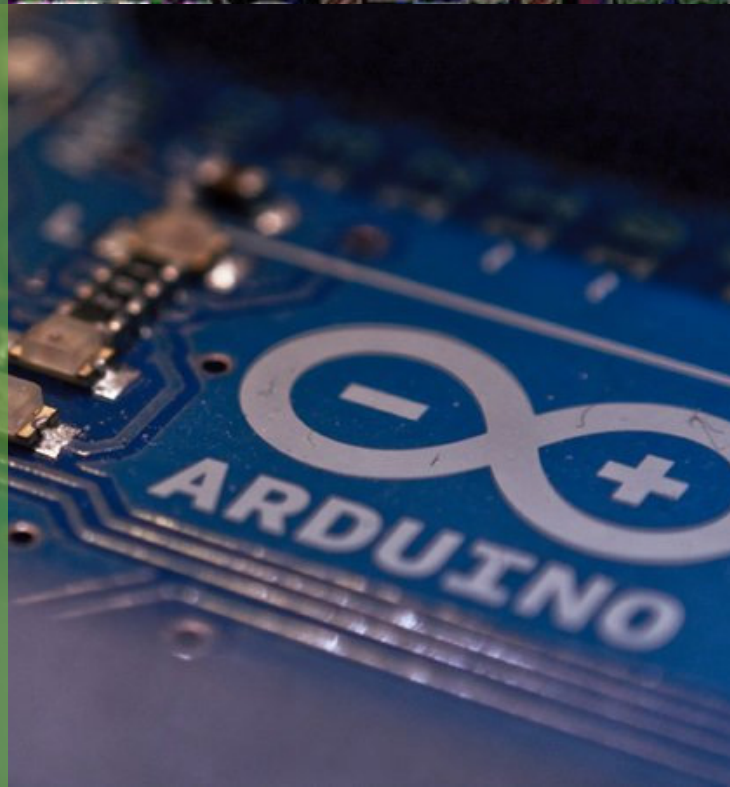


Charles Henry
Cabello

Project Technical Sheet



Domotic Piggy Bank

Project Technical Sheet
Course “Automazione Industriale”

By the student: Charles Henry Cabello

Milan, 2021 - 2022

INDEX

- Introduction.....5
- Components Used.....7
- Hardware Diagram.....9
- Code Explanation.....11
 - Libraries.....11
 - Void Setup Pinmode.....13
 - Loading Animation.....15
 - Door Open Condition.....17
 - Coin Animation.....19
 - Door Close Condition.....21
- Conclusion.....23

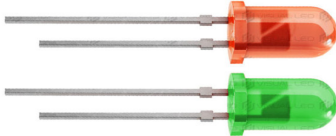



Introduction

In this technical report of my project, I will explain the functioning of my project, called the “Domotic Piggy Bank.” I will describe the components I used to build it and, finally, explain the operation of my code step by step. When my project is turned on, a loading animation will begin on the OLED display. At the end of the loading process, the message “INSERT COIN” will appear on the screen, and along with the message, the red LED will turn on. If I place my hand above the distance sensor, the servo motor will open a hole where coins can be inserted. As the hole opens, the red LED will turn off, the green LED will turn on, and an animation of a coin entering the hole will appear on the OLED display.

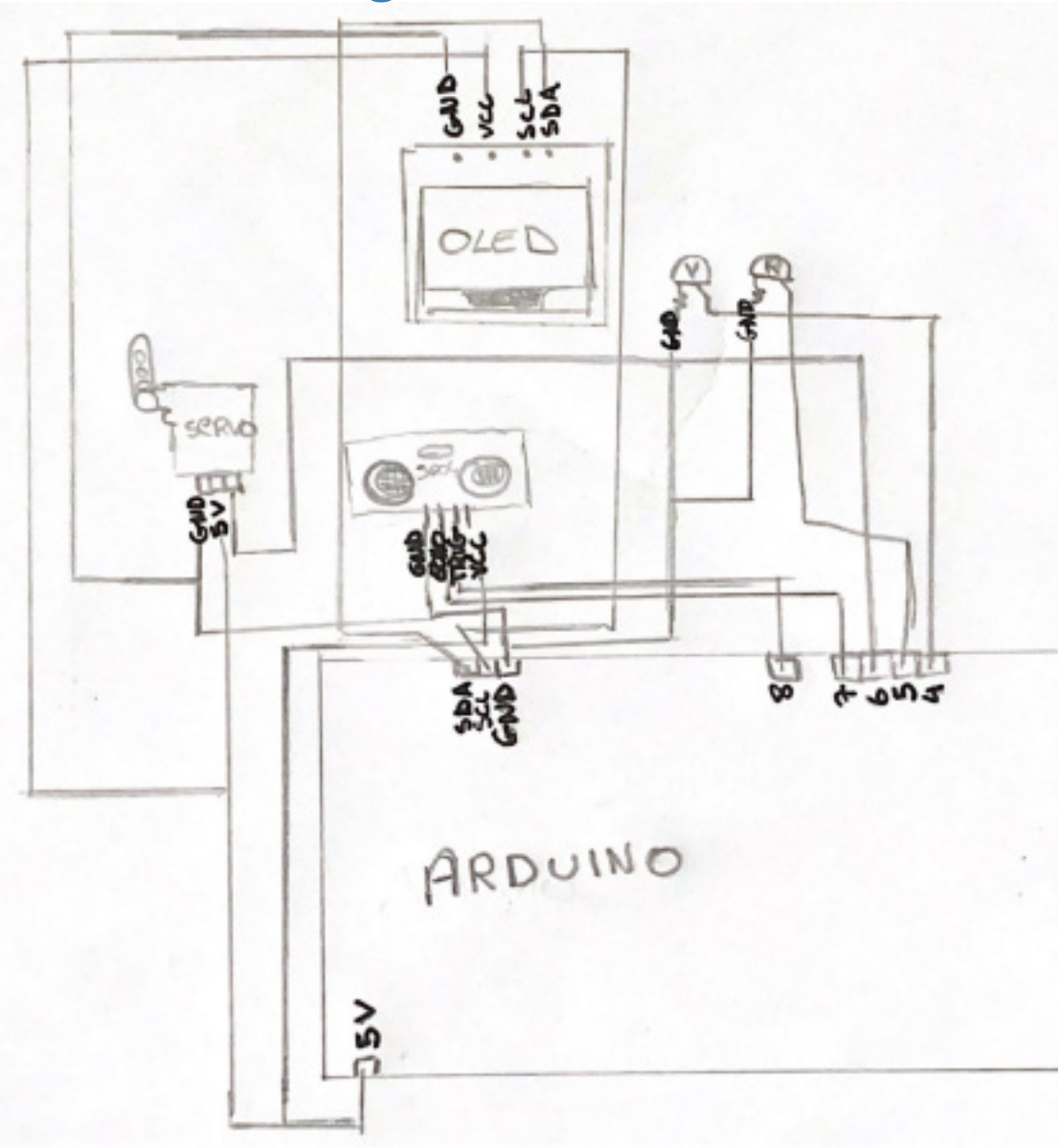


Components Used

The components I used for this project are:

	<p>2 Led: Those are components that produce light. They will be used to indicate to people when the door of the piggy bank is open or closed.</p>
	<p>1 display Oled: It's a component that allows us to display information for our projects. it will be used to show a loading screen with an animation and the message "INSERIRE LA MONETA"(Insert the coin) as well the coin animation</p>
	<p>1 Servo Motor: It is a rotating electric motor that allows precise control of the angular position. It will be used to move the door of the piggy bank</p>
	<p>1 Proximity Sensor: It is a sensor capable of detecting the presence of object nearby. It will be used so that when I place my hand near the sensor, it send a signal to the servo to open the door.</p>

Hardware Diagram





First test of the piggy bank

Code Explanation

Libraries

```
//:Oled;
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 display(128, 64,
&Wire, -1);
//:Servo;
#include <Servo.h>
Servo myservo;
//:Distanza;
#include "SR04.h"
SR04 sr04 = SR04(7, 8); //: Echo & Trig;
long valore;
```

Explanation

Before working on the void setup and the void loop, we first need to include the libraries for the OLED display, the servo motor, and the proximity sensor, which will allow us to use the commands for these components. If we do not include these libraries, the commands for our components will be treated as errors. Along with the libraries, we also create a variable called `valore`, which we will use later, and set the pins for the proximity sensor.



Second test of the piggy bank with the addition of OLED display

Explanation

After including the libraries, we move on to working with the void setup. In the void setup, we provide the board with all the necessary information before executing the program. Inside of the void setup, we include the pinMode commands, which configure specific pins and set them as INPUT or OUTPUT. The pinMode commands we include in the setup are for the red and green LEDs, which are connected to digital pins 5 and 4, and we define them as OUTPUT. Finally, we also include the pin for the servo motor and add a command to active the serial monitor, which we will use later.

Void Setup Pinmode

```
void setup() {  
  pinMode(5, OUTPUT); //:ROSSO;  
  pinMode(4, OUTPUT); //: VERDE;  
  //:Servo;  
  myservo.attach(6);  
  //:Monitor Seriale;  
  Serial.begin(9600);  
}
```



Third test of the piggy bank on wood

Loading Animation

```
//:Oled;

display.begin(SSD1306_
SWTCHCAPVCC, 0x3C);

display.clearDisplay();
display.display();
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(0, 25);
display.print("WELCOME!!!");
display.setCursor(0, 45);
display.setTextSize(1);
display.print("Made by Charles");
for (int x = 0; x < 128; x++) {
  display.drawPixel(x, 60, WHITE);
  display.display();
}
delay(500);
display.clearDisplay();
display.display();
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(25, 25);
display.print("Loading");
display.display();
display.setCursor(45, 45);
delay(1000);
```

```
display.print("$");
display.display();
delay(1000);
display.print("$");
display.display();
delay(1000);
display.print("$");
display.display();
display.clearDisplay();
display.display();
```

Explanation

Within the void setup, we also include the loading animation for the OLED display when the project is powered on. In the display code, we can also observe a “For” loop. The “For” loop is used to perform an operation or a series of operations a specific number of times. I used this “For” loop to create the loading bar animation.

Explanation

After finishing the work on the void setup, I begin working on the void loop. In the void loop, we include all the commands that are repeatedly executed in order, and once the code finishes, the void loop restart from the beginning. Inside the void loop, we use the variable we created earlier and set it equal to the value read from the proximity sensor. Then, we display this value on the serial monitor. after the proximity sensor detects a distance of less than 9 cm, the servo motor will move 40 degrees, the green LED will turn on, and the red LED will turn off.

Door Open Condition

```
void loop() {  
  //:Distanza;  
  valore = sr04.Distance();  
  Serial.println(valore);  
  if (valore < 9) {  
    myservo.write(40);  
    digitalWrite(4, HIGH);  
    digitalWrite(5, LOW);  
  }  
}
```


Coin Animation

```
//: 1° Parte;

display.clearDisplay();

display.fillCircle(65, 10, 20, WHITE);
display.fillCircle(65, 10, 15, BLACK);
display.setTextSize(3);
display.setTextColor(WHITE);
display.setCursor(58, 0);
display.print("$");

display.drawRoundRect(40, 50, 50, 9,
8, WHITE);

display.display();

//: 2° Parte;

delay(500);

display.clearDisplay();

display.fillCircle(65, 20, 20, WHITE);
display.fillCircle(65, 20, 15, BLACK);
display.setTextSize(3);
display.setTextColor(WHITE);
display.setCursor(58, 10);
display.print("$");

display.drawRoundRect(40, 50, 50, 9,
8, WHITE);

display.display();

//: 3° Parte;

delay(500);

display.clearDisplay();
display.fillCircle(65, 30, 20, WHITE);
```

```
display.fillCircle(65, 30, 15, BLACK);
display.setTextSize(3);
display.setTextColor(WHITE);
display.setCursor(58, 20);
display.print("$");

display.drawRoundRect(40, 50, 50, 9,
8, WHITE);

display.display();

//: 4° Parte;

delay(500);

display.clearDisplay();

display.fillCircle(65, 35, 20, WHITE);
display.fillCircle(65, 35, 15, BLACK);
display.setTextSize(3);
display.setTextColor(WHITE);
display.setCursor(58, 25);
display.print("$");

display.drawRoundRect(40, 50, 50, 9,
8, WHITE);

display.display();

delay(500);
}
```

Explanation

Inside the condition statement, we create two circles, one white and one black, stacked on top of each other to represent the coin. We continue drawing them, moving each circle by 10 pixel each time. This way, it will appear as though the coin is falling.

Door Close Condition

Explanation

At the end of everything, still inside the void loop, we include an else statement. If the distance detected by the proximity sensor is greater than 9 cm, the red LED will turn on, while the green LED will turn off. The servo motor will rotate back to 0 degrees, and finally, on the OLED display, the coin animation will no longer appear. Instead, it will be replaced with the message “INSERIRE MONETA”.

```
else {  
    myservo.write(0);  
    digitalWrite(5, HIGH);  
    digitalWrite(4, LOW);  
    display.clearDisplay();  
    display.setTextSize(2);  
    display.setTextColor(WHITE);  
    display.setCursor(15, 15);  
    display.print("INSERIRE");  
    display.setCursor(25, 35);  
    display.print("MONETA");  
    display.display();  
}  
}
```


Conclusion

I had a lot of fun creating this project. I didn't encounter any issue while writing the code, but I faced several challenges during the construction of my piggy bank. For example, when I had to create the holes to place the proximity sensor, I had to be careful. If I made the hole too small, the sensor wouldn't fit, and if I made it too large, the sensor wouldn't be positioned correctly. Another problem I encountered was attaching the components to the box of my project. Every time I secured them with adhesive tape, they would eventually come loose. Despite these challenges, I was very satisfied with the result of the project, and it gave me the opportunity to deepen my knowledge of the Arduino IDE.



Final Project with a smaller container and added LEDs