

# DFRobot Examples Programming standard

From Robot Wiki

---

## Contents

- 1 Example with SHT1x Library
    - 1.1 Source Code
  - 2 Example of SHARP GP2Y0A41SK0F IR ranger sensor
    - 2.1 Source Code
  - 3 Example of CO2 Sensor
    - 3.1 Source Code
  - 4 Example of CO2 Sensor with smooth algorithm
    - 4.1 Source Code
- 

## Example with SHT1x Library

ReadSht1x\_Test.ino

8/31/14, 10:34 PM

```

/*****
Digital Temperature and Humidity sensor (With Stainless Steel Probe)
<http://www.dfrobot.com/index.php?route=product/product&path=48&product_id=
912#.UniMMJH7k8M>
*****/
This example reads temperature and humidity from SHT1x Humidity and
Temperature Sensor.
Created 2014-8-28
By Angelo qiao <Angelo.qiao@dfrobot.com>
Modified 2014-8-29
By Angelo qiao Angelo.qiao@dfrobot.com>

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.Connection and Diagram can be found here
<http://www.dfrobot.com/wiki/index.php/
Digital_Temperature_and_Humidity_sensor_(With_Stainless_Steel_Probe)_SKU:
SEN0148#Connecting_diagram>
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
3.SHT1x library is created by jonoxer.
See <https://github.com/practicalarduino/SHT1x> for details.
*****/

#include "SHT1x.h"
const int SHT1xDataPin = 10;
const int SHT1xClockPin = 11;

SHT1x sht1x(SHT1xDataPin, SHT1xClockPin);

void setup()
{
  Serial.begin(9600);
  Serial.println("Starting up...");
}

void loop()
{
  int temperatureC=sht1x.readTemperatureC(); //store temperature in Centigrade
  int temperatureF=sht1x.readTemperatureF(); //store temperature in Fahrenheit
  int humidity=sht1x.readHumidity(); //store humidity

  Serial.print("Temperature: ");
  Serial.print(temperatureC);
  Serial.print("C / ");
  Serial.print(temperatureF);
  Serial.print("F. Humidity: ");
  Serial.print(humidity);
  Serial.println("%");

  delay(2000);
}

```

**Name of the product**

**Product web page**

**Description**

**Create Date**

**Name and email**

**Modify Date**

**License**

**Put the important notice here including pin connection Arduino platform, and the Library included. Or some Trouble shooting or warning.**

**Avoid use DataPin. That may conflict from Other Sensor's pin name. Use SHT1xDataPin instead.**

**Use const instead of #define**

**Use camel case, no underscore.**

**If the value is const, use uppercase on the first letter.**

**Use low case of the Class Name. This rule is Recommended**

**Put the important variables at the beginning of the loop.**

**Don't forget put the comment here**

**Use variables instead of functions here.**

**Print block should be tidy and easy to read**

Page 1 of 1

## Source Code

```

?
1 /*****
2 Digital Temperature and Humidity sensor (With Stainless Steel Probe)
3 <http://www.dfrobot.com/index.php?route=product/product&path=48&product_id=912#.UniMMJH7k8M>
4
5 *****/
6 This example reads temperature and humidity from SHT1x Humidity and Temperature Sensor.
7
8 Created 2014-8-28
9 By Angelo qiao <Angelo.qiao@dfrobot.com>
10 Modified 2014-8-29
11 By Angelo qiao Angelo.qiao@dfrobot.com>

```

```

12
13 GNU Lesser General Public License.
14 See <http://www.gnu.org/licenses/> for details.
15 All above must be included in any redistribution
16 *****/
17
18/*****Notice and Trouble shooting*****/
19 1.Connection and Diagram can be found here
20 <http://www.dfrobot.com/wiki/index.php/Digital_Temperature_and_Humidity_sensor_(With_Stainless_Steel_Probe)_(SKU:SEN0148)#Connecting_diagram
21 2.This code is tested on Arduino Uno, Leonardo, Mega boards.
22 3.SHT1x library is created by jonoxer.
23 See <https://github.com/practicalarduino/SHT1x> for details.
24 *****/
25
26#include "SHT1x.h"
27
28const int SHT1xDataPin = 10;
29const int SHT1xClockPin = 11;
30
31SHT1x sht1x(SHT1xDataPin, SHT1xClockPin);
32
33void setup()
34{
35  Serial.begin(9600);
36  Serial.println("Starting up...");
37}
38
39void loop()
40{
41  int temperatureC=sht1x.readTemperatureC(); //store temperature in Centigrade
42  int temperatureF=sht1x.readTemperatureF(); //store temperature in Fahrenheit
43  int humidity=sht1x.readHumidity(); //store humidity
44
45  Serial.print("Temperature: ");
46  Serial.print(temperatureC);
47  Serial.print("C / ");
48  Serial.print(temperatureF);
49  Serial.print("F. Humidity: ");
50  Serial.print(humidity);
51  Serial.println("%");
52
53  delay(2000);
54}

```

## Example of SHARP GP2Y0A41SK0F IR ranger sensor

GP2Y0A41SK0F\_Test.ino

8/31/14, 11:17 PM

```

/*****
  SHARP GP2Y0A41SK0F IR ranger sensor (4-30cm)
  <http://www.dfrobot.com/wiki/index.php/SHARP\_GP2Y0A41SK0F\_IR\_ranger\_sensor\_\(4-30cm\)\_SKU:SEN0143>

  *****/

This example reads distance value from SHARP GP2Y0A41SK0F

Created 2014-8-28
By Angelo qiao <Angelo.qiao@dfrobot.com>
Modified 2014-8-29
By Angelo qiao Angelo.qiao@dfrobot.com>

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.The distance of the IR ranger sensor is 4-30cm
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
3.The DFPlainprotocol is included and you can see this page for details:
<http://www.dfrobot.com/wiki>
*****/

#include "Arduino.h"

//PIN definition
const int DistancePin = A1;

//Functions
float readDistance(); //read distance from SHARP GP2Y0A41SK0F

void setup()
{
  Serial.begin(9600);
  Serial.println("Start...");
}

void loop()
{
  float distance = readDistance(); //Store the distance value

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println("cm");

  delay(1000);
}

float readDistance()
{
  int sensorValue = analogRead(DistancePin);
  sensorValue = constrain(sensorValue, 80, 530);

  float distanceValue = 2076.0/(sensorValue-11.0);

  return distanceValue;
}

```

**This one naming method of Pin**

**Use A1 instead of 1.  
Use A1 instead of A0,  
as button array on Romeo use A0**

**Only the functions used in setup()  
or loop() can be put in this block.  
Remove this block is acceptable but  
not recommended, as you can write  
some comments to tell how to use.**

**Use full and everyday words.  
It is tough to name a function.  
But try your best! And use Google  
Translation or YouDao Dict for help.**

**Use "sensorValue" buffering the analogRead() data  
Use "sensorState" buffering the digitalRead() data**

**Change the sensorValue into  
Distance through this function**

**There is no voltage data or analogRead() data output.  
As this example is only for beginners.  
For advanced users, they can modify it by themselves.**

Page 1 of 1

## Source Code

```

?
1 /*****
2   SHARP GP2Y0A41SK0F IR ranger sensor (4-30cm)
3   <http://www.dfrobot.com/wiki/index.php/SHARP\_GP2Y0A41SK0F\_IR\_ranger\_sensor\_\(4-30cm\)\_SKU:SEN0143>
4
5   *****/
6   This example reads distance value from SHARP GP2Y0A41SK0F
7
8   Created 2014-8-28
9   By Angelo qiao <Angelo.qiao@dfrobot.com>
10  Modified 2014-8-29
11  By Angelo qiao Angelo.qiao@dfrobot.com>

```

```

12
13 GNU Lesser General Public License.
14 See <http://www.gnu.org/licenses/> for details.
15 All above must be included in any redistribution
16 *****/
17
18/*****Notice and Trouble shooting*****/
19 1.The distance of the IR ranger sensor is 4-30cm
20 2.This code is tested on Arduino Uno, Leonardo, Mega boards.
21 3.The DFPlainprotocol is included and you can see this page for details:
22 <http://www.dfrobot.com/wiki>
23 *****/
24
25#include "Arduino.h"
26
27//PIN definition
28const int DistancePin = A1;
29
30//Functions
31float readDistance(); //read distance from SHARP GP2Y0A41SK0F
32
33void setup()
34{
35  Serial.begin(9600);
36  Serial.println("Start...");
37}
38
39void loop()
40{
41  float distance=readDistance(); //Store the distance value
42
43  Serial.print("Distance: ");
44  Serial.print(distance);
45  Serial.println("cm");
46
47  delay(1000);
48}
49
50float readDistance()
51{
52  int sensorValue = analogRead(DistancePin);
53  sensorValue = constrain(sensorValue, 80, 530);
54
55  float distanceValue = 2076.0/(sensorValue-11.0);
56
57  return distanceValue;
58}

```

## Example of CO2 Sensor

ReadMG811\_Test.ino

8/31/14, 11:50 PM

```

/*****
C02 Sensor (Arduino compatible)
<http://www.dfrobot.com/index.php?route=product/product&product\_id=1023&search=co2&description=true>

*****
This example read and caculate the C02 percentage from MG-811 Gas Sensor

Created 2014-8-28
By Angelo qiao <Angelo.qiao@dfrobot.com>
Modified 2014-8-29
By Angelo qiao Angelo.qiao@dfrobot.com>

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.The percentage of the C02(carbon dioxide) is only for reference.
You need to calibrate it in industrial implement.
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
3.Through exponential curve fitting based on the data from datasheet,
we get the function,which calculates the percentage of the C02 from voltage.
*****/

#include "Arduino.h"

//PIN definition
const float PercentageC02Pin = A0;

//Functions
float readPercentageC02(); //read percentage of the C02 from MG811

void setup()
{
  Serial.begin(9600);
  Serial.println("Starting...");
}

void loop()
{
  float percentageC02 = readPercentageC02();

  Serial.print("C02: ");
  Serial.print(percentageC02);
  Serial.println("PPM");

  delay(1000);
}

//Functions implement
//Co2=2.718281828459045*e^(-0.0558861525*sensorVoltage)
float readPercentageC02()
{
  const float VotageGain = 8.5;

  int sensorValue = analogRead(PercentageC02Pin);
  float sensorVoltage = sensorValue/1024.0*5.0;

```

Although the using of A0 is not recommended, considering the connection SVG picture has finished, don't change it until you have time to change the connection picture.

A proper Naming method.  
for CH4 use readPercentageCH4();  
for H2 use readPercentageH2();

This const variable is derived from the hardware circus. In order to keep the codes before setup() as simple as possible, move the variables into the function or move it down. Also it can avoid user changing it by accident.

Page 1 of 2

ReadMG811\_Test.ino

8/31/14, 11:50 PM

```

    sensorVoltage = sensorVoltage*1000.0/VotageGain;
    // Serial.println(sensorVoltage);
    float percentageC02Value = 25784989641.0468*pow(2.718281828459045,
    -0.0558861525*sensorVoltage);
    return percentageC02Value;
}

```

Convert the sensorVoltage into percentage through this function

Page 2 of 2

## Source Code

```

?
1 /*****
2 C02 Sensor (Arduino compatible)
3 <http://www.dfrobot.com/index.php?route=product/product&product_id=1023&search=co2&description=true>
4
5 *****/
6 This example read and caculate the C02 percentage from MG-811 Gas Sensor
7
8 Created 2014-8-28
9 By Angelo qiao <Angelo.qiao@dfrobot.com>
10 Modified 2014-8-29
11 By Angelo qiao Angelo.qiao@dfrobot.com>
12
13 GNU Lesser General Public License.

```



```

14 See <http://www.gnu.org/licenses/> for details.
15 All above must be included in any redistribution
16 *****/
17
18/*****Notice and Trouble shooting*****/
19 1.The percentage of the CO2(carbon dioxide) is only for reference.
20 You need to calibrate it in industrial implement.
21 2.This code is tested on Arduino Uno, Leonardo, Mega boards.
22 3.Through exponential curve fitting based on the data from datasheet,
23 we get the function,which calculates the percentage of the CO2 from voltage.
24 *****/
25
26#include "Arduino.h"
27
28//PIN definition
29const float PercentageCO2Pin = A0;
30
31//Functions
32float readPercentageCO2(); //read percentage of the CO2 from MG811
33
34void setup()
35{
36   Serial.begin(9600);
37   Serial.println("Starting...");
38}
39
40
41void loop()
42{
43   float percentageCO2 = readPercentageCO2();
44
45   Serial.print("CO2: ");
46   Serial.print(percentageCO2);
47   Serial.println("PPM");
48
49   delay(1000);
50}
51
52//Functions implement
53//Co2=2.718281828459045*e^(-0.0558861525*sensorVoltage)
54float readPercentageCO2()
55{
56   const float VotageGain = 8.5;
57
58   int sensorValue = analogRead(CO2Pin);
59   float sensorVoltage = sensorValue/1024.0*5.0;
60   sensorVoltage = sensorVoltage*1000.0/VotageGain;
61//   Serial.println(sensorVoltage);
62   float percentageCO2Value = 25784989641.0468*pow(2.718281828459045, -0.0558861525*sensorVoltage);
63   return percentageCO2Value;
64}

```

## Example of CO2 Sensor with smooth algorithm



ReadMG811\_Smooth.ino

9/1/14, 12:15 AM

```

/*****
C02 Sensor (Arduino compatible)
<http://www.dfrobot.com/index.php?route=product/product&product\_id=1023&search=co2&description=true>

*****
This example read and caculate the C02 percentage from MG-811 Gas Sensor

Created 2014-8-28
By Angelo qiao <Angelo.qiao@dfrobot.com>
Modified 2014-8-29
By Angelo qiao Angelo.qiao@dfrobot.com>

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.The percentage of the C02(carbon dioxide) is only for reference.
You need to calibrate it in industrial implement.
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
3.Through exponential curve fitting based on the data from datasheet,
we get the function,which calculates the percentage of the C02 from voltage.
4.The range of the MG811 is 400-10000ppm
*****/

#include "Arduino.h"

//PIN definition
const float PercentageC02Pin = A0;

//Functions
float readPercentageC02(); //read percentage of the C02 from MG811

void setup()
{
  Serial.begin(9600);
  Serial.println("Starting...");
}

void loop()
{
  float percentageC02 = readPercentageC02();

  Serial.print("C02: ");
  if (percentageC02 == 400.0) {
    Serial.print("<=400");
  }
  else if (percentageC02 == 10000.0){
    Serial.print(">=10000");
  }
  else{
    Serial.print(percentageC02);
  }
  Serial.println("PPM");

  delay(1000);
}

```

The range of the MG811 is 400-10000PPM.  
 So some constrains should be done to meet the need.

Page 1 of 2

ReadMG811\_Smooth.ino

9/1/14, 12:15 AM

```

//Functions implement
//Co2=2.718281828459045*e^(-0.0558861525*sensorVoltage)
float readPercentageCO2()
{
  #define ANALOG_SAMPLING_TIMES (50) //sampling time in one reading process
  #define ANALOG_SAMPLING_INTERVAL (5) //sampling interval of the two samples

  const float VoltageGain = 8.5; //Voltage Gain of the circuit
  long sensorSumValue = 0; //Don't forget to add brackets.
  //This is one of the reasons why we prefer const to #define.

  for (int samplingIndex; samplingIndex<ANALOG_SAMPLING_TIMES; samplingIndex++)
  {
    sensorSumValue += analogRead(PercentageCO2Pin);
    delay(ANALOG_SAMPLING_INTERVAL);
  }

  float sensorVoltage = sensorSumValue*5.0/(1024.0*ANALOG_SAMPLING_TIMES);

  sensorVoltage = sensorVoltage*1000.0/VoltageGain;
  // Serial.println(sensorVoltage);
  float percentageCO2Value = 25784989641.0468*pow(2.718281828459045,
    -0.0558861525*sensorVoltage);
  percentageCO2Value = constrain(percentageCO2Value, 400, 10000);
  return percentageCO2Value;
}

```

Use Uppercase and underscore in #define.

Don't forget to add brackets.

This is one of the reasons why we prefer const to #define.

Use full name here

This function averages the analogRead() data to get a more steady result, but it increases the CPU timing. It costs about 250ms to run.

Page 2 of 2

## Source Code

```

?
1 /*****
2 CO2 Sensor (Arduino compatible)
3 <http://www.dfrobot.com/index.php?route=product/product&product_id=1023&search=co2&description=true>
4
5 *****/
6 This example read and caculate the CO2 percentage from MG-811 Gas Sensor
7
8 Created 2014-8-28
9 By Angelo qiao <Angelo.qiao@dfrobot.com>
10 Modified 2014-8-29
11 By Angelo qiao Angelo.qiao@dfrobot.com>

```

```

12
13 GNU Lesser General Public License.
14 See <http://www.gnu.org/licenses/> for details.
15 All above must be included in any redistribution
16 *****/
17
18/*****Notice and Trouble shooting*****/
19 1.The percentage of the CO2(carbon dioxide) is only for reference.
20 You need to calibrate it in industrial implement.
21 2.This code is tested on Arduino Uno, Leonardo, Mega boards.
22 3.Through exponential curve fitting based on the data from datasheet,
23 we get the function,which calculates the percentage of the CO2 from voltage.
24 4.The range of the MG811 is 400-10000ppm
25 *****/
26
27#include "Arduino.h"
28
29//PIN definition
30const float PercentageCO2Pin = A0;
31
32//Functions
33float readPercentageCO2(); //read percentage of the CO2 from MG811
34
35void setup()
36{
37  Serial.begin(9600);
38  Serial.println("Starting...");
39}
40
41void loop()
42{
43  float percentageCO2 = readPercentageCO2();
44
45  Serial.print("CO2: ");
46  if (percentageCO2 == 400.0) {
47    Serial.print("<=400");
48  }
49  else if (percentageCO2 == 10000.0){
50    Serial.print(">=10000");
51  }
52  else{
53    Serial.print(percentageCO2);
54  }
55  Serial.println("PPM");
56
57  delay(1000);
58}
59
60//Functions implement
61//Co2=2.718281828459045*e^(-0.0558861525*sensorVoltage)
62float readPercentageCO2()
63{
64#define ANALOG_SAMPLING_TIMES  (50)    //sampling time in one reading process
65#define ANALOG_SAMPLING_INTERVEL (5)    //sampling interval of the two samples
66
67  const float VoltageGain = 8.5;        //Voltage Gain of the circus
68
69  long sensorSumValue = 0;
70
71  for (int samplingIndex; samplingIndex<ANALOG_SAMPLING_TIMES; samplingIndex++) {
72    sensorSumValue += analogRead(PercentageCO2Pin);
73    delay(ANALOG_SAMPLING_INTERVEL);
74  }
75
76  float sensorVoltage = sensorSumValue*5.0/(1024.0*ANALOG_SAMPLING_TIMES);
77
78  sensorVoltage = sensorVoltage*1000.0/VoltageGain;
79//  Serial.println(sensorVoltage);
80  float percentageCO2Value = 25784989641.0468*pow(2.718281828459045, -0.0558861525*sensorVoltage);
81  percentageCO2Value = constrain(percentageCO2Value, 400, 10000);
82  return percentageCO2Value;
83}

```

Retrieved from "http://www.dfrobot.com/wiki/index.php?title=DFRobot\_Examples\_Programming\_standard&oldid=27140"

- This page was last modified on 1 September 2014, at 00:34.
- This page has been accessed 111 times.