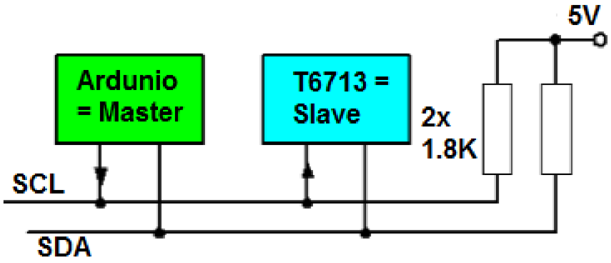
Application Note AN161:
Arduino I2C Interface to TelAire T6713 CO2 Sensor

Introduction



The Arduino Uno, Mega 1280 or Mega2560 are ideal microcontrollers for reading a TelAire T6713 CO2 sensor. The interface to the sensor is referred to as I2C or TWI (Two Wire Interface). To read the output, we recommend using the Arduino software Graphical User Interface (GUI).

If you are new to Arduino, these low cost development boards are available from many sources. We recommend you start with genuine Arduino products.

The TelAire T6713 CO2 Sensor is the latest generation of compact non-dispersive infrared (NDIR) carbon dioxide sensors on the market. It measures CO2 levels up to 5,000ppm using a gold-plated optical path to insure accuracy and long-life. The sensor requires 4.5-5 VDC, 200mA peak current and 25mA average current. Although both UART (serial) and I2C interfaces may be used to communicate with the sensor, in this example we use I2C.

Run the Blink Example

The best way to become familiar with the GIU is to verify your Arduino board is operating properly is to create an Arduino project and run the example **Blink**. This simple test program confirms that a number of connection details and the GUI are working properly.

Caution: Do not connect your Arduino board to your computer's USB port until the Arduino software is installed. Otherwise Windows will install a generic driver and the Arduino board will not operate.

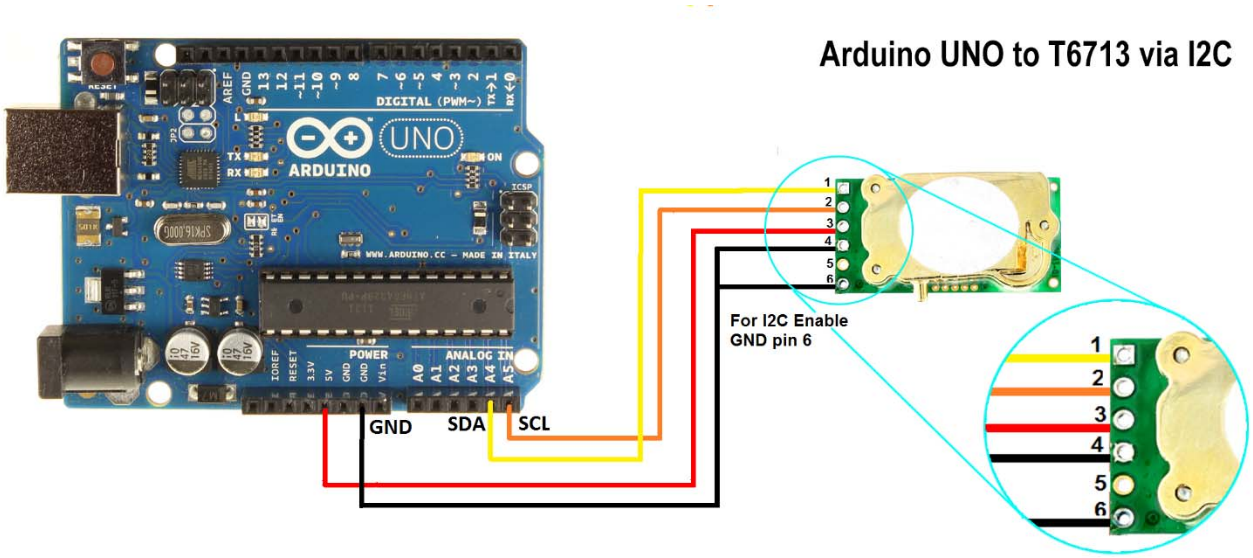
Step 1: Install Arduino software on your computer. From this page select the **Windows Installer.**
https://www.arduino.cc/en/Main/Software

Step 2: To run the Blink example follow these instructions:
https://www.arduino.cc/en/Tutorial/Blink

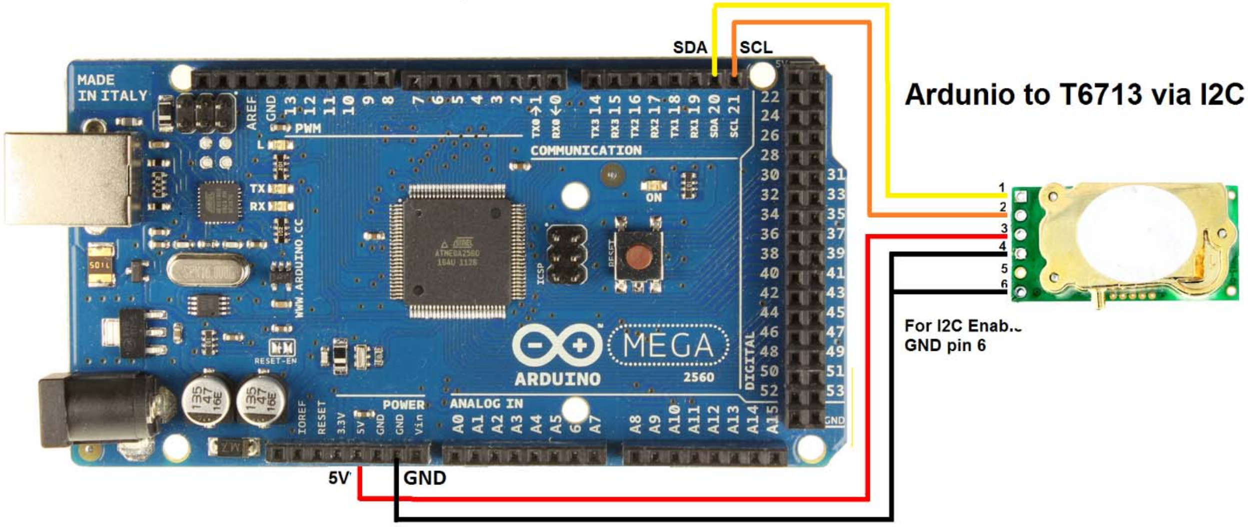
Note that the MEGA Arduinos have a LED on board. The Arduino UNO may require a LED and resistor be added as specified in the tutorial. Once Blink runs properly, you can connect the sensor as shown below.

Connecting the T6713 Sensor

Refer to the wiring diagram below for the Arduino Uno



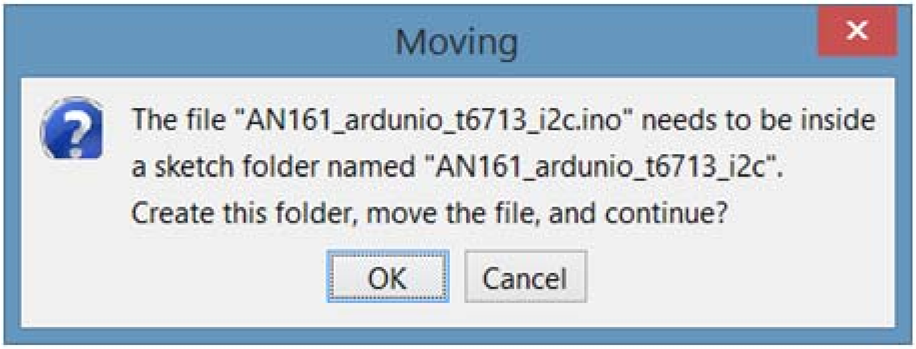
Refer to the wiring diagram below for the Arduino MEGA or Arduino Mega 2560. The I2C connections for Arduino MEGA and the Mega2560 are the same.



Create an Arduino Project

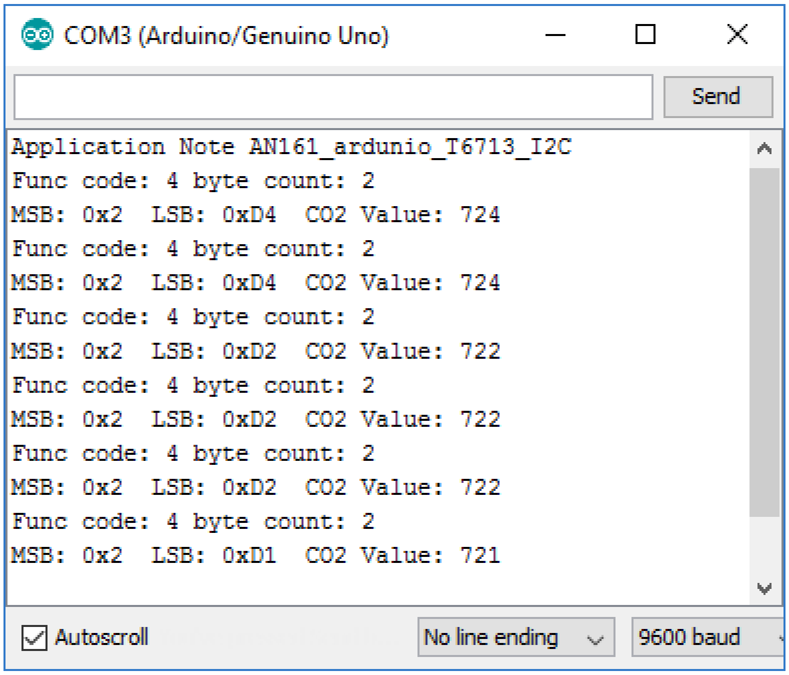
1. Check that your Arduino is connected to your USB port and that a LED on the board is on.
2. Connect the T6713 sensor to the Arduino board as shown above.
3. Go to CO2Meter.com, select App Notes, and download AN161 Sample Code.zip or click this link:
4. http://www.co2meters.com/Documentation/AppNotes/AN161-T6713-arduino-i2c.zip

5. The AN161-T6713-arduino-i2c.zip file will be downloaded to your Downloads directory.
6. Open the file and select the AN161_ardunio_t6713_i2c.ino file.



The file "AN161_ardunio_t6713_i2c.ino" needs to be inside a sketch folder named "AN161_ardunio_t6713_i2c".
Create this folder, move the file, and continue?

7. Click OK. Observe the Arduino project code is displayed.
8. Click Sketch >> Verify/Compile. The project should compile without errors.
9. Verify that your Arduino board is recognized correctly:
   a. Select Tools. Set board type to UNO or MEGA.
   b. Confirm that Processor matches your Arduino: UNO, MEGA or MEGA 2560.
10. Click Upload. When you see “Done uploading” your project is running on the Arduino board.
11. To view program operation, click on Tools >> Serial Monitor. Observe the following:



```
Application Note AN161_ardunio_T6713_I2C
Func code: 4 byte count: 2
MSB: 0x2  LSB: 0xD4  CO2 Value: 724
Func code: 4 byte count: 2
MSB: 0x2  LSB: 0xD4  CO2 Value: 724
Func code: 4 byte count: 2
MSB: 0x2  LSB: 0xD2  CO2 Value: 722
Func code: 4 byte count: 2
MSB: 0x2  LSB: 0xD2  CO2 Value: 722
Func code: 4 byte count: 2
MSB: 0x2  LSB: 0xD2  CO2 Value: 722
Func code: 4 byte count: 2
MSB: 0x2  LSB: 0xD1  CO2 Value: 721
```

☑ Autoscroll   No line ending   9600 baud

The contents of .txt file is shown below. You can copy/paste this text into a new Arduino empty sketch, but take care to remove this app note page numbers at the bottom left corner.

Sample Source Code

```
// CO2 Meter Amphenol T6713 i2c Example Interface

// By Marv Kausch, 12/2016 at CO2 Meter <co2meter.com>
// Revised by John Houck, 05/01/2018
// Talks via I2C to T6713 sensors and displays CO2 values
// We will be using the I2C hardware interface on the Arduino in
// combination with the built-in Wire library to interface.
// Arduino analog input 5 - I2C SCL
// Arduino analog input 4 - I2C SDA
/*
  In this example we will do a basic read of the CO2 value and
checksum verification.
  For more advanced applications please see the I2C Comm guide.
*/
#include <Wire.h>  // Although Wire.h is part of the Ardunio GUI
library, this statement is still necessary
#define ADDR_6713  0x15 // default I2C slave address
int data [4];
int CO2ppmValue;

// This is the default address of the CO2 sensor, 7bits shifted
left.
void setup() {
  Wire.begin ();
  Serial.begin(9600);
  Serial.println("Application Note AN161_ardunio_T6713_I2C");
}

//////////////////////////////////////////////////////////////////
/
// Function : int readCO2()
// Returns : CO2 Value upon success, 0 upon checksum failure
// Assumes : - Wire library has been imported successfully.
// - LED is connected to IO pin 13
// - CO2 sensor address is defined in co2_addr
//////////////////////////////////////////////////////////////////
/

int readC02()
{
  // start I2C
  Wire.beginTransmission(ADDR_6713);
  Wire.write(0x04); Wire.write(0x13); Wire.write(0x8B);
Wire.write(0x00); Wire.write(0x01);
  // end transmission
  Wire.endTransmission();
  // read report of current gas measurement in ppm
  delay(2000);
  Wire.requestFrom(ADDR_6713, 4);    // request 4 bytes from slave
device
  data[0] = Wire.read();
  data[1] = Wire.read();
  data[2] = Wire.read();
  data[3] = Wire.read();
    Serial.print("Func code: "); Serial.print(data[0],HEX);
    Serial.print(" byte count: "); Serial.println(data[1],HEX);
    Serial.print("MSB: 0x");  Serial.print(data[2],HEX);
Serial.print("  ");
    Serial.print("LSB: 0x");  Serial.print(data[3],HEX);
Serial.print("  ");
   CO2ppmValue = ((data[2] * 0xFF ) + data[3]);
}

void loop() {

  int co2Value =readC02();
  {
    Serial.print("CO2 Value: ");
    Serial.println(CO2ppmValue);
  }

  delay(2000);
}
```