Basic Concept of Using Arduino for Conducting Chemistry Experiment

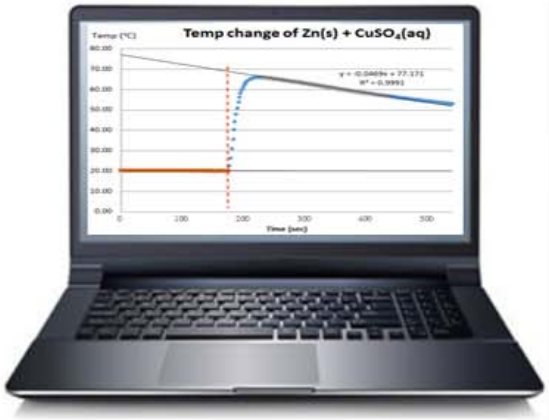
What is Arduino ?

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- Arduino boards are able to
  - read inputs - light on a sensor, a finger on a button
  - turn it into an output - activating a motor, turning on an LED, publishing something online.
  - You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

Input → Arduino Microcontroller → Output

Temperature sensor

Arduino UNO



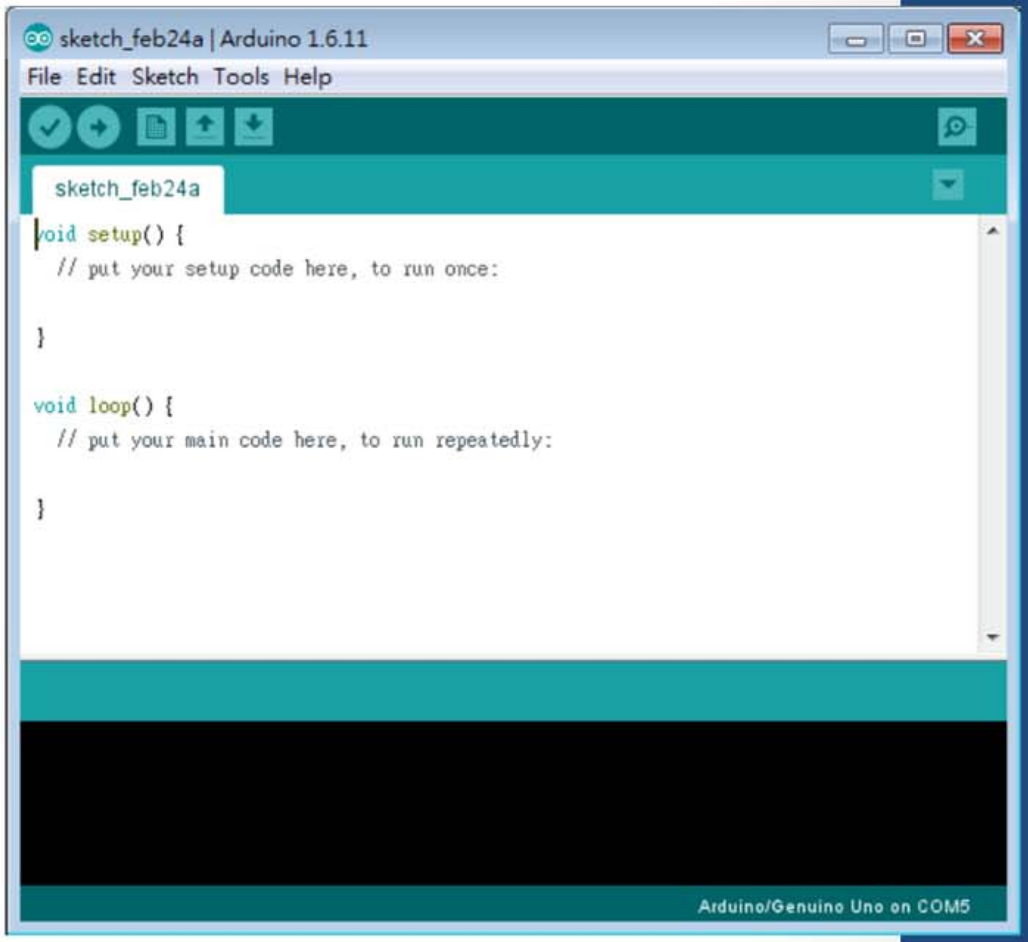
Arduino code

Software

- Arduino IDE Software
- Excel Plugins (PLX-DAQ)
- Arduino Code

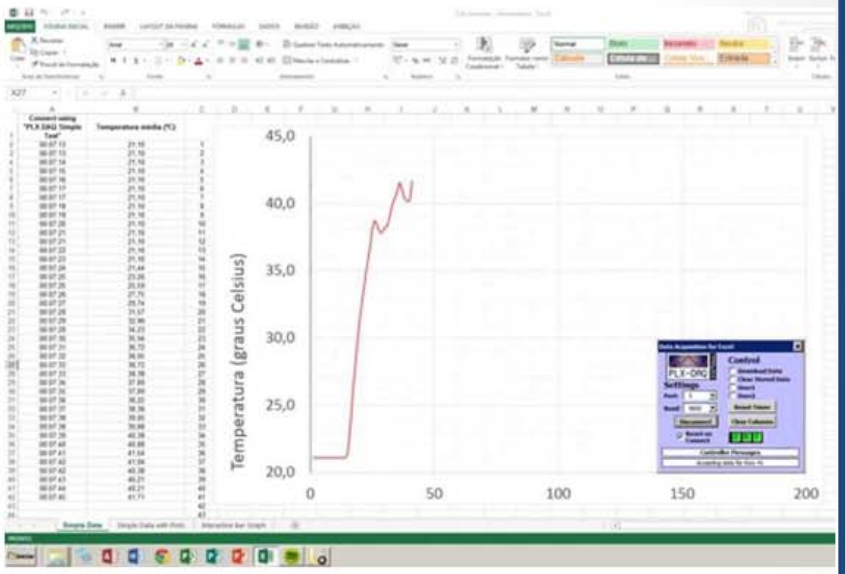
Arduino IDE Software

- Open-source
- Free to use
- Compiles and uploads codes to our Arduino boards
- Pre-loaded with examples for self-learning
- With Compiler to check your error(s)



Excel Plug-ins (PLX-DAQ)

- Plug-ins for Microsoft Excel
- For data acquisition
- Can act as data-loggers' interface

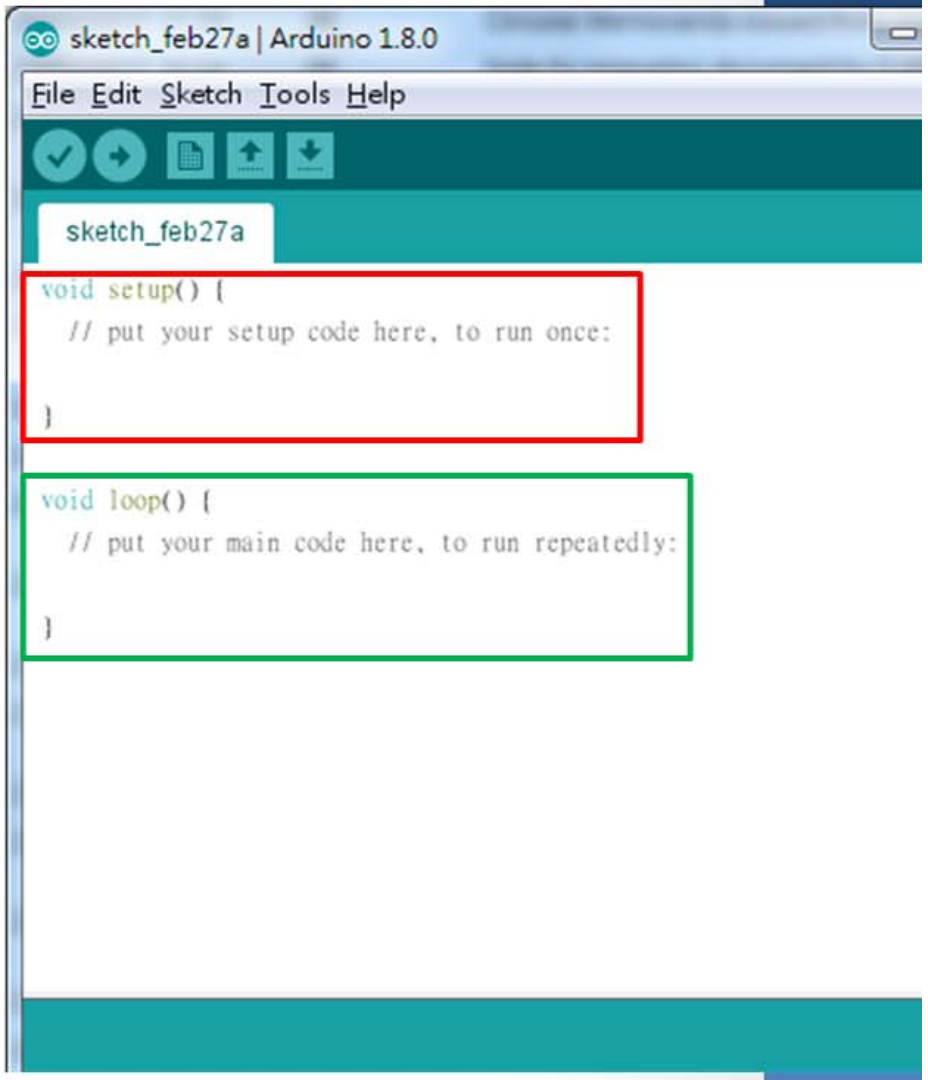


- Display the sensor reading simultaneously
- Record sensor reading data (up to 1048576) for further analysis

Arduino Coding

Basic Sketch

- 2 major functions
  - setup (run once only)
  - Loop (run continuously when executed)

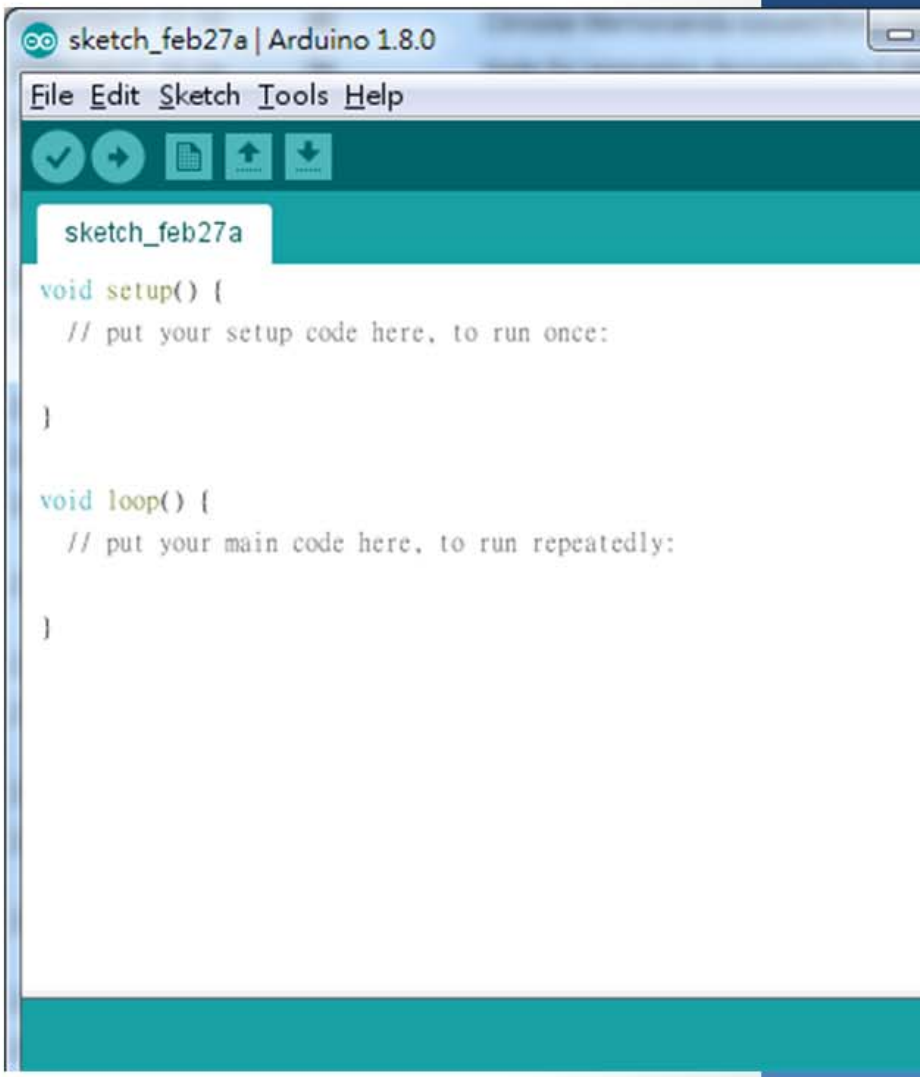


Basic Sketch

- 2 major functions
  - setup (run once only)
  - Loop (run continuously when executed)

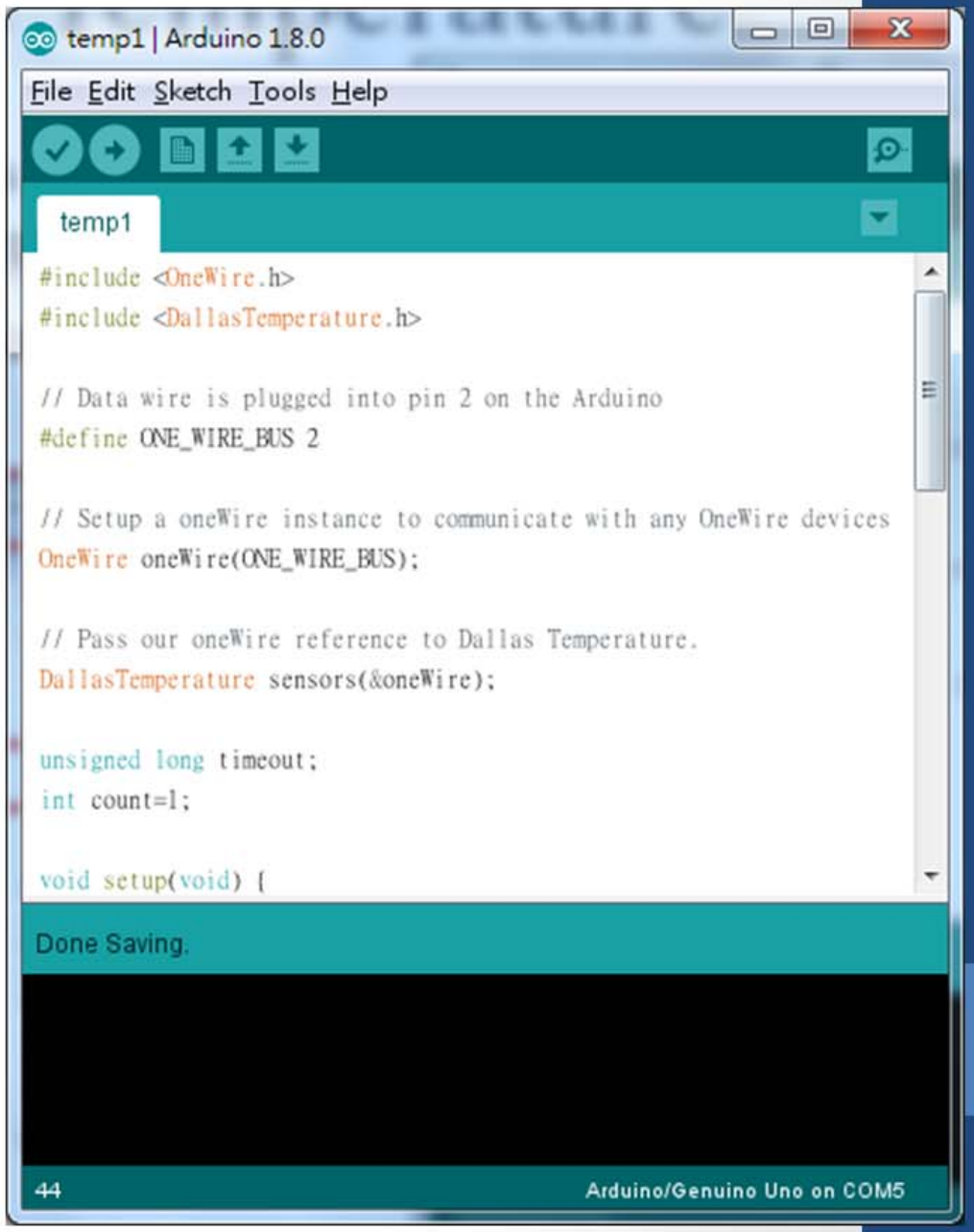
- ( ) – end of function name
- The body of a function consists of an { }
- Each line usually ends with a semi-colon ;

- // comment
  - Only one line



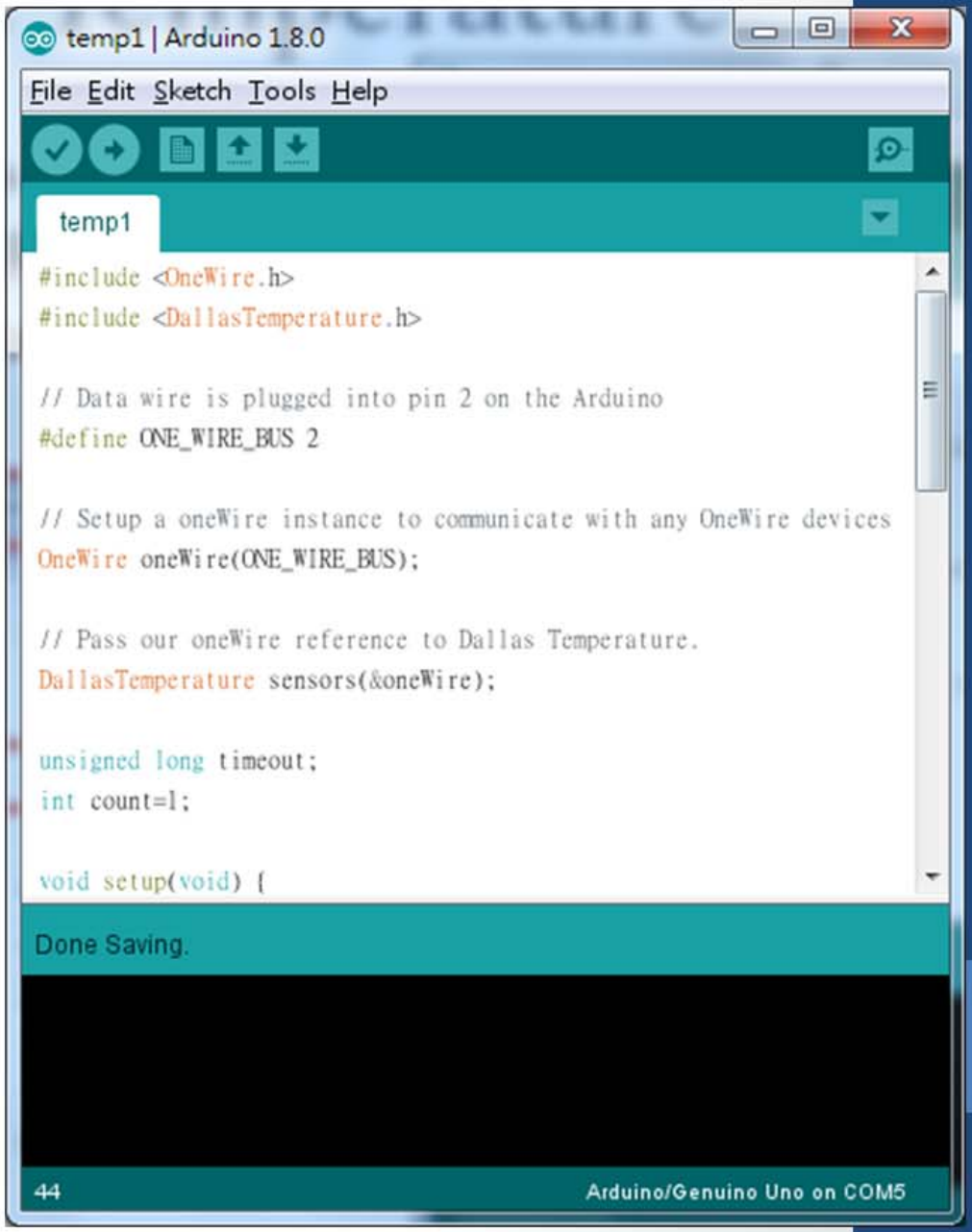
Example: Temperature

- Include 2 Library's
  - #include <OneWire.h>
  - #include <DallasTemperature.h>



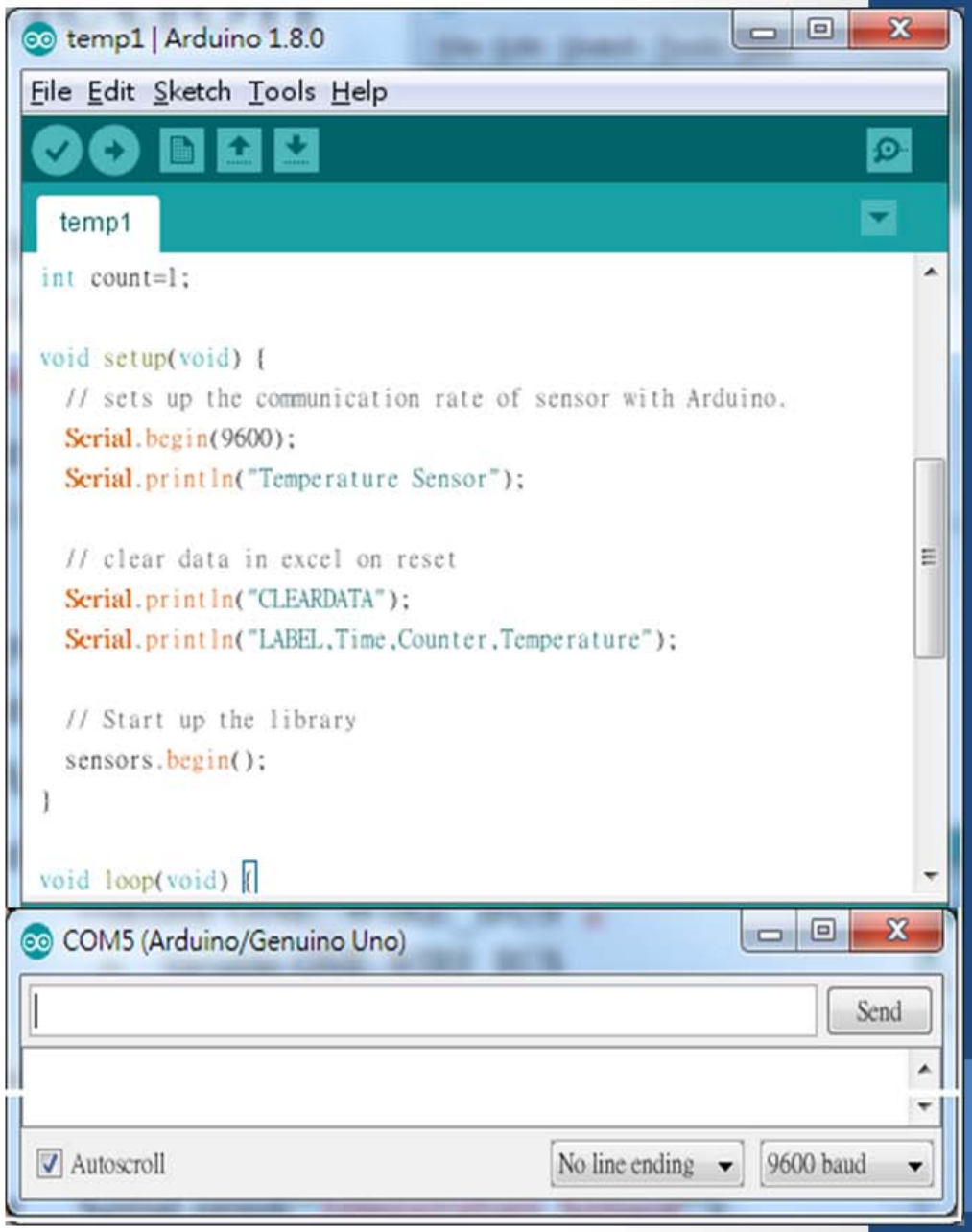
Example: Temperature

- Data wire is plugged into pin 2
  - #define ONE_WIRE_BUS 2



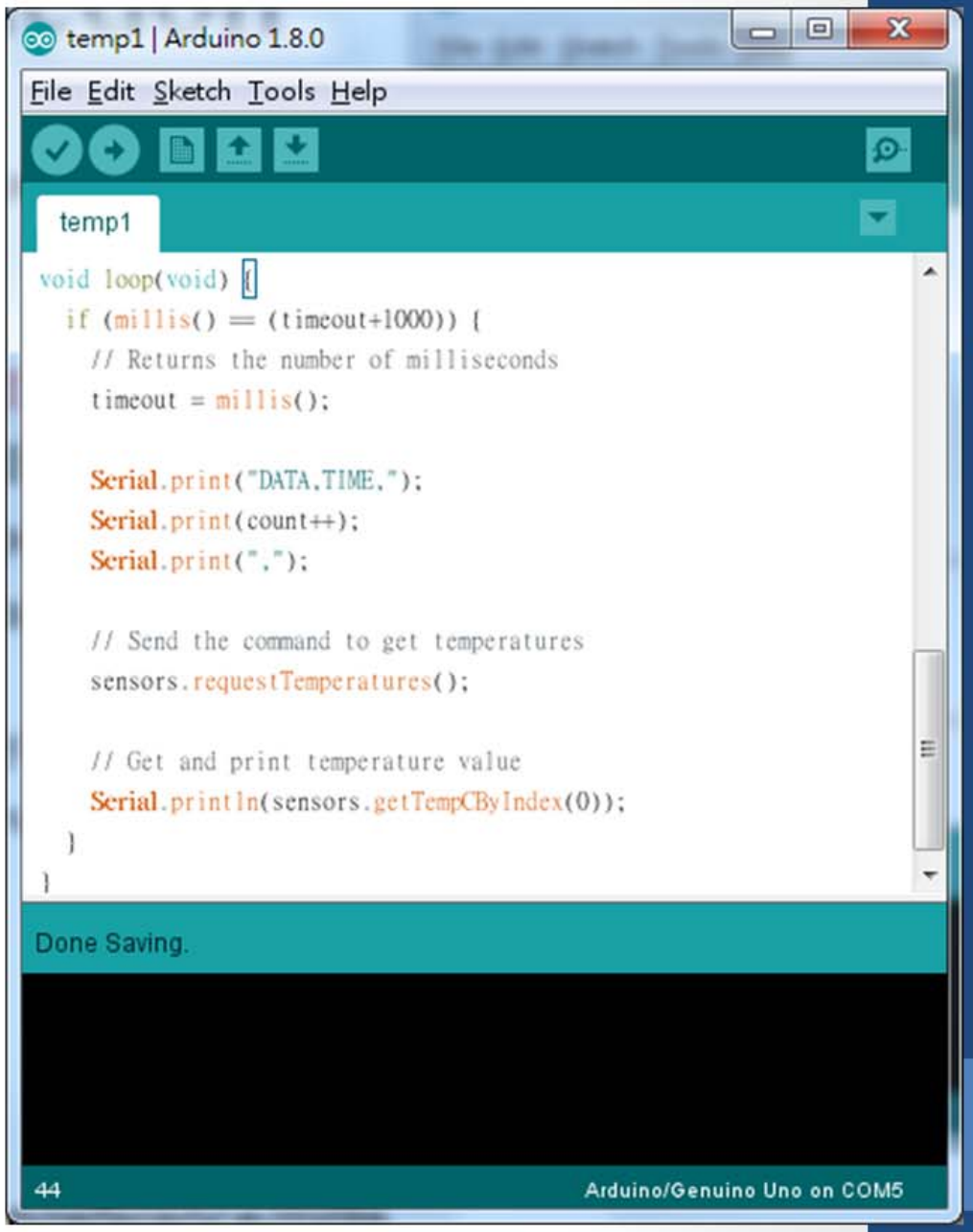
Setup Function

- Serial.begin(9600);
  - sets up the communication rate of sensor with Arduino.
- Serial.println("Temperature Sensor ");
  - Print the sentence "Temperature Sensor" and then start with a new line
  - Any text can be put between the opening and closing quotes (" ")
- Serial.println("CLEARDATA");
  - clear data in Excel plug-ins (PLX-DAQ) on reset



Loop Function

- if (millis() == (timeout + 1000))
  - Sampling rate at 1000 milliseconds (1 sec)
- Serial.print("DATA,TIME,");
  - Print DATA,TIME at the same line
- sensors.requestTemperatures();
  - Send the command to get temperatures
- sensors.getTempCByIndex(0)
  - Get temperature value



https://www.arduino.cc/en/reference/millis