

ÁREA DE TECNOLOGÍA ELECTRÓNICA - URJC

Notebook 3

Notebook: Air Conditioning

Description of the notebook

In this booklet you have, on the one hand, the practice that you must perform and show the PWM teacher.

On the other hand, 2 annexes. One is more theoretical and practical content that helps you understand the control of the PWM. And another second Anex for you to practice using an RGB led, which does not come in the material, but you can simulate it. This last Annex is only to complement your training should not deliver anything.

Stage 1: PWM

**3. PWM DC**

**3.1. Name of the system: Air Conditioning PWM**.

El funcionamiento del PWM se basa en variar el ángulo de giro cada un determinado tiempo de tal manera que en función de este, cambiará de ángulo su posición.

**3.2. DEPLOY - MATERIAL**

Material necessary for the development of Phase 1 and Phase 2:

- Kit of practices.

- The Arduino board.

- Transistor BC547.

- Resistance of 1K or 10K.

- 1.5-3V DC motor.

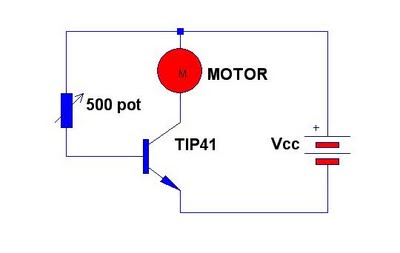
- A temperature sensor like the LM35.

- Insertion cables.

- Protoboard plate.

At the moment in which the program is executed, the temperature sensor starts measuring the ambient temperature. If the temperature is very low, the engine will turn to the minimum level, while if the temperature increases, the speed also does so proportionally to it.

The electronic assembly is the following, where TIP41 is the transistor BC547 and "pot" is the value of the resistance associated with the transistor, in our case 1K:

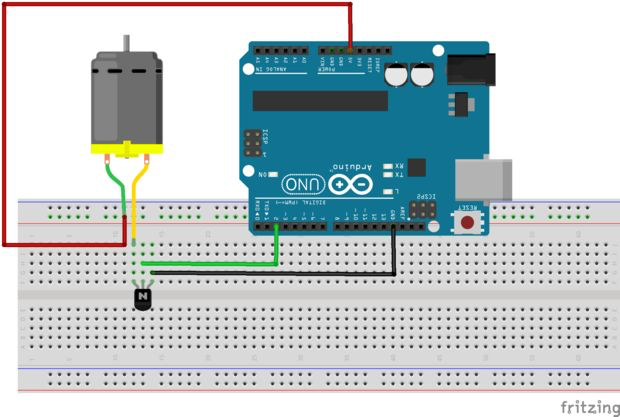


**Fig. 1: electronic assembly. Be careful, because what goes on the opposite side of the resistance to the base should not go to 5V in your assembly if not to the PWM pin of your Arduino board (see next figure).**

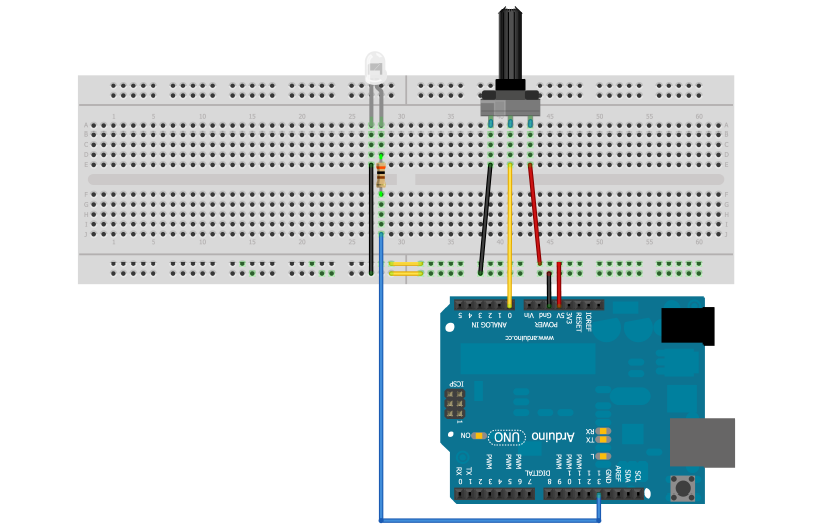
To carry out this session, the assembly of the following image will be followed, in which you will have to add to this assembly to that described in booklet 2 to insert a potentiometer. You can also support yourself in the second Figure, where we have the assembly of an Arduino using a potentiometer (you can use it for the simulation with the 123dcircuits). In class you can only use the potentiometer of the Grove kit.

Once you have done your assembly with the motor and the potentiometer goes to the next section. Remember that these figures are by way of example with an Arduino Uno, to do it with a mega Arduino you can use the same pins.

**Fig. 2: Arduino with a motor and a BC547 - BEWARE if you look at Fig. 1 you have to put a resistance that there must be between the collector and the PWM pin of your Arduino board.**



**Fig. 3: ARDUINO WITH a motor and a BC547**



**Fig. 4: Arduino with potentiometer and a resistance of 1K.**

The potentiometer will serve to simulate the change of temperature in PHASE 1 we will use (variable resistance that you have in the kit), in booklet 2 came an example of use, check it.

After all, the objective is for the motor to spin faster or slower depending on the temperature read, so the potentiometer depending on where it turns will simulate that the temperature rises or falls. Once you finish Phase 1, where the potentiometer is, you will have to go to Phase 2 and put a temperature sensor.

**3.3. BLOCK DIAGRAMS OF STAGE 1**

Next, we describe the block diagram you must implement on the assembly described above. Do it first like this, and when it has worked without the temperature, do it with the temperature.

**Start**

Move potenciometer

Read value from potenciometer

Translate value from potenciometer to PWM

Move the ventilator according to PWM

**3.4. PSEUDOCODE OF STAGE 1**

The next will be the pseudocode.

Init potenciometro=A0;

init motor=3;

Init shaft;

configure

Serial baudrate (9600);

pin mode (potenciometer, input);

Pin mode(motor,out);

bucle

shaft=read analog input(potenciometer);

velocity=map (shaft,0,5,0,255); // see reference of Arduino web

write (motor,velocity);

**3.5. RECOMMENDATIONS FOR STAGE 1**

The first step to carry out the program is to declare the variables that will be used with the "int" command.

int potentiometer = A0; // The potentiometer will be connected to analog pin A0.

int motor = 3; // Motor connected to digital pin 3.

int shaft; // Initialize a variable called shaft.

Within the void setup, the inputs and outputs of the system must be configured, specifically the potentiometer as an input using the "pinMode" command and the INPUT mode, because depending on what we are going to, we will obtain a speed of rotation u other.

The motor, on the other hand, is the output of the system, so it is set with the OUTPUT mode.

  pinMode (potentiometer, INPUT); // The potentiometer is configured as an input.

  pinMode (engine, OUTPUT); // The motor is configured as an output.

Through the command "analogRead" we will read the value of the potentiometer connected to the analog pin A0 and save it in the variable shaft initialized previously.

By last. and from the command "map", we will vary the speed in a range of intensity from zero to five depending on the value of the variable shaft and by means of the instruction "analogWrite" we will assign said speed to the motor.

// To read the value of the potentiometer

shaft = analogRead (potentiometer); // We store the potentiometer value in this variable.

// Motor control by PWM

// If we enter the variable shaft within the "map" command, the fan speed

// will rotate according to what we will turn the potentiometer.

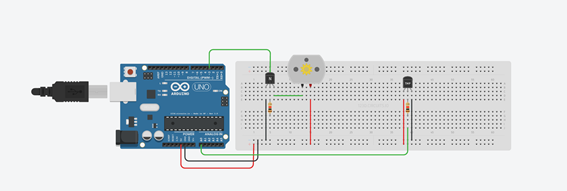
int velocity = map (shaft, 0,5,0,255); // Check in the Arduino libraries what

// means each parameter

analogWrite (engine, speed);

**3.5. RECOMMENDATION FOR STAGE 2**

At this point, you must add the temperature sensor to the assembly, and test the complete system for measuring the temperature and controlling the motor. The recommendation is that you finish the program from which you will vary the speed of an engine depending on the temperature (higher temperature, higher speed of the engine). Although you can take into account other considerations. The program starts from an initial state in which the fan does not rotate. If the temperature exceeds twenty degrees, the fan rotates and sets to level two of speed, which corresponds to a slight turning speed. If the temperature increases more, until reaching a value higher than 30 degrees, the fan increases speed up to level 3. Finally, if the temperature is excessively high (above 30 degrees), the fan is programmed to rotate at maximum speed. To carry out this phase you must carry out the assembly of the following figure. We recommend that you first do so with the 123dcircuits.io before doing it on the actual board.



NOTA: There is a fairly complete example to use an assembly similar to the practice but with the aim of turning on a light bulb based on a potentiometer at http://bildr.org/2012/03/rfp30n06le-arduino/al

**3.5. BLOCK DIAGRAMS OF STAGE 2**

**START**

VELOCITY OF VENTILATOR = 5

VENTILATOR IS OFF

NO

If the temperature > 30

VELOCITY OF VENTILATOR = 3

NO

If the temperature > 25

VELOCITY OF VENTILATOR = 2

YES

NO

If the temperatura is < 20

YES

YES

**3.4. PSEUDOCODE STAGE 2**

Declare next variables:

a;

del=1000;

temperature;

b=3975;

resistance;

motor=3;

c;

velocity;

setup

configure Baudrate (9600);  
 pin mode (motor, output);

bucle

a=read analog input (A0);

// The next code is for LM35, The Simulator has the TMP36 sensor

// Each sensor has different formulas. Search on the web

// http://www.prometec.net/sensor-tmp36/

Float resistance = (value/1023.0) \* 5000; // For 5V

Float temperature=resistance / 10;

Wait (del);

If (temperature >20 and temperature=<25) then

c=2;

velocity=map (c,0,5,0,255);

write (motor,velocity);

end if

if (temperature >25 and temperature<=30) then

c=3;

velocity=map (c,0,5,0,255);

write (motor,velocity);

end if

if (temperature >30) then

c=5;

velocity=map (c,0,5,0,255);

write (motor,velocity);

end if

else { // Think what happened? Fixed}

**3.5. FOR THE TEACHER**

For the delivery of the PWM to be approved it is necessary to show the operation following the same instructions that we have proposed to the teacher

In addition, a delivery to the teacher of the results of this booklet in which you have to respect the delivery date of the following will be enabled.

The delivery must be made in the SAME ZIP (NumeroGrupoX-sed.zip) with the next elements:

o Section 1- PWM: The first page with the name and surnames of the group members.

o Section 2- PWM: Electronic circuit type Figure 1 of your system of Phase 1 and Phase 2 together with the image or photograph of your circuit (if possible made from the 123dcircuits).

o Section 3- PWM: State / block diagram to explain the functioning of your block and the description you consider to explain how it works.

o Section 4- PWM: Table of Truth that represents the functioning of your code plus the development of the logical gates that would result when simplifying by the methods seen in class.

o Section 5- PWM: The code of your Arduino file copied within the same document.

or [optional] Section 6- PWM: A link to a video that demonstrates how it works (or pictures of how it works in that file).