Master Jump

Weixuan Zhang

Contents

1 Introduction

Jump is a game on mobile phones, in which one controls a chess piece to jump between platforms. The longer the screen is pressed, the further the piece jumps. If the piece falls off a platform, game overs.

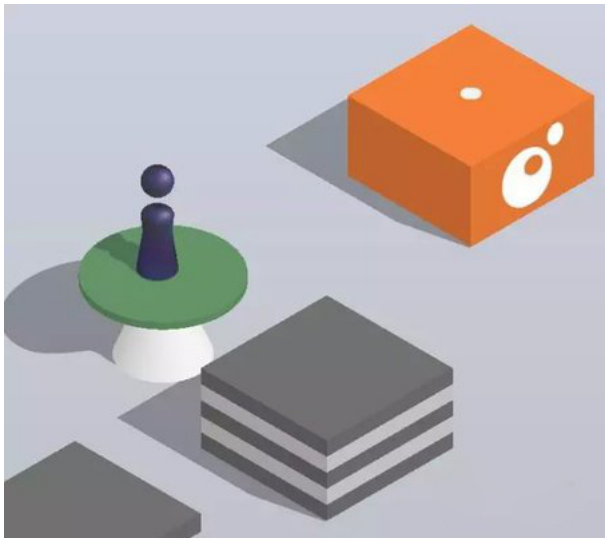


Figure 1: A screenshot taken in *Jump*

The goal of this project is to master *Jump* by mimicking finger pressing the touch-screen for a certain amount of time. The time can be calculated automatically from the input, which is generated by a potentiometer placed on the screen measuring the distance between two platforms.

2 Specification

- Operates via a 9V battery
- Reads an analogue input from a potentiometer
- Has both qualitative and quantitative outputs
- Controls a capacitative touch-screen

3 Code

3.1 Mode of Operation

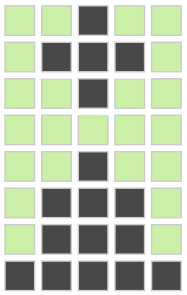
Please also see the comments on the code, this section only discusses several important points. The code has changed in the course of the building of the circuit, alteration mentioned in section 3.1.4 was carried out after running the code on an actual Arduino. Voltage-time formula is section 3.1.3 was changed after the whole circuit was completed, discussed in section 5.1.

3.1.1 Animation

There is a a three-step animation on the start-up screen, in which the chess piece in *Jump* jumps across the LCD screen. To make the animation look more natural, many frames are drawn. Each frame is a custom character, whose appearance is specified by an array of eight bytes, one for each row. The five least significant bits of each byte determine the pixels in that row [1]. 1 for turning on a pixel, 0 for setting a pixel to off.
For example

```
byte customChar[8] = {
  0b00100,
  0b01110,
  0b00100,
  0b00000,
  0b00100,
  0b01110,
  0b01110,
  0b11111
};
```

(a) Code for defining a character



(b) Character produced

(c) The chess piece in Jump

Figure 2: An example of defining custom characters

The character in the animation is the chess piece in *Jump*, shown in Figure 2c.

3.1.2 Analogue Reading

The voltage from the potentiometer is an analogue input, which is converted into a digital value between 0 and 1023 by the analogue-to-digital converter inside the microcontroller. The function `voltageCalc` converts the voltage back. As the potentiometer is connected to the 5V pin, the voltage is converted from 0-5V to 0-1023. In order to convert it back, the coefficient should be $\frac{5}{1023}$.

3.1.3 Voltage-time Equation

Since the time-controlling of the output is achieved using the `delay` command, whose input must be in milliseconds, the formula's output needs to be in milliseconds.
As the circuit is not built at this time, the voltage-time equation cannot be determined, so for simplicity, the equation is set to be a linear function with a reasonably large coefficient in front of the variable (i.e. voltage) making the change in time clearly noticeable while testing.
The equation at this stage is set to be

$$t(V) = 500 \times V$$

The equation is coded as a function called `timeCalc`, so can be easily changed when the actual one is determined and called later in the code.

3.1.4 Switches

The `INPUT_PULLUP` parameter is added in the `pinMode` command.
As mentioned in the comments on the code, this prevents "floating", where when the switch is open (i.e. not pressed), the pin is connected to nothing. As a result of "floating", the input can be affected by noises, leading to unexpected behaviour of the switch. The `INPUT_PULLUP` parameter physically connects a built-in resistor.

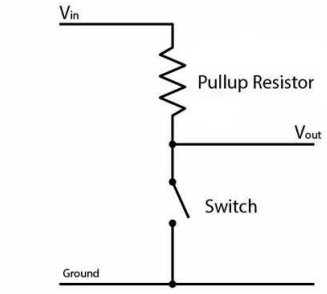


Figure 3: Circuit diagram of a pullup resistor [7][Altered]

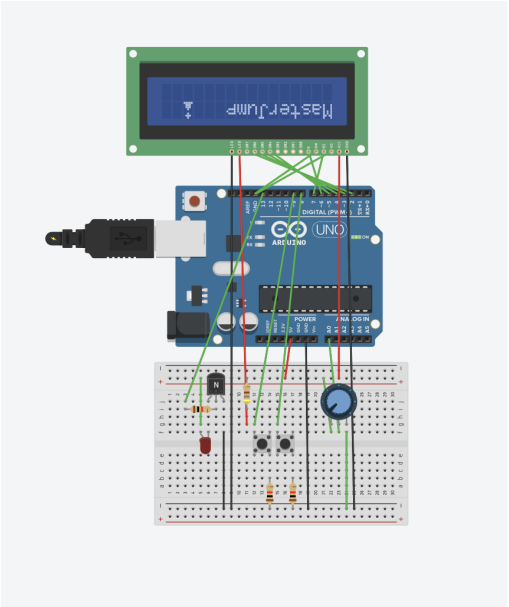
This ensures that the output of the switch is either HIGH or LOW. When the switch is open, the pin connects to 5V through the pullup resistor, and is HIGH. When the switch is pressed, the pin is LOW.

3.2 Prediction and intention of the code

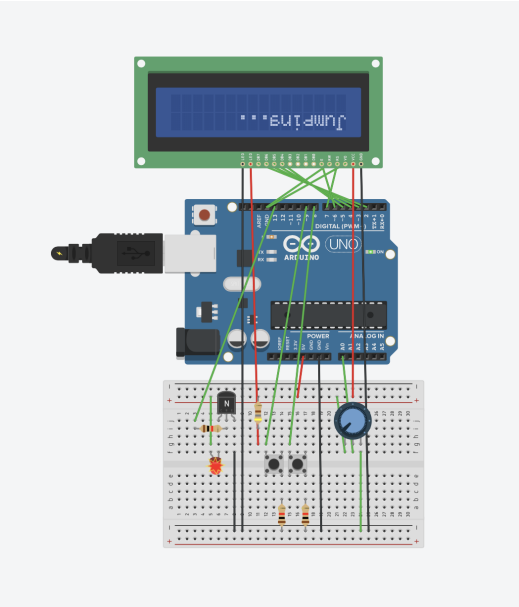
When the Arduino is first turned on, the LCD display should display "MasterJump", next to which there should be an animation that features a the chess-piece (character in *Jump*) jumping across the display. Then voltage readings in V and the time needed to be "pressed" by the device in ms should be displayed, to the nearest 0.01. The voltage and time should be in real-time, meaning if the potentiometer is adjusted, the voltage and time should change accordingly immediately.
When the red switch next to the LCD display is pressed, the display should print "Jumping..." and the screen should be grounded, as well as the LED below pin 13 being turned on for the length of time previously displayed, after which LED is turned off and the display is no longer grounded with the display returning to normal.
When the black switch is pressed, the display should show "Restarting...", while the LED is switched on, and the screen grounded for 500ms to mimic the finger pressing the restart button on the screen. Then, it is turned off and the grounding is stopped, the display returns to normal as well.

3.3 Simulation

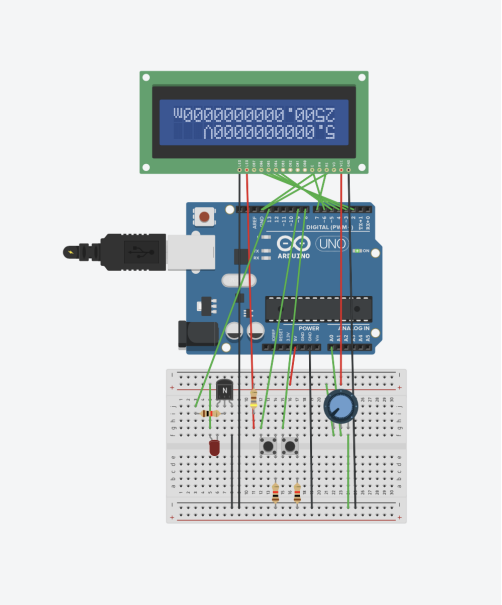
As the code was written during the time in between two sessions, simulations were run.



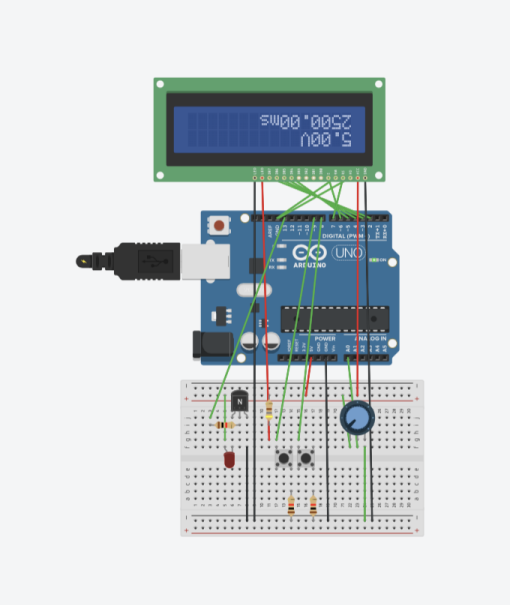
(a) Start-up screen



(b) Jumping



(c) Number display



(d) Number display after removing DEC

Figure 4: Simulation on `thinkercad.com`

Note that the LED was connected so that when the screen would have been grounded, it would light up.

The simulation showed that the transistor was working as predicted. However, the values displayed were not to the nearest 0.01 as predicted, shown in Figure 4c. This problem was later identified (when testing on an actual Arduino) as being caused by the `DEC` parameter in the `lcd.print` command, which was then removed. The circuit was also changed (a relay is added) as the touching didn't work with the transistor, but the code remained the same.

4 Circuit

4.1 Circuit Diagram

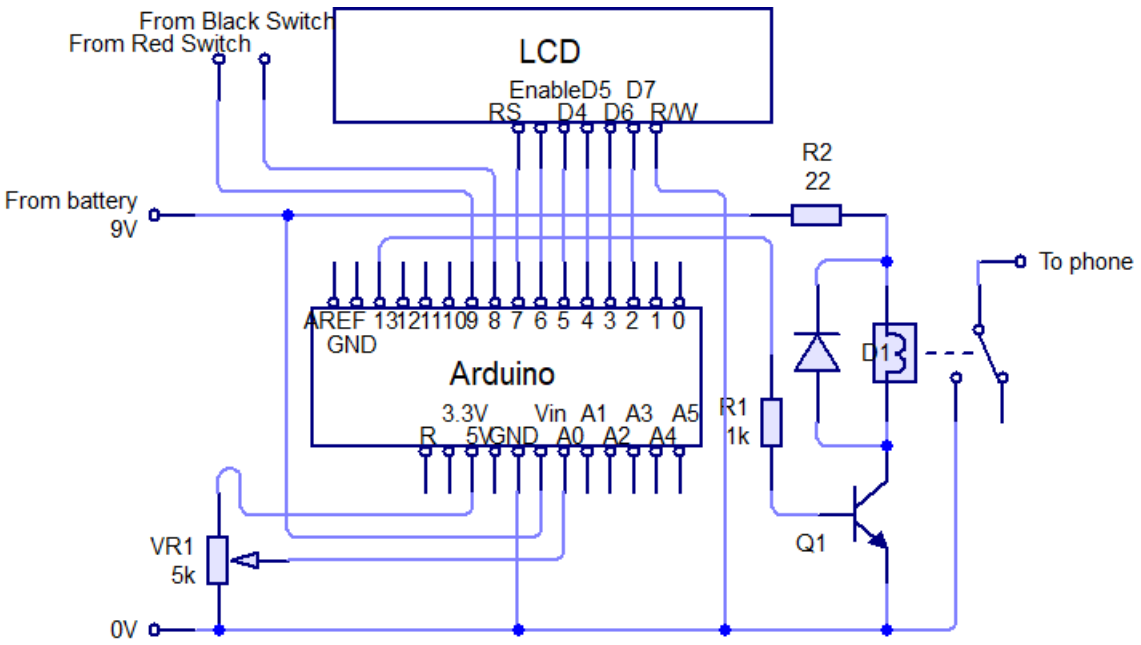


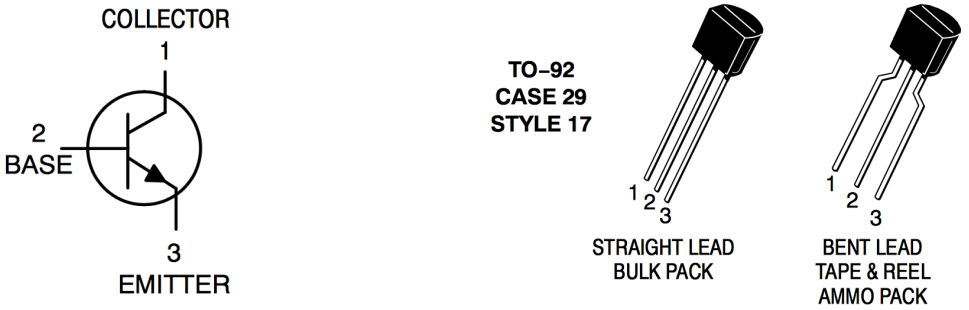
Figure 5: Circuit diagram

The components initially connected on the Arduino board had been omitted from the circuit diagram above.

4.2 Connecting Instructions

4.2.1 Transistor

The transistor used is BC337 which should be connected as shown below



(a) Circuit diagram symbol (b) Physical appearance

Figure 6: BC337 transistor appearance and circuit diagram symbol[4]

The collector should be connected to the relay, and the emitter should be connected to ground.

4.2.2 Diode

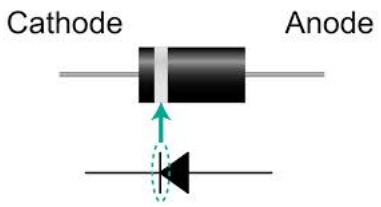


Figure 7: Diode appearance and circuit diagram symbol[5]

The cathode should be connected to the resistor. For the reason that the diode is connected in this sense, see section 4.4.4.

4.3 Physical Circuit Layout

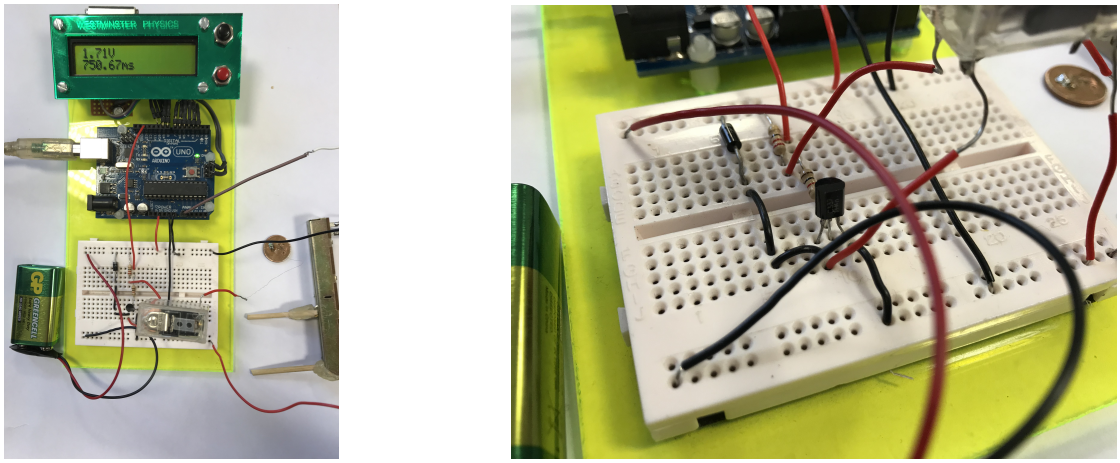


Figure 8: Physical circuit layout

4.4 Components, Purposes and Methods of Function

4.4.1 Resistor to Relay

The resistor connected to the relay is chosen to be 22Ω. The current through the relay when connected to a 6V power supply at which the relay normally works is measured to be 0.15A. According to $R_c = \frac{V}{I}$, the resistance of the coil is calculated to be 40Ω, so from $V_r = \frac{R}{R+R_c}$, the nearest sufficient value of resistors is 22Ω.It divides the voltage preventing the relay from over-heating.

4.4.2 Potentiometer

The potentiometer used is a linear slide potentiometer, as a voltage divider.

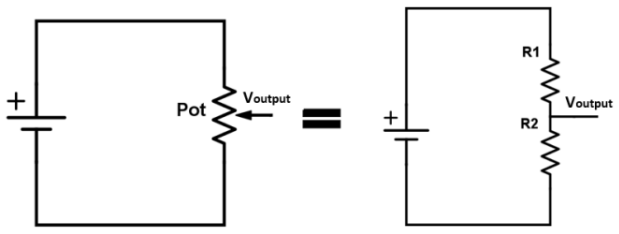


Figure 9: Potentiometer as a voltage divider[6]

$$V_{out} = \frac{R_1}{R_1 + R_2} \times V_{in} = \frac{\frac{\rho \times l_0}{A}}{\frac{\rho \times (l - l_0)}{A} + \frac{\rho \times l_0}{A}} \times V_{in} = \frac{l_0}{l} \times V_{in}$$

As shown above, the output voltage is directly proportional to the distance between the wiper and one side of the potentiometer, making it suitable for measuring distances.

4.4.3 Transistor

The chip used in Arduino is ATmega328P.

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

Figure 10: *Absolute maximum ratings* section of ATmega328P datasheet[2]

As the resistance of the relay coil is relatively small (40Ω, mentioned earlier in section 4.4.1), a large current will flow through it. If it is connected directly to one of the I/O pins on the Arduino board, because when the output pin on the Arduino is high, it will provide a voltage of 5V, the current drawn will exceed the current limit of each I/O pin and may damage the chip $\big(I = \frac{V}{R} = \frac{5}{40} = 125mA > 40mA\big)$(maximum current is shown in Figure 10).
Therefore, a transistor which only draws a small current from the Arduino is used to switch the relay on and off. We decided to use BC337 NPN transistor, as the maximum collector current is 800mA (in Figure 11), it satisfies the requirements.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector – Emitter Voltage	V_{CEO}	45	Vdc
Collector – Base Voltage	V_{CBO}	50	Vdc
Emitter – Base Voltage	V_{EBO}	5.0	Vdc
Collector Current – Continuous	I_C	800	mAdc
Total Device Dissipation @ T_A = 25°C Derate above 25°C	P_D	625 5.0	mW mW/°C
Total Device Dissipation @ T_C = 25°C Derate above 25°C	P_D	1.5 12	W mW/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−55 to +150	°C

Figure 11: BC337 transistor datasheet[4]

When the base voltage of the NPN transistor is 0, the transistor acts as an open switch, where no current flows through it and the relay is off. If a large enough positive current is driven into the base to saturate the NPN transistor, the current flowing from base to emitter controls the larger relay coil current flowing through the transistor from the collector to emitter, and the relay closes.
A $1k\Omega$ resistor is placed between the Arduino output and the base of the transistor to protect them.

4.4.4 Diode

The diode, called a “flyback diode”, is used to prevent damage to the transistor from back EMF (Electro-Motive Force) generated when the relay coil switches off.

As a relay coil is not only an electromagnet but also an inductor, when a current flows through the coil, a magnetic field is created around it. When the supply is switched off, the current flowing through the coil decreases to zero and the magnetic field collapses. This causes a voltage to be impressed across the coil (positive at the lower end of the coil and negative at the upper end) and because of the speed at which the field collapses this voltage can be very high.
The cathode of the diode is connected to the supply of the relay so that when the transistor closes the circuit, the diode remains in reverse biased and inactive. When the current is switched off, the diode conducts, forming a momentarily loop with the inductor, which limits the back EMF to the forward voltage drop of a diode.

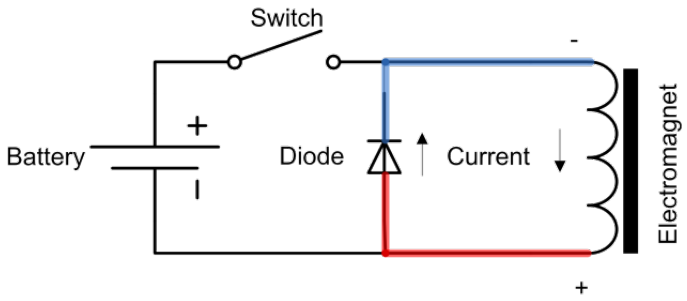


Figure 12: Inductor and diode momentarily forming a loop[3]

4.4.5 Touch-screen Controller

This project is designed to operate on a capacitative touch-screen, especially those on iPhones.

1. Grounding
   The simplest form of a capacitor is a dielectric capacitor, which consists of two conductors, separated by an insulator.

   In a touch-screen, there are two perpendicular layers of conductive coatings, forming a grid of electrodes below the insulative glass, represents one plate of such a capacitor. The other plate is represented by the surroundings of the sensor electrode (giving a parasitic capacitance C_0) and another conductive object, for example, a finger (forming touch capacitance C_T). A measurement circuit measures capacitance, a change is registered as a touch.

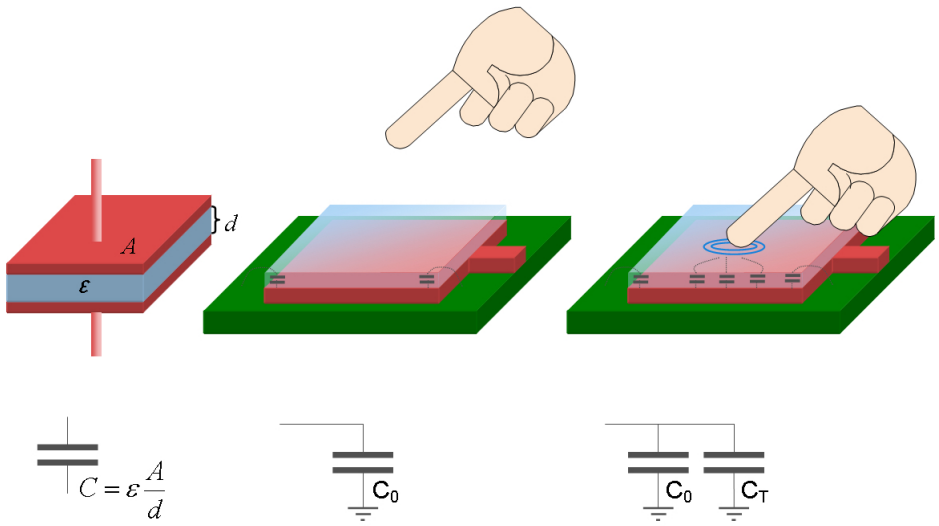


Figure 13: Untouched touch-screen with parasitic capacitance C_0 , touched touch-screen with additional touch capacitance C_T [8]

In order for the sensor to measure the capacitance, the circuit must be completed, as shown the

Figure 13, the capacitors are grounded. As humans are good electrical conductors, and usually in contact with the ground, the finger is always grounded.

To mimic finger pressing on a touch-screen, a conductive material which is grounded when pressing is intended can be used. Therefore, a relay is used to ground the material during which the output of the Arduino is high.

2. Touch-screen contact
   The algorithm calculating touches in smartphones usually specifies a lower threshold of the area in contact, in order to register as a touch, so the wire from the relay cannot be stick directly to the touch-screen (since the area is too small). A contact made of conducting materials with an area of about $50mm^2$ is needed.

   Experiments were done with conductive tape, aluminium foil and copper coin. The copper coin worked the best, so was chosen.

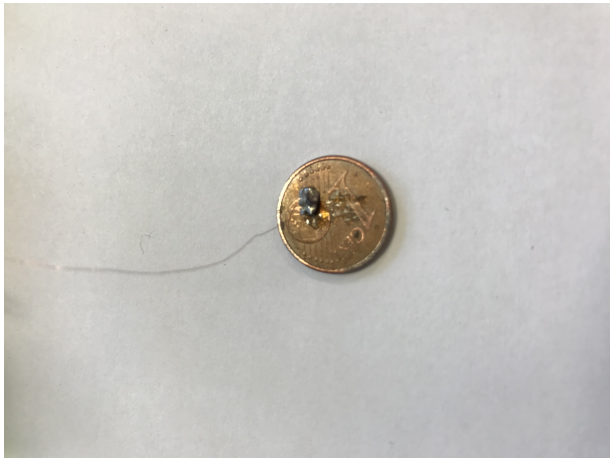


Figure 14: touch-screen controller

A thin soft copper wire is soldered onto the coin, so it will remain in contact with the touch-screen even if the circuit is moved a little. When soldering, a flux dispensing pen is used for precision control. The contact area was increased by sanding down the contacting side of the coin to make a flat surface.

~~Note: the coin was a euro, the royal portrait didn't get defiled.~~

5 Equation and Testing

5.1 Determining the Voltage-time Equation

To determine the voltage-time formula, a time-voltage graph is plotted, and the line of best fit is drawn. Though the error in determining each set of data is relatively larger, as the line of best fit is an "average" of all the data, the equation determined by this method should be accurate enough.
The data sets are determined by steps below:

1. Place one pointer of the potentiometer to the bottom of the chess piece and the other pointer to the centre of the next platform

2. Press the screen by hand the measure the time pressed using a stopwatch

3. Adjust the potentiometer to the real distance jumped by placing one pointer at the centre of the new platform, hold the potentiometer still and adjust the pointer selected to the bottom of the chess piece (compensating the difference between the actual distance and the one measured to the centre in step 1)

4. Note down the time and voltage on the screen

To increase the accuracy of the equation, 49 sets of data were obtained.

Voltage/V	Time/ms
0.64	540
1.07	560
1.28	600
1.36	620
1.90	820
1.46	620
1.34	630
1.08	590
0.20	400
0.70	530
0.77	490
1.34	650
1.95	830
1.28	700
1.91	790
1.34	630
0.89	560
1.71	800
1.51	700
1.06	650
1.18	630
1.52	700
0.64	500
1.14	570
0.95	550

Voltage/V	Time/ms
0.58	490
1.82	830
1.26	630
2.16	850
0.50	470
1.24	600
1.19	630
0.88	590
1.29	650
1.11	640
1.29	620
0.86	590
1.55	740
1.30	630
1.14	670
1.35	750
2.08	910
0.22	490
0.50	550
1.05	560
1.27	620
1.30	640
1.90	800
1.17	630

Table 1: Voltage displayed and pressing time needed

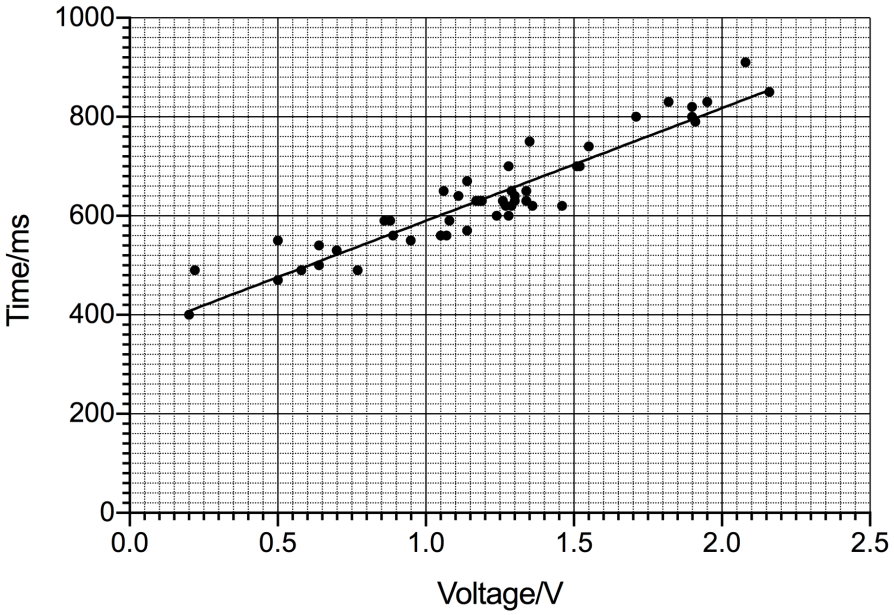


Figure 15: Time-voltage graph

The equation for the line of best fit is given by the plotting software to be

$$t(V) = 227.59 \times V + 362.45$$

The equation in `timeClac` is changed to the one above.

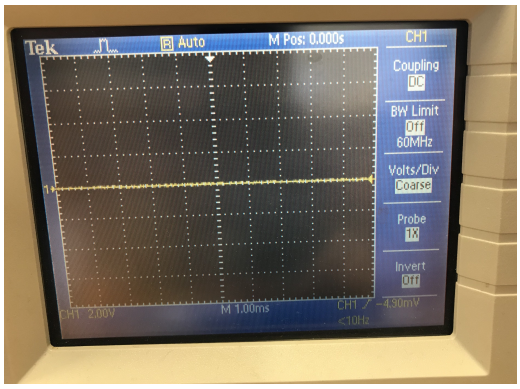
5.2 Testing

As predicted in Section 3.2, the start-up screen consists of "MasterJump" and a jumping-chess-piece animation.

Figure 16: Start-up screen on actual Arduino

To test the grounding of the touch-screen controller, the coin is connected to 5V, and an oscilloscope is connected to the coin and 0V.

(a) Oscilloscope display when not grounded



(b) Oscilloscope display when grounded

Figure 17: Grounding of the touch-screen controller

When the buttons are pressed, the LED below pin 13 lights up, the relay closes, and the coin is grounded (shown in Figure 17b).
Then, the circuit is tested with an iPhone, playing *Jump*. It worked perfectly, and reliably.
The video of playing the game can be found **Here**.

6 Evaluation

The circuit satisfies the specification (Section 2). However, the equation is determined by back box testing, through limited sets of data, so it may not be very accurate. Furthermore, the distance measurements are carried out by hand, which is time-consuming and potentially inaccurate.

Future Improvements
For further improvements, the distance can be measured using a camera. Since the platform always has a colour in contrast to the background, platforms can be detected by measuring pixels colour differences.

References

[1] Arduino documentation. https://www.arduino.cc/en/Reference/LiquidCrystalCreateChar.

[2] Atmega328/p datasheet complete. www.microchip.com.

[3] Back emf suppression. https://progeny.co.uk/back-emf-suppression/.

[4] Bc337, bc337-25, bc337-40 amplifier transistors. http://onsemi.com.

[5] Diode as a circuit element. https://www.khanacademy.org/science/electrical-engineering/ee-semiconductor-devices/ee-diode/a/ee-diode-circuit-element.

[6] Electronics basics – how a potentiometer works. https://randomnerdtutorials.com/electronics-basics-how-a-potentiometer-works/.

[7] Pullup / pulldown. http://midihacker.com/pullup-pulldown/.

[8] Fujitsu Microelectronics Europe Dirk Fischer. Capacitive touch sensor technology.