UNIT 8: OTHER SENSORS

AIMS

There are a huge number of physical phenomena in nature such as speed, temperature, visible and invisible light, humidity, acceleration, weight and many others. These days there's a large range of sensors on the market capable of detecting and measuring these phenomena. They provide data that can be treated and processed by any kind of controller like Arduino for example.

We can't study and work with all of them: we'd never get through them all and we'd certainly go over budget. So we're going to work with two sensors in this unit; they're capable of measuring temperature and humidity. They represent a realistic sample of what's available on the market and we've chosen them because they're simple and accessible. We're also going to explain the fundamentals of acceleration sensors.

You'll be doing a series of practical exercises with them and I hope they'll give you some basic ideas and help you develop your own projects and applications.

THEORY SECTION

- ACCELEROMETER
- TYPES OF ACCELEROMETERS
  - Mechanical accelerometers
  - Piezoelectric accelerometers
  - Thermal accelerometers
  - Capacitive accelerometers
  - Micro electro-mechanical systems (MEMS)
- THE NTC TEMPERATURE SENSOR
  - Basic circuit
- HUMIDITY SENSORS
  - Mechanical sensors
  - Condensation sensors
  - Infrared sensors

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

PRACTICE MATERIALS

-Lap top or desk top computer

-Arduino IDE work environment; this should include the supplementary material already installed and configured.

-Arduino UNO controller board

-A USB cable

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

TABLE OF CONTENTS

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information therein."

THEORY SECTION

1. ACCELEROMETER

Accelerometers are devices that measure acceleration or, in other words, the rate of change in the speed of an object. It measures this in metres per second squared (m/s^2) or in terms of standard gravity (g). The force of gravity on Earth is equivalent to 9.8 m/s2 or 1g although this can vary slightly with altitude. The force of gravity on each planet is different.

As usual, we're going to avoid complex physics principles and mathematical calculations. It's not really difficult to understand the basic working principles of an accelerometer. Imagine a hollow cylinder with a ball, the mass, inside (Figure 1). The ball hangs from a spring and can move up or down inside the cylinder. If you move the cylinder up or down you create a force that displaces the ball. The distance of the displacement is proportional to the force applied which in turn is proportional to the acceleration of the cylinder when you moved it upwards.

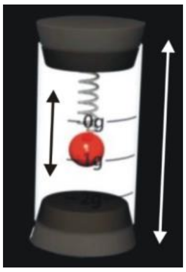


Figure 1

Now imagine you've got three identical cylinders positioned just like in the Figure 2. The ball inside each one of them will move according to the direction the cylinder is moved in: up or down, to the right or to the left, forwards or backwards. You can therefore measure the acceleration for each one of the axes X, Y and Z. This is known as a "three axis accelerometer".

Comentado [X1]:

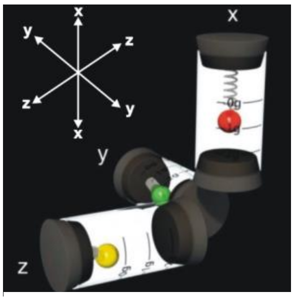


Figure 2

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

Acceleration is defined as the rate of change in the speed of an object compared to the time this takes. In other words, it measures how quickly an object changes speed.

Acceleration is a vector. In other words, it shows what the acceleration is and in which direction. Accelerometers can therefore detect movements, vibrations and impacts. The airbags in cars are a good example of their use. They can also detect direction and position. I'm sure you've noticed that when you turn your Smartphone, Tablet or e-book and you put it in a horizontal or vertical position the screen adjusts automatically.

2. TYPES OF ACCELEROMETERS

There are a number of different types of accelerometer sensors; the differences lie in their design and practical application. Here's a brief summary:

A. MECHANICAL ACCELEROMETERS

These accelerometers use an inert mass (similar to the ball we talked about before) and some elastic springs. The changes are measuring with strain gauges and include suspension systems that prevent excessive movement. The gauge is made of copper or nickel and platinum or silicon alloys that create a piezoresistive effect. In other words, when the gauge warps its resistance changes; this causes the intensity that goes through it to change too and also the voltage. Remember, I=V/R

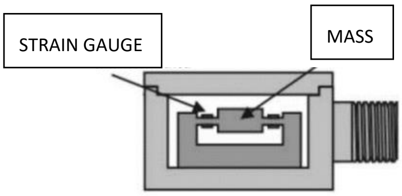


Figure 3

When the sensor moves the mass moves too and "warps" the gauge; this causes variations in the resistance, intensity and voltage. These variations are directly proportionate to the acceleration of the movement applied to the sensor.

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

B. PIEZOELECTRIC ACCELEROMETERS

These accelerometers are based on the piezoelectrical effect: the physical distortion of a material modifies its crystal structure thus changing its electrical properties.

These accelerometers are made of piezoelectrical crystals (like quartz) which produce an electrical charge when a force or "pressure" that modifies their structure is applied to them.

A movement from the sensor makes the inertial mass put pressure on the crystal which changes its structure. This generates an electrical charge proportionate to the pressure applied which in turn is proportionate to the acceleration or the strength of the movement.

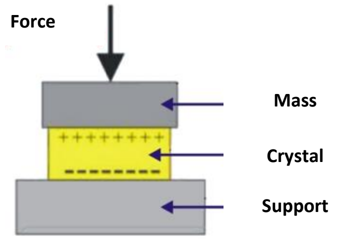


Figure 4

C. THERMAL ACCELEROMETERS

These accelerometers detect and measure heat. They consist of a receptacle with a small heater (a resistor) that warms a gas bubble; there are also a number of temperature sensors or thermal batteries. The Figure 5 shows an accelerometer with two axes and four temperature sensors.

When the sensor is totally level the hot gas bubble remains right in the middle of the receptacle. The four temperature sensors or thermal batteries show the same thermal value.

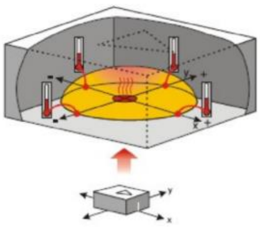


Figure 5

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

When one of the sensor's axes moves or turns, the hot gas bubble is displaced within the casing; this causes a rise in temperature in some of the thermal batteries and a fall in others. By comparing these temperatures, we obtain a result proportional to the acceleration of the movement.

D. CAPACITIVE ACCELEROMETERS

In general terms, it consists of two metal plates separated by an insulating or dielectric material such as paper, a ceramic or air. Its capacity depends on the size of these plates and/or the distance between them: the bigger they are and/or the closer together, the greater their capacity. They must never actually touch though.

These are the basic principles of capacitive accelerometers. They modify the relative position of the plates in the condenser when it's subjected to acceleration. The parallel movement of one of the condenser's plates causes its capacity to vary.

Two elastic springs keep the inertial mass in place. The moveable plate in the condenser is joined to the inertial mass. When then accelerometer is at rest the moveable plate remains right in the middle of the two fixed plates. The C1 and C2 capacitors are identical as the dielectric material or, in this case, the air that separates the moving plate from the fixed ones is the same.

When a force is applied to the sensor from the left, the mass moves and displaces the moveable plate (shown in red) to the right. The distance between the mass and the fixed plate (shown in green) on the right is now less than the distance between the mass and the fixed plate on the left. C1's capacity is less than C2's. The difference between the two capacitors generates an electric signal that is proportionate to the acceleration.

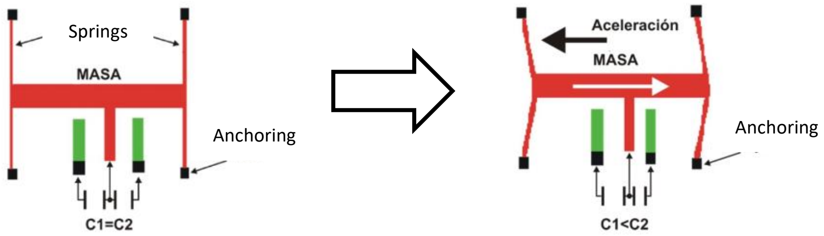


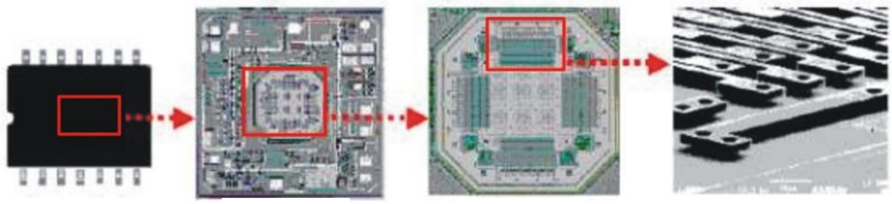
Figure 6

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

E. MICRO ELECTRO-MECHANICAL SYSTEMS (MEMS)

Technological advances have enabled us to produce microscopic mechanical devices known as “micro electro mechanical systems” or just plain MEMS. You may have heard expressions such as “micro machines”, “nanotechnology”, “nanomachines” etc.

Okay then, this technology is used to manufacture accelerometers, especially capacitive ones. The illustration may give you an idea of just what we’re talking about: a chip that contains a capacitive micro mechanized accelerometer. On the right, there are two plates of about a millionth of a metre; the distance between them creates the condenser that changes according to the acceleration.



Casing	Chip	Sensor	Plates
<1 cm^2	2 mm^2	400 μm^2	1 µm

3. THE NTC TEMPERATURE SENSOR

You’ve already used a temperature sensor, the LM35. Bear in mind that there are several types of sensors; some are more accurate than others and there are various ways of using them. One of these is known as the “thermistor” or negative temperature co-efficient resistor, commonly abbreviated to “NTC”. We’re going to have a look at these sensors now because they’re very easy to use and, most importantly, they’re very economical.

The resistance of NTC thermistors *decreases* as the temperature rises. In other words, the hotter they get, the lower their level in ohms and vice versa. They’re made of semiconducting materials such as ferric oxide, nickel oxide or cobalt oxide. You can see their electronic symbol Figure 7:

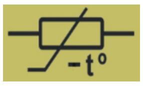


Figure 7

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

There's a graph in the Figure 8 showing how an NTC resistor works. Of course each manufacturer provides graphs, percentages and tables with exact values with their product. As you can see from the figure, when T, the temperature, rises, the value of R, the resistance, falls.

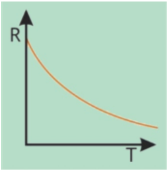


Figure 8

A. BASIC CIRCUIT

The basic circuit for using an NTC as a temperature sensor is both economical and straightforward. The most basic one consists of a voltage divider with a fixed resistor connected in series with the NTC resistor; we apply a voltage of 5 V. Have a look at the diagram Figure 9.

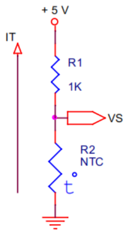


Figure 9

We calculate the TI, or total intensity, that travels round the circuit using Ohm's Law:

$$I_T = \frac{V}{RT} = \frac{V}{R1 + R2} = \frac{5}{1000 + NTC}$$

If we want to know the output voltage at the extremes of the R2 resistor, or NTC, then we use the following formula:

$$V_S = R2 \cdot I_T;\ V_S = R2 \cdot \frac{V}{R1 + R2}$$

As R2 is the NTC resistance and this varies with the temperature, then there's an output voltage, or OV, that also varies with the temperature according to the preceding equation. To sum up, the analogue voltage depends on the temperature. You can of course measure this voltage using the Arduino analogue converter.

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

Just so you can get an idea of what's going on, suppose there's an NTC resistance that corresponds to the graph shown in the Figure 10. You can complete the table below by looking at the diagram of the basic circuit and the preceding equations. The Arduino ADC converter always uses the reference voltage, 5 V, so the resolution will be 0.00488 V (5 / 1023).

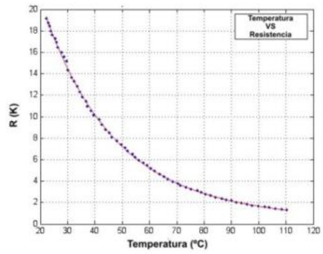


Figure 10

TEMPERATURE	25º	40º	55º	70º	80º	90º	100º	110º
R2-NTC (Ω)	18KΩ	10KΩ	6KΩ	4KΩ	3KΩ	2KΩ	1K7Ω	1K5Ω
OV	4.73V	4.54V	4.28V	4V	3.75V	3.33V	3.14V	3V
ADC OUTPUT	969	930	877	819	768	682	643	614

4. HUMIDITY SENSORS

Humidity is a natural molecular phenomenon. It is essentially related to the number of water molecules present in a solid or gas (e.g. the air).

The amount of moisture in the atmosphere doesn't affect the daily lives of human beings all that much but measuring and monitoring it can contribute to our well being and comfort. It can however have an important effect on other chemical, physical and biological processes:

- food preservation
- soil humidity
- the humidity of building materials
- and many other areas…

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

Sensors called “hygrometers” provide us with information on humidity. We can find out the relative humidity (RH) in the atmosphere, for example. Relative humidity is the amount of water vapour in the air. For instance, 50% RH at 23ºC means that the air contains 50% of the maximum humidity possible at that temperature. 100% RH means that the air is at maximum saturation point. When it comes into contact with cooler air or a cooler surface the water vapour turns into drops of water. This is called “*dew point*”.

There are various types of humidity sensors. The differences lie in the way they’re manufactured and the way they’re used. Let’s have a look at a few of them and get an idea.

A. MECHANICAL SENSORS

These sensors rely on the changes in size that certain materials experience when exposed to humidity. These materials include some synthetic and organic fibres such as human hair for example. When the RH or relative humidity increases these fibres experience changes in size - they get longer. These changes are measured and controlled by mechanical or electronic means and adjusted in proportion to the relative humidity. Does the friar in the figure ring a bell? You must have seen him some time.

A fibre changes size according to the humidity in the atmosphere. There’s a fibre attached to the monk’s arm which moves its pointer along the graduated scale.

There are industrial hygrometers based on the same idea. These hygrometers have a spring which expands or contracts according to the humidity moving a needle around an appropriately graduated scale.

These expansions and contractions can also affect the size of a “strain gauge” and thus affect its resistance. As you know, if the circuit resistance changes the intensity and the voltage change too. The analogue voltage is therefore proportionate to the humidity.

B. CONDENSATION SENSORS

These sensors are a little bit more complex. Have a look at the Figure 11 to get an idea of what’s going on. The air circulates through a chamber with a mirror inside (2). This air can be warmed or chilled using a cooling (3) or heating (1) system This way the vapour condenses on the mirror or the water evaporates from it.

There’s also a light source (4) directed at the mirror. The mirror reflects the light onto a photo resistor (5a). This same light falls directly on a second photoresistor (5b). So there are two measurements: the 5b phototransistor measures the real luminous intensity and another luminosity measurement which is

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

distorted by the amount of condensation on the mirror (5a). The difference between the two intensities is multiplied and used on the power regulator that controls the heater and measures the humidity.

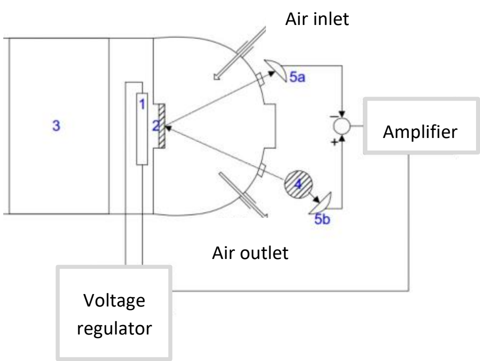
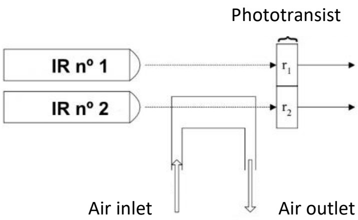


Figure 11

C. INFRARED SENSORS

Do you remember the electromagnetic spectrum? Good, because it turns out that water molecules absorb wavelengths of around 1400 to 1930 nm which corresponds to the infrared region of the electromagnetic spectrum. Here's a diagram of how these types of humidity sensors work based on this idea.

They've got two sources that emit infrared light, IR nº 1 and IR nº 2, and their corresponding receivers, r1 and r2. The light emitted by IR nº 1 goes directly to the r1 phototransistor. This is the reference point. The light emitted by IR nº 2 travels through the air that contains the water vapour that has to be measured. As this vapour absorbs part of the light the r2 phototransistor receives a smaller amount of light. The difference between r1 and r2 makes it possible to calculate the degree of humidity in the air.



"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

D. RESISTIVE SENSORS

These are very economical and easy to use. They consist of two conducting grids with parallel fingers separated from each other and mounted on a smooth, non conducting surface. The number of water molecules is proportionate to the degree of RH. Water is a conductor so the resistance between the two ends of the grid varies in proportion to the amount of humidity. As always we apply Ohm's law: if the resistance varies, the intensity and the voltage in circulation do too.

5. THE DHT11 SENSOR

This device integrates a temperature sensor based on an NTC resistor and a resistive humidity sensor. It was developed by the Chinese company Aosong (www.aosong.com); you'll find the manufacturer's datasheet amongst the supplementary material for this unit.

Whilst it may not be the most accurate sensor on the market, it is very cost effective and convenient to connect up. You can use it for almost any domestic task you can think of.

You should also be aware that the device includes a controller. It reads the analogue intakes of both the NTC temperature sensor and relative humidity or RH resistive sensor. It oversees the calculations, adjustments, calibrations and processes necessary to provide your Arduino with accurate information whenever it receives a request.

This is one smart gadget!

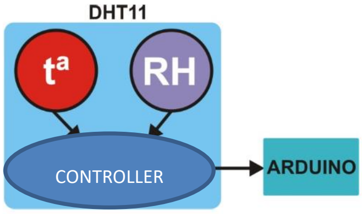


Figure 12

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

A. FEATURES

Here's a summary of the main features:

PARAMETER	HUMIDITY	TEMPERATURE
Resolution	8 bits	8 bits
Repeatability	± 1% RH	± 0.2ºC
Accuracy	±5% to 25ºC	± 1ºC
Range	20% -90% from RH to 25ºC	0ºC – 50 ºC

- ✓ Supply voltage: 3.5 – 5 V
- ✓ Consumption: 0.3 mA during measurement and 60 µA at rest.
- ✓ Sampling period: more than 2 seconds

B. INTRODUCTION

The DHT11 that you're going to use comes installed on a small printed board or module that facilitates its connection. The Microbot company (www.microbot.it) markets it in a module and the reference number is MR003-005.1. This is the one you're going to use for these experiments. It comes in a kit with two strips of three pins: a strip of elbow pins and another of straight pins.

As on prior occasions, you'll be using the strip of straight pins. There's a picture of the device and a simplified diagram below that and the electronic symbol too (Figure 13).

It's only got three pins:

Figure 13

1. GND:

2. +5V: voltage + 5V

3. DATA: Information outlet (relative humidity and temperature)

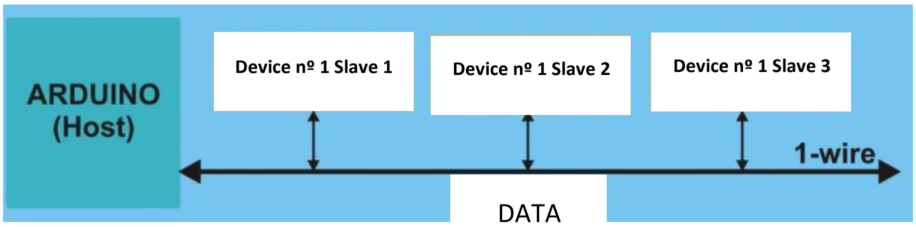
"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

C. 1-WIRE COMMUNICATION

Apart from its convenient connection and cost effectiveness, another sound reason for choosing this device is the communication system it uses to transmit the humidity and temperature measurements to our Arduino controller. You're going to learn something new here.

In both this course and previous one – if you did it – you've already done communication exercises. In particular, you've used what's known as standard serial communication. But there are other forms of serial communication used by lots of other temperature and humidity sensors, our subject of the moment. Now you're going to study and use serial communication known as "1-wire".

Just imagine this: Arduino can communicate with a number of devices like this sensor and others using just one pin. Have a look at the figure



Arduino acts as the main controller or "Host". It initiates and completely controls communication with all of the devices which function as slaves. Communication is conducted using a single wire ("1 Wire") and can be bidirectional: Arduino - device or device - Arduino. Obviously there can only be communication in one direction with a single device at a time.

There's a wide range of 1-wire devices on the market: analogue to digital (ADC) converters, data storage devices, peripherals, a number of sensors and many others. Each one has its own individual features and you should always refer to the documentation supplied by the manufacturer. In this case we're going to try and study how our DHT11 sensor communicates with the Arduino controller using a single cable or pin.

To start off with, when the host controller, Arduino, requests data from the sensor, the latter returns 40 bit (5 bytes) of data through the data pin.

The first byte is an integer that represents the relative humidity as a percentage (%Rh) and the third is the temperature in degrees Celsius. Bytes 2 and 4 return 0x00. They're used by other versions of the sensor with greater resolution to express the fractional part of the humidity and the temperature. Byte 5

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

represents the Checksum. Its value is the sum of the four previous bytes. It’s a way of guaranteeing the reliability of the data transmitted. Here are some examples:

DATA	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Binary	0011 1100	0000 0000	0001 0010	0000 0000	0100 1110
Hexadecimal	0x3C	0x00	0x12	0x00	0x4E
Decimal	60 %RH	0	18 ºC	0	78

DATA	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Binary	0011 0101	0000 0000	0001 1000	0000 0000	0100 1101
Hexadecimal	0x35	0x00	0x18	0x00	0x4D
Decimal	53 %RH	0	24 ºC	0	77

It’s always the host controller, Arduino, that performs the initial sequence whenever it needs data from the sensor. This is how our DHT11 does it:

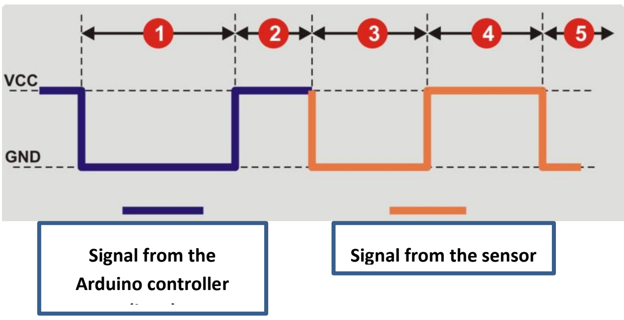


Figure 14

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

1. The Host (Arduino) sets the data pin at level “0” for a minimum of 18 mS.

2. Next it releases the data pin which goes to level “1” and waits on a reply from the DHT11 sensor (between 20 – 40 µS).

3. In reply, the DHT11 sensor sets the same data pin at level “0” and it keeps it there for 80 µS.

4. Next the sensor releases the data pin which goes to level “1” for another 80 µS.

5. From that moment on, the sensor sends the 40 bits of data, beginning with the most significant bit from byte 1, the one that indicates the relative humidity as a percentage (%RH).

After this exchange between the host and the DHT11, the latter starts to transmit the 40 bits of data from its internal humidity and temperature sensors. Remember that it’s got its own controller. It’s time for the host, the Arduino controller, to collect the data.

All the bits begin with a level “0” followed by a level “1” and their duration establishes the bit’s exact value. It’s somewhat like the “*Our*” protocol we used for the reception of bits using infrared radiation. Do you remember?

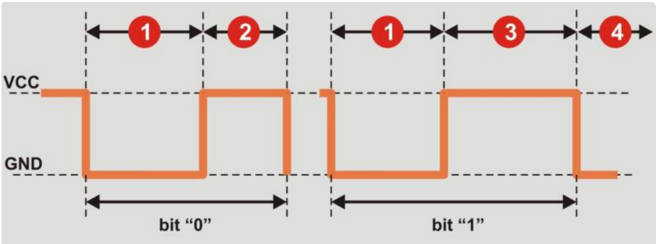


Figure 15

1. A level “0” indicates the beginning of a new bit.

2. This is followed by a level “1” that lasts for about 28 µS and represents a 0 bit.

3. This is followed by a level “1” that lasts for about 70 µS and represents a 1 bit.

4. The start of the following bit.

Last of all, there's the final sequence of the transmission; it's very straight forward.

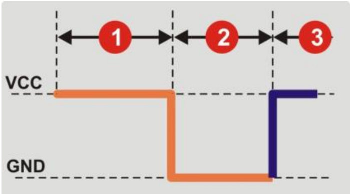


Figure 16

1. The sensor transmits bit nº 40, the final one.

2. The data line goes to level "0" for 50 µS and terminates the transmission.

3. The data pin is released and goes to level "1". The Arduino (Host) regains control of the data pin and performs a new start sequence whenever necessary.

D. THE DHT1 LIBRARY

Just as we did when we designed the "Our" protocol in the previous unit, we could create another library of functions so that our Arduino was able to transmit the start and finish sequences as well as receive the 40 bits provided by the DHT11 device we're using.

But that won't be necessary: there's a link on the website of the manufacturer of our MR003-005.1 module with DHT11 (Microbot) sensor where you can download the "DHT11" library (http://www.microbot.it/sketches/DHT11.zip). This is a library developed by Mirko Prosseda which will facilitate the use of this device. We'd like to take this opportunity to thank him for this contribution.

There are more libraries to be found on the net. You'll come across other options but were familiar with this function and it's easy to use. You can also find it in the supplementary material together with the exercises.

- **<u>The read() Function</u>**

This is the only function in the library. Each time it's executed it performs the entire initial and read sequence of 40 bits that it receives from the DHT11 sensor. It returns an error code and two variables with the humidity and temperature.

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

Syntax:

read(pin);

pin: the data pin that's connected to the DHT11 sensor; the data is received here.

Returns:

0: Correct device read

-1: Checksum error. The data received is not correct.

-2: Timeout. time overrun without response error.

-3: Busy. Error, the data line is engaged (possibly by another device).

humidity: 8 bit integer variable expressing the value of relative humidity as a percentage (%RH)

temperature: variable expressing the value of the temperature in degrees Celsius.

EXAMPLE:

```
byte error;                             //Error variable
#include <dth11.h>                      //Includes the library
dth11 DTH11;                            //Variable used
error=DTH11.read(2);                    //Reads the values of the DHT11 device
Serial.println(error);                  //Prints the error code
Serial.println(DHT11.humidity);         //Prints the humidity level (%RH)
Serial.println(DHT11.temperature);      // Prints the temperature level (ºC)
```

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

PRACTICE SECTION

6. EXAMPLE 1: Measuring temperature and humidity

Let's get stuck into the first exercise on using the DHT11 device contained in the MR003-005.1 module. The idea is to measure the humidity and atmospheric temperature and then display them on an LCD screen.

You can assemble the circuit by following the diagram below Figure 17

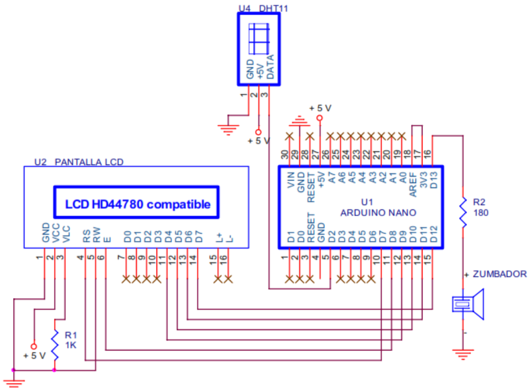


Figure 17

Note that the sensor data pin, number 3, is connected to the D2 Arduino pin, number 5. When assembling the circuit, as on other occasions, start by connecting the wires as they're no longer visible when the module is inserted into the sensor (Figure 18).

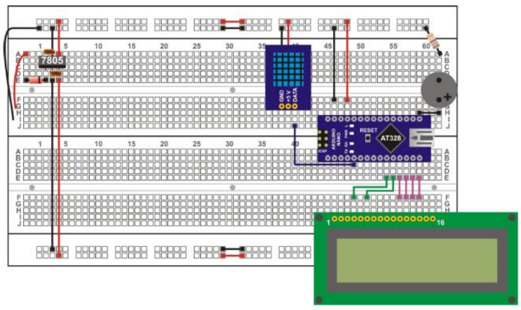


Figure 18

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

Once you've assembled the circuit and checked all the connections, record the program and make sure it works properly. Now you've got a hydrometer and a thermometer.

You can vary the temperature and humidity by warming the sensor with your hand or with a fan or a hair dryer or by breathing on it; try different ways of doing it but be careful not to burn yourself.

If you compare the measurements that appear on the LCD screen with any other instrument that you have at hand, you should bear the margin of error in mind. The DHT11 has a margin of error of ±5% at 25ºC for humidity and ±1 ºC for temperature.

7. EXAMPLE 2: More measurements

You're going to keep measuring humidity and temperature. In this exercise the temperature will be displayed in degrees Celsius (ºC), Fahrenheit (ºF) and Kelvin (ºK). Here's the assembly illustration (Figure 19):

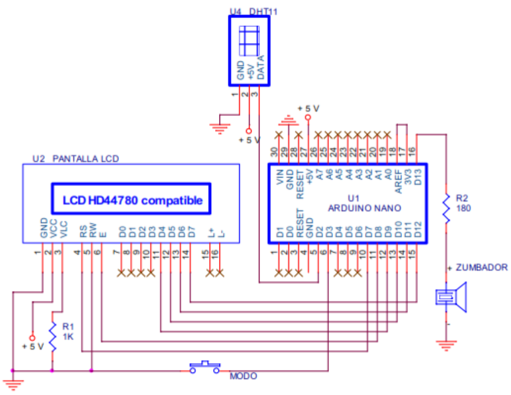


Figure 19

The only thing that's been added is the "MODE" pushbutton that's connected to the input pin D3 which is also the INTI interrupt input.

Have a close look at the program in the exercise. There some things that you may remember from exercises in previous units.

1. The "grade" variable consists of a matrix that defines the graphic nature of the symbol º (grades).

2. We use the lcd.createChar() command to create the degree symbol º.

3. The D3 input pin is configured as an input with a “pull-up” resistor.

4. The same D3 pin is configured as an INTI interrupt input enabled at level “0”. The Inter_1() function is executed every time the interrupt is triggered.

5. Interrupts() trigger the interruptions.

Each time someone presses the pushbutton connected to the D3 pin there’s an interruption and the Inter_1() treatment is executed. The “mode” variable increases; it ranges between 1, 2 and 3. The temperature will be displayed in Cº, Fº or Kº depending on its value at any given moment.

The main program is very simple. It reads the DHT11 sensor to get the humidity and temperature. Both are displayed on the LCD screen. The temperature may be displayed in Cº, Fº or Kº depending on “mode” variable’s value.

MODE	UNIT	DESCRIPTION
0	Cº	Temperature displayed directly in Cº
1	Fº	Temperature displayed in Fº: F = Cº(1.8) + 32
2	Kº	Temperature displayed in Kº: Kº = 273,15 + Cº

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

8. EXAMPLE 3: A simple idea for a weather station

You're going to simulate a simple weather station using the same circuit diagram and assembly illustration as in the previous exercise. As well as measuring the humidity and the temperature the weather station will be capable of recording maximums and minimums.

The first thing it does is take a sample of the current humidity and temperature and records them as maximums and minimums in the variables “Hmax”, “Tmax”, “Hmin” and “Tmin”.

From that time on the main body of the program takes samples from the DHT11 sensor every second. It compares these samples with the corresponding maximums and minimums and, where appropriate, records them.

The pushbutton is connected to the D3 pin. This is configured as an input with a pull-up resistor and an interrupt capacity enabled at level “0”. Each time someone presses the button the treatment program displays the maximums and minimums for two seconds. It then goes back to the main program to continue displaying the current values.

There's nothing very original about this program but it may give you some ideas for more ambitious projects. Now you've got a temperature and humidity sensor and you know how to read it. There are numerous applications for this sensor: everything depends on your imagination and your knowledge of the areas you want to use it in. Here are some examples any way:

- ✓ Well-being
- ✓ Air conditioning
- ✓ Humidifiers
- ✓ Dehumidifiers
- ✓ Ventilation
- ✓ Thermostats
- ✓ Temperature control
- ✓ Weather stations
- ✓ Food preservation
- ✓ Greenhouses
- ✓ Saunas
- ✓ And lots more

There are a million possibilities; you can experiment with them and learn from them and then apply them.

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."

REFERENCES

BOOKS

[1]. **EXPLORING ARDUINO**, Jeremy Blum, Ed.Willey
[2]. Practical ARDUINO, Jonathan Oxer & Hugh Blemings, Ed.Technology in action
[3]. Programing Arduino, Next Steps, Simon Monk
[4]. Sensores y actuadores, Aplicaciones con ARDUINO, Leonel G.Corona Ramirez, Griselda S. Abarca Jiménez, Jesús Mares Carreño, Ed.Patria
[5]. ARDUINO: Curso práctico de formación, Oscar Torrente Artero, Ed.RC libros
[6]. 30 ARDUINO PROJECTS FOR THE EVIL GENIUS, Simon Monk, Ed. TAB
[7]. Beginning C for ARDUINO, Jack Purdum, Ph.D., Ed.Technology in action
[8]. ARDUINO programing notebook, Brian W.Evans

WEB SITES

[1]. https://www.arduino.cc/
[2]. https://www.prometec.net/
[3]. http://blog.bricogeek.com/
[4]. https://aprendiendoarduino.wordpress.com/
[5]. https://www.sparkfun.com

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."