |           |-----|-----|-----| >=200: +/-%5
* | TMF8701   | DISTANCE mode           | 10~60cm       | 100~200mm: +/-10mm
* |           |-----|-----|-----|
* |           | PROXIMITY and DISTANCE  | 0~60cm        |
* |           | hybrid mode             |                |
* |-----|-----|-----|-----|
* *
* @n hardware conneted table:
* -----
* | TMF8x01   | MCU                      |
* |-----|-----|
* | I2C       | I2C Interface           |
* |-----|-----|
* | EN        | not connected, floating |
* |-----|-----|
* | INT       | not connected, floating |
* |-----|-----|
* | PIN0      | not connected, floating |
* |-----|-----|
* | PIN1      | not connected, floating |
* |-----|-----|
*
* @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
* @licence The MIT License (MIT)
* @author [Arya](xue.peng@dfrobot.com)
* @version V1.0
* @data 2021-04-02
* @get from https://www.dfrobot.com
* @url https://github.com/DFRobot/DFRobot\_TMF8x01
*/

#include "DFRobot_TMF8x01.h"

```

```

#define EN      -1                //EN pin of of TMF8x01 module is floating,
#define INT     -1                //INT pin of of TMF8x01 module is floating

DFRobot_TMF8701 tof(/*enPin =*/EN,/*intPin=*/INT);

void setup() {
  Serial.begin(115200);
  while(!Serial){
  }

  Serial.print("Initialization ranging sensor TMF8x01.....");
  while(tof.begin() != 0){
    Serial.println("failed.");
    delay(1000);
  }
  Serial.println("done.");

  Serial.print("Software Version: ");
  Serial.println(tof.getSoftwareVersion());
  Serial.print("Unique ID: ");
  Serial.println(tof.getUniqueID(),HEX);
  Serial.print("Model: ");
  Serial.println(tof.getSensorModel());

/**
 * @brief Config measurement params to enable measurement. Need to call stopMeasurement
 * @param cailbMode: Is an enumerated variable of eCalibModeConfig_t, which is to c
 * @n      eModeNoCalib   :           Measuring without any calibration data.
 * @n      eModeCalib     :           Measuring with calibration data.
 * @n      eModeCalibAndAlgoState : Measuring with calibration and algorithm state.
 * @param disMode : the ranging mode of TMF8701 sensor.(this mode only TMF8701 support)
 * @n      ePROXIMITY: Raing in PROXIMITY mode,ranging range 0~10cm
 * @n      eDISTANCE: Raing in distance mode,ranging range 10~60cm
 * @n      eCOMBINE: Raing in PROXIMITY and DISTANCE hybrid mode,ranging range 0~60cm
 */
  tof.startMeasurement(/*cailbMode =*/tof.eModeCalib, /*disMode =*/tof.ePROXIMITY);
}

void loop() {
  if (tof.isDataReady()) {
    Serial.print("Distance = ");
    Serial.print(tof.getDistance_mm());
    Serial.println(" mm");
  }
}

```

## Expected Result

COM4



Send

Initialization ranging sensor TMF8x01.....done.

Software Version: 2.0.22.3

Unique ID: 384016DF

Model: unknown

Distance = 23 mm

Distance = 22 mm

Distance = 22 mm

Distance = 22 mm

Distance = 22 mm

Distance = 23 mm

Distance = 23 mm

Distance = 23 mm

Distance = 23 mm

Distance = 23 mm

Distance = 23 mm

Distance = 23 mm

Autoscroll  Show timestamp

No line ending

115200 baud

Clear output