Modify Address and Multi-machine Linkage of TFmini_I2C

Catalog

1. Experiment objective and principle ..... 错误!未定义书签。
1.1 Experiment objective..... 错误!未定义书签。
1.2 Experiment principle 3
2. Experiment devices..... 3
3. Cable connection ..... 错误!未定义书签。
3.1 Modify the address of one set and testing wiring diagram ..... 错误!未定义书签。
3.2 Multiple sensors linkage test wiring diagram..... 5
4. TFmini I2C version address modifying command descriptions 6
4.1 Modifying address commands 6
4.2 Reset command..... 7
5. Modify address and testing process..... 8
6. Related code and analysis..... 8
6.1 Address modifying code(Arduino IDE) 9
6.2 Reset code(Arduino IDE) 10
6.3 Read data code(Arduino IDE)..... 10

1. Purpose and Principle

1.1 Purpose

Provides a stable and reliable process and code examples that can be referenced to modify the address of the I2C version of TFmini. This I2C version of the TFmini multi-machine linkage experiment will use Arduino UNO as the experimental carrier. For other carriers, please refer to this article for port development or use Arduino UNO to change the TFmini address and use it for other carriers.

1.2 Principle

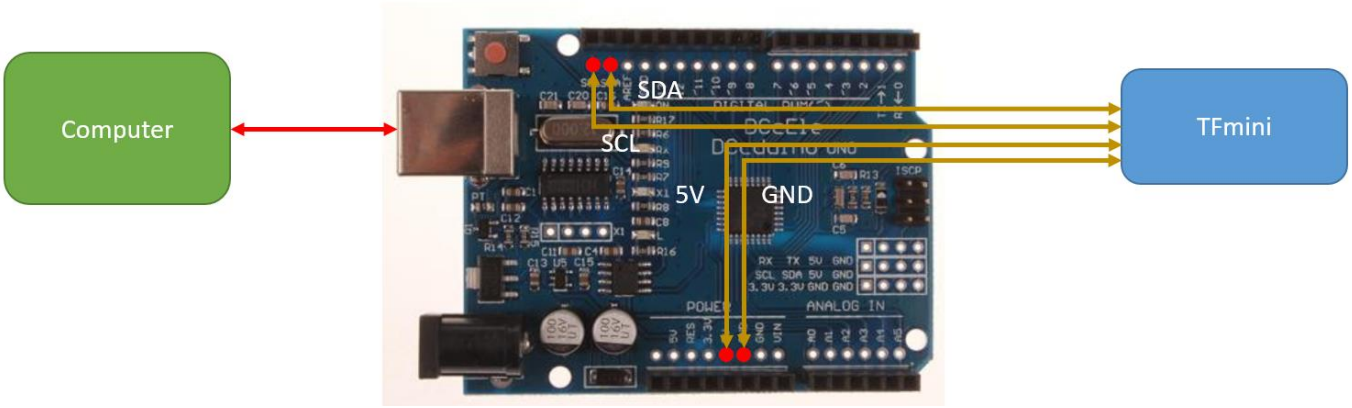
Use Arduino UNO to modify the address of each TFmini in turn, and then modify the multi-machine collaborative test. In this experiment, two TFminis will be used as test samples for multiple machines. If more TFminis are connected, simply follow the procedure to continue to modify the address.

2. Tools

- Arduino UNO *1
- TFmini I2C version ∗2
- 5V Power Source ∗1
- Dupont cable *N(It depends on the real needs)
- Arduino IDE

3. Wiring

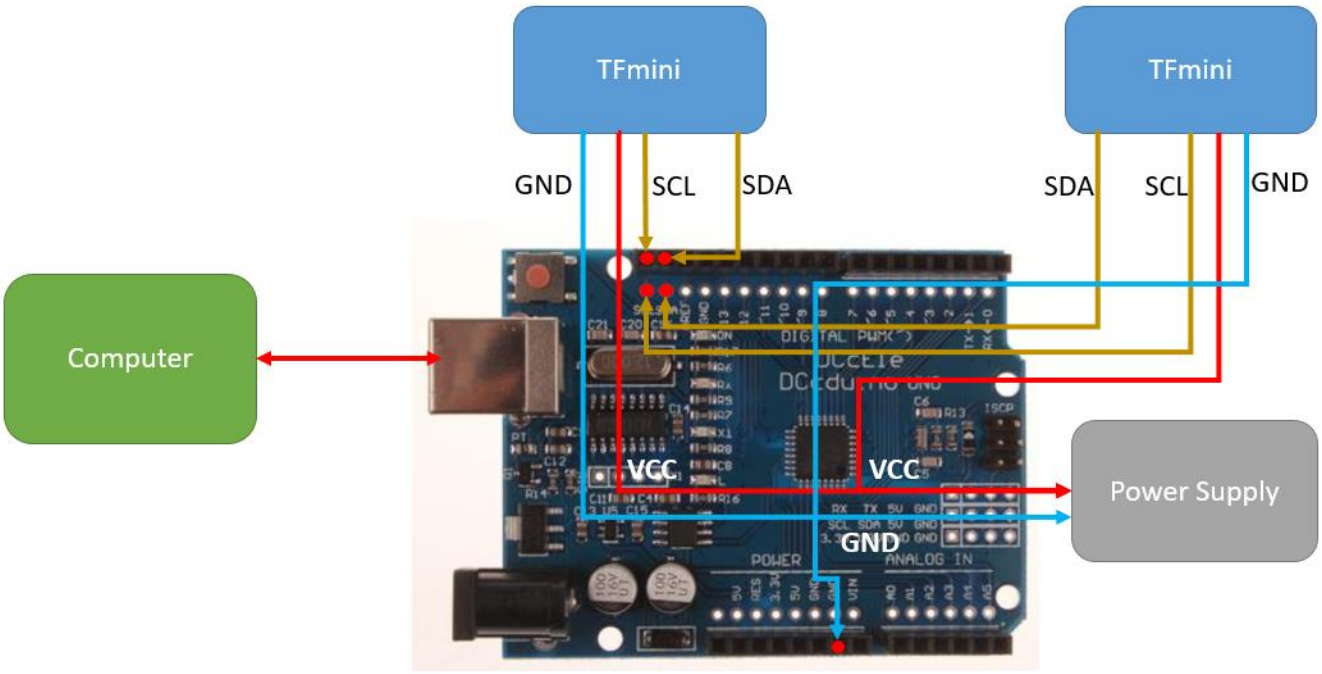
3.1 Single machine modification address and test wiring



When testing the TFmini stand-alone and modifying the address, you can wire it as follows:

Computer	Arduino UNO		TFmini
USB connection	USB connection	GND	GND
		5V	5V
		SCL	SCL
		SDA	SDA

3.2 Multiple sensors linkage test wiring

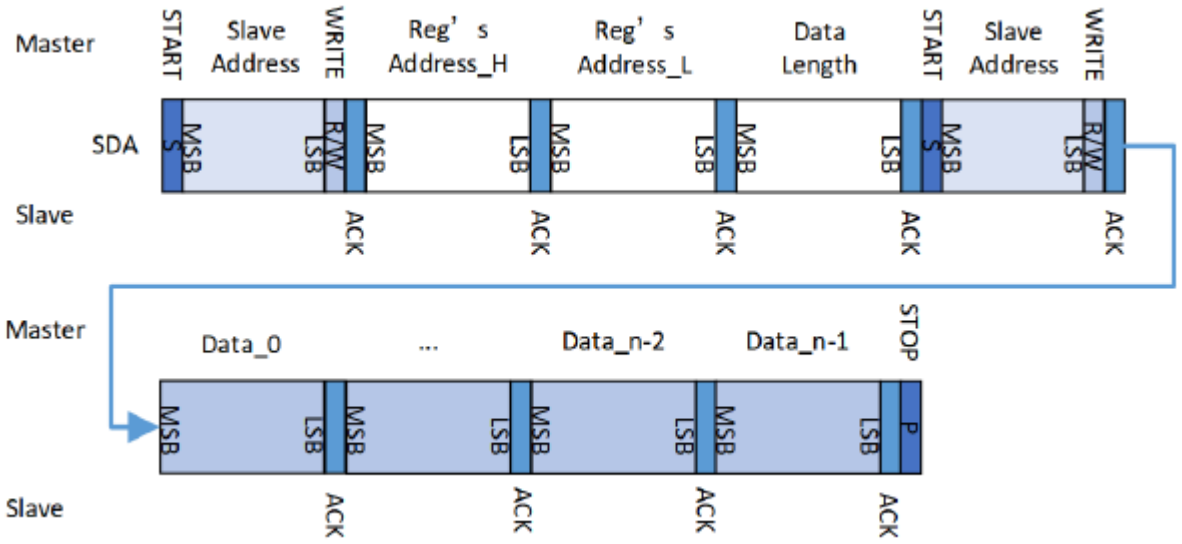


Cable connection could be referred to the following table when multiple sensors linkage testing:

Computer	Arduino UNO		TFmini-1	TFmini-2
USB connection	USB connection	GND(External Power Source)	GND(External Power Source)	GND(External Power Source)
		5V	5V(External Power Source)	5V(External Power Source)
		SCL	SCL	SCL
		SDA	SDA	SDA

Note: At this time, the power supply of the Arduino will not be able to meet the two TFminis at the same time. It is necessary to provide the TFmini with an external power supply of 5V and a peak current of 800mA, and the external power supply and the Arduino UNO should be common-ground.

4. Command Descriptions



Time sequence of TFmini-I2C parameters configuration

See figure above for time sequence configuration of TFmini-I2C, we can know that there are two command need to be sent to slave if any parameter need to be changed.

The first command:

- Declare slave address
- Declare slave register high eight byte address
- Declare slave register lower eight byte address
- Declare data length will be transmitted

The second command:

- Declare slave address
- Input every bit of data value according to the data length written by the first command

According to the product manual, The default slave address of TFmini is 0x10, modified slave address will take effect after power on or system reset. So there are two commands need to be executed for modifying slave address: address modifying and system reset.

4.1 Modifying address commands

Slave address is saved in register 0x0026(decimal 38), so register modifying commands could

be broken up into:

The first command:

- Declare slave address: 0x10(default setting)
- Declare high eight byte address:0x00
- Declare lower eight byte address:0x26
- Declare data length:1

The second command:

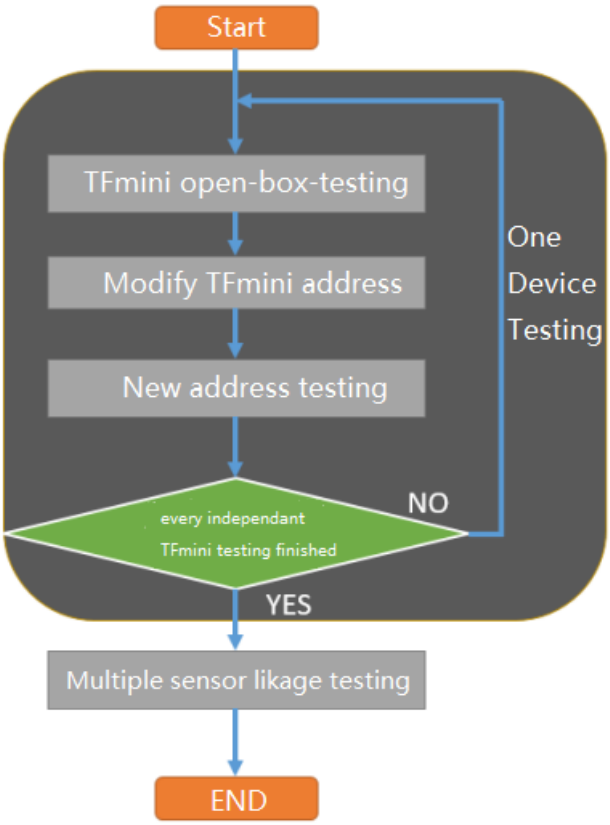
- Declare slave address:0x10
- Read-in data: new address value

4.2 Reset command

Time sequence of reset command is different with other commands, only send data 0x06 to 0x00 address for reset operation, that is:

- Declare slave address:0x00
- Read-in data:0x06

5. Modify Address and Testing



There are some important experiment point:

1. TFmini open-box testing: default address and data reading and writing function of every TFmini need to be tested independently, please contact with the related sales and FAE if any error appears.
2. Modify TFmini address: after open-box testing, TFmini address need to be modified by the devices of Arduino UNO or other devices.
3. New address testing: Check whether the TFmini with its new address could be read and wrote by the devices of Arduino UNO or other devices.
4. Test single device one by one in cycle until every TFmini address is unique and could read distance information normally.
5. Test prepared TFmini-I2C by linking.(It may need independent 5V external power source)

6. Code

Address modifying, resetting, reading and writing code will be introduced in this chapter could be used in your Arduino development environment.

6.1 Address modifying code(Arduino IDE)

```
/**
 * @Description: Modify TFmini address from @Addr to @value
 * Params:
 *          @Addr: current address of TFmini, ie 0x10
 *          @value: new address, ie 0x11
 */
void SET_TFMini_I2CAddr(byte Addr, byte value) {
  //Start a command, declare @Addr, Reg_H, Reg_L, DATA Length
  Wire.beginTransmission(Addr); // Begin a transmission to the I2C Slave device with the given address.
  Wire.write(0); // Reg's Address_H
  Wire.write(38); // Reg's Address_L , OCT 38 == HEX 0x26
  Wire.write(1); // Data Length
  Wire.endTransmission(0);   // Send a START Sign
  //Start a command, store new value to Addr
  Wire.beginTransmission(Addr); // Begin a transmission to the I2C Slave device with the given address.
  Wire.write(value); // I2C new Address
  Wire.endTransmission(1);   // Send a STOP Sign
  Serial.print("Set Addr ");
  Serial.print(Addr);
  Serial.print(" as ");
  Serial.print(value);
  Serial.print(".\r\n");
  delay(500);
}
```

This piece of code in parameter:

Addr (byte),be used for declaring current TFmini address

value (byte),be used for declaring TFmini new address

There are two pieces of communication, the first paragraph is used for declaring slave address, register lower eight byte, register upper eight byte, and data length. The second paragraph is used for writing data to slave.

6.2 Reset code(Arduino IDE)

```
/**
 * @Description: Reset TFmini
 */
void RESET_I2CBus() {
   //send 0x06 to address 0x00
   Wire.beginTransmission(0);
   Wire.write(6);
   Wire.endTransmission(1);
   delay(1000);
}
```

It's unnecessary to declare read and writing,0x06 will be written to address 0x00 when calling ,all of TFmini in this BUS will be reset by resetting I2CBus.

6.3 Read data code(Arduino IDE)

```
/**
 * @Description: Get and print reading data from TFmini
 *   data_0   data_1 data_2 data_3 data_4        data_5        data_6
 *   Trigger     0      Dist_L Dist_H Strength_L Strength_H    Mode
 */
void GetAndPrintTFminiData(byte Addr) {
   byte rx_Num;
   byte rx_buf[7] = {0};
   byte i = 0;

   //Start a command, tell TFmini at Addr to measure distance
```

```
  Wire.beginTransmission(Addr); // Begin a transmission to the I2C Slave device with the given address.
  Wire.write(1); // Reg's Address_H
  Wire.write(2); // Reg's Address_L
  Wire.write(7); // Data Length
  Wire.endTransmission(0);    // Send a START Sign

  // Wire.requestFrom（AA,BB）;receive the data form slave.
  // AA: Slave Address ; BB: Data Bytes
  rx_Num = Wire.requestFrom(Addr, 7);

  // Wire.available: Retuens the number of bytes available for retrieval with read().
  while ( Wire.available())
  {
    rx_buf[i] = Wire.read(); // received one byte
    i++;
  }
      // OUTPUT
  Serial.print("Addr in OCT = ");
  Serial.print(Addr);
  Serial.print(":TrigFlag= ");
  Serial.print(rx_buf[0]);
  Serial.print(",Dist= ");
  Serial.print(rx_buf[2] | (rx_buf[3] << 8));
  Serial.print(",Strength= ");
  Serial.print(rx_buf[4] | (rx_buf[5] << 8));
  Serial.print(",Inttime= ");
  Serial.print(rx_buf[6]);
  Serial.print("\r\n");
}
```

This piece of code in parameter:

Addr, byte, is used for declaring TFmini reading and writing data of slave address. It has been sent one time and received one time in the code.

Sending: send data 0x07 to register 0x0102 of TFmini.

Receiving: declare array length which is 7 is used for saving 7 bit data returned from TFmini.

The meaning of every bit data is as the followings:

data_0	data_1	data_2	data_3	data_4	data_5	data_6
Trigger	0	Dist_L	Dist_H	Strength_L	Strength_H	Mode

Gather distance and signal strength from data_2 to data_5.