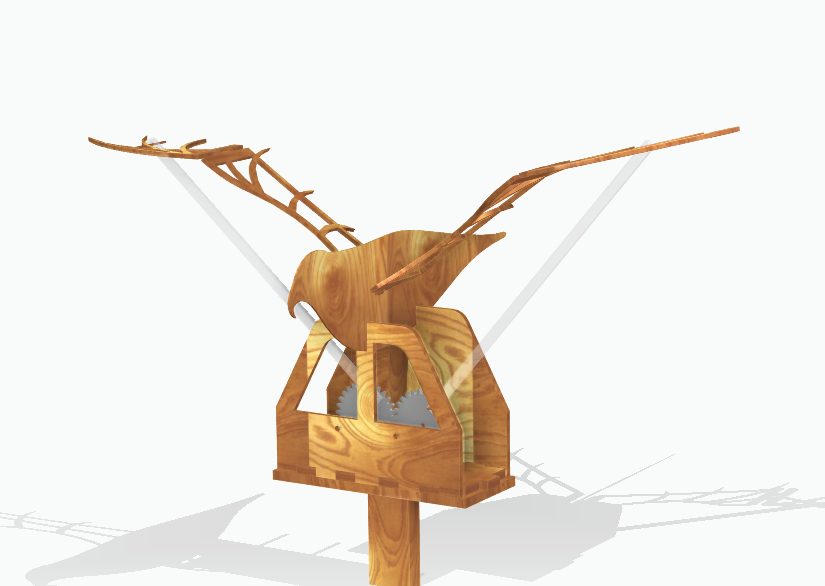
****

Gesture controlled Kinetic Art

Group 3

MF 2104 | 15 Dec 2022

Submitted by

Hala Mawas

Rimmie Duraisamy

Yawen Deng

Atocha Anaya

Table of Contents

[Introduction 2](#_Toc121927203)

[Background 2](#_Toc121927204)

[Motivation 2](#_Toc121927205)

[Scope 3](#_Toc121927206)

[Concept Realization 3](#_Toc121927207)

[Pulse Width Modulation (PWM) 3](#_Toc121927208)

[Transformation of Electric Signals to Movement 4](#_Toc121927209)

[Processing 4](#_Toc121927210)

[Methodology 5](#_Toc121927211)

[Design and Prototype 5](#_Toc121927212)

[Electronic System Design 5](#_Toc121927213)

[Hardware Design 5](#_Toc121927214)

[Movement Control Circuit 6](#_Toc121927215)

[IMU Sensor 6](#_Toc121927216)

[Software Design 7](#_Toc121927217)

[Results 9](#_Toc121927218)

[Discussion 9](#_Toc121927219)

[References 10](#_Toc121927220)

# Introduction

Gesture control technology is developing quickly and changing many aspects of our life. These devices are used in a much wider range, from research experiments and prototypes to day-to-day commercial products (Birmingham).

In this project, an attempt has been made to develop and demonstrate the prototype of a gesture-controlled kinetic art.

## Background

In the current scenario, with the rising popularity of ubiquitous and ambient devices, (i.e. PlayStations), and equipment, which allows users to grasp virtual objects, hand or body gesture are becoming essential: gesture controls have become central to the human computer interaction system (Birmingham). Gesture control and touchless user interfaces both provide the ability to interact with devices without physically touching them. Gesture control can be used in various amazing applications in automotive industry, entertainment and consumer electronics sector, healthcare sector, assisting people with disabilities or education sector (IMAGIMOB, u.d.).

The desktop computing paradigm limits the users' flexibility by forcing them to interact using a 2-Degree-Of-Freedom device (the mouse), while gestures allow the user to handle multiple points of input and even define several parameters at once. They are, therefore, a more natural form of communication (Attwenger). Unlike traditional buttons and menus, gesture control does not require any additional devices, the command and the parameters can be specified using simple movements. The possibility of interaction is narrowed down by the input device like a mouse while, gestures are versatile and offer a range without any constraints. The user-interaction with gesture control devices is more intuitive and enjoyable as they mirror experiences of the real world (Attwenger).

Even though gesture control devices offer a wide range of possibilities, it opens up difficulties in learning and remembering executable gestures on the user’s end. The developer must take into account the lead-time required by the device to rapidly and correctly recognize the gestures (Attwenger). Gesture control devices also struggle with accuracy and the need for more processing power.

## Motivation

This project is inspired by the concept of moving and interacting arts. Interactive art responds to the actions of its interactants. Such works are typically either visual or sonic in nature, involve digital technology, and respond to the movements, sounds, or input (via a computer interface) of the interactant. This creates a bidirectional flow of information between the interactant and the work. The interactant's actions are, therefore, an integral part of interactive art. It examines four properties: (1) the number of controllable parameters in the interaction; (2) the use of fantasy in the work; (3) the timescale on which the work responds; and (4) the amount agency ascribed to the work (Michael Krzyzaniak, 2022). An interactive art features interfaces and sensors to respond to motion, sound, heat or other changes as type of input to respond to by the interactant.

For this project, an example of kinetic origami is taken as a reference (VietBui). Based on this example, a prototype of a gesture controlled kinetic bird is chosen.

## Scope

A model of the kinetic bird in designed using Solid Edge. Many iterations were made to ensure the structural integrity of the bird design. A prototype of the design is then developed using 3D printers, laser cutting and other means of infrastructure.

The goal of this project is to control the flight of the bird with hand gestures. An attempt has been made to realize this gesture control as wireless using Bluetooth/ nRF modules. More detailed description of the involved hardware and software design follows in the next chapter of the report.

## Concept Realization

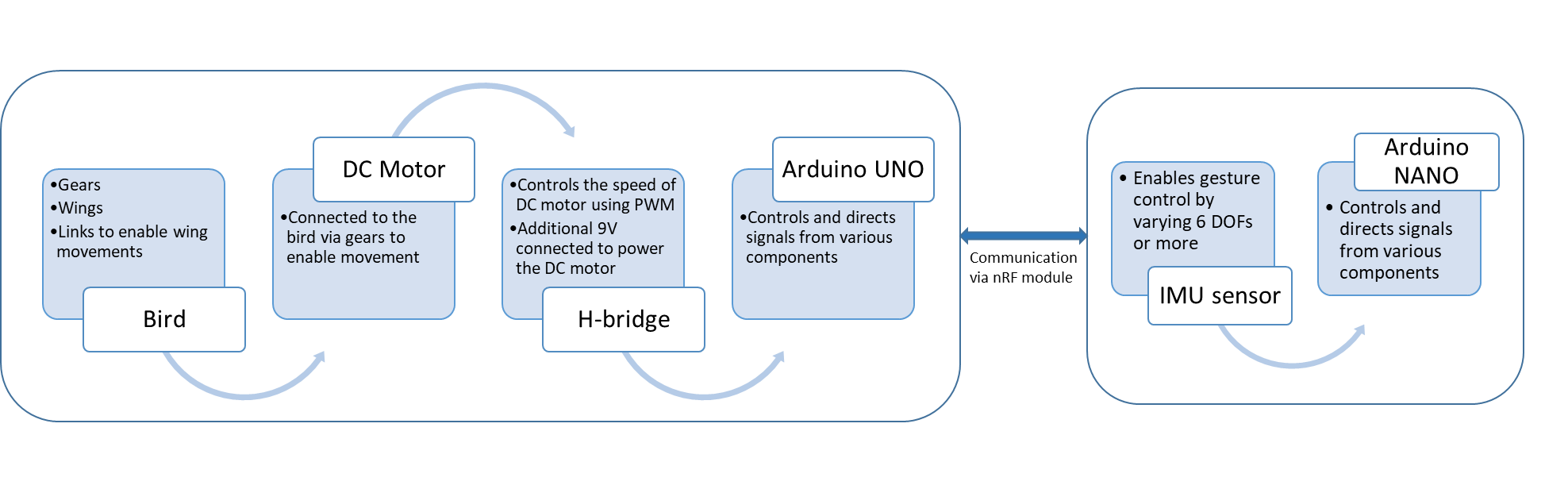
In order to control the movement of the bird, a DC motor is used. The speed of the motor is controlled via H-bridge L298N using PWM (Pulse Width Modulation). To include gesture control, an IMU (Inertial Measurement Unit) sensor is connected to the Arduino board. A Bluetooth/ nRF module is used to make the gesture control wireless. Figure 1 shows the block diagram of the systems involved to realize this concept. In addition to this, to give an effect of that the bird is flying under a night sky and make the experience even further interactive, a full moon is also controlled using the IMU sensor via Processing. 

Figure 1: Schematic of the entire system

### Pulse Width Modulation (PWM)

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between the full input voltage of the board (e.g., 5 V on Arduino UNO) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is referred to as the pulse width. To get varying analog values, you change, or modulate, that pulse width (Hirzel). The PWM is characterized by a parameter referred to as ‘duty cycle’. The width of the square wave determines the percentage of the duty cycle. Figure 2 below shows the PWM for different duty cycles. The ‘analogWrite(xxx)’ converts the digital value ‘xxx’ to its corresponding analog value. Here, with reference to Arduino UNO, 0 is OFF and 255 is ON at full voltage.

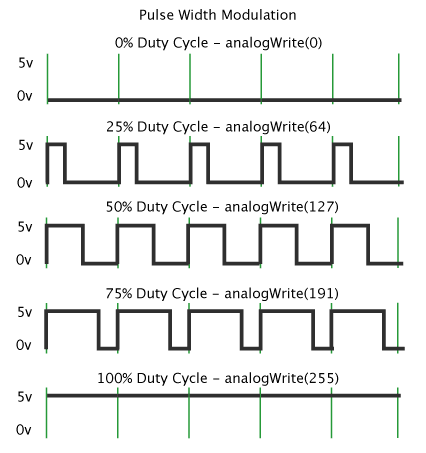


Figure 2: Pulse Width Modulation Duty Cycle (Hirzel)

#### PWM and H-bridge: Controlling speed of a DC motor

The speed of the motor can be easily controlled by varying the resistance and changing the current passing through the motor as shown in Figure 3(a). While this works, a lot of heat is generated and power is wasted in the resistance. Hence, PWM is a better way to control the voltage across the motor thereby changing its speed (Pulse Width Modulation, u.d.). To control a DC motor, PWM is often used in conjunction with an H-bridge as shown in Figure 3(b).

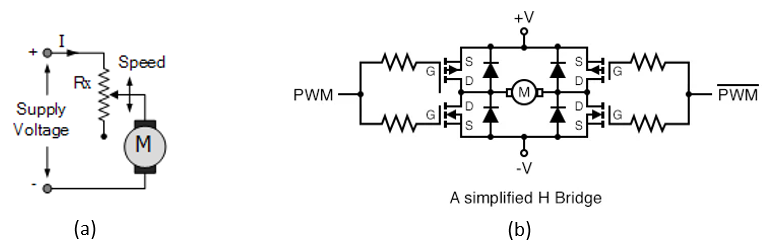


Figure 3: (a) Controlling speed of a motor (Pulse Width Modulation, u.d.); (b) Schematic of H-bridge (DC Motor Drives)

### Transformation of Electric Signals to Movement

### Processing

# Methodology

## Design and Prototype

## Electronic System Design

The movement of the bird, visual and audio effects are, in real time, controlled by human gestures. Specifically, this involves controlling both in the physical world and in the virtual world. To achieve this, the electronic system also consists of two parts, the software system and the hardware system, shown in Figure 3.

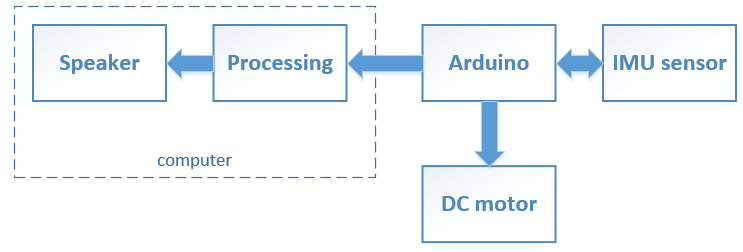


Figure 4: Block diagram of the electronic system

The software system includes Processing and its related virtual serial ports, and the speaker. In this project, a laptop speaker is utilized, which can be replaced by a bluetooth speaker as well. The main function of the software system is to control the visual and audio effects according to the gesture-related signals sent from Arduino.

The hardware system includes an Arduino, an IMU sensor and a DC motor. The IMU sensor is the physical input of the whole system, and the motor, including its drive, is one of the outputs. The main function of the hardware system is to receive human gestures and adjust the movement of the mechanical structure accordingly.

In this subsection, we will explain both the hardware development and the software development and motivate the reasons behind.

### Hardware Design

The hardware system needs to interact with the physical world and communicate with the virtual world. To be able to detect human gestures, an IMU sensor, specifically MPU 6050, is chosen for the circuits. And to be able to move the wings of the bird, a DC motor is utilized. The reason why we typically select a DC motor rather than a servo motor is that the DC motor can easily turn 360°, while a servo motor usually turns 180° only. This choice would make the flapping of the wings smoother and more aesthetically pleasant.

Table 1 is a list of main components actually used in the circuits. Wires including the jumper wires and USB cables are not included.

Table 1: List of main components

|  |  |  |
| --- | --- | --- |
| Component | Model | Function |
| Arduino | Arduino UNO | control the motor and the sensor, communicate with Processing |
| DC motor | DG01D-L | move the wings at different speed |
| DC motor drive | L298N | control the speed and direction of the motor; separate motor with Arduino to protect the circuit |
| IMU sensor | MPU6050 | receive input from different gestures |

### Movement Control Circuit

Figure 2 is the circuit diagram of the DC motor, its drive and Arduino.  Arduino powers the drive and sends input signals for speed and direction control; and the drive generates PWM (Pulse Width Modulation) signal to actually control the speed and direction of the motor.

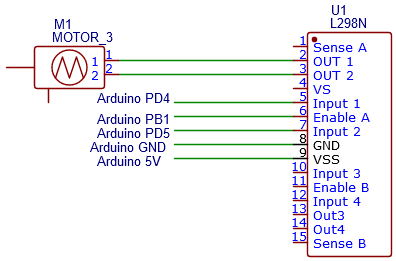


Figure 5: The movement control circuit

There are two things we should pay attention to when implementing this circuit. First, the stall current of a 4.5V 48:1 DC motor may reach 1A, which is dangerous for the chip of the drive and Arduino. Though the L298N model has a heat sink, it’s essential to cut off the power when the motor gets blocked. Second, the cap of the pins in L298N should be taken off, if we want to adjust the speed and direction via Arduino. For instance, the motor would always work at the maximum speed if the cap of Enable A is on.

The direction of the motor can be adjusted via Input 1 and Input 2. In this project, we set Input 1 to be digital HIGH and Input 2 to be digital HIGH. The speed input for the Enable A pin can be adjusted via Arduino. In the program, the speed should be in an integer, between 0-255. This controls the duty cycle of PWM, in which 0 is the slowest speed (not spinning) and 255 is the highest.

### IMU Sensor

Figure 3 shows the circuit diagram of the IMU sensor and its connection to the Arduino board. In this project, an IMU sensor with six DOFs (Degrees of Freedom) is used.

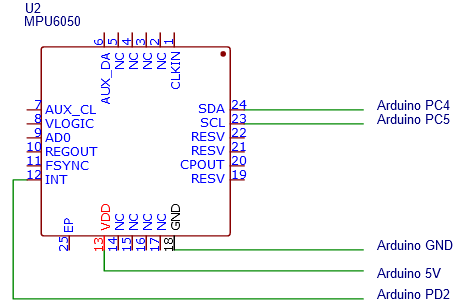


Figure 6: Circuit diagram of the IMU sensor and its connection to the Arduino board

## Software Design

The software system is in charge of the virtual world and the input from the physical world. It takes the gesture input from the IMU sensor and reacts with visual and audio effects. In particular, we set the interaction to be: the faster the hand turns, the bigger the moon behind the bird is, and the higher volume the music plays, and vice versa. To achieve this, Processing is utilized to be the software environment. Because it’s an integrated environment especially designed to realize visual and audio arts, and can easily manage the communication between Arduino.

The Processing used in running the final prototype is Processing 4.1.1 (Windows) and the programming language is Java. Libraries imported in the program are processing.serial and processing.sound. They are imported to communicate with Arduino via the serial port and to manipulate music playing respectively.

Figure 4 is the flow diagram of the program. Similar to Arduino IDE, Processing also has the form of first setting up the parameters, then running the loop program.

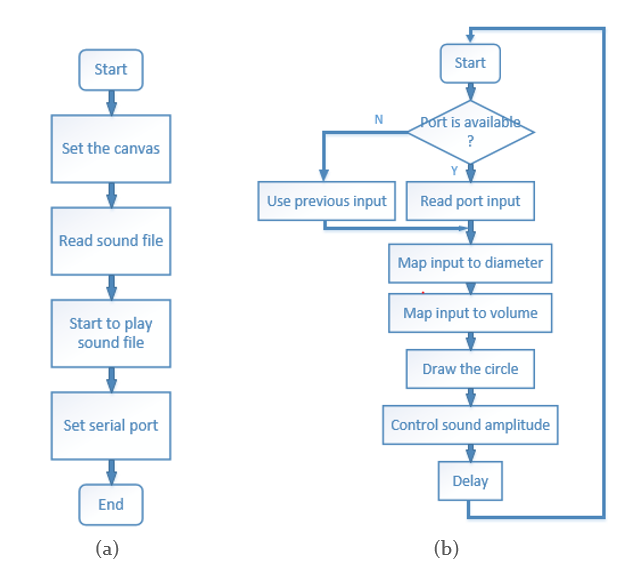


Figure 7: The flow diagram of the program. (a) The flow chart of setup(). (b) The flow chart of the draw()

In the setup() function, the size and background of the canvas are set. And the sound file is read from the folder and played in a loop mode. Then the serial port is set to the same serial port that Arduino occupies, with the same baud rate as Arduino, which is 9600b/s here.

The loop is called draw() in Processing. In the loop, we first check if the value in the serial port is ready. If it’s ready, Processing will read the new value; if not, Processing will keep the latest value. Then we map the input value to the diameter of the circle and the volume separately. The mapping functions are designed to fit a reasonable and aesthetical diameter of the circle in the background; and to fit the parameter of sound.amplitude, which is a float between 0 and 1. After calculating the diameter and amplitude, we draw the circle and adjust the sound. After a delay, the loop starts from the beginning again.

A delay is set, due to the pleasantness of the visual effect. Without the delay, the circle may flash big and small rapidly, which influences the user experience. Since Processing reads the datastream slower than the speed of Arduino sending, the main loop of arduino has to include a delay as well. When setting the delay value, there is a trade-off between the reaction time of Arduino to IMU sensor and the real-time of the data stream transmission. If the delay is long in Arduino, Processing would be able to keep pace with the newest data; but the motor would not be able to react to the change of gestures fast enough. And if the delay is short in Arduino, the motor would be able to react fast; but Processing would not be able to read the latest data from Arduino, with the delay between the physical world and virtual world accumulating. After testing, we set the delay of Processing to be 100ms, and that of Arduino to be 500ms, which works for most of the time.

# Results

a list of finished functions, non-finished function

What's the formula that maps from imu input to the speed of the motor, does it work well?

What's the formula that maps from imu input to the visual features (the diameter of the moon?) Does it work well?

What's the formula that maps from imu input to the audio features, does it work well?

# Discussion

In terms of telecom algorithms, the delay solution between Arduino and Processing can be replaced by better algorithms. As mentioned in Methods, the delay makes it a trade-off between the reaction speed of the gesture input and the synchronization between Arduino and Processing. In fact, this problem could be optimized by two methods, customized downsampling and smoothing algorithms.

In the downsampling method, the data streaming is only triggered when the IMU sensor witnesses a turbulence of value. In this way, the signal is short-time stationary; and Processing has less data to read. And without the delay, the motor can keep pace with the IMU sensor.

Among the smoothing algorithms, Mean Filtering is one of the easiest and most classic. The principle is to replace the value sampled at t0, with the average of the values sampled between t0-τ and t0+τ. So pulses in the signal are reduced, and the signal becomes smoother. In this way, there is no delay in Processing needed, and a little delay in Arduino, making sure of the synchronization. And both smoothing and downsampling can improve the performance of the electronic system.

# References

Attwenger, A. (n.d.). *Advantages and Drawbacks of Gesture-based Interaction.* Retrieved from https://www.grin.com/document/369514

Birmingham, U. o. (n.d.). *Gesture Control Technology: an investigation on the potential use in Higher Education.* Retrieved from https://intranet.birmingham.ac.uk/it/innovation/documents/public/Gesture-Control-Technology.pdf

*DC Motor Drives.* (n.d.). Retrieved from All about circuits: https://www.allaboutcircuits.com/textbook/semiconductors/chpt-11/pulse-width-modulation/

Hirzel, T. (n.d.). *Basics of PWM (Pulse Width Modulation).* Retrieved from Arduino.cc: https://docs.arduino.cc/learn/microcontrollers/analog-output

IMAGIMOB. (n.d.). *The Future is Touchless: Radical Gesture Control Powered by Radar and Edge AI*. Retrieved from https://www.imagimob.com/blog/the-future-is-touchless-radical-gesture-control-powered-by-radar-and-edge-ai#:~:text=Gesture%20control%20devices%20have%20the,system%20without%20direct%20physical%20contact.

Michael Krzyzaniak, Ç. E. (2022). What Makes Interactive Art Engaging? *Frontiers in Computer Science*.

*Pulse Width Modulation*. (n.d.). Retrieved from Electronics-Tutorials: https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html

VietBui. (n.d.). *Origami Interactive Kinetic Art.* Retrieved from Instructables: https://www.instructables.com/Origami-Interactive-Kinetic-Art-parallel-Processin/