

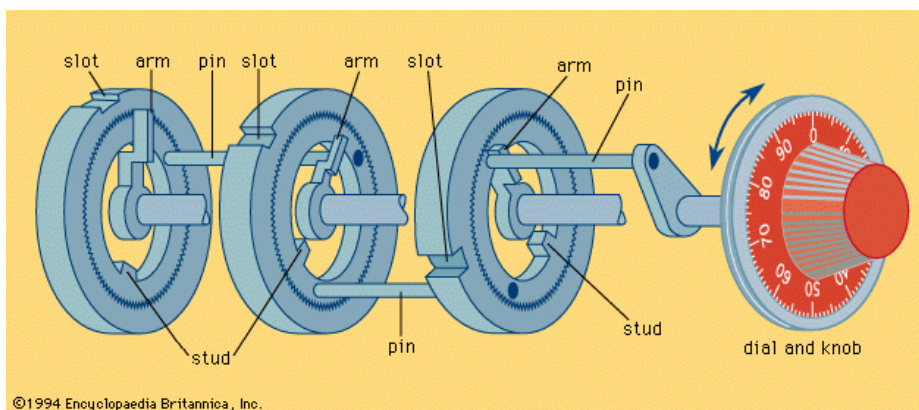


Combination Dial Lock Workshop

What is a Combination Dial Lock?



A single dial combination lock is a type of locking mechanism found on padlocks, lockers, or safes. It uses a single dial which interacts with several aligned discs or cams. A lock of this type is opened by rotating the dial clockwise to the first number, then counterclockwise to the second, and so on alternating direction until the last numeral is reached.



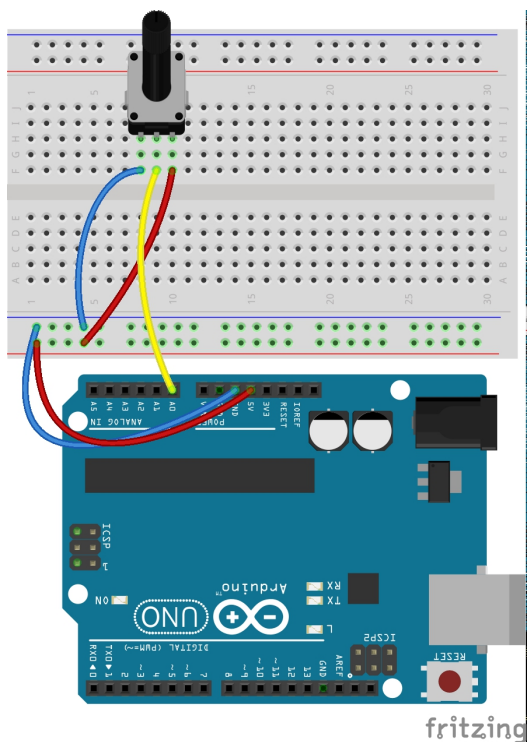
Below is a link to a great video of a home made wooden dial lock. Shows the inner workings of a dial locking mechanism.
<https://www.youtube.com/watch?v=1-fCrLwhPo>

This is how a mechanical dial locks work. Today we will be simulate the inner workings of a combination lock with electronic components and Arduino.

1. Analog input - Receiving the value of a potentiometer

Question: Why is the potentiometer an analog input?

Build the circuit



Add the code

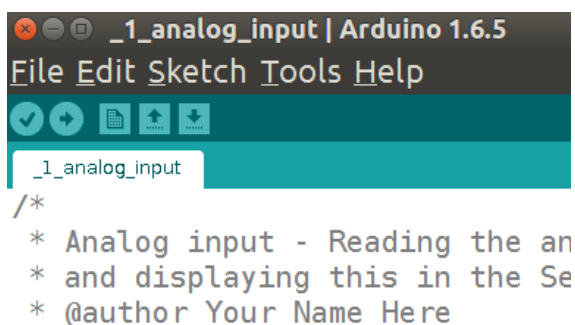
```
void setup() {
  // Start the serial communication between the Arduino and computer.
  Serial.begin(9600);
}

void loop() {
  // Store the main analog input values.
  int potValue = analogRead(A0);

  // Concatenate/join together strings for a neat output.
  String output = "Sensor value: ";
  output += potValue;

  // Output into serial monitor for debugging.
  Serial.println(output);
}
```

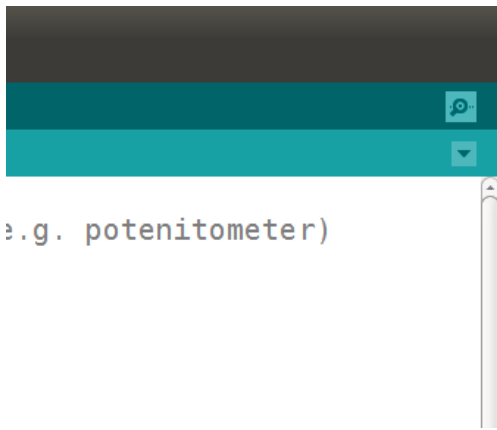
Compile & upload your code



Clicking on the tick icon/button will compile your code. Converting the code you've written into a format that Arduino understands. It will also tell you if you have any errors in your code.

If everything is ok then click the 'right arrow' icon/button. This will upload your code to the Arduino.

Open Serial Monitor (*ctrl + shift + m*)

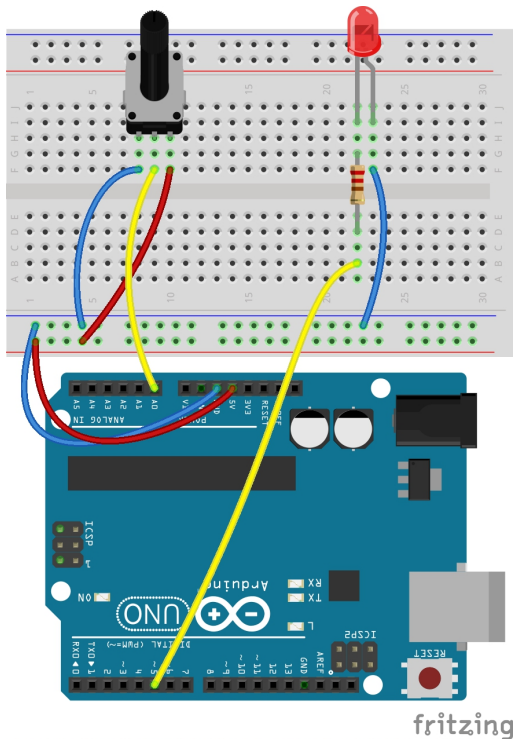


Clicking on the magnifying glass will open up the Serial Monitor and allow you to see what position the potentiometer is currently at.

2. Analog input and Analog output - Controlling an LED

Question: What do the `setup()` and `loop()` functions do?

Build the circuit



Change your code

(1) Add this line to your `setup()` function

```
pinMode(ledPin, OUTPUT);
```

(2) Add this line to your `loop()` function

```
analogWrite(ledPin, potValue / 4);
```

Compile and upload

Any errors? If so what do the error messages say?

Change your code

(3) **Replace** this line of code:

```
analogWrite(ledPin, potValue / 4);
```

with this:

```
if(potValue > 500 && potValue < 600){  
  // Tell the LED pin to turn on  
  digitalWrite(ledPin, HIGH);  
} else {  
  // Tell the LED pin to turn off  
  digitalWrite(ledPin, LOW);  
}
```

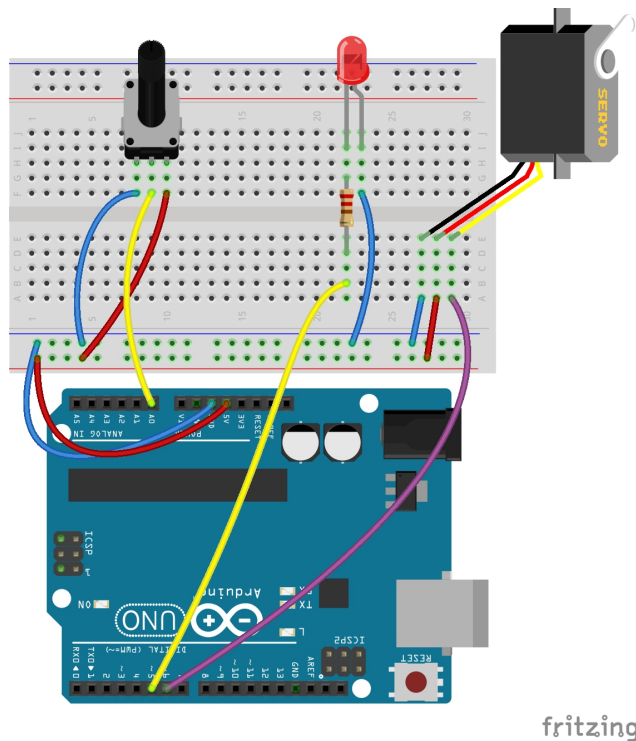
Compile and upload

Any errors? No! Super.

3. Analog input and servo - Controlling a servo!

Question: What is a function?

Build the circuit



Add some code

Discussion

Let's take a look at the code and see what has changed. What do we expect to happen when we upload our code?

Compile and upload

Compile errors? Upload errors?

Change the code

(1) We don't want the servo to `open()` when the sketch starts. How do we stop that from happening?

(2) We only want the servo to `open()` when the potentiometer is at a certain point. How do we make this happen? Look at line number 35.

Compile & upload

4. Create our dial positions

Question: How many dial positions shall we have? And how does this change the strength of the lock?

Add our code

Discussion

What's going on? How has the code changed and what are we expecting to happen?

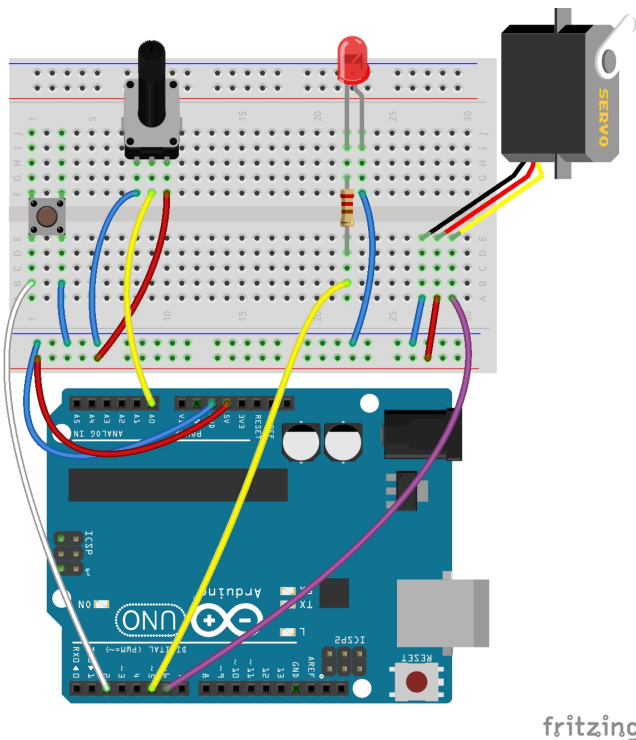
Compile & upload

What is the result? We can change the output from an LED to a piezo?

5. Digital input - Read the value from our button

Question: Why is the button a digital input? **Question:** How does a button work in a circuit?

Build the circuit



Change the code

(1) We want to prepare the buttonPin on Arduino to be a special input (`INPUT_PULLUP`). Add this line to your `setup()` function

```
pinMode(buttonPin, INPUT_PULLUP);
```

(2) Add this line to your `loop()` function

```
int buttonState = digitalRead(buttonPin);  
Serial.println(buttonState);
```

Compile and upload

Compile errors? It happens. Not to worry. Where in our code do we define `buttonPin` ? How does the code know to read from pin number 6?

Open the Serial Monitor (*ctrl + shift + m*)

Let's take a look at the values that we get back from the button. What is interesting about this? Nothing?! Just ones and zeros! Perhaps but how does this effect our next coding task?

Change the code

(1) Add a new function

```
void reset(){  
    isOpen = false;  
}
```

(2) Replace the following lines

```
if(potValue > minPotValue && potValue < maxPotValue){  
    open();  
}
```

With the following

```
if( buttonState == LOW && lastButtonState == HIGH ){  
    lastButtonState = LOW;  
    open();  
}  
if( lastButtonState == LOW && buttonState == HIGH ){  
    lastButtonState = HIGH;  
    reset();  
}
```

Discussion

What is our `reset()` function doing? What is the `lastButtonState` variable doing?

6. Collecting the dial positions[]

Question: What is an array?

Add the code

Compile and upload

Compile errors? Any bugs in our code? Gremlins in the system?

Open the Serial Monitor (*ctrl + shift + m*)

Move the potentiometer to any dial position. Push the button. Every time you push the button you should see the values you are storing on the serial monitor.

Discussion

Why does the count start at 0 and not 1? Because...

7. Compare our dial positions with the lock CODE[]

Change the code

(1) Add this new variable at the top of our sketch. Above the `setup()` function

```
int CODE[] = { 1, 2, 3 }; // Super secret code. Probably best to change this.
```

(2) Add the following code to the bottom of our `loop()` function

```
// If we have reached 3 values and the combination is correct.
if( count == 3 ){
    if( CODE[0] == combinationAttempt[0]
        && CODE[1] == combinationAttempt[1]
        && CODE[2] == combinationAttempt[2]){
        open();
    }
    reset();
}
```

Discussion

What are we doing here? What do we expect to happen?

Compile & upload

We should now have a lockable box for your precious things. You can change the `CODE[]` to anything you want.