TDS Sensor Water Quality Detection Module
BM25S4021-1
Example Description
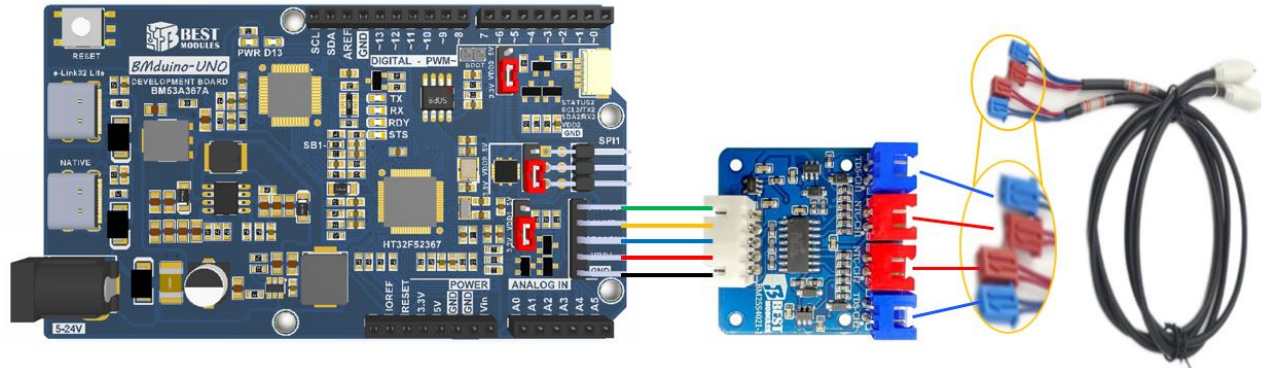
Version：V0.01   Date：2025-03-19

Introduction

BM25S4021-1 is a TDS detection module launched by Best Modules, which uses UART for control. The example demonstrates the use of the BM25S4021-1 module to read TDS values and set the module ID, among other functions.

Arduino Example

Physical Connection Diagram

Example 1: readTDS

Function: Read the TDS value.

> Note: This example demonstrates reading the module with ID 1. If the ID does not match, refer to Example 3 for setting the ID.

    Open Example: Select Example (readTDS)

1.   Example Description:

a. Construct object & initialize module

```
#include <BM25S4021_1.h>
#include <SoftwareSerial.h>
//BM25S4021_1 myTDS(7,6);//SoftwareSerial RX TX pin of arduino UNO and BMduino
BM25S4021_1 myTDS(&Serial1);//HardwareSerial of BMduino BMCOM1
uint8_t channel,ID;/*TDS chanel*/
float TDSValue ;
float TempValue ;

void setup()
{
   ID = 1;
   myTDS.begin();
   Serial.begin(115200);
    delay(600);
}
```

b. Read TDS and Temperature Values from Channel 1 (CH1) Every Second and Print to Serial Monitor

Read the TDS value and temperature value from Channel 1 (CH1) every second and print them to the serial monitor.

```
void loop()
{
  channel = 1;  //CH1
  TDSValue  = myTDS.readTDS(ID,channel);
  Serial.print("TDS");
  Serial.print(channel);
  Serial.print(" value is ");
  Serial.print(TDSValue,1);
  Serial.print(" ppm");
  Serial.print("(");
  Serial.print(TDSValue*2,1);
  Serial.println(" uS/cm)");
  TempValue = myTDS.readTemp(ID,channel);
  if(TempValue == 6548.6)
  {
     Serial.print("NTC interface is open circuit");
  }
  else if(TempValue == 150)
  {
     Serial.print("NTC interface is short circuit");
  }
  Serial.print("temperature");
  Serial.print(" value is ");
  Serial.print(TempValue,1);
  Serial.println(" ℃");
  delay(1000);
}
```

c. Open the serial monitor, set the baud rate to 115200; the serial monitor will display the TDS and temperature values of CH1 as follows：

```
Output   Serial Monitor  ×

Message (Enter to send message to 'BM53A367A (BMduino-UNO HT32F52367)' on 'COM4')   New Line  ▼   115200 baud  ▼

11:22:16.708 -> TDS1 value is 153.9 ppm(307.8 uS/cm)
11:22:16.708 -> temperature value is 21.9 ℃
11:22:17.300 -> TDS1 value is 154.2 ppm(308.4 uS/cm)
11:22:17.334 -> temperature value is 21.9 ℃
11:22:18.350 -> TDS1 value is 154.4 ppm(308.8 uS/cm)
11:22:18.397 -> temperature value is 21.9 ℃
11:22:19.418 -> TDS1 value is 154.7 ppm(309.4 uS/cm)
11:22:19.418 -> temperature value is 21.9 ℃
11:22:20.435 -> TDS1 value is 155.2 ppm(310.4 uS/cm)
11:22:20.469 -> temperature value is 21.8 ℃
11:22:21.502 -> TDS1 value is 155.5 ppm(311.0 uS/cm)
11:22:21.502 -> temperature value is 21.9 ℃
11:22:22.555 -> TDS1 value is 155.8 ppm(311.6 uS/cm)
11:22:22.555 -> temperature value is 21.9 ℃
11:22:23.584 -> TDS1 value is 156.2 ppm(312.4 uS/cm)
11:22:23.619 -> temperature value is 21.9 ℃
```

Example 2：reset

Function: Reset the module.
Open Example: Select Example (reset)
Example Description:
a. Construct object & initialize module
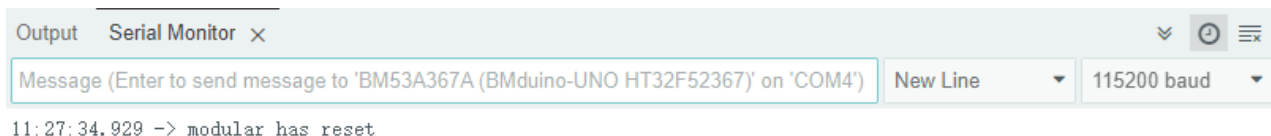
```
#include <BM25S4021_1.h>
#include <SoftwareSerial.h>
//BM25S4021_1 myTDS(7,6);//SoftwareSerial RX TX pin of arduino UNO and BMduino
BM25S4021_1 myTDS(&Serial1);//HardwareSerial of BMduino BMCOM1
uint8_t ID;
void setup()
{
   ID = 1;
   myTDS.begin();
   Serial.begin(115200);
   delay(600);
}
```

b. Reset module

```
void loop()
{
   myTDS.reset(ID);
   Serial.println("modular has reset");
   //Serial.println(myTDS.reset(ID));

   while(1);
}
```

c. Open the serial monitor, set the baud rate to 115200; the serial monitor will display a successful reset message as follows:

Output Serial Monitor ×	⩢ ⊙ ☰ₓ
Message (Enter to send message to 'BM53A367A (BMduino-UNO HT32F52367)' on 'COM4') New Line ▼ 115200 baud ▼	

```
11:27:34.929 -> modular has reset
```

Example 3: setID

Function: Set the module ID.
Open Example: Select Example (setID)
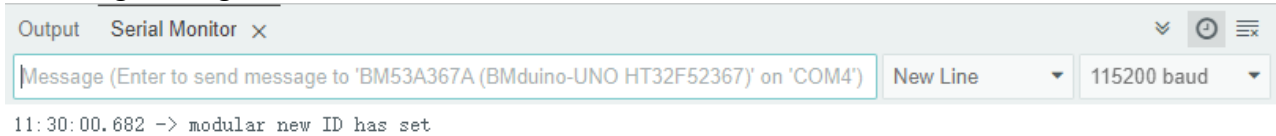**Example Description:**
a. Construct object & initialize module

```cpp
#include <BM25S4021_1.h>
#include <SoftwareSerial.h>
//BM25S4021_1 myTDS(7,6);//SoftwareSerial RX TX pin of arduino UNO and BMduino
BM25S4021_1 myTDS(&Serial1);//HardwareSerial of BMduino BMCOM1
uint8_t oldID;
uint8_t newID;
void setup()
{
    oldID = 1;
    newID = 2;
    myTDS.begin();
    Serial.begin(115200);
    delay(600);
}
```

b. Change the module ID from 1 to 2.

```cpp
void loop()
{
    myTDS.setID(oldID,newID);
    Serial.println("modular new ID has set");
    //Serial.println(myTDS.setID(oldID,newID));
    while(1);
}
```

c. Open the serial monitor, set the baud rate to 115200; the serial monitor will display a successful ID setting message as follows:

Output Serial Monitor ×			≫ ⊙ ≡x
Message (Enter to send message to 'BM53A367A (BMduino-UNO HT32F52367)' on 'COM4')	New Line ▾	115200 baud ▾	

```
11:30:00.682 -> modular new ID has set
```

Example 4: setWarnValue

2. Function: Set the TDS alarm value.
3. Open Example: Select Example (setWarnValue)

Example Description:
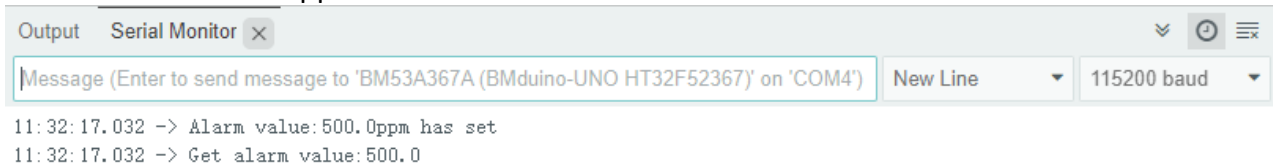
a. Construct object & initialize module

```
#include <BM25S4021_1.h>
#include <SoftwareSerial.h>
//BM25S4021_1 myTDS(7,6);//SoftwareSerial RX TX pin of arduino UNO and BMduino
BM25S4021_1 myTDS(&Serial1);//HardwareSerial of BMduino BMCOM1
uint8_t channel,ID;
float warnValue;
void setup()
{
   ID = 1;
   channel = 1;
   warnValue = 500;
   myTDS.begin();
   Serial.begin(115200);
   delay(600);
}
```

b. Set the TDS alarm value to 500 and read the alarm value to confirm.

```
void loop()
{
        myTDS.setWarn(ID,channel,warnValue);
        Serial.print("Alarm value:");
        Serial.print(warnValue,1);
        Serial.println("ppm has set");
        warnValue = 0;
        warnValue = myTDS.getWarn(ID,channel);
        Serial.print("Get alarm value:");
        Serial.println(warnValue,1);
        while(1);
}
```

c. Open the serial monitor, set the baud rate to 115200; the serial monitor will display the module's alarm value set to 500 ppm as follows:

```
Output    Serial Monitor  ×                                                    ⤓  ⏱  ⌸

Message (Enter to send message to 'BM53A367A (BMduino-UNO HT32F52367)' on 'COM4')   New Line  ▼   115200 baud  ▼

11:32:17.032 -> Alarm value:500.0ppm has set
11:32:17.032 -> Get alarm value:500.0
```

Example 5: setWorkMode

Function: Set the work mode.
Open Example: Select Example (setWorkMode)
Example Description:
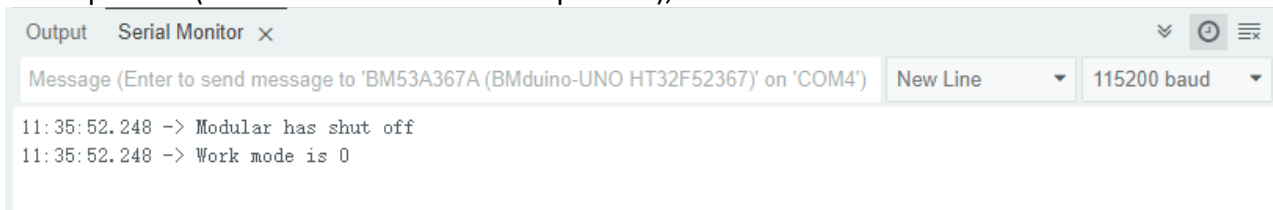a. Construct object & initialize module

```
#include <BM25S4021_1.h>
#include <SoftwareSerial.h>
//BM25S4021_1 myTDS(7,6);//SoftwareSerial RX TX pin of arduino UNO and BMduino
BM25S4021_1 myTDS(&Serial1);//HardwareSerial of BMduino BMCOM1
uint8_t ID,status;
void setup() {
   ID = 1;
   myTDS.begin();
   Serial.begin(115200);
   delay(600);
}
```

b. Set the work mode to sleep mode.

```
void loop() {
   myTDS.setWorkMode(ID,SLEEP_MODE);
   Serial.println("Modular has shut off");
   status = myTDS.getWorkMode(ID);
   Serial.print("Work mode is ");
   Serial.println(status);
   while(1);
}
```

c. Open the serial monitor, set the baud rate to 115200; the serial monitor will display the module in sleep mode (work mode 0 indicates sleep mode), as follows:

```
Output    Serial Monitor  ×                                                    ⤓  ⏱  ⌸

Message (Enter to send message to 'BM53A367A (BMduino-UNO HT32F52367)' on 'COM4')   New Line  ▼   115200 baud  ▼

11:35:52.248 -> Modular has shut off
11:35:52.248 -> Work mode is 0
```