

Alexandra Salois & Véronique Pesant

Etude 4: Fall 2019

Professor Elio & Sabine

October 25th 2019

Etude 4: Game Controller

For this assignment, we decided to have our game controller used in a more physically active manner. In fact, paired with the game Veloce, we will attach the controller on your foot, and paired with it will be a capacitive sensor that will detect where you are kicking your leg, in order to control the dot on the screen and not get touched by the enemies. Equipped with this will be an active buzzer, and every 15 seconds, the buzzer will ring, and the controls of the game will be inverted: to go left, kick right, to go up, direct your foot down, and so on. This will allow the player to be physically and mentally agile, being able to coordinate the movements in real-time, and staying alert to the new controls changing constantly. This will make the game much more interesting and exciting for learning.

Given our little knowledge about electronics, circuits and sensors overall, we looked up our specific sensors to get a better awareness about its functions. The active buzzer, according to Bowling Green State University, requires only a battery or power from an Arduino board to work, although a 100Ω resistor is recommended, but not obligatory. It can turn on and off, to vary the sound output, and has a built-in oscillating source to produce noise. As for the capacitive sensor, it does not require physical contact to detect movement or closeness, by creating an electrical field and detecting the change in composition (Moermond). The one we have however, given its small size, does not have a big range.

Sources

2172 ARTC 4330/5330 CS. "Tutorial: Active Buzzer", Bowling Green State University, <https://bgsu.instructure.com/courses/1157282/pages/tutorial-active-buzzer>

Moermond, Jack. "What is a Capacitive Sensor?", Automation Insights, <https://automation-insights.blog/2017/06/07/what-is-a-capacitive-sensor/>

EXAMPLE OF CODE

TURN BUZZER ON / OFF:

```
int buzzerPin = 4;
```

```
// the setup function runs once
```

```
void setup() {
```

```
    // initialize digital pin 7 as an output.
```

```
    pinMode(buzzerPin, OUTPUT);
```

```
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
    digitalWrite(buzzerPin, HIGH); // turn the buzzer on (HIGH is the voltage level)
```

```
    delay(1000);                // wait for a second
```

```
    digitalWrite(buzzerPin, LOW); // turn the buzzer off by making the voltage LOW
```

```
    delay(1000);                // wait for a second
```

```
}
```

CAPACITIVE SENSOR?????

```
#include <CapacitiveSensor.h>
```

```
CapacitiveSensor cs_4_8 = CapacitiveSensor(4,8); // 1M resistor between pins 4 & 8,  
pin 8 is sensor pin, add a wire and or foil
```

```
void setup()
```

```
{
```

```
    cs_4_8.set_CS_Autocal_Millis(0xFFFFFFFF); // turn off autocalibrate on channel 1 -  
just as an example
```

```
    Serial.begin(9600);
```

```
    pinMode(7,OUTPUT);
```

```
}
```

```
void loop()
{
  long sensor1 = cs_4_8.capacitiveSensor(50);

  Serial.println(sensor1); // print sensor output
  if(sensor1 >= 1000)
  {
    digitalWrite(7,HIGH);
  }
  else{
    digitalWrite(7,LOW);
  }
}
```

<https://www.hackster.io/TheGadgetBoy/arduino-capacitive-sensor-in-less-than-2-minutes-1c0277>

https://www.pjrc.com/teensy/td_libs_CapacitiveSensor.html