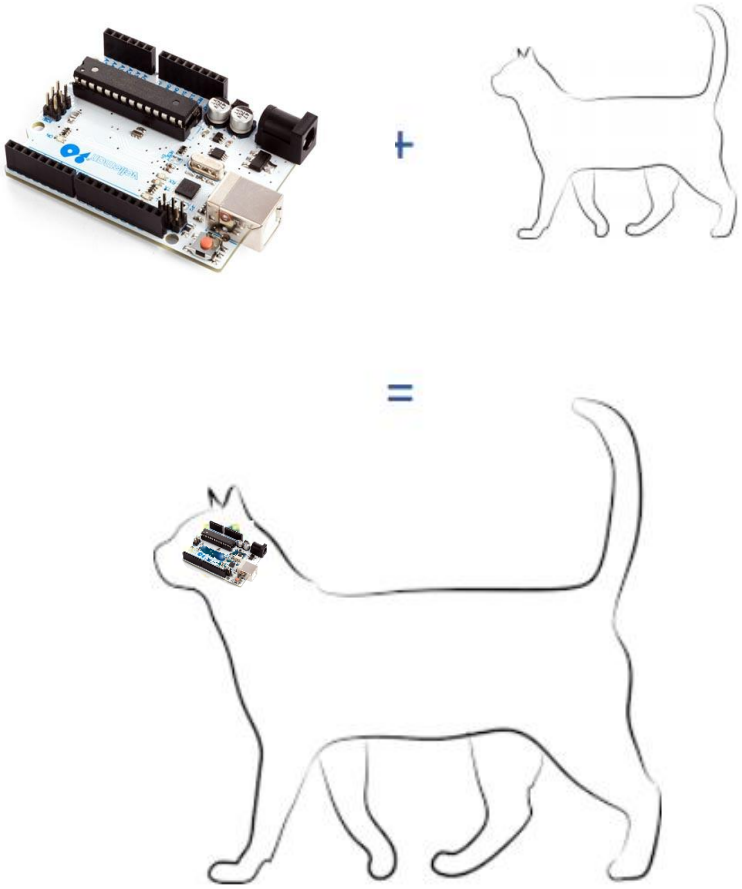
CoALA – *Code a Little Animal*

WORKSHOP

Level 2 (Lower) Secondary Education

CordeMaria.cat
escoles



WORKSHOP C

Level 2 (Lower) Secondary Education

Information:

The list of materials refers to using an Arduino board.

It can be used either by a single student, by a team or by a small group of students.

- Arduino board
- laptop or pc
- USB cable for the Arduino
- battery (9.5v) for the Arduino
- connection cables
- resistors
- LEDs
- RGB LEDs
- 8x8 LEDs matrix (MAX7219-based)
- buzzer
- light sensor
- temperature sensor
- humidity sensor
- NFC reader (grove NFC)
- three NFC-Cards
- material for pet construction (cardboard, porexpan…)

WORKSHOP C

Level 2 (Lower) Secondary Education

learning area	number	worksheet	done
Get to know the Arduino			
	1	Arduino board	☐
	2	coding	☐
	3	Arduino interfaces	☐
CoALA: Code A Little Animal			
	4	your pet	☐
	5	needs of your pet	☐
	6	vital functions of your pet	☐
	7	state of your pet	☐
	8	gazing	☐
	9	touching	☐
	10	tasting (1) - food	☐
	11	tasting (2) - drink	☐
	12	body of your pet	☐
	13	life of your pet	☐

WORKSHEET 1. ARDUINO

With Arduino you have countless creative options at your fingertips. Do you want to build a robot or transmit messages? With a microprocessor like Arduino you can create a series of instructions (or program) so that Arduino does exactly what you write in the program. This is called *programming* or *coding*. Now, look closely at your Arduino.

Figure 1: **Arduino UNO R3 board**

Arduino is based on a microcontroller (an electronic processor that executes a single program). It has the following connection ports:

- 14 digital pins (on / off) that can be configured as inputs or outputs. Of these, 6 outputs can be configured as PWM outputs (~).
- 6 analog input pins.
- 1 USB connection.
- 1 external power connector (maximum 12 V).
- You can find more information at:

https://www.arduino.cc/

Task 1: Identify on the Arduino board the main components that integrate the board and find out what they are and what their function is.

Task 2: Identify on the Arduino board the different connection ports / pins and indicate what the terms “digital”, “analog” and “PWM” mean.

WORKSHEET 2. CODING

With Arduino you will be able to write programs for creating digital projects. The purpose of a program is to perform a sequence of instructions that will automate a given task on a computer. Writing this sequence of instructions is called coding. The instructions are written in one of the so-called programming languages. Arduino uses an own programming language and an own integrated development environment (Arduino IDE). The Arduino Software (IDE) allows you to write programs and upload them to your board.

2.1. ARDUINO IDE

In the Arduino page (https://www.arduino.cc/en/Guide/HomePage) you will find two options:

a) **Online IDE** (Arduino Web Editor). It will allow you to save your sketches in the cloud, having them available from any device and backed up. You will always have the most up-to-date version of the IDE without the need to install updates or community generated libraries. It is available at: https://create.arduino.cc/. It requires to create a user account.

Figure 2: **Online IDE (Arduino Web Editor)**

b) **Desktop IDE**. It is installed in your own computer, so you can work offline. You can get the latest version from the page: https://www.arduino.cc/en/Guide/HomePage. Once the IDE has been installed, it can be accessed through the icon:

2.2. ARDUINO PROGRAMS

When the Arduino Editor opens, the following screen appears:

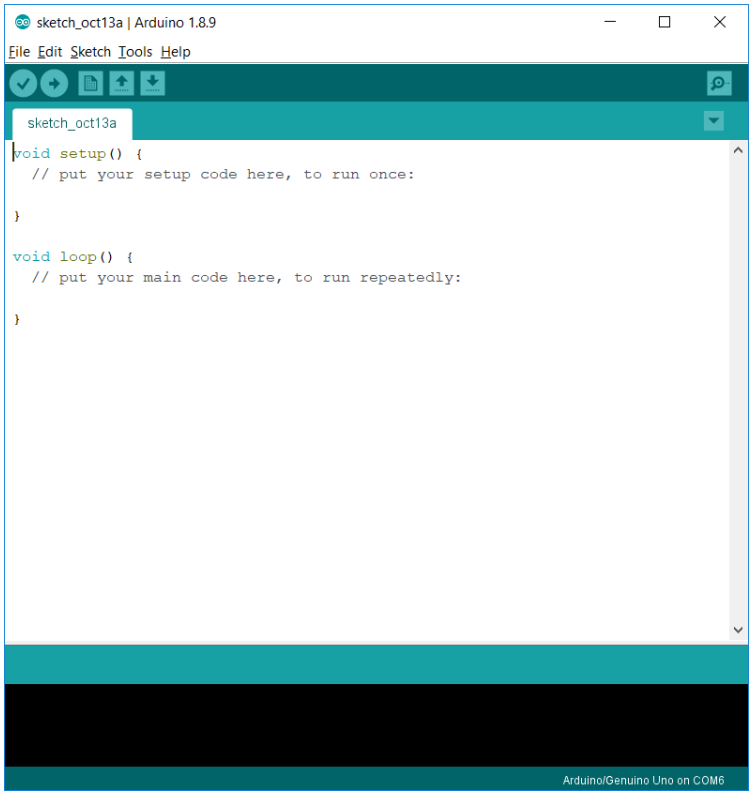


Figure 3: **Arduino program structure**

The screen shows already the Arduino program window. An Arduino program (also called sketch) has two main parts: the **setup** function and the **loop** function.

The **setup** function is used to initialize variables, pin modes, start using libraries, etc. The **setup** section will only run once, after each powerup or reset of the Arduino board.

The **loop** function does precisely what its name suggests: loops consecutively, allowing your program to change and respond.

Comments are lines in the program that are used to inform yourself or others about the way the program works. A single line comment is written: ***// comment*** while a multi-line comment is written: ***/* comment #1 … */***

Task 1: Program the Arduino board. To program the board, follow these steps:

1. Connect the Arduino board to the computer via USB cable
2. Install the drivers (https://www.arduino.cc/en/Guide/HomePage) on the Arduino board if necessary
3. Start the Arduino web editor or the desktop IDE
4. Open an example: Blink (blinks the integrated LED on the Arduino board)

Figure 4: **Opening Arduino program**

5. Select the type of board
6. Select the serial communication port if necessary
7. Upload the program on the Arduino board

WORKSHEET 3. ARDUINO INTERFACES

As we have seen in previous WORKSHEET, the Arduino board is a quite powerful device to control and execute complex tasks according to coded instructions. However, by itself, the Arduino board is limited since the Arduino only interfaces are electrical signals at the pins on the board. Therefore, to interface with the real world some additional elements are needed: sensors and actuators.

A **sensor** is a device that detects physical characteristics of the environment and send the information to other electronics, frequently a processor such as: voltages, currents, temperature, light, sound, …

An **actuator** is a component of a system, that is responsible for moving and / or controlling a mechanism or some machinery.

These sensors and actuators are usually prepared to connect / communicate with the Arduino board using the pins on the board. As we have seen in WORKSHEET 1, these pins are of different types, digital and analogic.

3.1. DIGITAL INPUTS / OUTPUTS

Open the Programa_1 in the editor. You will see the two sections, setup() and loop(). Connect the Arduino as shown in the figure (with a cable connecting pin 2 and the GND pin) and load Programa_1 in the Arduino board.

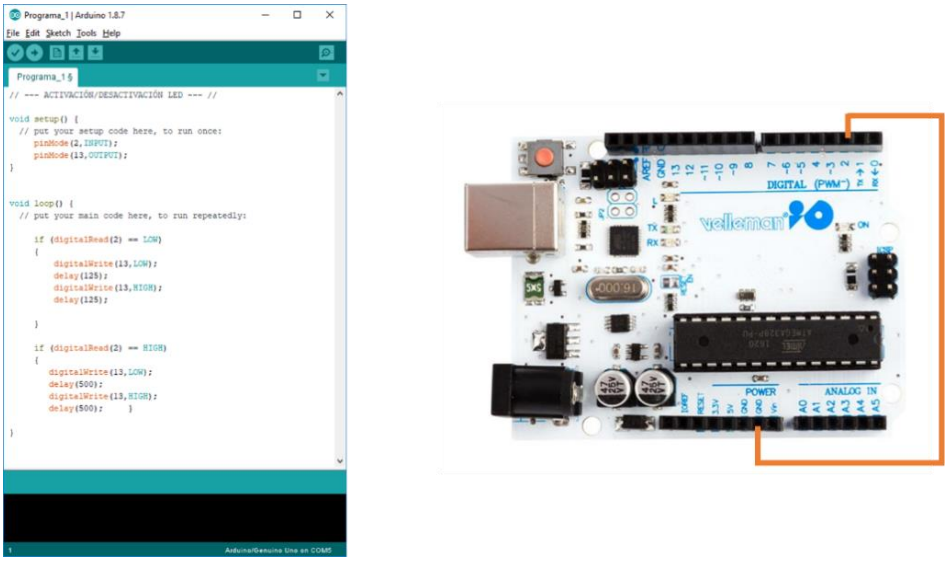


Figure 5: **Digital Inputs / Outputs**

Task 1: Analyze the operation of the program. The Arduino board has an integrated LED connected to pin 13. Why does the board LED turn on and off? What do you think the program does in each sections, **setup ()** and **loop ()** and what does each of the instructions do?

Task 2: If you have analyzed the program, how would we have to connect the cable to change the blinking frequency? What changes would we have to make in the program so that the blink goes slower? And faster?

Task 3: Could you make it work with pin 3 instead of pin 2? And could you make blink at four frequencies? (hint: two wires are needed).

3.2. ANALOG INPUTS

Open the Program_2a in the Arduino editor. We are now working with analog inputs, that is, they can vary continuously between 0V and 5V. To do this, we will work with an ambient light sensor. Disconnect the Arduino board to change the connections.

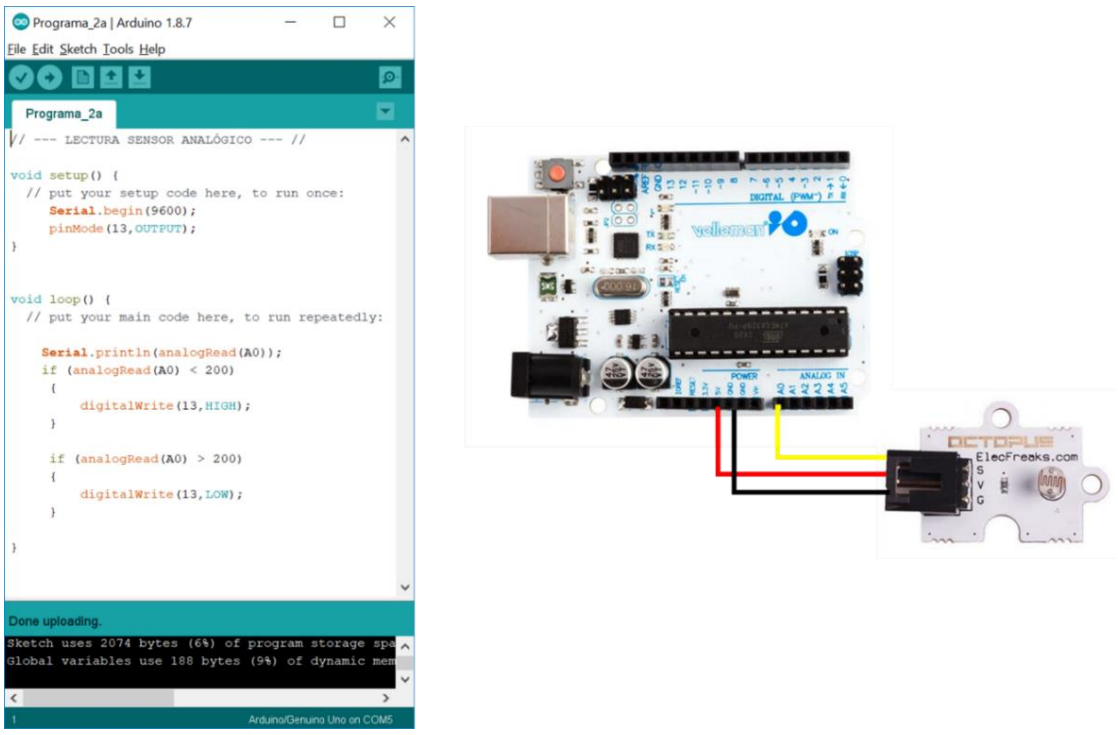


Figure 6: **Analog Inputs**

Connect the Arduino to the sensor as shown in the next figure: black (ground) wire connected to GND, red (power supply) wire connected to 5V, and analog (sensor output) signal connected to input pin A0 and upload the program.

Task 1: Analyze the program. Why the LED turns on and off?

Task 2: Activate the "Serial Monitor" in the "Tools" tab and see the maximum and minimum values sensed. Notice when the LED turns on and off with respect to the value of the “Serial Monitor”. How could you make the light meter more or less sensitive?

3.3. ANALOG OUTPUTS (PWM)

Open the Program_2b in the Arduino editor. We are now working with the analog outputs, that is, they can vary continuously between 0V and 5V. This is achieved by generating a pulse train with variable width called PWM. The output is digital, but it varies so fast that the average value can take any value between 0 and 5V.

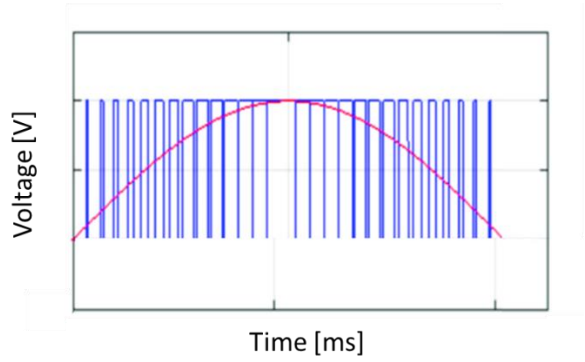


Figure 7: **PWM Signal**

To do this, we will work with the ambient light sensor that we will use to control the brightness of an LED.

Connect the Arduino to the sensor as shown in the next figure: black (ground) wire connected to GND, red (power supply) wire connected to 5V, and analog (sensor output) signal connected to input pin A0 and the LED plus a 330Ω series resistor to the pin 13 and load the program on the board.

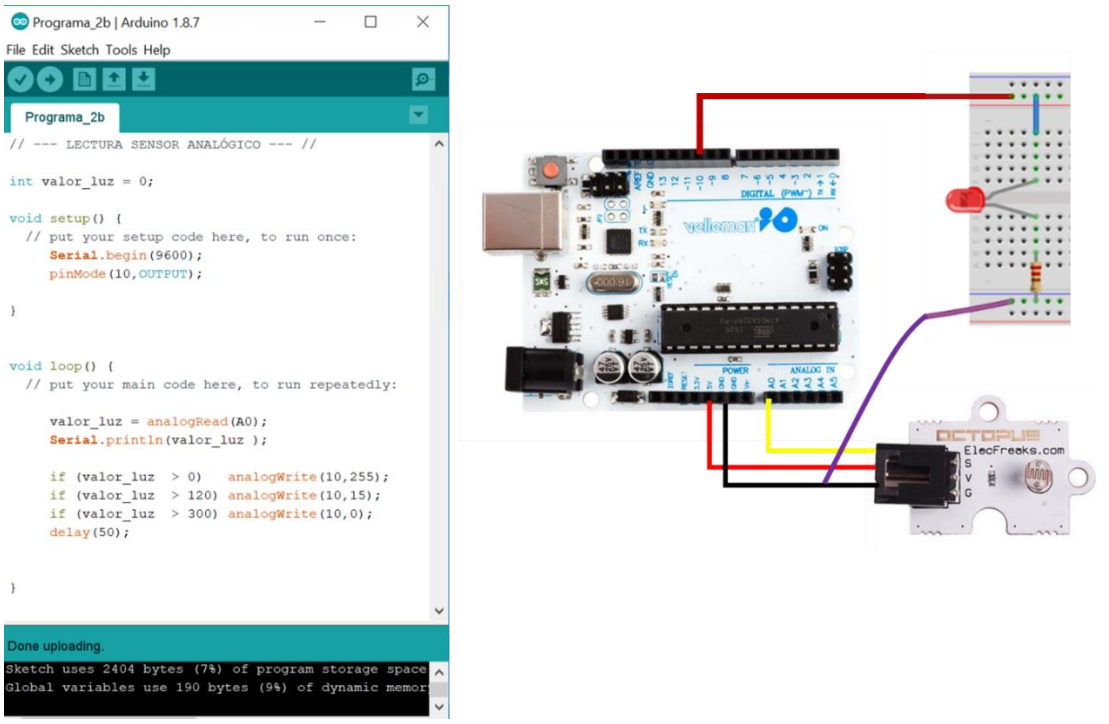


Figure 8: **Analog Outputs**

Task 1: Analyze the program. How many levels of brightness have we programmed? How could we add more levels?

Task 2: How would you change the program so that the brightness of the LED is proportional to the light intensity?

NOTE: To test the connections, we use a BreadBoard by connecting the Arduino pins 5V (supply) and GND (ground) to the BreadBoard as shown in the Figure. In this way, we can make tests more easily (to connect each analog sensor we will need a 5V and a GND connection).

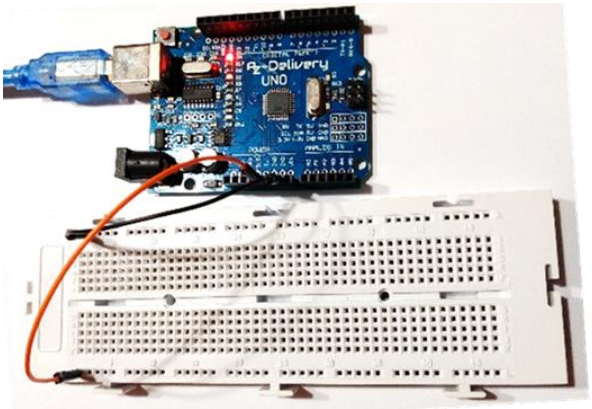


Figure 9: **Breadboard**

WORKSHEET 4. YOUR PET

A pet (dog, cat, rabbit, ...) is a domesticated animal that has the purpose of giving company.

Task 1: Choose a pet.

Talk about your choice with your teacher. Note the pet you choose with your teacher here:

_____

Task 2: Write a profile of your pet.

General

Name: _____

Taxonomy: _____

Characteristics: _____

Comfort temperature: _____

Lifespan: _____

Physical characteristics

Body length: _____

Body height: _____

Weight: _____

Lifestyle

Diet (food and drink): _____

Activity (much or little): _____

Sleep (much or little): _____

WORKSHEET 5. NEEDS OF YOUR PET

By writing the profile you already learned a lot about your pet. However, before you consider getting a pet it is important to know what the pet needs. A pet, like any other living being, has different vital functions. A vital function is, in biology, any of the three processes or functions that all living beings perform: nutrition (which includes breathing), interaction (relationship) and reproduction.

Task 1: What does your pet need?

Write down the five most important points.

1. _____
2. _____
3. _____
4. _____
5. _____

Soon, you will be able to have a pet (using the Arduino board). You will be able to make and take proper care of your pet. You will simulate how your pet 'senses' and 'feels' using sensors and actuators and coding/programming the behavior of the pet on the next worksheets.

Task 2: How make your pet happy?

Think on how to do the care or nurturing of the pet and how the state or feelings of the pet could be known and how could be modified or altered by the behaviour of the owner.

WORKSHEET 6. VITAL FUNCTIONS OF YOUR PET

Using electronic devices, we want to simulate some of the pet's vital functions (basically nutrition and relationship functions). The functions to simulate are:

- Nutrition → food / drink
- Relationship → look, caresses, …

In this worksheet, we will look among the different types of sensors and actuators available and choose several among them to simulate these functions.

Task 1:

a) How can you know how your pet feels?

_____

_____

b) How can you check if the room is hot enough for your pet?

_____

_____

c) How can you verify if your pet is being caressed?

_____

_____

d) You give your pet two different foods. How do you know what your pet prefers?

_____

_____

e) How can you know if your pet is thirsty?

_____

_____

WORKSHEET 7. STATE OF YOUR PET

To visualize the physical and emotional state of your pet we will use an 8x8 LEDs matrix controlled a the MAX7219 device.

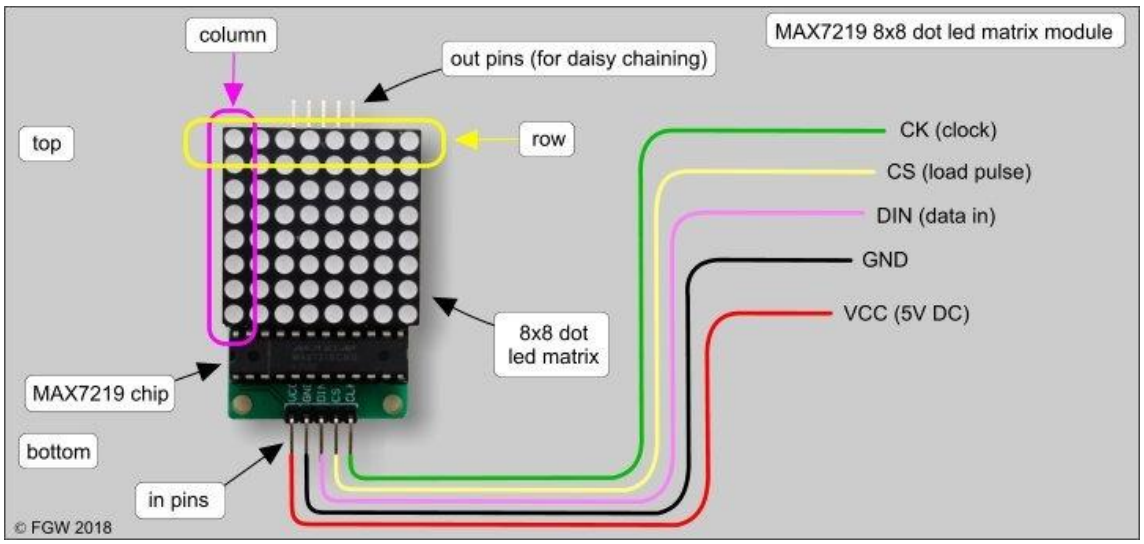


Figure 10: **8x8 LEDs matrix**

The interfacing between the MAX7219 and the Arduino is done using a communication bus with three wires (Clock, CS and DI). The connection scheme is shown in next Figure:

CS -pin 10

CLOCK- pin 11

DIN- pin 12

VCC- 5v

GND

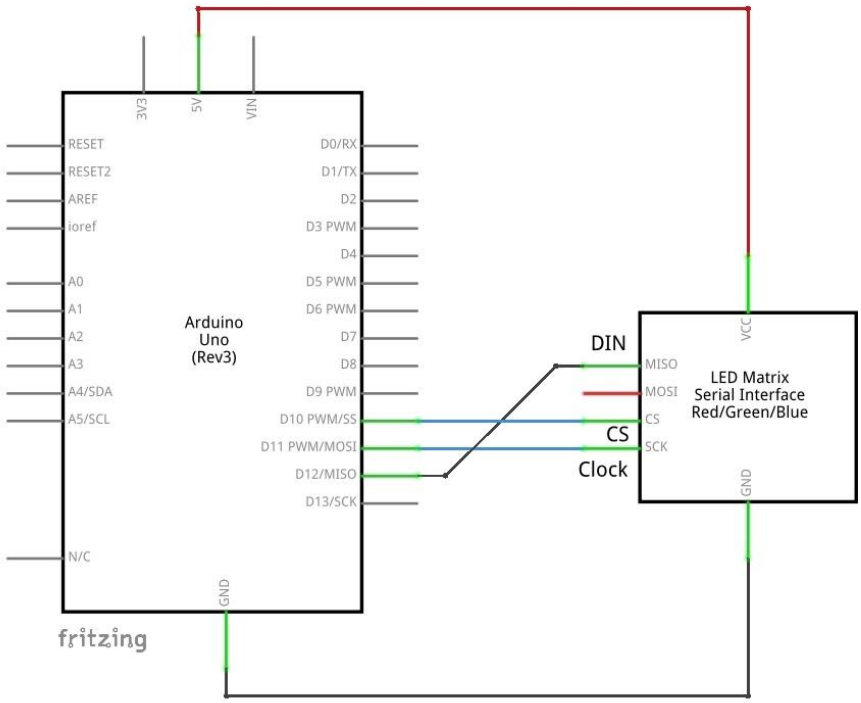
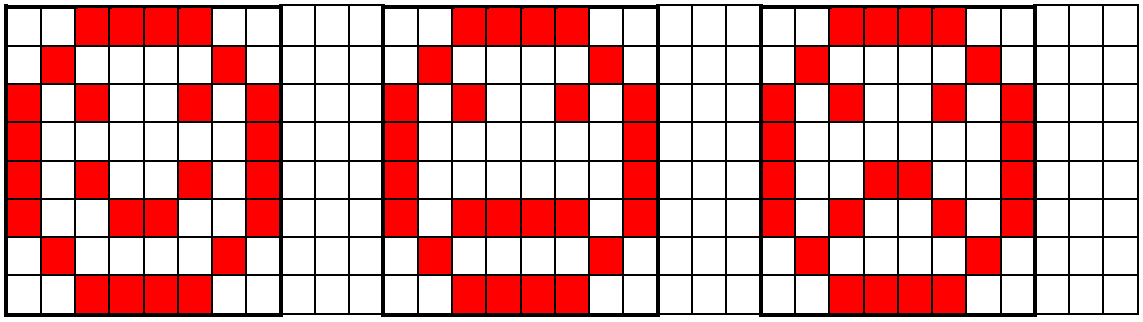


Figure 9: **MAX7219-Arduino Interface**

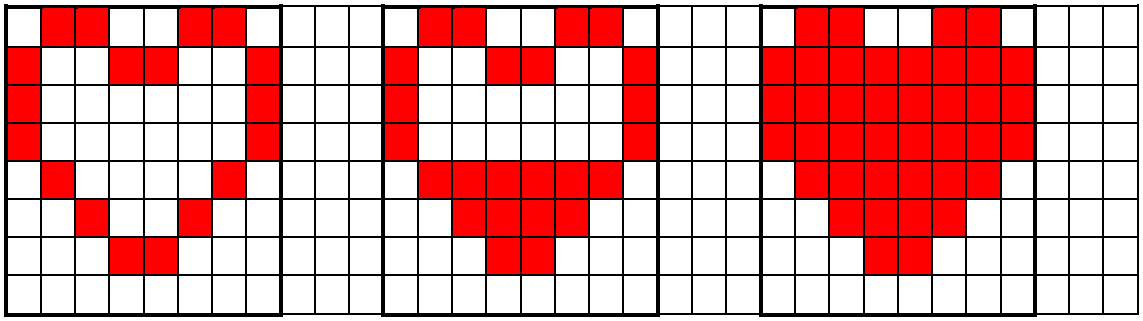
Task 1: Connect the LEDs matrix as indicated and install the library (LedControl.h) for managing the LEDs following the instructions available at:

https://www.instructables.com/id/LED-Matrix-with-Arduino/

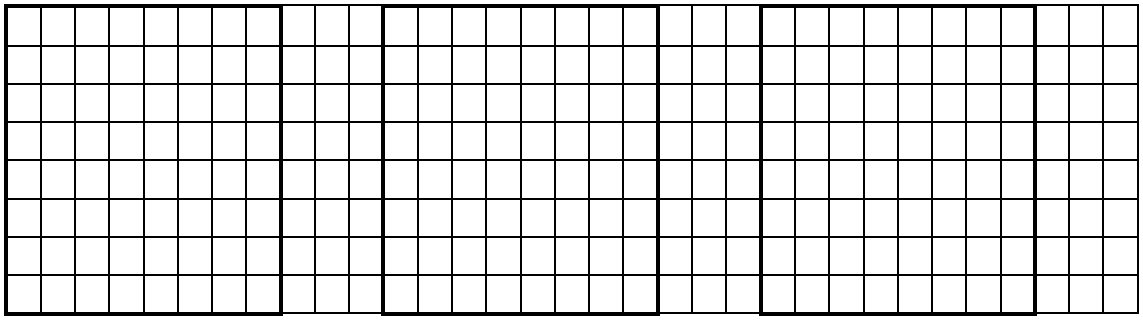
Once the library is installed, open the Arduino editor and in the menu **File/Example/LedControl/** you can find different examples to understand its programming. Open and upload the program **Programa_3a** that will show different faces in the LED matrix:



Task 2: Modify the program so that it draws a full heart that is emptying. Use the square template as template. Save it with the name **Programa_3b**.

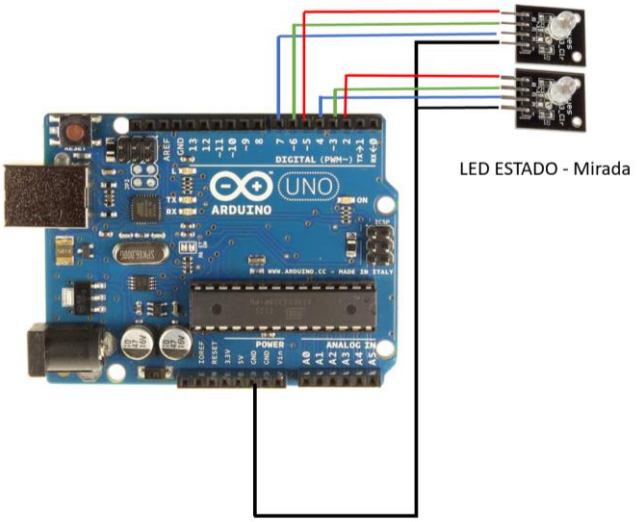


Task 3: Modify the program so that you draw on the screen different types of food that your pet likes and that your pet doesn't. Save it with the name **Programa_3c.**



WORKSHEET 8. GAZING

Another way to show the mood of our pet is using RGB LEDs simulating the gazing and programming them to change colour according to the state.



LED RGB 1

R (red) – pin 2
G (green) – pin 3
B (blue) – pin 4
– GND

LED RGB 2

R (red) – pin 5
G (green) – pin 6
B (blue) – pin 7
– GND

It generates three different programs, one that the LEDs are green (the pet feels good 🟢🟢) named **Program_4a**, another that the LEDs are blue (the pet is changing its status 🔵🔵) named **Program_4b** and another with red eyes (the pet feels bad 🔴🔴) named **Program_4c**.

Once the pins are indicated as output: `pinMode(2, OUTPUT)` in the `setup()` function; it is indicated in the `loop()` function; Which pins are active or not in each program:

`digitalWrite(2, LOW);` or `digitalWrite(3, HIGH);`

If you use coloured wires, it will be much easier to know what each wire corresponds to.

You can also try programming a buzzer (your pet will complain when he doesn't feel well) or some other actuator with the same purpose. Program_5. Simulate different sounds and advise the teacher to select one of them.

ACTUADOR BUZZER

– GND
Pin 8
+ 5V

Answer the following questions:

1. Are the devices you have used input or output?

RGB LED: ……

Buzzer sensor: ……

2. Are they digital or analog?

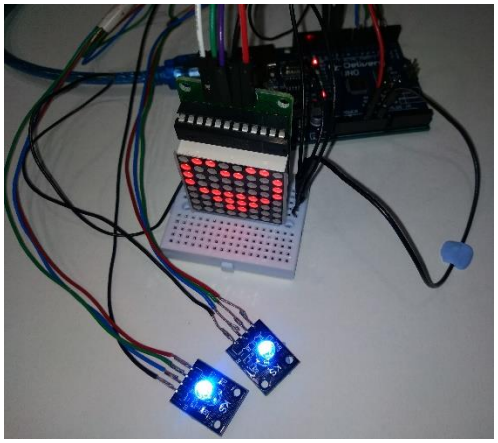
RGB LED: ……

Buzzer sensor: ……

Justify the answers.

You may use more than one sensor to indicate how your pet feels:

Task 1: You can relate what happens in the LEDS matrix with RGB LEDs, for example, when the heart is emptying, the eyes change color, from green, to blue and finally red. **Program_6**. (Combine **Program_3c** with **Program_4).** As you will have to connect many cables, use a breadboard.



Task 2: You can link what happens in the LEDS matrix with the buzzer sensor and the pet could produce a sound when the heart is almost empty. **Program_7**.

WORKSHEET 9. TOUCHING

Your pet wants to be petted, otherwise it feels lonely. Use the light sensor to simulate this action. When light does not reach the sensor (covered sensor) the pet knows that it is being petted. If it passes a while without caressing it, your pet feels sad and sends us a signal: the heart begins to empty, and the eyes change colour.

Since the light sensor is an analog sensor, it must be connected to GND (black G-cable), 5V (red V-cable) and the output signal S should be connected to an analog input pin (in our case we will use A0). See **Program_2a**.

SENSOR LUZ

S/S – A0
V/ – 5V
G/- – GND

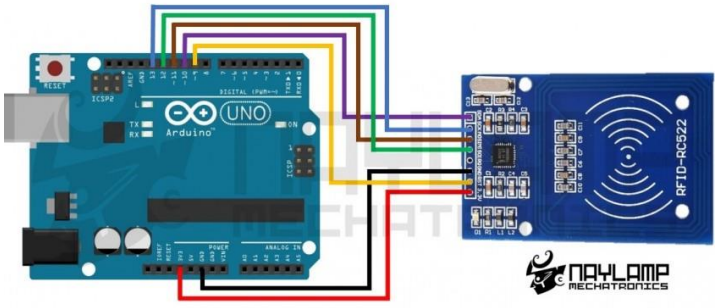
Task 1: Connect the light sensor as indicated (without disconnecting the LED array or RGB LEDs). Open your saved program, **Program_6** and add the light sensor code to fill the heart when the sensor is covered. Save it as **Program_8**.

We will let the heart empty out unless we touch the pet. Code this behaviour using an if function:

```
if (analogRead (A0)> 200)
{
   heart_stat = 5;
}
```

WORKSHEET 10. TASTING (FOOD)

Your pet needs food to survive. We will simulate the different kinds of food using a wireless NFC card reader. In our case we will use an RFID reader RC522.



LECTOR RFID RC522
RST – pin 9
SDA(SS) – pin 10
MOSI – pin 11
MISO – pin 12
SCK – pin 13
GND – GND
3.3 V – 3.3 V

For the RFID reader RC522 to function properly, it is necessary to download and install the library "MFRC522.h", following the instructions of:

https://naylampmechatronics.com/blog/22_Tutorial-Lector-RFID-RC522.html

Once the library is downloaded, open the Arduino editor and in the menu File/Examples/LedControl/ you can try different programs to see how it works.

Task 1:

a) Connect the NFC reader to an Arduino board as indicated in the previous Figure and execute the example program **rfid_read_personal_data**, open the "Serial Monitor" from the "Tools" menu to be able to read the contents of each card or keychain. Ask the teacher for the prepared cards and draw an appropriate image for the respective text message on each card. Use a non-permanent marker!

b) You can write on cards or keychains what you consider appropriate for your pet. (Remember that on sheet 7A we already draw 3 foods. **Program_3c**). Now install the example program **rfid_write_personal_data**, open the "Serial Monitor" from the "Tools" menu to write the contents of each card or keychain.

Task 2:

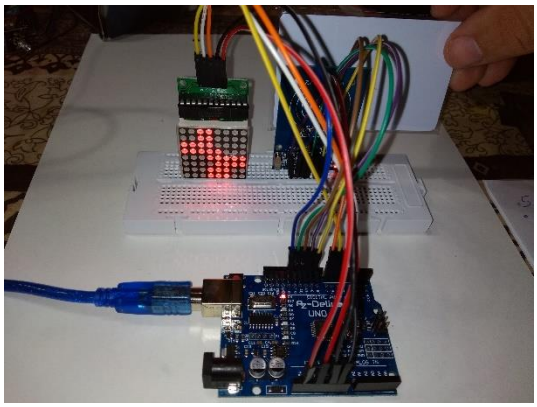
a) Upload the Program_9 and check how it works.

b) As we do not have enough digital pins, we will use the same pins for the two RGB LEDs (pin 2, pin 3 and pin 4), we connect them in parallel. And we will use

pins 5, 6, 7 to connect the LEDS matrix. Once the connections have been modified, modify **Program_8** and save it as **Program_8b**. Check that everything works.

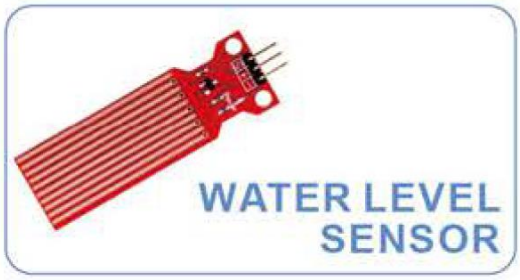
c) Add the NFC reader to your pet's device. We must modify the program so that it becomes happy when we give it food that it likes (that is, it fills the heart). Show the program to your teacher.

To make it more visual, we can merge this program with the **Program_3c** in which we draw the food that the NFC reader reads in the LEDs matrix. Save it as **Program_10.**



WORKSHEET 11. TASTING (DRINK)

Your pet also needs to drink (water) to survive. We will use a humidity sensor to simulate the tongue of the pet. Use a humidity sensor to measure if you have enough water. Depending on the humidity, it offers values between 0 and 800.



SENSOR DE AGUA

S/S – A0
V/ – 5V
G/- – GND

Task 1: Connect the humidity sensor to an Arduino board as indicated in the previous Figure and use a glass of water to determine the output value:

- when the sensor is not submerged in water (value: _____)

- when half of the sensor is submerged in water (value: _____)

- when the sensor is completely submerged in water (value: _____)

Attention: make sure not to immerse the sensor beyond the electrical contacts.

Task 2: Program a sequence in which to fill the heart when you consider that the value of the sensor reading is appropriate. Save the program as **Program_11**. Show your program to your teacher.

WORKSHEET 12. BODY OF YOUR PET

In this worksheet, we will design the body of the pet using some templates of animals such as dog, cat, …

Task 1:

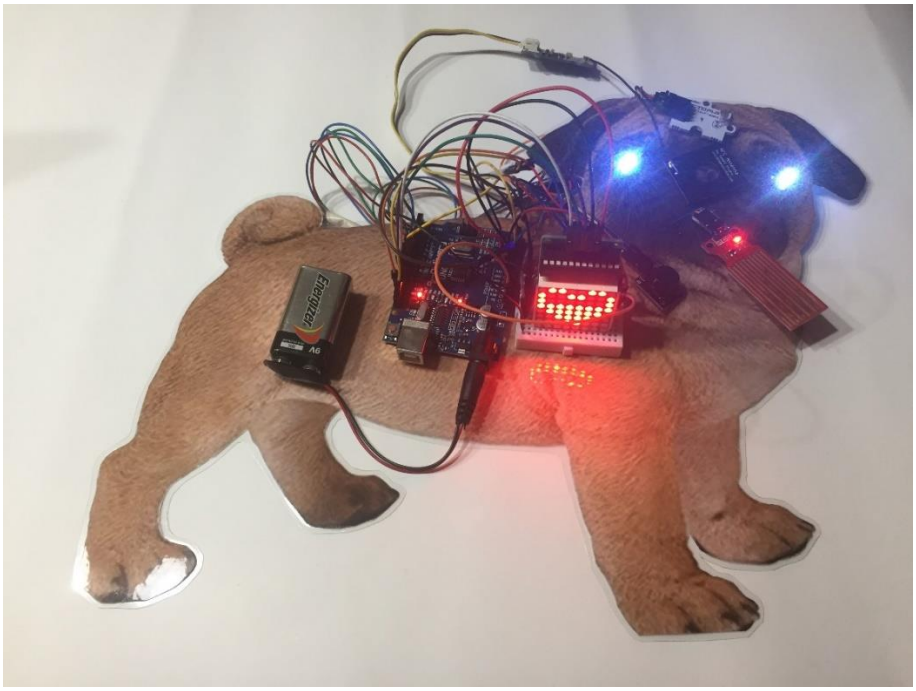
a) You need the right template for your pet (templates 4-11). Cut out your pet's body and stick it on a cardboard. Leave enough space on the top of the card for Arduino.

b) Cut along the red lines (not the red areas) with a cutter. Let your teacher help you if there is any difficulty.

c) Paste the different devices in the appropriate areas of your pet.

d) Hold the Arduino board above the pet. Leave some space.

e) Put your tongue, eyes, the NFC reader and the light sensor on your pet's head. Put your heart on your pet's body. Connect the devices to the Arduino board. Attention: each cable must be in the right place. Check the cards for instructions.



WORKSHEET 13. LIFE OF YOUR PET

Once all is on place, let's check the behaviour of our pet. Turn on your Arduino.

Task 1:

a) Does your pet react?

b) Touch your pet.

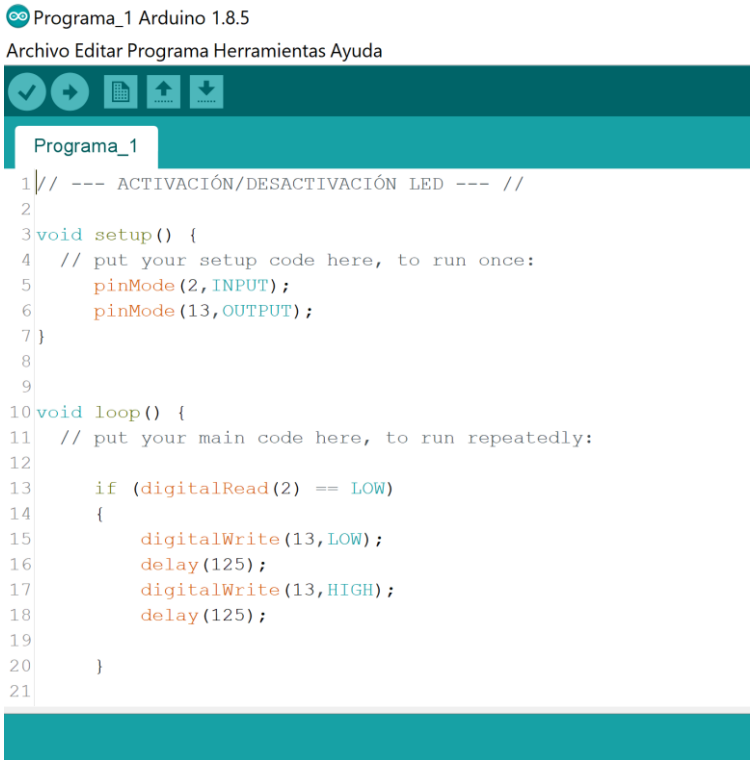
c) Feed your pet.

d) Give water to your pet.

Task 2: Think of other sensors and devices that could be used to simulate other characteristics of your pet and discuss your proposal with the teacher.

PROGRAMS

Program 1




```
// --- ACTIVACIÓN/DESACTIVACIÓN LED --- //

void setup() {
  // put your setup code here, to run once:
     pinMode(2,INPUT);
     pinMode(13,OUTPUT);
}


void loop() {
  // put your main code here, to run repeatedly:

     if (digitalRead(2) == LOW)
     {
          digitalWrite(13,LOW);
          delay(125);
          digitalWrite(13,HIGH);
          delay(125);

     }

```

Program 2a

Programa_2a Arduino 1.8.5

Archivo Editar Programa Herramientas Ayuda

Programa_2a

```
// --- LECTURA SENSOR ANALÓGICO --- //

void setup() {
  // put your setup code here, to run once:
     Serial.begin(9600);
     pinMode(13,OUTPUT);
}


void loop() {
  // put your main code here, to run repeatedly:

    Serial.println(analogRead(A0));
    if (analogRead(A0) < 200)
     {
         digitalWrite(13,HIGH);
     }

     if (analogRead(A0) > 200)
     {
         digitalWrite(13,LOW);
```

Program 2b

Programa_2b Arduino 1.8.5

Archivo Editar Programa Herramientas Ayuda

Programa_2b §

```
// --- LECTURA SENSOR ANALÓGICO --- //

int valor_luz = 0;
void setup() {
  // put your setup code here, to run once:
     Serial.begin(9600);
     pinMode(10,OUTPUT);


}

void loop() {
  // put your main code here, to run repeatedly:

     valor_luz = analogRead(A0);
     Serial.println(valor_luz );

     if (valor_luz  > 0)   analogWrite(10,255);
     if (valor_luz  > 120) analogWrite(10,15);
     if (valor_luz  > 300) analogWrite(10,0);
     delay(50);
}
```

Programa_3a – FICHA 7A (caritas)

#include "LedControl.h"
#include "binary.h"

```
/*
 DIN connects to pin 12
 CLK connects to pin 11
 CS connects to pin 10
*/
LedControl lc=LedControl(12,11,10,1);

// delay time between faces
unsigned long delaytime=1000;

// happy face
byte hf[8]=
{B00111100,B01000010,B10100101,B10000001,B10100101,B10011001,B01000010,B00111100};
// neutral face
byte nf[8]={B00111100,
B01000010,B10100101,B10000001,B10111101,B10000001,B01000010,B00111100};
// sad face
byte sf[8]=
{B00111100,B01000010,B10100101,B10000001,B10011001,B10100101,B01000010,B00111100};

void setup() {
  lc.shutdown(0,false);
  // Set brightness to a medium value
  lc.setIntensity(0,8);
  // Clear the display
  lc.clearDisplay(0);
}

void drawFaces(){
  // Display sad face
  lc.setRow(0,0,sf[0]);
  lc.setRow(0,1,sf[1]);
  lc.setRow(0,2,sf[2]);
  lc.setRow(0,3,sf[3]);
  lc.setRow(0,4,sf[4]);
  lc.setRow(0,5,sf[5]);
  lc.setRow(0,6,sf[6]);
  lc.setRow(0,7,sf[7]);
  delay(delaytime);

  // Display neutral face
  lc.setRow(0,0,nf[0]);
  lc.setRow(0,1,nf[1]);
  lc.setRow(0,2,nf[2]);
  lc.setRow(0,3,nf[3]);
  lc.setRow(0,4,nf[4]);
  lc.setRow(0,5,nf[5]);
  lc.setRow(0,6,nf[6]);
  lc.setRow(0,7,nf[7]);
  delay(delaytime);

  // Display happy face
  lc.setRow(0,0,hf[0]);
  lc.setRow(0,1,hf[1]);
  lc.setRow(0,2,hf[2]);
  lc.setRow(0,3,hf[3]);
  lc.setRow(0,4,hf[4]);
  lc.setRow(0,5,hf[5]);
  lc.setRow(0,6,hf[6]);
```

```
  lc.setRow(0,7,hf[7]);
  delay(delaytime);
}

void loop(){
  drawFaces();
}
```

Programa_3b – FICHA 7A (corazón)

```
#include "LedControl.h"

LedControl lc=LedControl(12,11,10,1);  // Pin and # of Displays

unsigned long delayTime=1000;  // Delay between Frames
int heart_stat=5;
int time_stat=0;

// Put values in arrays

byte heart5_icon[] =
{
  B01100110,
  B11111111,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart4_icon[] =
{
  B01100110,
  B10011001,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart3_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart2_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
```

```
  B10000001,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart1_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00111100,
  B00011000,
  B00000000
};

byte heart0_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00100100,
  B00011000,
  B00000000
};

void setup()
{
  lc.shutdown(0,false);  // Wake up displays
  lc.shutdown(1,false);
  lc.setIntensity(0,5);  // Set intensity levels
  lc.setIntensity(1,5);
  lc.clearDisplay(0);  // Clear Displays
  lc.clearDisplay(1);

}

//  Take values in Arrays and Display them

void heart5()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart5_icon[i]);
  }
}

void heart4()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart4_icon[i]);
  }
}

void heart3()
```

```
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart3_icon[i]);
  }
}

void heart2()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart2_icon[i]);
  }
}

void heart1()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart1_icon[i]);
  }
}

void heart0()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart0_icon[i]);
  }
}

void loop() {
  // put your main code here, to run repeatedly:
unsigned char ii;

heart5();

delay(1000);

heart4();

delay(1000);

heart3();

delay(1000);

heart2();

delay(1000);

heart1();

delay(1000);

heart0();

delay(1000);

}
```

Programa_3c – FICHA 7A (comida)

```
#include "LedControl.h"
#include "binary.h"

/*
 DIN connects to pin 12
 CLK connects to pin 11
 CS connects to pin 10
*/
LedControl lc=LedControl(12,11,10,1);

// delay time between faces
unsigned long delaytime=1000;

// pollo
byte hf[8]=
{B00000000,B00001110,B00011110,B00011110,B0011110,B0011100,B01100000,B00100000}
;
// cirera
byte nf[8]={B00000000,
B00011000,B00011000,B00100100,B01000010,B11100111,B11100111,B00000000};
// galeta
byte sf[8]=
{B00111100,B01111110,B11111111,B11111111,B11111111,B11111111,B01111110,B001111
00};
// os
byte os[8]=
{B00001100,B00001100,B00001111,B00011111,B11111000,B11110000,B00111000,B001110
00};


void setup() {
  lc.shutdown(0,false);
  // Set brightness to a medium value
  lc.setIntensity(0,8);
  // Clear the display
  lc.clearDisplay(0);
}

void drawFood(){
  // Display galeta
  lc.setColumn(0,0,sf[0]);
  lc.setColumn(0,1,sf[1]);
  lc.setColumn(0,2,sf[2]);
  lc.setColumn(0,3,sf[3]);
  lc.setColumn(0,4,sf[4]);
  lc.setColumn(0,5,sf[5]);
  lc.setColumn(0,6,sf[6]);
  lc.setColumn(0,7,sf[7]);
  delay(delaytime);

  // Display cirera
  lc.setColumn(0,0,nf[0]);
  lc.setColumn(0,1,nf[1]);
  lc.setColumn(0,2,nf[2]);
  lc.setColumn(0,3,nf[3]);
  lc.setColumn(0,4,nf[4]);
  lc.setColumn(0,5,nf[5]);
  lc.setColumn(0,6,nf[6]);
  lc.setColumn(0,7,nf[7]);
```

```
  delay(delaytime);

  // Display pollo
  lc.setColumn(0,0,hf[0]);
  lc.setColumn(0,1,hf[1]);
  lc.setColumn(0,2,hf[2]);
  lc.setColumn(0,3,hf[3]);
  lc.setColumn(0,4,hf[4]);
  lc.setColumn(0,5,hf[5]);
  lc.setColumn(0,6,hf[6]);
  lc.setColumn(0,7,hf[7]);
  delay(delaytime);

    // Display os
  lc.setColumn(0,0,os[0]);
  lc.setColumn(0,1,os[1]);
  lc.setColumn(0,2,os[2]);
  lc.setColumn(0,3,os[3]);
  lc.setColumn(0,4,os[4]);
  lc.setColumn(0,5,os[5]);
  lc.setColumn(0,6,os[6]);
  lc.setColumn(0,7,os[7]);
  delay(delaytime);
}

void loop(){
  drawFood();
}
```

Programa_4a/Programa_4b/Programa_4c – FICHA 7B

Programa_4a §

```
void setup()
{
  Serial.begin(9600);


  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

 }
void loop(){

  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);


}
```

Programa_4b §

```
void setup()
{
  Serial.begin(9600);


  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

 }
void loop(){

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);


}
```

Programa_4c

```
void setup()
{
  Serial.begin(9600);


  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

 }
void loop(){

  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);



}
```

Programa_6

```
#include "LedControl.h"


LedControl lc=LedControl(12,11,10,1);  // Pin and # of Displays

unsigned long delayTime=1000;  // Delay between Frames
int heart_stat=5;
int time_stat=0;
```

```
// Put values in arrays

byte heart5_icon[] =
{
  B01100110,
  B11111111,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart4_icon[] =
{
  B01100110,
  B10011001,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart3_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart2_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart1_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00111100,
  B00011000,
```

```
  B00000000
};


byte heart0_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00100100,
  B00011000,
  B00000000
};


void setup()
{
  lc.shutdown(0,false);  // Wake up displays
  lc.shutdown(1,false);
  lc.setIntensity(0,5);  // Set intensity levels
  lc.setIntensity(1,5);
  lc.clearDisplay(0);  // Clear Displays
  lc.clearDisplay(1);

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

}


//  Take values in Arrays and Display them

void heart5()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart5_icon[i]);
  }
}

void heart4()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart4_icon[i]);
  }
}

void heart3()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart3_icon[i]);
  }
}
```

```
void heart2()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart2_icon[i]);
  }
}

void heart1()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart1_icon[i]);
  }
}

void heart0()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart0_icon[i]);
  }
}



void loop() {
  // put your main code here, to run repeatedly:
unsigned char ii;


heart5();
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
delay(1000);

heart4();

delay(1000);

heart3();

delay(1000);

heart2();
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
delay(1000);

heart1();

delay(1000);
```

```
heart0();

  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
delay(1000);


}
```

Programa_8

```
#include "LedControl.h"


LedControl lc=LedControl(12,11,10,2);  // Pin and # of Displays

unsigned long delayTime=1500;  // Delay between Frames
int heart_stat=5;
int time_stat=0;

// Put values in arrays

byte heart5_icon[] =
{
  B01100110,
  B11111111,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart4_icon[] =
{
  B01100110,
  B10011001,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart3_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart2_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};
```

```
byte heart1_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00111100,
  B00011000,
  B00000000
};


byte heart0_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00100100,
  B00011000,
  B00000000
};


void setup()
{
  Serial.begin(9600);

  lc.shutdown(0,false);  // Wake up displays
  lc.setIntensity(0,5);  // Set intensity levels
  lc.clearDisplay(0);  // Clear Displays

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

}


//  Take values in Arrays and Display them

void heart5()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart5_icon[i]);
  }
}

void heart4()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart4_icon[i]);
  }
}
```

```
void heart3()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart3_icon[i]);
  }
}

void heart2()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart2_icon[i]);
  }
}

void heart1()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart1_icon[i]);
  }
}

void heart0()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart0_icon[i]);
  }
}


void loop() {
  // put your main code here, to run repeatedly:
unsigned char ii;


 if (heart_stat == 0)
 {
    heart0();
    heart_stat= 0;
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);

 }

 if (heart_stat == 1)
 {
    heart1();
    heart_stat= 0;
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
```

```
   digitalWrite(7, HIGH);
 }

 if (heart_stat == 2)
 {
   heart2();
   heart_stat= 1;
   digitalWrite(2, LOW);
   digitalWrite(3, LOW);
   digitalWrite(4, HIGH);
   digitalWrite(5, LOW);
   digitalWrite(6, LOW);
   digitalWrite(7, HIGH);
 }

 if (heart_stat == 3)
 {
   heart3();
   heart_stat= 2;
   digitalWrite(2, LOW);
   digitalWrite(3, HIGH);
   digitalWrite(4, LOW);
   digitalWrite(5, LOW);
   digitalWrite(6, HIGH);
   digitalWrite(7, LOW);
 }
 if (heart_stat == 4)
 {
   heart4();
   heart_stat= 3;
   digitalWrite(2, LOW);
   digitalWrite(3, HIGH);
   digitalWrite(4, LOW);
   digitalWrite(5, LOW);
   digitalWrite(6, HIGH);
   digitalWrite(7, LOW);
 }

 if (heart_stat == 5)
 {
   heart5();
   heart_stat= 4;
   digitalWrite(2, LOW);
   digitalWrite(3, HIGH);
   digitalWrite(4, LOW);
   digitalWrite(5, LOW);
   digitalWrite(6, HIGH);
   digitalWrite(7, LOW);

 }

  delay(delayTime);


 /* SENSOR LUZ */
if (analogRead(A0) > 200)
{
  heart_stat= 5;
}
```

}

Program_8b

```
#include "LedControl.h"
LedControl lc=LedControl(7,6,5,2);  // Pin and # of Displays
```

(::::)

Program_9 or Programa_rfid_read

```
/*
 * Initial Author: ryand1011 (https://github.com/ryand1011)
 *
 * Reads data written by a program such as "rfid_write_personal_data.ino"
 *
 * See: https://github.com/miguelbalboa/rfid/tree/master/examples/rfid_write_personal_data
 *
 * Uses MIFARE RFID card using RFID-RC522 reader
 * Uses MFRC522 - Library
 * -----------------------------------------------------------------------------------------
 *             MFRC522      Arduino       Arduino   Arduino    Arduino          Arduino
 *             Reader/PCD   Uno/101       Mega      Nano v3    Leonardo/Micro   Pro Micro
 * Signal      Pin          Pin           Pin       Pin        Pin              Pin
 * -----------------------------------------------------------------------------------------
 * RST/Reset   RST          9             5         D9         RESET/ICSP-5     RST
 * SPI SS      SDA(SS)      10            53        D10        10               10
 * SPI MOSI    MOSI         11 / ICSP-4   51        D11        ICSP-4           16
 * SPI MISO    MISO         12 / ICSP-1   50        D12        ICSP-1           14
 * SPI SCK     SCK          13 / ICSP-3   52        D13        ICSP-3           15
*/

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         9          // Configurable, see typical pin layout above
#define SS_PIN          10         // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance

//*****************************************************************************************//
void setup() {
  Serial.begin(9600);                                           // Initialize serial communications with the PC
  SPI.begin();                                                  // Init SPI bus
  mfrc522.PCD_Init();                                           // Init MFRC522 card
  // Serial.println(F("Read personal data on a MIFARE PICC:"));    //shows in serial that it is ready to read
}

//*****************************************************************************************//
void loop() {

  // Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  //some variables we need
  byte block;
  byte len;
  int aliment = 11;
  MFRC522::StatusCode status;

  //-------------------------------------------

  // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
```

```
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }



  //-------------------------------------- GET FOOD NAME

  byte buffer2[18];
  block = 1;
  len = 18;

  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
&(mfrc522.uid)); //line 834
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

  status = mfrc522.MIFARE_Read(block, buffer2, &len);
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

 Serial.print(F("Food: "));
// PRINT LAST NAME
  for (uint8_t i = 1; i < 16; i++) {
    Serial.write(buffer2[i] );
  }

aliment = buffer2[0];

if (aliment == '0')
{
     Serial.print(F("BONE is GOOD\n"));
}

if (aliment == '1')
{
     Serial.print(F("CHERRY is BAD\n"));
}

if (aliment == '2')
{
     Serial.print(F("CHICKEN is GOOD\n"));
}

if (aliment == '3')
{
     Serial.print(F("COOKIE is GOOD\n"));
}

  //----------------------------------------

  // Serial.println(F("\n**End Reading**\n"));

  delay(1000); //change value if you want to read cards faster

  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
}
```

//***//

Program_10

```
#include "LedControl.h"

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         9          // Configurable, see typical pin layout above
#define SS_PIN          10         // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance
LedControl lc=LedControl(7,6,5,2);  // Pin and # of Displays

unsigned long delayTime=1500;  // Delay between Frames
int heart_stat=5;
int time_stat=0;

char menjar[15];

// Put values in arrays

byte heart5_icon[] =
{
  B01100110,
  B11111111,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart4_icon[] =
{
  B01100110,
  B10011001,
  B11111111,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart3_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B11111111,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart2_icon[] =
{
  B01100110,
```

```
  B10011001,
  B10000001,
  B10000001,
  B01111110,
  B00111100,
  B00011000,
  B00000000
};

byte heart1_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00111100,
  B00011000,
  B00000000
};


byte heart0_icon[] =
{
  B01100110,
  B10011001,
  B10000001,
  B10000001,
  B01000010,
  B00100100,
  B00011000,
  B00000000
};

// pollo
byte hf[8]=
{B00000000,B00001110,B00011110,B00011110,B0011110,B0011100,B01100000,B00100000}
;
// cirera
byte nf[8]={B00000000,
B00011000,B00011000,B00100100,B01000010,B11100111,B11100111,B00000000};
// galeta
byte sf[8]=
{B00111100,B01111110,B11111111,B01111111,B11111111,B11111111,B01111110,B001111
00};
// os
byte os[8]=
{B00001100,B00001100,B00001111,B00011111,B11111000,B11110000,B00111000,B001110
00};


void setup()
{
  Serial.begin(9600);
  SPI.begin();                                  // Init SPI bus
  mfrc522.PCD_Init();                                 // Init MFRC522 card
  // Serial.println(F("Read personal data on a MIFARE PICC:"));    //shows in serial that it is
ready to read

  lc.shutdown(0,false);  // Wake up displays
  lc.setIntensity(0,5);  // Set intensity levels
```

```
  lc.clearDisplay(0);  // Clear Displays


}


//  Take values in Arrays and Display them

// Display galeta
void cirera()
{
byte nf[8]={B00000000,
B00011000,B00011000,B00100100,B01000010,B11100111,B11100111,B00000000};
  lc.setColumn(0,0,nf[0]);
  lc.setColumn(0,1,nf[1]);
  lc.setColumn(0,2,nf[2]);
  lc.setColumn(0,3,nf[3]);
  lc.setColumn(0,4,nf[4]);
  lc.setColumn(0,5,nf[5]);
  lc.setColumn(0,6,nf[6]);
  lc.setColumn(0,7,nf[7]);
}

void galeta()
{
  byte sf[8]=
{B00111100,B01111110,B11011011,B11111111,B10101101,B11111111,B01110110,B001111
00};
  lc.setColumn(0,0,sf[0]);
  lc.setColumn(0,1,sf[1]);
  lc.setColumn(0,2,sf[2]);
  lc.setColumn(0,3,sf[3]);
  lc.setColumn(0,4,sf[4]);
  lc.setColumn(0,5,sf[5]);
  lc.setColumn(0,6,sf[6]);
  lc.setColumn(0,7,sf[7]);
}

void pollo()
{
  byte hf[8]=
{B00000000,B00001110,B00011110,B00011110,B0011110,B0011100,B01100000,B00100000}
;
  lc.setColumn(0,0,hf[0]);
  lc.setColumn(0,1,hf[1]);
  lc.setColumn(0,2,hf[2]);
  lc.setColumn(0,3,hf[3]);
  lc.setColumn(0,4,hf[4]);
  lc.setColumn(0,5,hf[5]);
  lc.setColumn(0,6,hf[6]);
  lc.setColumn(0,7,hf[7]);
}


void oset()
{
byte os[8]=
{B00001100,B00001100,B00001111,B00011111,B11111000,B11110000,B00111000,B001110
00};
  lc.setColumn(0,0,os[0]);
  lc.setColumn(0,1,os[1]);
```

```
  lc.setColumn(0,2,os[2]);
  lc.setColumn(0,3,os[3]);
  lc.setColumn(0,4,os[4]);
  lc.setColumn(0,5,os[5]);
  lc.setColumn(0,6,os[6]);
  lc.setColumn(0,7,os[7]);
}

void heart5()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart5_icon[i]);
  }
}

void heart4()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart4_icon[i]);
  }
}

void heart3()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart3_icon[i]);
  }
}

void heart2()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart2_icon[i]);
  }
}

void heart1()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart1_icon[i]);
  }
}

void heart0()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setColumn(0,i,heart0_icon[i]);
  }
}


void loop() {
  // put your main code here, to run repeatedly:
unsigned char ii;
```

```
// Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  //some variables we need
  byte block;
  byte len;
  int aliment = 11;
  MFRC522::StatusCode status;

  //-------------------------------------------

  // Reset the loop if no new card present on the sensor/reader. This saves the entire process
when idle.
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }



  //----------------------------------------- GET FOOD NAME

  byte buffer2[18];
  block = 1;
  len = 18;

  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
&(mfrc522.uid)); //line 834
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

  status = mfrc522.MIFARE_Read(block, buffer2, &len);
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

 Serial.print(F("Food: "));
// PRINT LAST NAME
  for (uint8_t i = 1; i < 16; i++) {
    Serial.write(buffer2[i] );
  }

aliment = buffer2[0];

if (aliment == '0')
{
      oset();
      delay(1000);
      heart_stat=5;
      Serial.print(F("BONE is GOOD\n"));
}
```

```
if (aliment == '1')
{
      cirera();
      delay(1000);
      heart_stat=0;
      Serial.print(F("CHERRY is BAD\n"));
}

if (aliment == '2')
{
      pollo();
      delay(1000);
      heart_stat=5;
      Serial.print(F("CHICKEN is GOOD\n"));
}

if (aliment == '3')
{
      galeta();
      delay(1000);
      heart_stat=5;
      Serial.print(F("COOKIE is GOOD\n"));
}

  //----------------------------------------

  // Serial.println(F("\n**End Reading**\n"));

  delay(1000); //change value if you want to read cards faster

  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();

//****************************************************************************************//


 if (heart_stat == 0)
 {
    heart0();
    heart_stat= 0;

 }

 if (heart_stat == 1)
 {
    heart1();
    heart_stat= 0;

 }

 if (heart_stat == 2)
 {
    heart2();
    heart_stat= 1;

 }

 if (heart_stat == 3)
 {
    heart3();
```

```
    heart_stat= 2;

 }
 if (heart_stat == 4)
 {
    heart4();
    heart_stat= 3;

 }

 if (heart_stat == 5)
 {
    heart5();
    heart_stat= 4;

 }

  delay(delayTime);


}
```