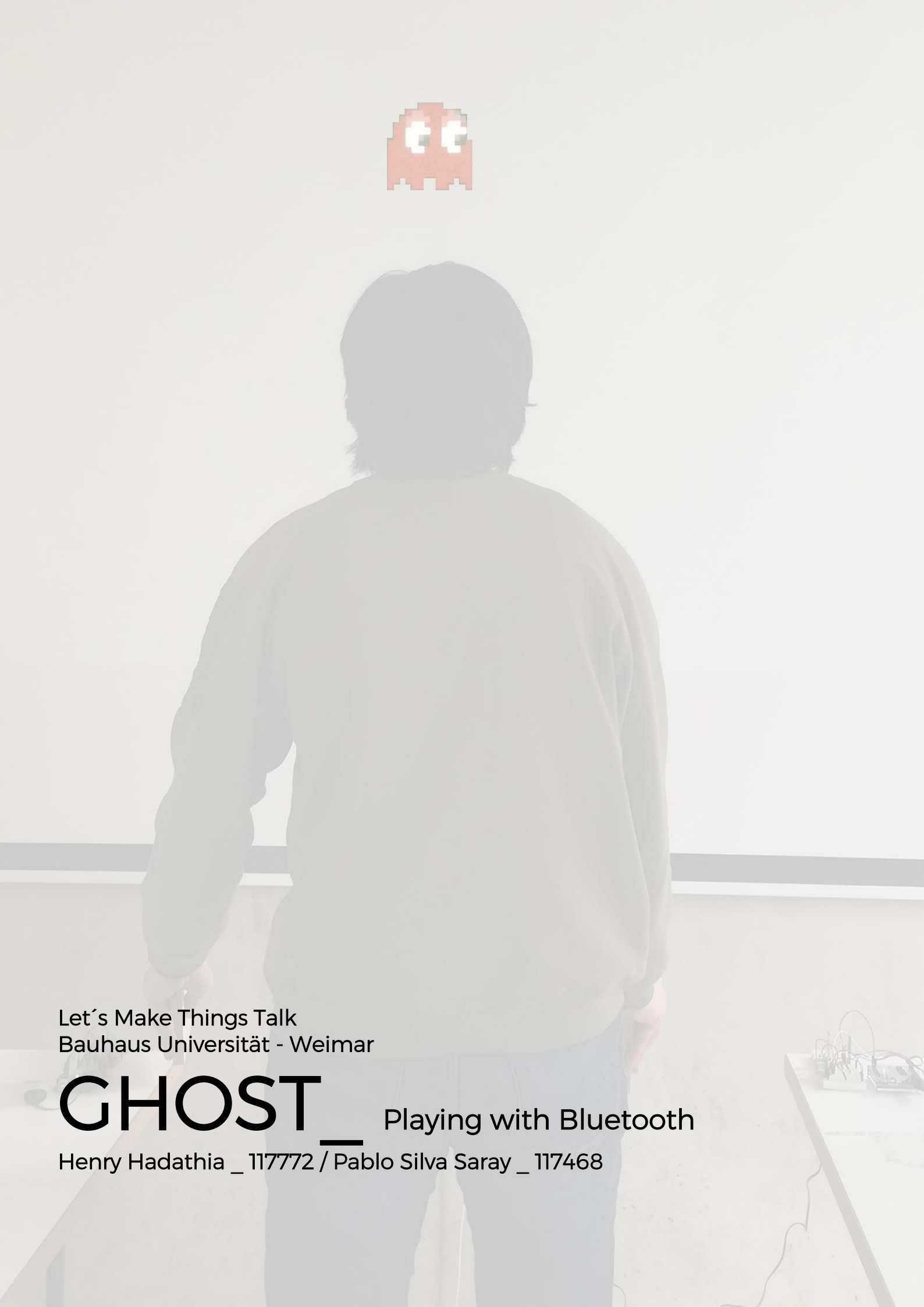


Let´s Make Things Talk
Bauhaus Universität - Weimar

GHOST_ Playing with Bluetooth

Henry Hadathia _ 117772 / Pablo Silva Saray _ 117468

GHOST_ Playing with Bluetooth

Henry Hadathia _ 117772 / Pablo Silva Saray _ 117468

https://www.youtube.com/watch?v=-9aHgxaNmR8&feature=youtu.be

Looking to understand and apply the use of Bluetooth systems, both software and hardware, we have applied what we have learned in class in a progression of examples. That in a didactic way have led us to the development interfaces that, although elementary, contain the potential to be developed in greater detail and purpose.

Test 001_ Henry Hadathia, Pablo Silva Saray

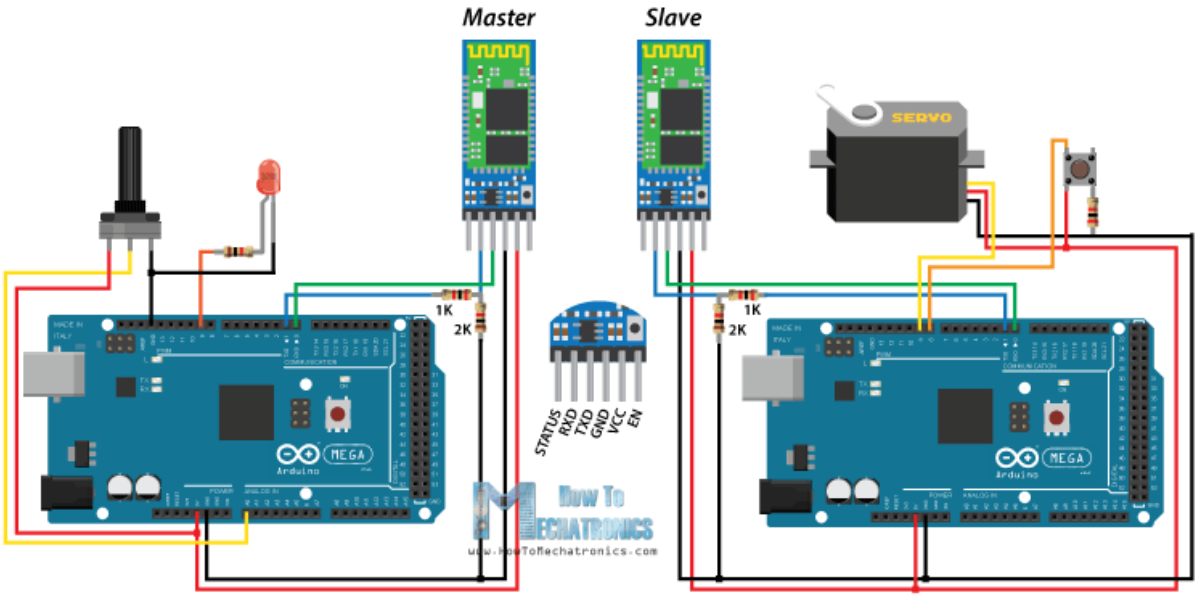
Example

As first approach to the task of connecting two Bluetooth devices, it was decided to try an example of configuration and pairing two HC-05 Bluetooth modules as master and slave.

Ref: *http://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/*

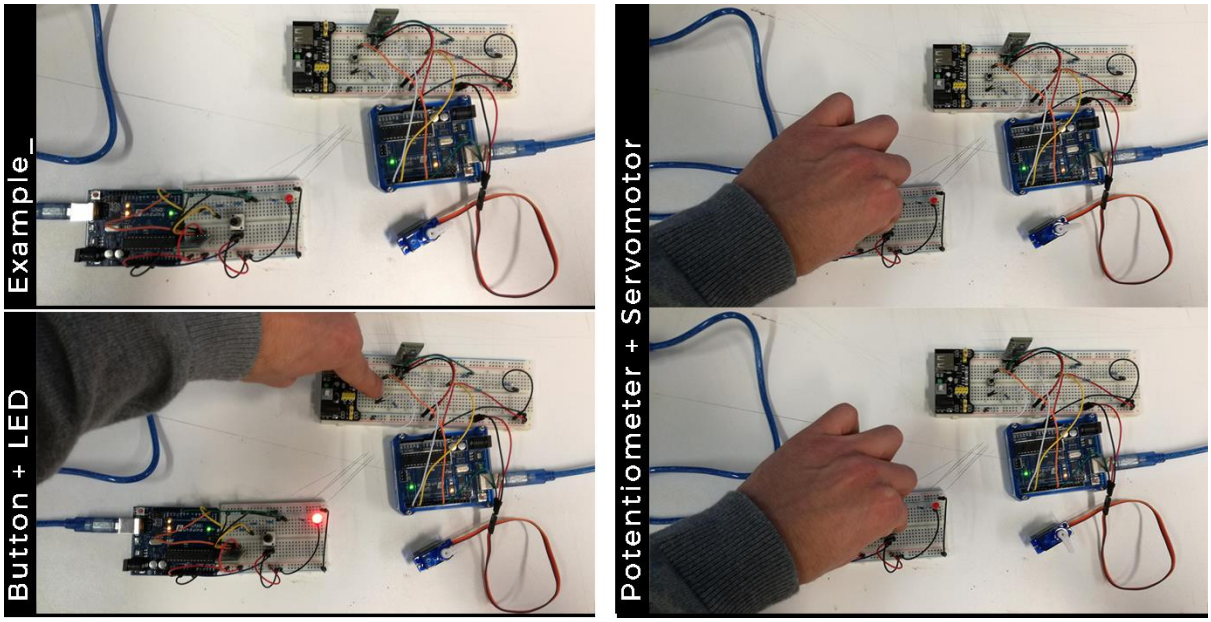
The example allow us to have a ground where to start to build the following programs. In this case two Bluetooth modules are configured as Master and Slave using the AT Command Mode of the modules.

Being this an example taken from internet is worthless to explain it in the present document. However, as result we understood some issues related with the software and hardware configuration, specially related with the use of sensors and actors as part of the program to develop.



Test 001 _ Setup

REF: http://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/



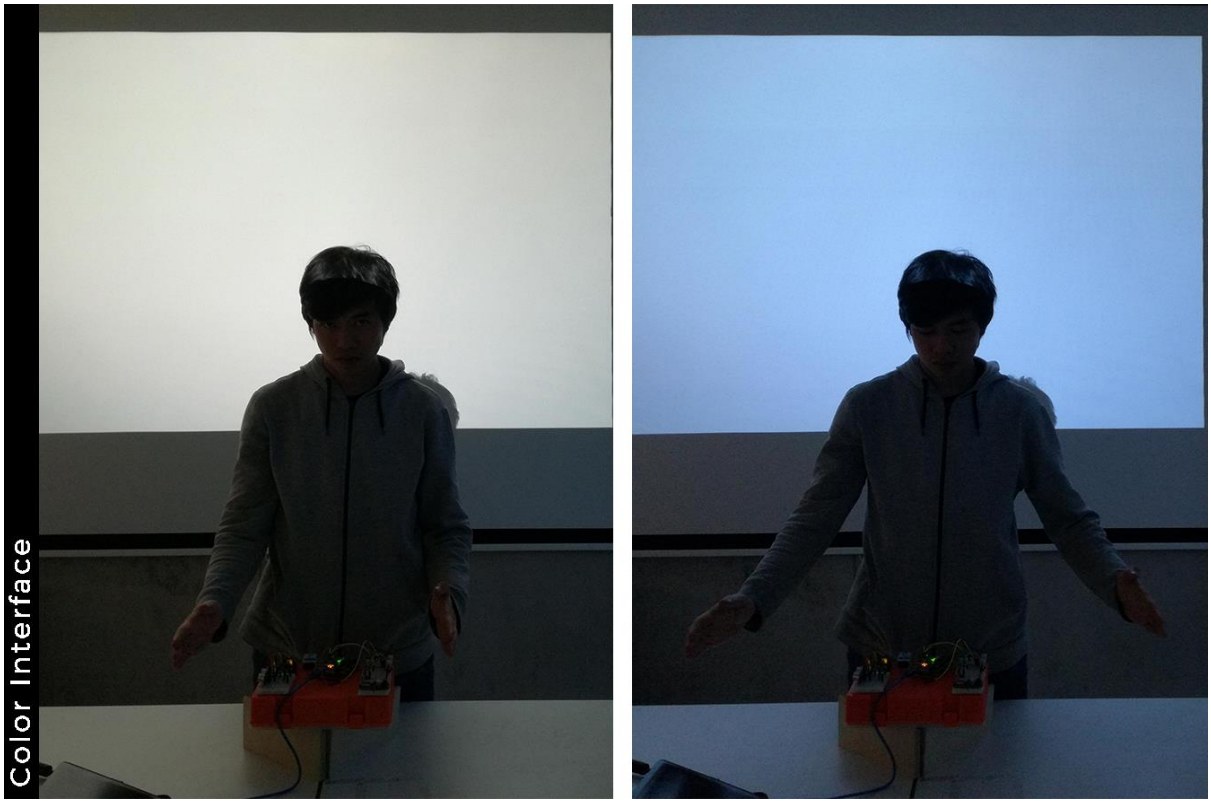
Test 001 _ Developed Example.

Test 002_ Henry Hadathia, Pablo Silva Saray

Color Interface:

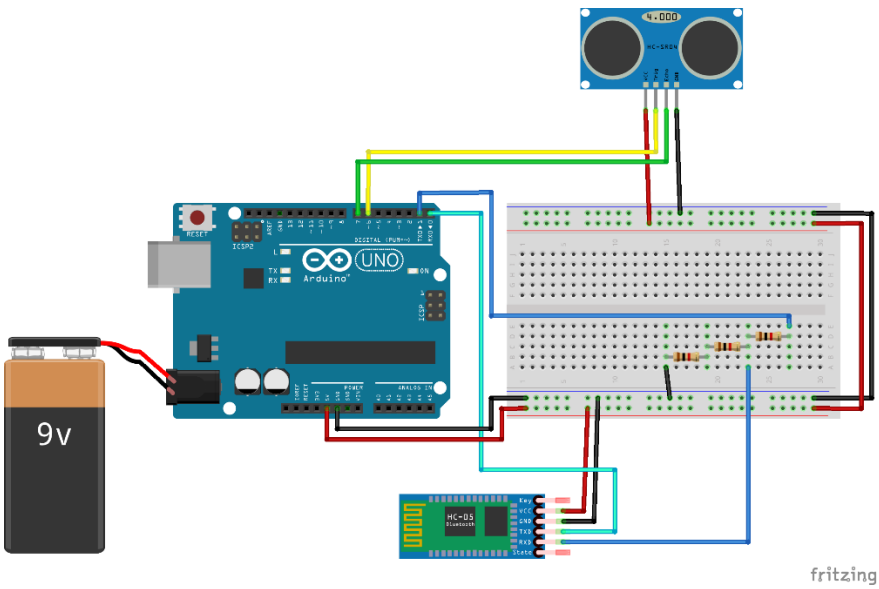
Interaction through Distance Sensors (HC-SR04 - Ultrasonic sensor)

Once a first test was done and the results were successful, we decided to work in an interface through distance sensors, where the movement of the user interacts with the system to modify the visualization on the screen.



For this it was necessary to set up two HC-SR04, Ultrasonic Sensors, working with an HC05 Bluetooth module each one. The data given by the first ultrasonic sensors is sent via Bluetooth to the second one, where is stored and collected along with the data given by the second sensor.

Thus, the complete data is sent **through serial cable** to processing from the second HC05 module, allowing us to create a visual interface, where the color hue is modified by the data received from the first sensor and the saturation by the data form the second one.



Test 002 _ Setup

Test 003 _ Setup

Arduino code _ Color Interface:

```
#define trigPin 6
#define echoPin 7

// BT communication
int state_1 = 0;
int state_2 = 0;

/*array value*/
float sensor_value[]={0, 1};
char* first[] = {"a", "b"};

void setup() {
 Serial.begin(9600); // Default communication rate of the Bluetooth module
 // Ultrasonic Sensor
   pinMode(trigPin, OUTPUT);
   pinMode(echoPin, INPUT);
}

void loop() {
  if(Serial.available() > 0){ // Checks whether data is comming from the serial port
    state_2 = Serial.read(); // Reads the data from the serial port
  }
  // Controlling the Ultrasonic Sensor
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  state_1 = distance;
  delay(50); //pause to let things settle
  store_data();
}

void store_data(){
  if (state_1 > 50){   state_1 = 50;  }
  if (state_2 > 50){   state_2 = 50; }
  //add result to array sensor_value
  sensor_value[0] = state_1;
  sensor_value[1] = state_2;

  for(int i = 0; i < 2; i++){
       Serial.print(first[i]);
       Serial.println(sensor_value[i]);
    }
}
```

Processing code _ Color Interface:

```
/*Variables Sensors*/
import processing.serial.*; //library for connecting processing to serial
Serial myPort; //assign myPort class
//float value;
String[] sensor_value = new String[6]; //make array for sensors' value

//sensor inputs
float state_1;
float state_2;

//variable for changing color
int hue;
int saturation;

void setup(){
  fullScreen();
  //size(displayWidth, displayHeight);
  background(255);

  /*Variables Sensors*/
  String portName="COM3"; //connection port of arduino, check it in arduino sketch
  myPort = new Serial(this, portName, 9600);
}

void draw(){
  colorMode(HSB, 360, 100, 100);
  manualTune();
  background(hue, saturation, 100);
}

void serialEvent()
{
  for(int i=0;i<6;i++)
  {
    //read string output from arduino
    sensor_value[i] = myPort.readStringUntil('\n'); // \n is line break

    if(sensor_value[i] != null)
    {
      //read first character and store it as data_type
      char data_type = sensor_value[i].charAt(0);

      //delete first charactar aka the letter a to f
      sensor_value[i] = sensor_value[i].substring(1,sensor_value[i].length()-1);

      //store each data type to the respective variable
      if(data_type == 'a')
      {
        state_1 = float(sensor_value[i]);
      }
```

```
      else if(data_type == 'b')
      {
        state_2 = float(sensor_value[i]);
      }
    }
  }
}

void manualTune()
{
  serialEvent(); //function for reading input values

  /*adjusting range, speed, and random circle from sound*/
  hue          = (int) map(state_1, 0, 50, 0, 360);
  saturation   = (int) map(state_2, 0, 50, 0, 100);
}
```

Test 003_ Henry Hadathia, Pablo Silva Saray

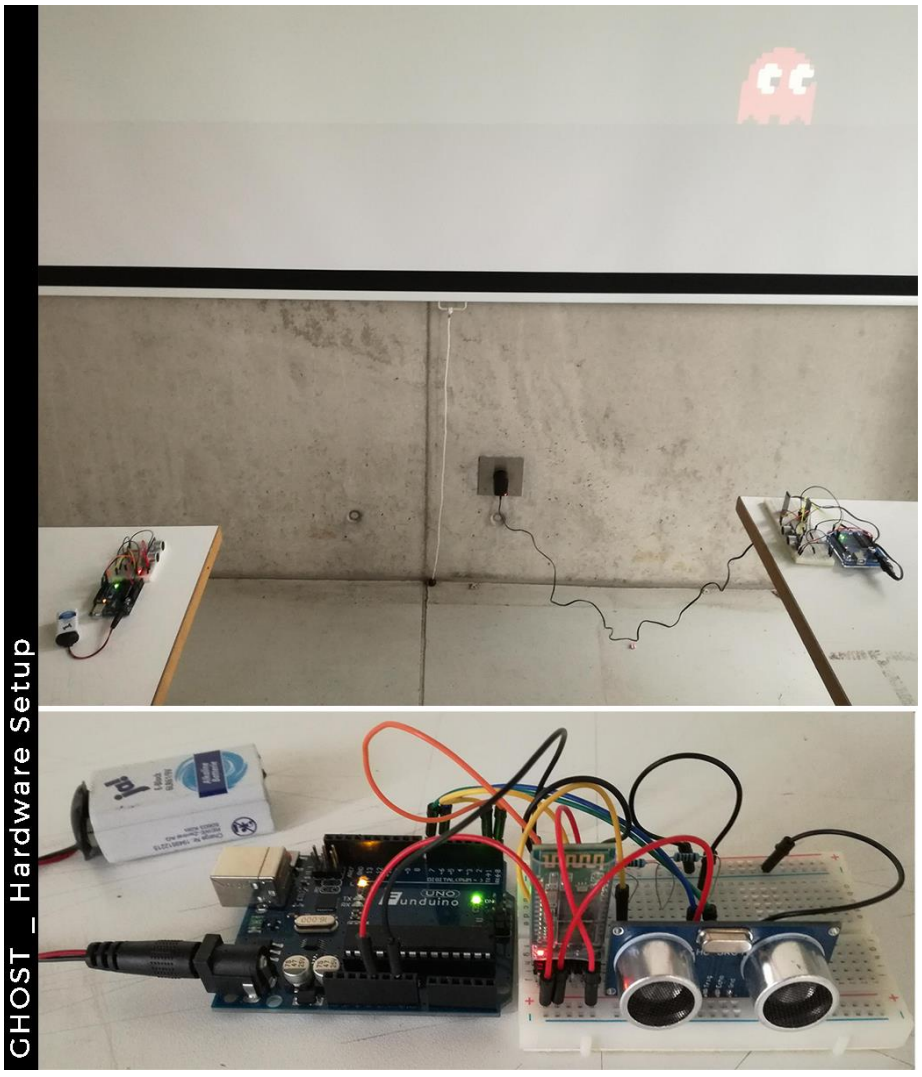
Ghost:

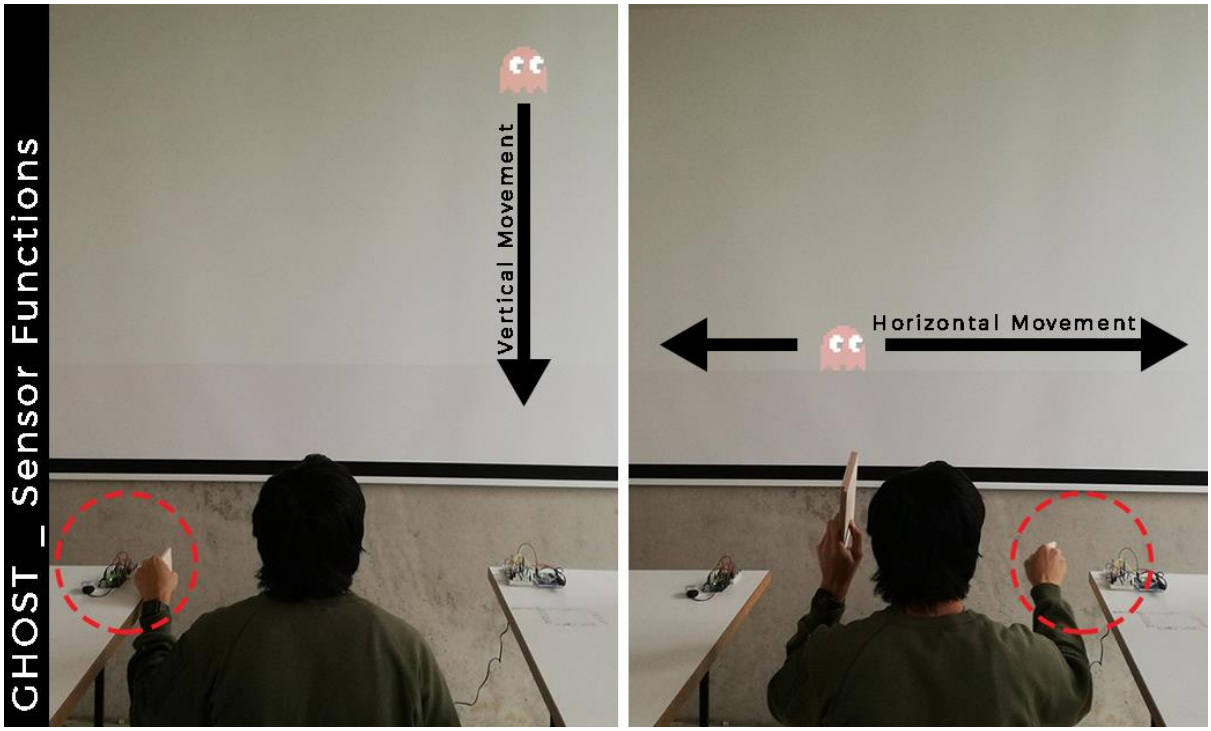
Interaction through Distance Sensors (HC-SR04 - Ultrasonic sensor)

Following the previous example, test 002, we decided to improve the interface of the visualization and more important to simplify the communication between the Bluetooth modules trying to avoid the need to use a wired connection to the computer and processing.

Although, the concept of Arduino program for the last example, test 002 is similar to this one, due to the use of the same hardware (HC-SR04 – Ultrasonic Sensor), the communication parameters were modified.
In this new code, the function void store_data, allow us to communicate directly to the computer, using only the Bluetooth modules.

Therefore, the designed interface is totally wireless, fulfilling the final goal of using Bluetooth in its full potential.





https://www.youtube.com/watch?v=-9aHgxaNmR8&feature=youtu.be

Arduino code _ Ghost _ Henry Hadathia

```
//ultrasonic pins
#define trigPin 6
#define echoPin 7

// for BT communication
int state = 0;
int sensor_value;

void setup() {
   Serial.begin(9600); // Default communication rate of the Bluetooth module


    // Ultrasonic Sensor
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop() {
  // Controlling the Ultrasonic Sensor
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  state = distance;
  delay(50); //pause to let things settle

  //process the data from ultrasonic sensor
  store_data();

  //just for checking the final result sent to computer
  //Serial.println(sensor_value);
}
void store_data(){
  // constraint the max value to 50
  if (state > 50){
    state = 50;
  }

  //add result to sensor_value
  sensor_value = state;

  //send the value through bluetooth to computer (processing)
  Serial.write(sensor_value);
}
```

Processing code _ Ghost

```
// setting for connecting bluetooth modules
import processing.serial.*;

Serial arduino1;
Serial arduino2;


// to store data from ultrasonic sensors
int val1, val2;


// variables for visualization
int player_xpos, player_ypos;
int alien_size = 300;
PImage img;

void setup()
{
  // screen mode of visualization (choose one of them)
  //size(1000,1000);
  fullScreen();

  // initial position of the ghost
  player_xpos = width/2;
  player_ypos = height/2;

  // the image's path (must be in folder "data")
  img = loadImage("pacman_ghost_v2.png");

  // setup for bluetooth connection
  printArray(Serial.list()); // show the list of available ports
  String arduino1Name = Serial.list()[1]; //change the number in the bracket to select
which port for each bluetooth
  arduino1 = new Serial(this, arduino1Name, 9600);
  String arduino2Name = Serial.list()[3]; //change the number in the bracket to select
which port for each bluetooth
  arduino2 = new Serial(this, arduino2Name, 9600);

}

void draw()
{
  background(0); //set black background
  image(img, player_xpos, player_ypos, alien_size,alien_size); //draw the image on
the screen
  player_position(); //reading the input data from bluetooth and process it

  // just to check the value of each input in console
  print('a');println(val1);
```

```
  print('b');println(val2);
}

void player_position()
{
  // store the input data to variables
  if ( arduino1.available() > 0) {
    val1 = arduino1.read();
  }
  if ( arduino2.available() > 0) {
    val2 = arduino2.read();
  }

  // remap the range data from 50 to the width and height of screen
  player_xpos    = (int) map(val1, 0, 50, 0, width-alien_size);
  player_ypos    = (int) map(val2, 0, 50, 0, height-alien_size);
}
```