# 16-722 Sensing and Sensors Lab 2
# 1/f Noise
# Lab Report

Zhiqi Gong
zhiqig

October 2018

## 1 Introduction

Pink noise or 1/f noise is a signal or process with a frequency spectrum such that the power spectral density (energy or power per frequency interval) is inversely proportional to the frequency of the signal. The simple description of the phenomenon is “the longer you wait, the more likely you are to see a large random fluctuation”. The target of this lab is to identify a phenomenon that the scientific/engineering literature says exhibits 1/f noise, design an experiment to observe it, and present your results. The experiment I conducted to show the existence of 1/f noise is based on the observation of a photo resistor. A sensitive photo resistor will change its resistance according to the light condition of its surroundings. But there is certain amount of noise in the circuit. Assuming thermal noise and white noise, we need to find the noise at a relatively lower frequency to show 1/f noise. We read from the output of our photo resistor sensor module, which shows the voltage across itself. From this sensitive output, we could tell the fluctuation of the reading. By repeating this for at a longer period of time, we could fine the pattern of the noise by looking at the power spectral density plot. We expected to see a higher power spectral density at a lower frequency according to the definition of 1/f noise and the decay is approximately 1/f.

## 2 Experiment

The experiment was conducted in an enclosed environment. All the sensors, Arduino and breadboard were placed inside a sealed paper box where outside light would not interfere the environment inside. The insed environment was not dark. I placed a LED red in it and use the Arduino to drive it and it will give the photo resistor a stable light beam. The breadboard and Arduino were fixed in the box so that any tiny movements would not interfere the results.

The arduino was attached to the usb port so that it could read from the analog continuously. My sampling frequency is 1 second in this case. The experiments I conducted the experiment twice: the first time lasted for more than 33 hours and the second one lasted for more than 27 hours in order to compare and observe the experiment under low frequency. By reading the output of the analog pin A0, we could determine the light condition in the box. We could see the output of the A0 is fluctuating due to the noise from time to time.

## 3 Hardware Design

### 3.1 Components

1 x Arduino UNO

1 x Breadboard

1 x Photo Resistor(embeded a 10kΩ Resistor)

1 x Red LED

1 x 10kΩ Resistor

Several Jump Wires

### 3.2 Photo Resistor Setup

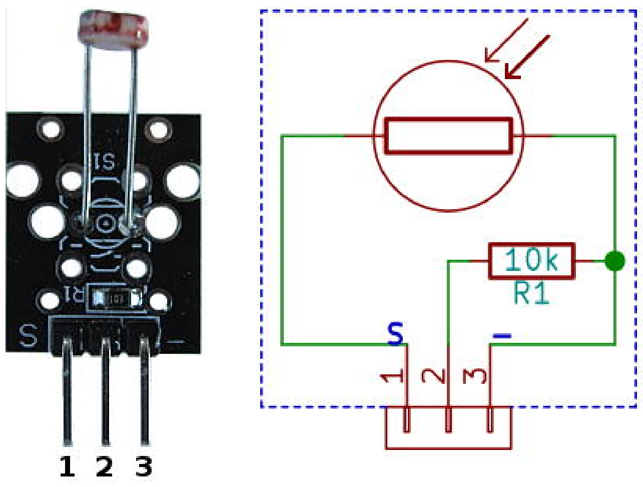


Figure 1: Photo Resistor Sensor Module Diagram

Configuration:
Pin 1 (S) of the module connects to the Arduino 5V pin.
Pin 2 of the module connects to the Arduino GND pin.
Pin 3 (-) of the module connects to an Arduino analog input pin. Arduino analog input A0 is used in the sketches below.

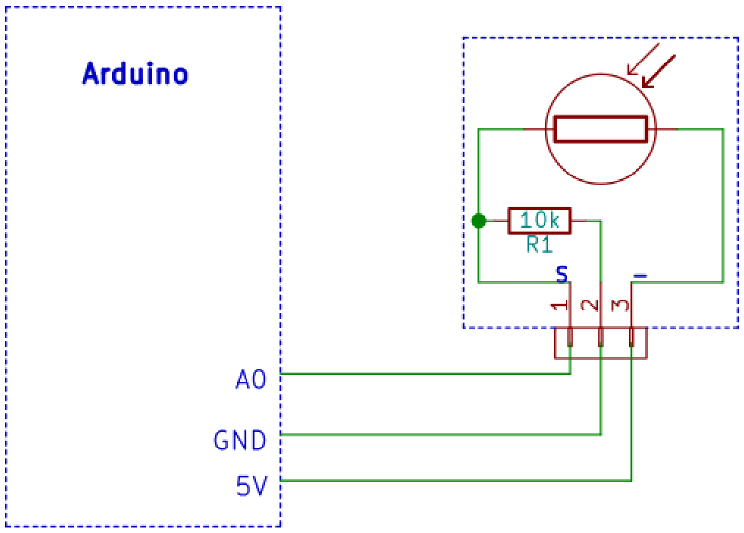


Figure 2: Photo Resistor Sensor Module Configuration Diagram

## 3.3 LED Setup

Setting up a Red Led on the board with the digital pin from Arduino:

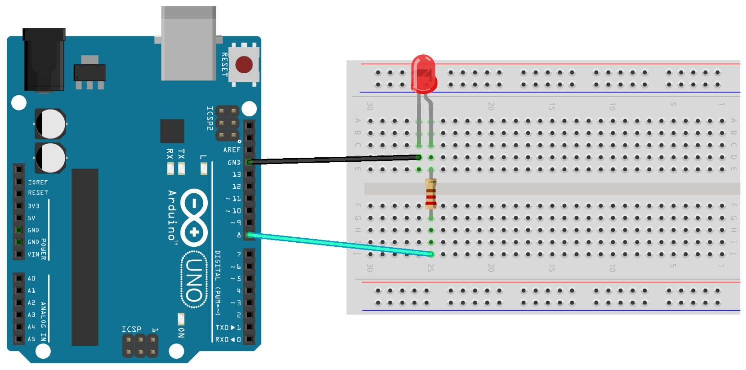


Figure 3: LED Diagram

## 3.4 Circuit Setup

We will connect our circuit according to this Diagram:

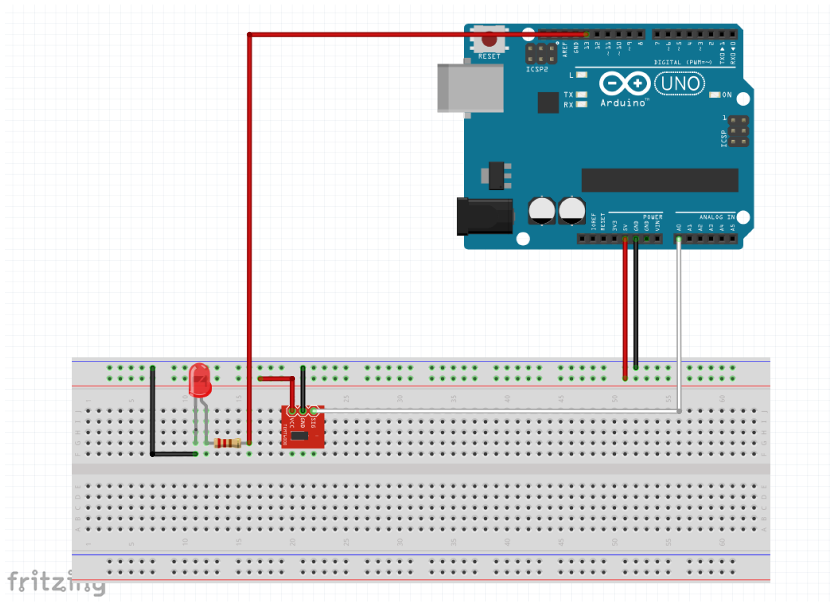


Figure 4: Circuit diagram

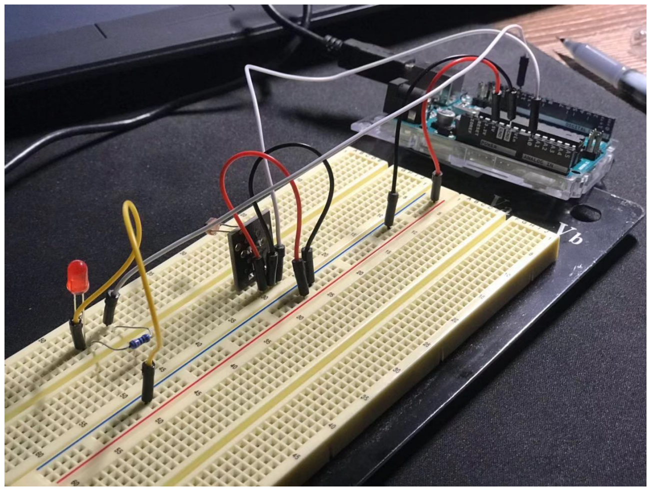


Figure 5: Circuit-1

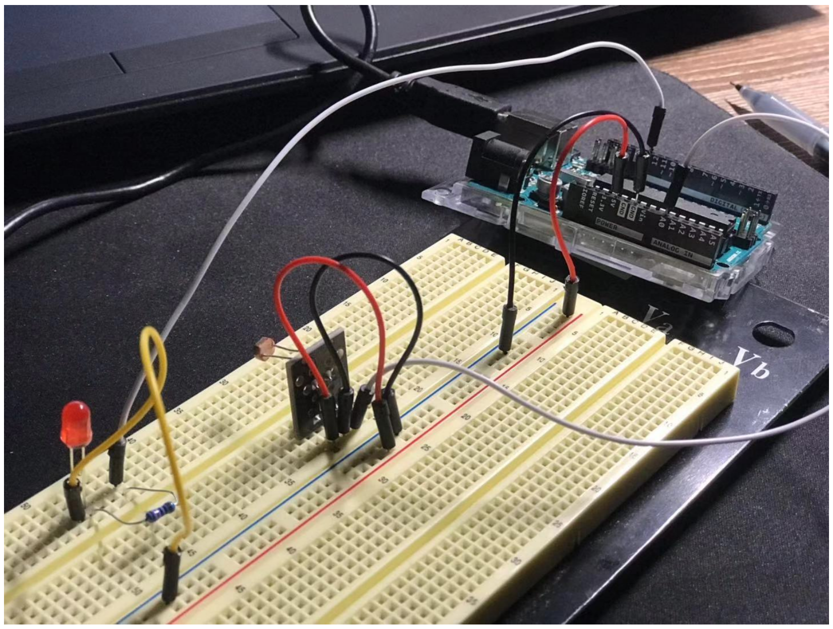


Figure 6: Circuit-2

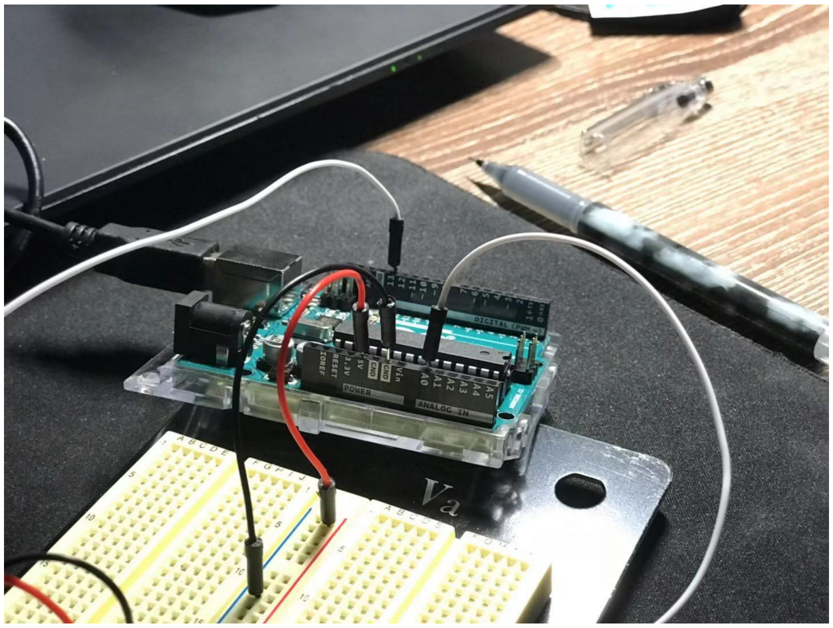


Figure 7: Circuit-3

## 4 Code Design

### 4.1 Photo Resistor

For the photo resistor, we could initialize the pin and analog read from the pin to get the results. The results is within the range of 0 and 1023. Given the voltage of the circuit is 5V, we could calculate the voltage by some maths. The code of reading the analog pin is given below:

```
int sensorPin = A0;   // select the analog input pin for the photoresistor

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(sensorPin));
  delay(200);
}
```

### 4.2 LED

We could use the following code to drive a led by setting the digital output as high:

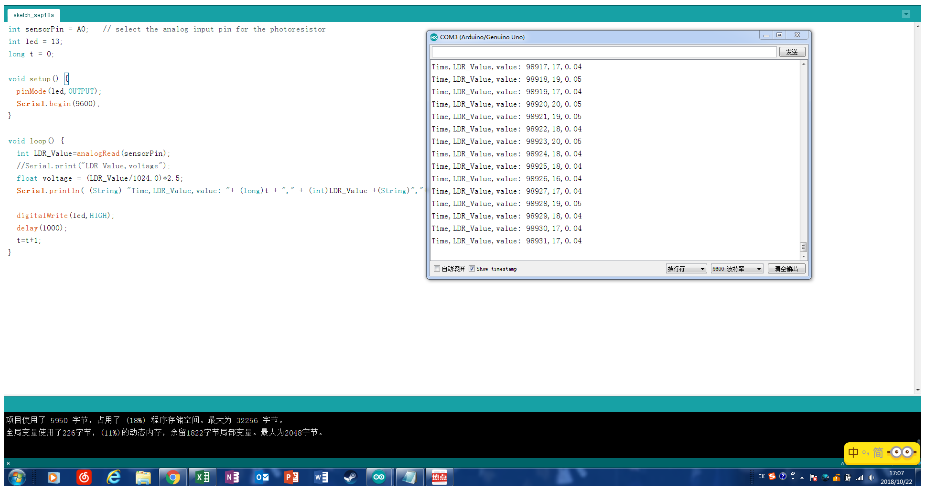
```
int led = 13;
void setup() {
  pinMode(led,OUTPUT);
}

void loop() {
  digitalWrite(led,HIGH);
}
```

### 4.3 Circuit

```
int sensorPin = A0;   // select the analog input pin for the photoresistor
int led = 13;
long t = 0; //Timestamp

void setup() {
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int LDR_Value=analogRead(sensorPin);
  digitalWrite(led,HIGH);
  float voltage = (LDR_Value/1024.0)*2.5;
  Serial.println( (String) "Time,LDR_Value,value: "
                           + (long)t + "," + (int)LDR_Value
                           +(String)","+(float)voltage);

  delay(1000);//Sample rate: 1 second
  t=t+1;
}
```

## 5 Observation

We could observe the output and fluctuation directly through serial output:



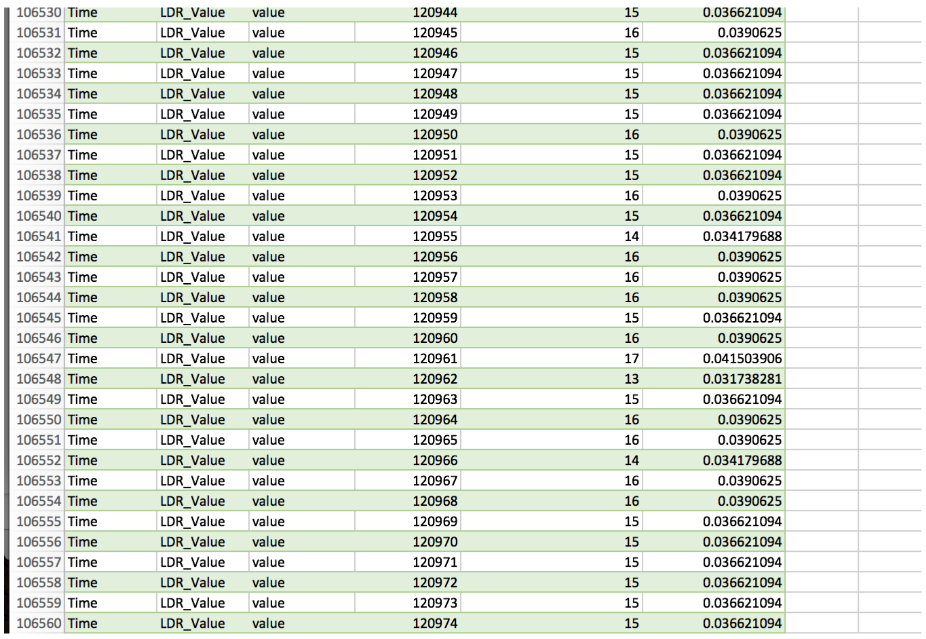
```
int sensorPin = A0;    // select the analog input pin for the photoresistor
int led = 13;
long t = 0;

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int LDR_Value=analogRead(sensorPin);
  //Serial.print("LDR_Value,voltage");
  float voltage = (LDR_Value/1024.0)*2.5;
  Serial.println( (String) "Time,LDR_Value,value: "+ (long)t + "," + (int)LDR_Value +(String)","+

  digitalWrite(led,HIGH);
  delay(1000);
  t=t+1;
}
```

```
Time, LDR_Value, value: 98917, 17, 0.04
Time, LDR_Value, value: 98918, 19, 0.05
Time, LDR_Value, value: 98919, 17, 0.04
Time, LDR_Value, value: 98920, 20, 0.05
Time, LDR_Value, value: 98921, 19, 0.05
Time, LDR_Value, value: 98922, 18, 0.04
Time, LDR_Value, value: 98923, 20, 0.05
Time, LDR_Value, value: 98924, 18, 0.04
Time, LDR_Value, value: 98925, 18, 0.04
Time, LDR_Value, value: 98926, 16, 0.04
Time, LDR_Value, value: 98927, 17, 0.04
Time, LDR_Value, value: 98928, 19, 0.05
Time, LDR_Value, value: 98929, 18, 0.04
Time, LDR_Value, value: 98930, 17, 0.04
Time, LDR_Value, value: 98931, 17, 0.04
```

```
项目使用了 5950 字节，占用了 (18%) 程序存储空间。最大为 32256 字节。
全局变量使用了226字节，(11%)的动态内存，余留1822字节局部变量。最大为2048字节。
```

Figure 8: Serial Output

We then copy the data and format the data into csv:



| | | | | | | |
|---|---|---|---|---|---|---|
| 106530 | Time | LDR_Value | value | 120944 | 15 | 0.036621094 |
| 106531 | Time | LDR_Value | value | 120945 | 16 | 0.0390625 |
| 106532 | Time | LDR_Value | value | 120946 | 15 | 0.036621094 |
| 106533 | Time | LDR_Value | value | 120947 | 15 | 0.036621094 |
| 106534 | Time | LDR_Value | value | 120948 | 15 | 0.036621094 |
| 106535 | Time | LDR_Value | value | 120949 | 15 | 0.036621094 |
| 106536 | Time | LDR_Value | value | 120950 | 16 | 0.0390625 |
| 106537 | Time | LDR_Value | value | 120951 | 15 | 0.036621094 |
| 106538 | Time | LDR_Value | value | 120952 | 15 | 0.036621094 |
| 106539 | Time | LDR_Value | value | 120953 | 16 | 0.0390625 |
| 106540 | Time | LDR_Value | value | 120954 | 15 | 0.036621094 |
| 106541 | Time | LDR_Value | value | 120955 | 14 | 0.034179688 |
| 106542 | Time | LDR_Value | value | 120956 | 16 | 0.0390625 |
| 106543 | Time | LDR_Value | value | 120957 | 16 | 0.0390625 |
| 106544 | Time | LDR_Value | value | 120958 | 16 | 0.0390625 |
| 106545 | Time | LDR_Value | value | 120959 | 15 | 0.036621094 |
| 106546 | Time | LDR_Value | value | 120960 | 16 | 0.0390625 |
| 106547 | Time | LDR_Value | value | 120961 | 17 | 0.041503906 |
| 106548 | Time | LDR_Value | value | 120962 | 13 | 0.031738281 |
| 106549 | Time | LDR_Value | value | 120963 | 15 | 0.036621094 |
| 106550 | Time | LDR_Value | value | 120964 | 16 | 0.0390625 |
| 106551 | Time | LDR_Value | value | 120965 | 16 | 0.0390625 |
| 106552 | Time | LDR_Value | value | 120966 | 14 | 0.034179688 |
| 106553 | Time | LDR_Value | value | 120967 | 16 | 0.0390625 |
| 106554 | Time | LDR_Value | value | 120968 | 16 | 0.0390625 |
| 106555 | Time | LDR_Value | value | 120969 | 15 | 0.036621094 |
| 106556 | Time | LDR_Value | value | 120970 | 15 | 0.036621094 |
| 106557 | Time | LDR_Value | value | 120971 | 15 | 0.036621094 |
| 106558 | Time | LDR_Value | value | 120972 | 15 | 0.036621094 |
| 106559 | Time | LDR_Value | value | 120973 | 15 | 0.036621094 |
| 106560 | Time | LDR_Value | value | 120974 | 15 | 0.036621094 |

Figure 9: CSV Example

Matlab provides numerous methods for taking my raw recorded signal and producing a power spectral density estimate. In this case, I obtain nonparametric power spectral density (PSD) estimates equivalent to the periodogram using fft. The sampling frequency is 1 Hz. The signal length is more than 98900 samples. Use the default settings of the random number generator for reproducible results. Obtain the periodogram using fft. The signal is real-valued and has even length. We could then use loglog() to do the log plot(it will calculate the log).

```
X = csvread('data_2.csv',2,4); \\Read Data from csv
rng default
Fs = 1; \\Sampling Frequency
x = X(:,1);


N = length(x);
xdft = fft(x);
xdft = xdft(1:N/2+1);
psdx = (1/(Fs*N)) * abs(xdft).^2;
psdx(2:end-1) = 2*psdx(2:end-1);
freq = 0:Fs/length(x):Fs/2;

loglog(freq,psdx);
```

```
grid on
title('Periodogram Using FFT')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/kHz)')
```

By running this matlab script, we could find the power spectral density of our noise:

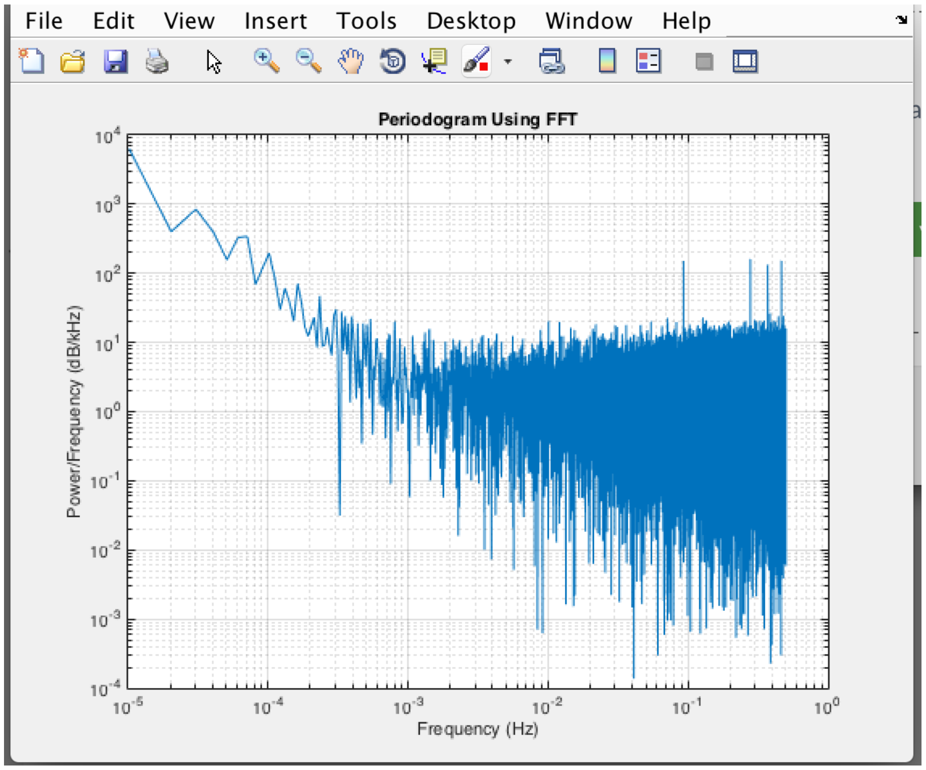


Figure 10: Power Spectral Density of the noise(27 hours)

By observing the image we could find the lower frequency part does have a higher power density of noise(red part) while the higher frequency part(blue part) are generally shot noise. 1/f noise only exists in a certain frequency:

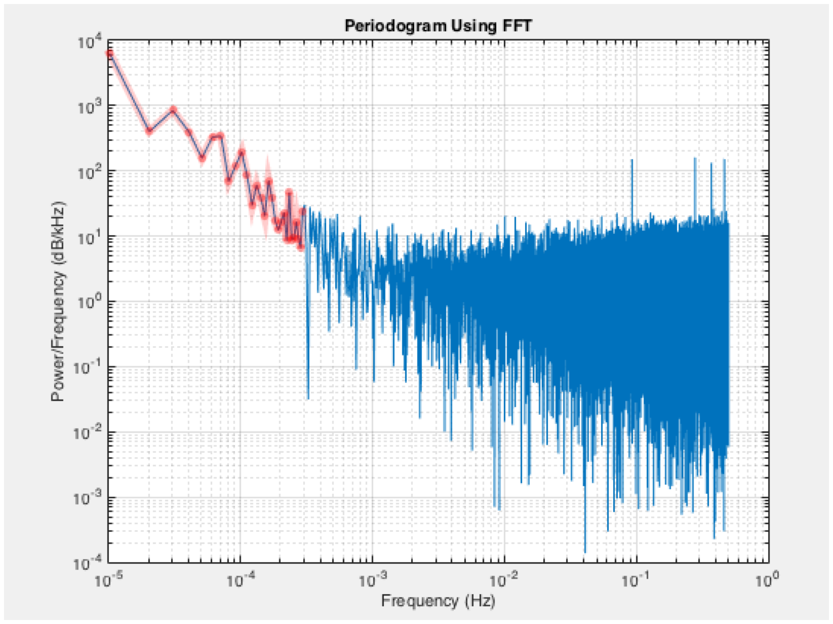


Figure 11: 1/f noise(27 hours)

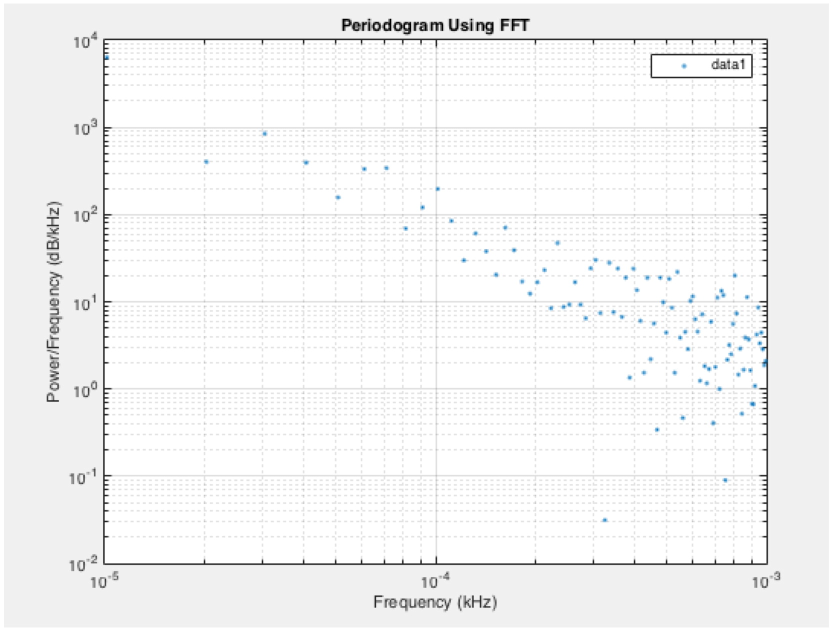


Figure 12: 1/f noise(27 hours)

From this figure, we could tell that the power spectral density of the noise is just the inverse of its frequency, which fulfills the definition of 1/f noise. We could use the fit function in matlab to find the fit line for these points:

```
a=freq(m);
b=psdx(m);
loglog(a,b,'.');
hold on;
p =fit(transpose(a),b,'poly1')
```

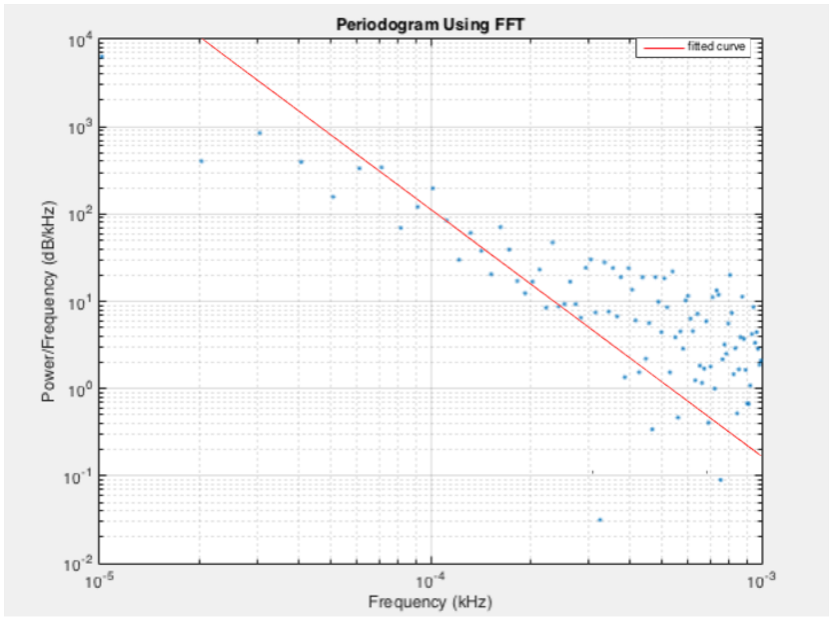


Figure 13: 1/f noise(27 hours)

```
p =
     Linear model Poly1:
     p(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =      -1.636  (-1.886, -1.386)
       p2 =      -10.79  (-12.78, -8.805)
```

This approximately -20db/decade.

## 6 Conclusion

From figure 11 through 12, we could identify the noise as 1/f noise according to the definition of 1/f noise:

$$S(f) = S(1) * \frac{1}{f^x}$$

where x typically is around 1. In our circuit, this means that the 1/f only exits in a low frequency. Through this lab, we could clearly see the existence and behavior of 1/f noise as well as how it will change our circuits.
Below is a graph of the comparison of the two datasets I collected, we could see that 33 hours do have more lower frequency data because we collected 6 more hours. The general trend of the two data are identical:

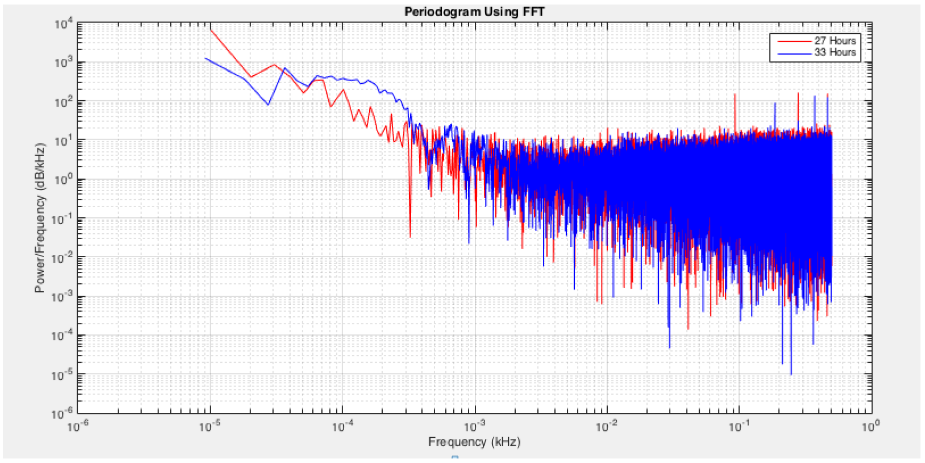


Figure 14: Comparison 27 hours VS. 33 hours

Knowing that flicker noise is a low-frequency effect, we could modulate it into a frequency band outside the signal band. This is known as chopping.