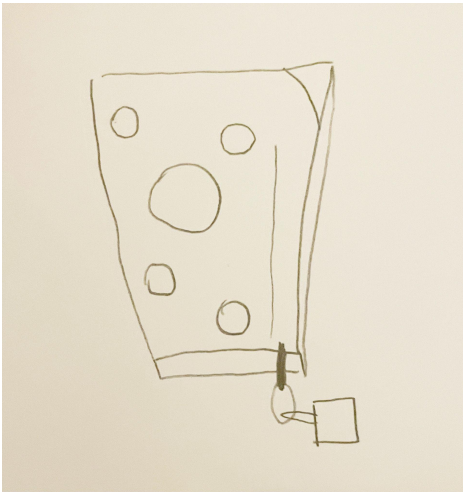
My project is named Table Pinball. Unlike the traditional pinball game in Hong Kong, which requires standing while playing, Table Pinball is a game that can be played on a surface. You do not need to use your hands to play with this, rather control it by an Arduino circuit. It is mainly composed of two parts, the non-electronic part (gaming box) and the electronic part (Arduino). I will talk about how this idea is generated first.

Idea Generation

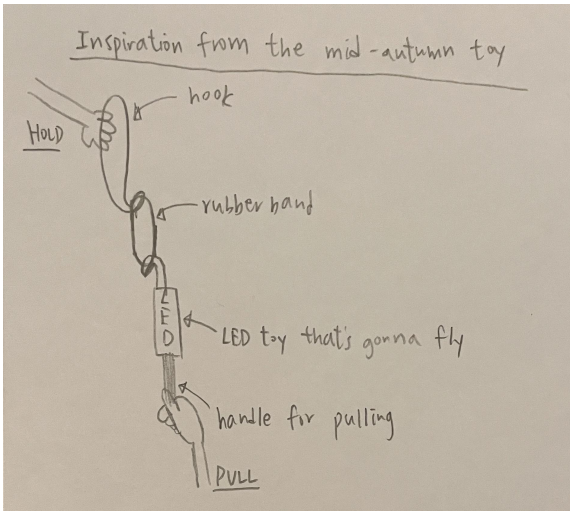


Initially, I came to the idea that moving this game to one that can be played at home on surface would be great, as this pinball game that is usually found in malls in public housing estates are reducing rapidly these years. The mechanism looks easy to build, so I chose this as what I am going to do with my Arduino.

First Draft:
The picture on the right is the first draft for how I think I can make Table Pinball workable. I would like to use a servo motor to pull the rubber band, so that when the servo motor turns in a direction that is opposite to how it pulls the band, the band will hit the ball that is inside the gaming box.

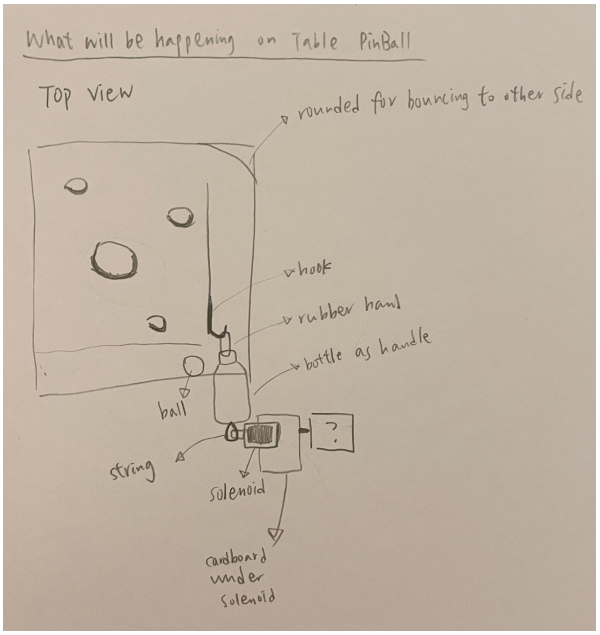
After consulting my professor's idea, he told me that the servo motor is not powerful enough to stretch the rubber band and it also cannot turn quickly when it releases the rubber band. I was suggested to use a solenoid for releasing the rubber band and use a motor that is more powerful to make Table Pinball work.

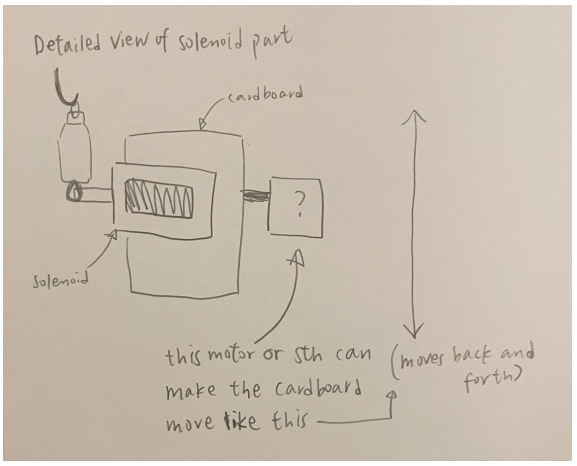
Second Draft:



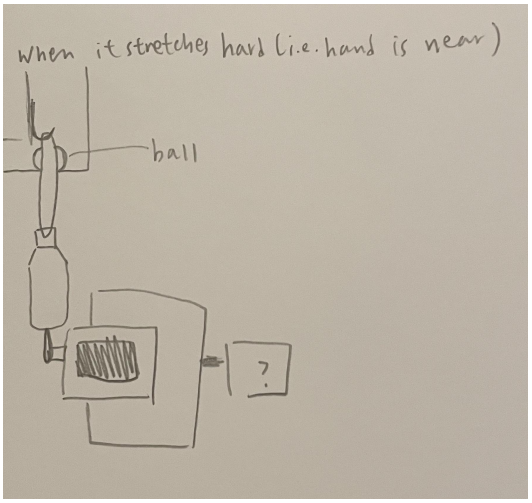
Link to how the mid-autumn toy works(filename: demo_midautumn.mov): https://drive.google.com/drive/u/0/folders/1T2XR6X1jEVwdPCQUzv11WqLJ9ywq28Bb

In my first draft, I have not clearly thought of how to push the ball precisely. After the consultation, I drew some pictures to illustrate how I expect it to work. From the above image, it shows the setup for how stress and tension can fly the LED toy to the sky. It is like a bow. I apply this concept to Table Pinball by changing the LED toy to a plastic bottle, so that it can fly into the hole on the gaming box.



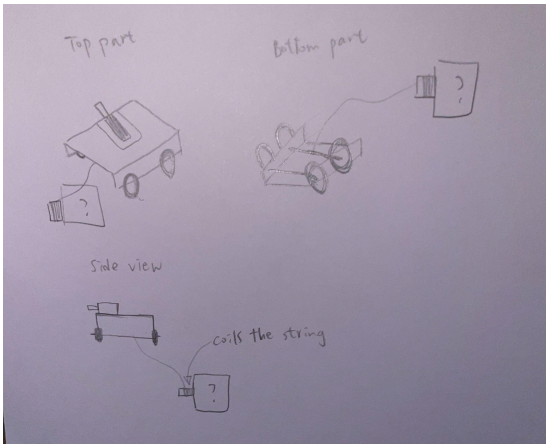


At this stage, the gaming surface design of Table Pinball is still adopting a “holes” approach, meaning that the ball will fall into one of the holes. As a servo motor is not a good option, I was curious and confused about what kind of motors can I use to make the solenoid move vertically. The picture on the right shows the detailed view of my ideation on the solenoid part in the second draft. These are also the images that I used to explain to my professor what exactly I am going to do.



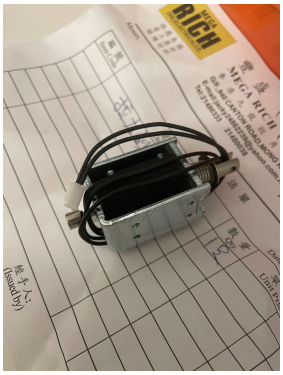
This shows the contrast before and after the change in position of the solenoid brought by a motor.

One of the ideas that I have is to substitute a motor that moves straight with another thing. What came to my mind at that time was to use a car. A thread will be coiled when the motor turns, so the car will move and bring the solenoid to other positions.



Third Draft:

Reviewing the course title, it is called “Hardware Hacking”. I am a hacker, that means I can do all these by hacking something. Professor had suggested using a linear motor or the linear encoder from an old printer. I started to find the materials I need for Table Pinball, and had gone to Ap Liu Street in Sham Shui Po a couple of times. For your reference, Ap Liu Street is a street in Hong Kong that has many hawkers and shops, selling electronic stuff and second-hand products. I found most of the materials but not the two most important ones. I could not find the solenoid and an old printer.



A customer in one of the electronic shops heard what I wanted to find, and suggested me to go to Reclamation Street in Mong Kok for the solenoid. Thanks to this helpful passerby, I found a 12V solenoid there.

Whereas I started to find an old printer online. I found it on Carousell(https://www.carousell.com.hk/p/%E4%BD%B3%E8%83%BD%E5%BD%B1%E5%8D%B0%E6%A9%9F-cannon-printer-187007215/), with an acceptable price of $40 hkd.
The printer looks like this. It is a pity that I was busy putting the pieces of components apart and did not take a picture of how it looks inside.

A step-by-step tutorial will be shown below.
Tools that you need throughout the process of making:

- AA super glue x1
- Glue gun x1
- Cutter x2 (a big one and a small one)
- Scissors x1
- Screws x~10
- Screwdriver kit x1
- Double-sided tape x1
- Soldering iron x1
- Solder roll x1

Gaming box

By the end of the creation of the gaming box, you can play Table Pinball manually without Arduino.

Step 1: Materials

Components that form the gaming box:

- Cardboards (cut from the box of iPad)
- Old plastic bottle x1
- Rubber band x1
- Hook x1
- String x1
- Table tennis ball x1
- Clear F4 file x1
- Wooden board x1

Step 2: Build the box

I stacked and stuck two frames of iPad boxes one on top of the other. Another cardboard should be placed like a slope, with one side landing on the ground and another side leaning on top of the side of the bottom frame.

Adjust the steepness of the slope that is nearer to yourself by rolling the table tennis ball. Make sure that the ball can roll back to the original position through this slope.

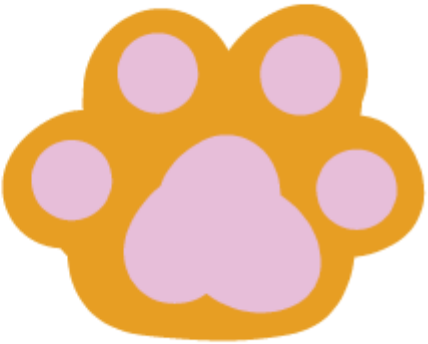
Make a hole on the right part to the side facing you for the bottle to get in slightly. Hang a hook on the upper frame and hook the rubber band with the bottle on it. Stick the hook with a glue gun.

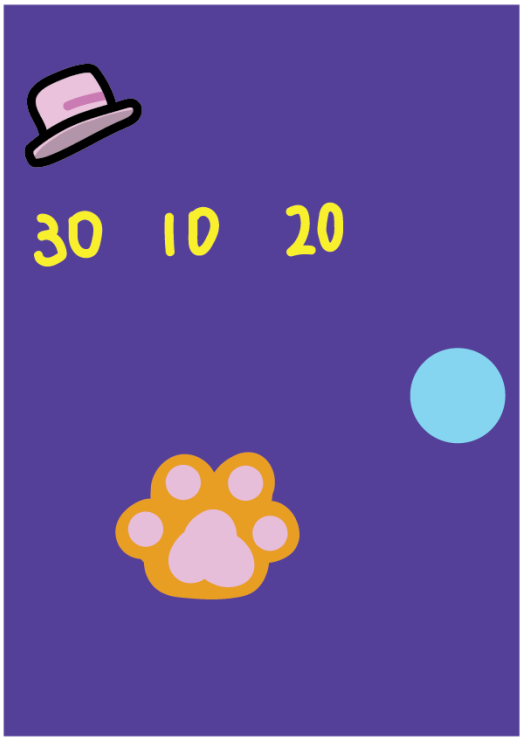
Tie the rubber band to the cap of the bottle and tie the string to the bottom of the bottle.

Cut the back of the upper frame, a height that papers can go in and come out easily is enough.

Step 3: Design a logo and gaming surfaces

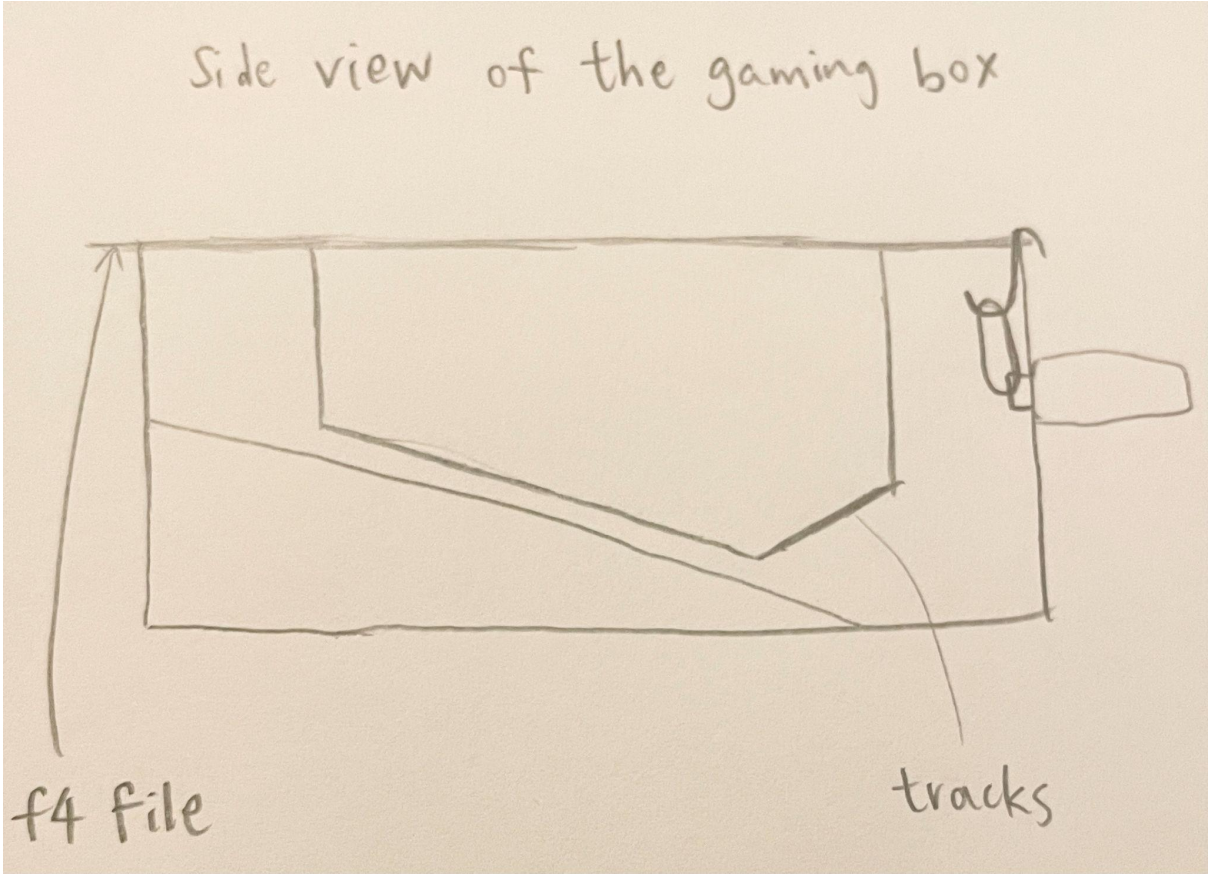
Design your own logo for a little bit of brand building and cover the logo or unwanted parts on your box with it. I designed mine as a claw of a cat, as one of the features of this game is it can be played without hands, meaning animals at home and also play with it.





Next, design your own gaming surface with the patterns you like. I have chosen yellow and purple to be my theme colors for Table Pinball, with my logo on it. These gaming surfaces can be inserted and taken out from where you cut your upper frame in step 2.

Step 4: Build the tracks



I made the tracks using cardboard from the iPad box too. As they needed to fit onto the slope, they were cut into trapeziums. However, the part that is near to the slope for passing the ball back to the original place might block the ball. To let the ball pass smoothly, I cut away another angle from the trapezium and it looks like the one in the above image. Stick these tracks on the F4 file. Place the F4 file on top of the frames but do not stick all the sides with any glue, as you may need to replace the rubber band in the future.

Using a piece of cardboard, curl it into an arc and stick it on the top right corner. This can send the ball to the other side of the gaming surface when it is popped.

Your gaming box should look fine now. Get to the next stage to the electronics part.

Electronics Part

The electronics part includes the solenoid, the linear encoder of the printer and the ultrasonic distance sensor.

I will connect these parts to the Arduino, so that how far the rubber band stretch can be controlled by detecting the distance between the ultrasonic distance sensor and your hand.

Step 1: Materials and Skills

Materials:

- Arduino Uno x1
- Breadboard x1
- Ultrasonic distance sensor x1
- Solenoid x1
- Old printer x1
- Jumper wires xMany
- Diode x2 (1N4007)
- Resistor x2 (1k x1, 10k x1)
- Mosfet x1 (IRF520)
- NPN power transistor x1 (TIP102)
- 12V battery supply

Skills:

- Knowledge on soldering
- Knowledge of how the breadboard works
- Know how to connect the Arduino to the computer
- Know basic C++ and syntax for Arduino
- Willing to try until success

Step 2: Draft the circuit

You can draft your circuit using Tinkercad, it is free and there are lots of diagrams for you to use.

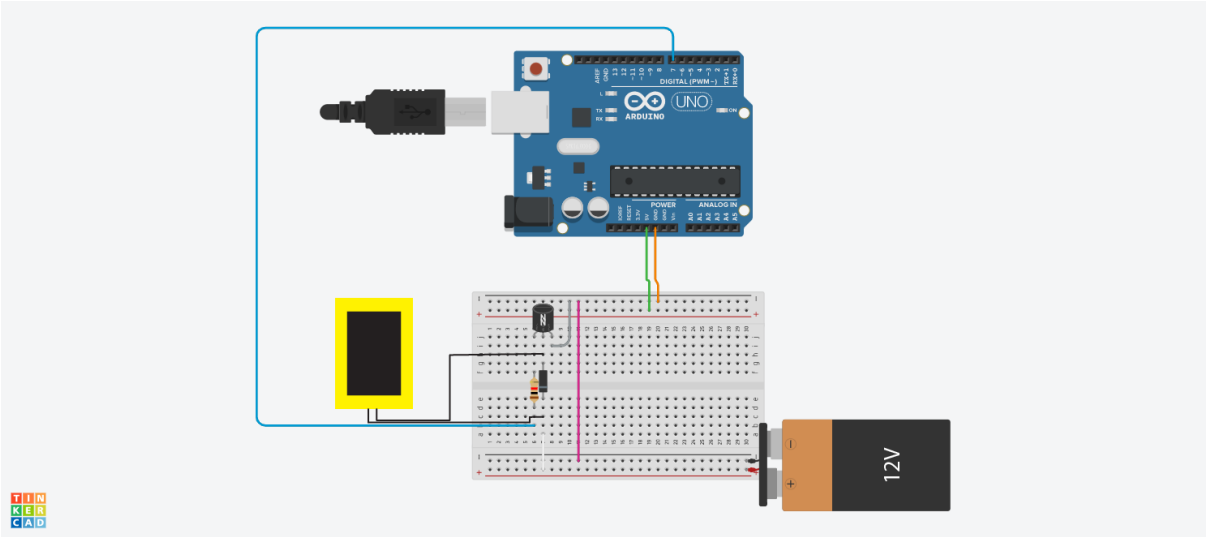
It is worth noticing that Arduino can only provide 5V. If you need to use a higher voltage, you have to use an external power source. There are two ways to do so.

1. Plug your battery with a 2.1mm center-positive plug cable onto the Arduino board.
2. Plug your battery on the side of the breadboard that is different from where you give 5V to the breadboard.

In my case, as I do not have a cable with a 2.1mm center-positive plug, I used the second way to give 12V to the breadboard. If I rotate the diagram below clockwise by 90 degrees, I can get 12V from the left and 5V from the right. It is important to remember that the 12V will only link to the device (i.e. solenoid and printer), but not other parts like the transistor or resistor. Too much voltage will burn parts other than the device.

My 12V power supply is built by a battery case that can be found in stores selling light-related goods and eight 1.5V AA batteries. Be aware of the contact for the wires. Mine is not that good, so I added some soldering iron to it to ensure the stability of contact.

For the solenoid:

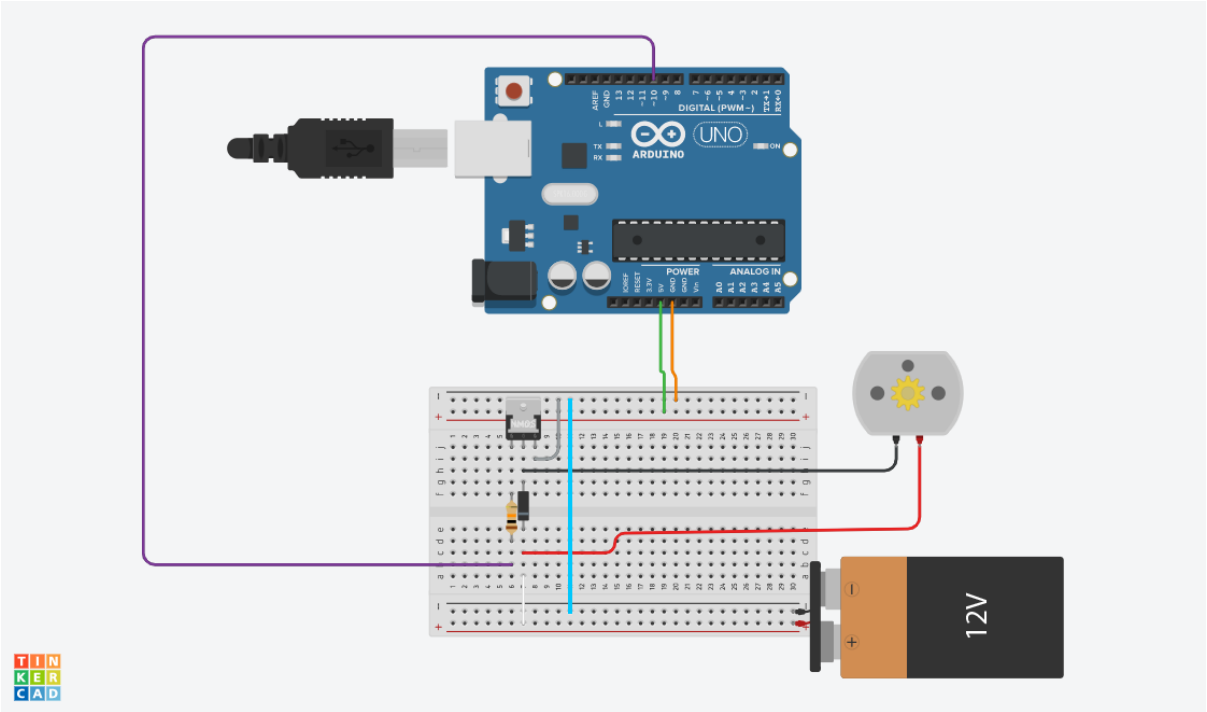


I used a NPN power transistor (TIP102) for the solenoid. I will mention the transistor as TIP102 in short. The left leg is the gate. I connect the left leg to the pin, which can control the on/off of the power supply for the solenoid. The middle leg is the drain. It should be connected to the positive power supply. The right leg is the source. In Arduino, we call the negative power supply a ground. This leg will be connected to the ground.

There was a small accident in my trial. The solenoid did not work when I tried to connect it to the circuit. I tried another solenoid that only needs to use 5V, and the same thing happened. This little experiment told me it was not the problem of the battery supply. It should be some issue regarding the transistor. I went to Sham Shui Po again and bought TIP102 and mosfet (IRF520). After changing the transistor to TIP102, it worked. A lesson learnt here.

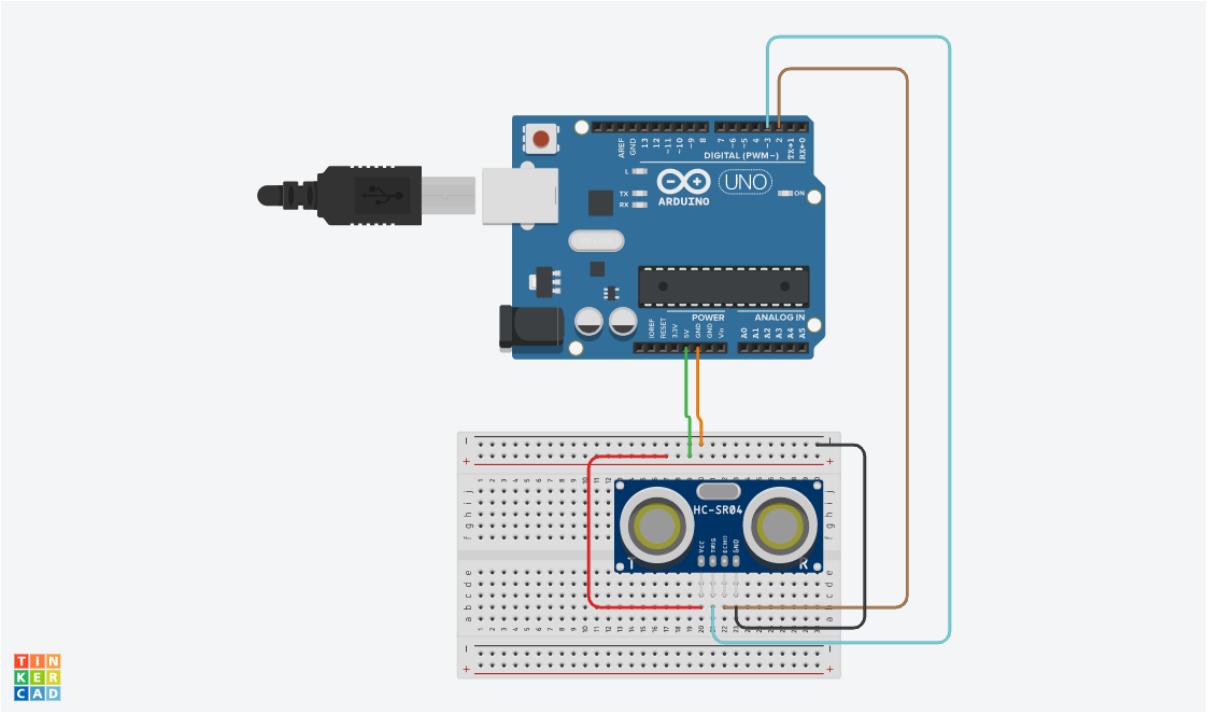
The pink wire brings 12V negative to 5V negative and is connected by the grey wire to link it to the right leg of TIP102. Place both legs of the diode in the middle leg of TIP102. Notice the direction of the diode. The side with a silver mark should be near to the positive power supply. Put the wires of the solenoid to both ends of the diode. And link the positive 12V power supply according to the white wire shown above. The light blue wire connects the pin to 7 in the Arduino, so I will connect pin number 7 in the Arduino for on or off. A 1k transistor is put between the pin and TIP102.

For the printer:



I used a mosfet (IRF520) for the printer. A mosfet works the same for the connections on different legs as the transistor used for the solenoid. The only difference between the connection of the solenoid and the printer is the pin on Arduino and the resistor. I chose pin number 10 as my pin for switching on and off. The resistor I used here is the 10k resistor. Other connections are the same as the solenoid. The motor in the linear encoder from the printer I used is a DC motor, so connecting which side to which polarity only has the difference in the direction of how the motor spins.

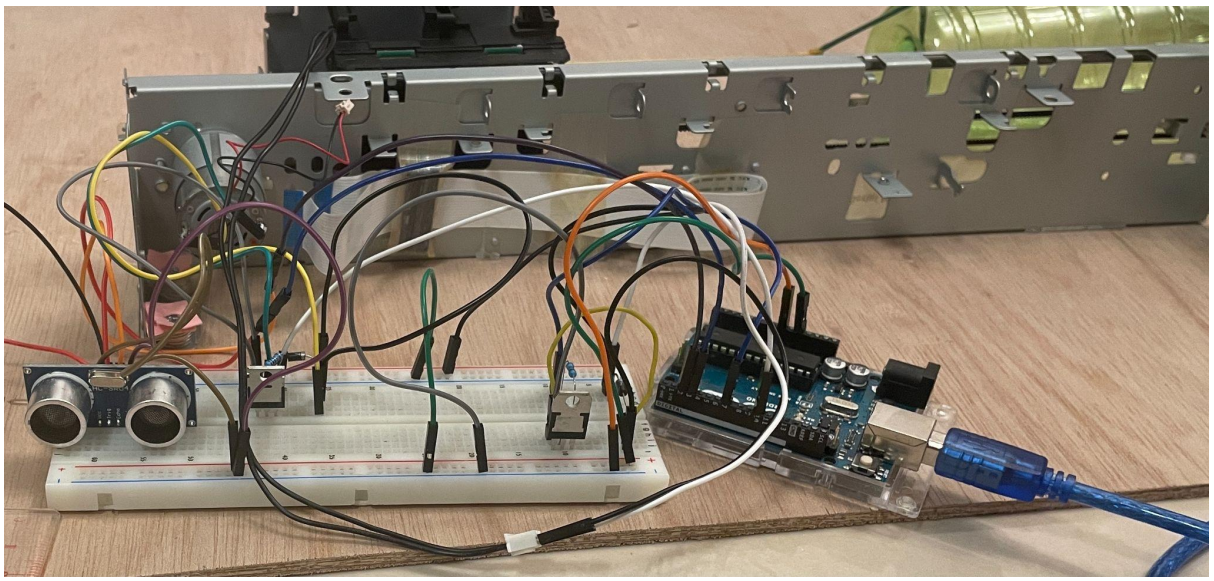
For the ultrasonic distance sensor:

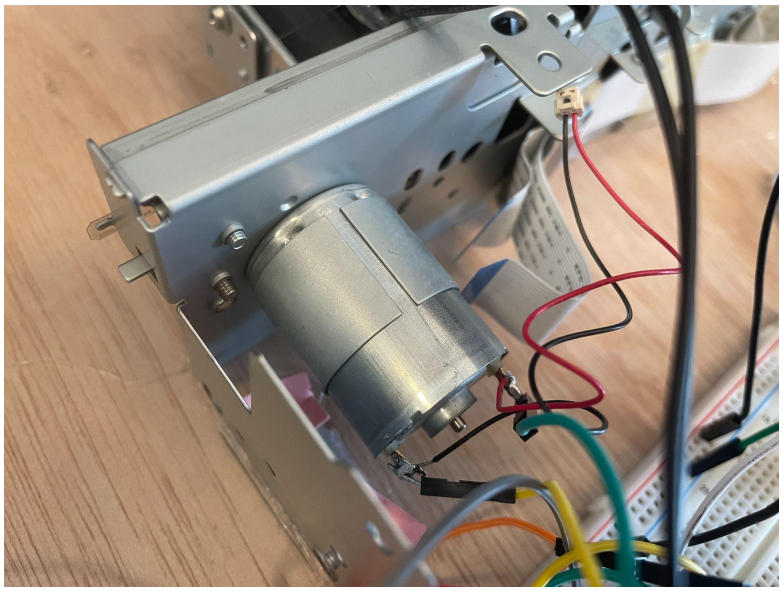


The ultrasonic distance sensor only requires 5V. Getting the power supply from the right of the breadboard is enough. The type that I use is with four legs. The left leg is VCC, connecting to the positive 5V power supply (red wire). The middle left leg is the trigger pin, connecting to one of the pins on the Arduino board. It triggers ultrasonic sound pulses. The middle right leg is the echo pin, also connecting to one of the pins on the board. It produces a pulse when it receives a signal. The right leg is the ground, connecting to the negative 5V power supply (black wire). I connected the trigger to pin number 3 (light blue wire) and echo to pin number 2 (brown wire).

Step 3: Build your circuit

To facilitate illustration and explanation, I divided them into three diagrams above. In reality, I put them all onto the same breadboard. This step is quite easy if you designed your circuit correctly in step 2.





The only part that is a bit hard but super important is to solder two jumper wires to the connection on the DC motor of the printer (yellow and green wires).

Step 4: Program your Arduino

Remember the flow of operations when programming.

1. Detect the distance by the ultrasonic distance sensor
2. The printer encoder moves
3. The solenoid clicks

To ensure all these come in the correct order, I put a delay after each operation, so that neither of these operations will crash together when running the program.

As mentioned above, the distance on how far the linear encoder of the printer goes is determined by how far is your hand from the ultrasonic distance sensor.

This is done by setting how long the circuit will be closed, meaning the duration of electricity going through. I set the maximum distance to 25cm.

```
void loop() {
  Serial.print("The Distance is: ");
  int x = sonar.ping_cm();
  Serial.println(x);
  delay(5000);
  if(x > 20){
    digitalWrite(printer, HIGH);
    delay(100);
    digitalWrite(printer, LOW);
    delay(500);
    for(int i = 0; i < 2; i++){
      digitalWrite(led, HIGH);
      digitalWrite(solenoide, HIGH);
      delay(50);
      digitalWrite(led, LOW);
      digitalWrite(solenoide, LOW);
      delay(50);
    }
  }
```

I print the distance between my hand and the ultrasonic distance sensor on the serial monitor. This also shows a loop for a distance of 21-25cm. I will explain sonar.ping_cm() in the setup part of the code.

For the range of 6-10cm, the circuit is closed for 1000ms. The linear encoder goes furthest.

For the range of 11-15cm, the circuit is closed for 700ms.

For the range of 16-20cm, the circuit is closed for 300ms.

For the range of 21-25cm, the circuit is closed for 100ms.

A led is added for convenience in debugging. When the led blinks, it means the solenoid should click too. If the solenoid clicks during times when the led is off, there must be some problem, either the program itself or the solenoid is out of order.

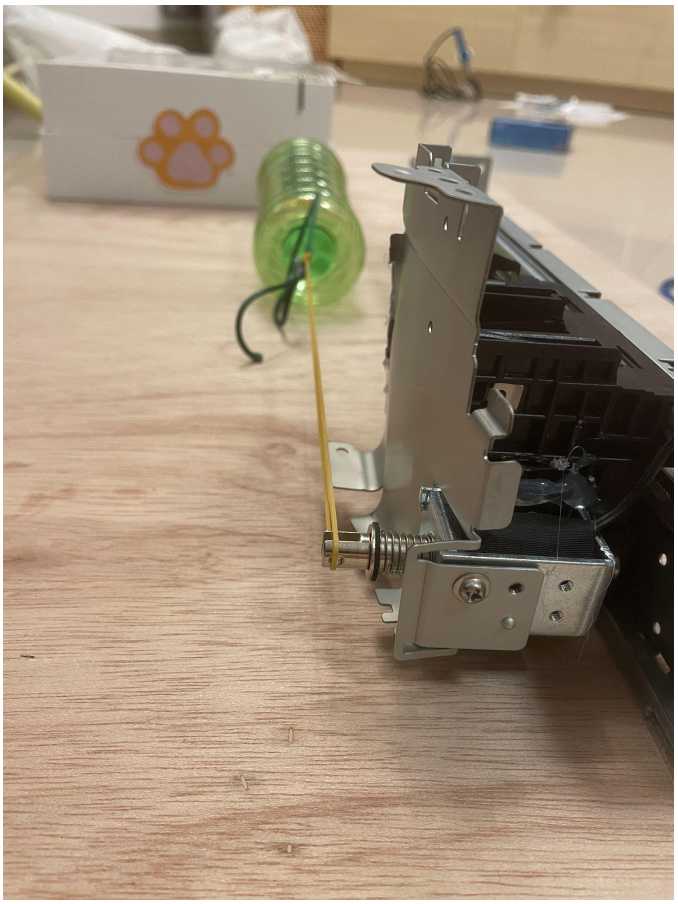
```
#include <NewPing.h>
NewPing sonar(3, 2, 25); //trigger, echo, max distance wanted to be tested
int led = 13;
int solenoide = 7;
int printer = 10;

void setup() {
  Serial.begin(9600);
  delay(50);
  pinMode(led, OUTPUT);
  pinMode(solenoide, OUTPUT);
  pinMode(printer, OUTPUT);
}
```

The setup of the program looks like this. For the ultrasonic distance sensor, I downloaded a library named NewPing. You can download this from “Tools” → “Manage Library” and type NewPing in the search bar. I do not know the steps for downloading on an Android device but it should be similar. After downloading, include the header file to the Arduino program. NewPing sonar requires three parameters, the first one is the pin number of the trigger, the second one is the pin number of the echo and the third one is the maximum distance to be tested by the ultrasonic distance sensor. In setup, set the led, solenoid and printer to be output. sonar.ping_cm() records the measured distance.

Step 5: Keep everything in position

The ruler should be placed 90 degrees to the ultrasonic distance sensor as a mark on how far you are from the device. The printer encoder is screwed onto the wooden board.

To fix the position of the solenoid, I stuck it with the glue gun onto the ink container and used the pieces from the printer to fix it with screws.



Congratulations, you have successfully built your Table Pinball by now.

Limitations

- The DC motor from the linear encoder of the printer is not powerful enough to drive to a far distance. A motor that is more powerful should be used.
- The clicking of the solenoid might not successfully loosen the string everytime, as the pulling force gets large when at a far distance. A few more clickings can solve the problem.
- I did not build a “tunnel” for the bottle, so it may hit the box instead of the ball when it flies.

Improvements and future development

For improvements, finding a motor that is more powerful or using some materials other than tension to fly the bottle would be great. As stretching the band by the linear encoder of the printer is too much of a burden, I have to deal with the mechanics to make Table Pinball work more smoothly. Maybe I should build a tunnel for the bottle, but I am thinking of a way of not blocking the solenoid.

In the future, using the linear encoder of the printer, I would like to keep the theme of games. What comes to my mind is to make a shooting game, controlling the position of the linear encoder using the same method as in Table Pinball. The gun will occupy the position of the solenoid. If the gun is light enough, I can use a servo motor to turn its direction. This might also be a fun game to make a series of games with the help of the linear encoder of the printer.

Table Pinball is published on Instructables now. It also enters a contest called “Trash to Treasure”.

Common questions:

Q: What is so special about Table Pinball?
A: Table Pinball provides a new way to control a pinball game, any solid can be the trigger of Table Pinball, as long as it is detectable.

Q: Why must I cut the upper frame for inserting papers?
A: Unlike other physical games, you can change the background of the gaming surface as you like at anytime, changing themes by yourself physically.

Q: Is this game suitable for little children?
A: Yes, as the operation is rather simple, everyone can learn how to play in a short period of time.

Q: Does controlling by the ultrasonic distance sensor mean I do not need hands to play this?
A: 95% yes. The time that requires you to use your hand is to put the string onto the solenoid and when your printer encoder needs to get to a far position.

Similar project:

I found a similar pinball project on Youtube. The channel is called “fluxwood”. The link is https://www.youtube.com/watch?v=xlxsT34q97c. The difference is the gaming surface is more complicated and there is a scoring board on the device. The way of controlling it is by hand, demonstrating a variety of usages of Arduino on pinball games.

Link to my Instructables page:
https://www.instructables.com/Table-Pinball/

Link to my Video demo on Youtube:
https://youtu.be/2lB0tzWWIOA