1. **实现思路：展示tel.wav的时域图，通过截取每1000ms的时域图，转化为频谱图，即可看到每一次按键所表示的频率叠加，再结合按键工作原理，确定对应的按键。**
2. **0-1s f1 = 1332, f2 = 849 corresponding to 8**

**1-2s f1 = 769, f2 = 1483 corresponding to 6**

**2-3s f1 = 850, f2 = 1208 corresponding to 7**

**4-5s f1 = 772, f2 = 1338 corresponding to 5**

**5-6s f1 = 694, f2 = 1477 corresponding to 2**

**6-7s f1 = 942, f2 = 1337 corresponding to 0**

**7-8s f1 = 847, f2 = 1469 corresponding to 9**

**按键1的代码：**

|  |
| --- |
| [x, Fs\_x] = audioread('tel.wav')  N = Fs\_x;  k = 0:1:N-1;    x = 0.5 \* sin(697\*2\*pi/N\*k) +0.5 \* sin(1209\*2\*pi/N\*k);  x = x(1:1000);  plot(x)  y = fft(x, N);  plot(k, real(y));  button_1 |

**按键3的代码：**

|  |
| --- |
| [x, Fs\_x] = audioread('tel.wav')  N = Fs\_x;  k = 0:1:N-1;    x = 0.5 \* sin(697\*2\*pi/N\*k) +0.5 \* sin(1477\*2\*pi/N\*k);  x = x(1:1000);  plot(x)  y = fft(x, N);  plot(k, real(y));  button_3 |

**按键5的代码：**

|  |
| --- |
| [x, Fs\_x] = audioread('tel.wav')  N = Fs\_x;  k = 0:1:N-1;    x = 0.5 \* sin(767\*2\*pi/N\*k) +0.5 \* sin(1333\*2\*pi/N\*k);  x = x(1:1000);  plot(x)  y = fft(x, N);  plot(k, real(y));  button_5 |