Código

main.m

|  |
| --- |
| close all**;**  clear**;**  clc**;**  clear **global;** #Limpa variavel **global**  **function** aperta**(**ObjH**,** eventdata**,** tecla**,** txt\_frm**,** g11**,** g12**,** g21**,** g22**)**  #disp**(**tecla**)**  % Call DTMF  fs **=** 8000**;** # Frequencia de amostragem **[**Hz**]**  sinal\_out **=** dtmf\_gen**(**tecla**,** fs**);**  % Adiciona ruido  sinal\_in **=** sinal\_out **+** 0.5 **\*** randn**(**size**(**sinal\_out**));**  #sinal\_in **=** sinal\_in**/**max**(**abs**(**sinal\_in**));** #Normalizando o sinal  % Grava audio  #audiowrite**(**'sinal.wav'**,** sinal**,** fs**);**  % Toca audio  playblocking**(**audioplayer**(**sinal\_out**,** fs**,** 24**));**  % Transformada de Fourrier para o sinal (Verificação)  #figure**([],** 'name'**,** 'FFT'**,** 'numbertitle'**,** 'off'**);**  #stem**(**1**:(**8000**/**length**(**sinal**)):**8000**,** abs**(**fft**(**sinal**)),** '.'**);**  %% Inserindo dados nos gráficos  l **=** length**(**sinal\_out**);**  x **=** 0**:**1**/**8000**:**l**/**fs**;**  x**(**end**)** **=** **[];**  % Gráficos de saida  set**(**g11**,** 'xdata'**,** x**);**  set**(**g11**,** 'ydata'**,** sinal\_out**);**  set**(**g12**,** 'xdata'**,** 1**:(**fs**/**length**(**sinal\_out**)):**fs**);**  set**(**g12**,** 'ydata'**,** abs**(**fft**(**sinal\_out**)));**  % Gráficos de entrada  set**(**g21**,** 'xdata'**,** x**);**  set**(**g21**,** 'ydata'**,** sinal\_in**);**  set**(**g22**,** 'xdata'**,** 1**:(**fs**/**length**(**sinal\_in**)):**fs**);**  set**(**g22**,** 'ydata'**,** abs**(**fft**(**sinal\_in**)));**  % Call servico  #texto **=** char**([**get**(**txt\_frm**,** 'string'**),** 10**,** servico**(**sinal**,** fs**)]);**  set**(**txt\_frm**,** 'string'**,** servico**(**sinal\_in**,** fs**));**  **endfunction**  % GUI janela  MainFrm **=** figure**(**1**,** 'name'**,** 'DTMF'**,** 'numbertitle'**,** 'off'**);**  % Container de texto  texto **=** uipanel**(**MainFrm**,** ...  'units'**,** 'normalized'**,** ...  'position'**,** **[**0.3**,** 0**,** 0.7**,** 0.5**]);**  txt\_frm **=** uicontrol**(**texto**,** ...  'style'**,** 'text'**,** ...  'string'**,** ''**,** ...  'verticalalignment'**,** 'top'**,** ...  'horizontalalignment'**,** 'left'**,** ...  'units'**,** 'normalized'**,** ...  'position'**,** **[**0**,** 0**,** 1**,** 1**],** ...  'backgroundcolor'**,** **[**1 1 1**]);**  % Conteiner plot 1  grafico1 **=** uipanel**(**MainFrm**,** ...  'title'**,** 'Sinal saída'**,** ...  'units'**,** 'normalized'**,** ...  'position'**,** **[**0**,** 0.5**,** 0.5**,** 0.5**]);**  #Setando Grafico 11  subplot**(**2**,** 1**,** 1**,** 'parent'**,** grafico1**);**  g11 **=** plot**([**1**],** 'marker'**,** 'none'**);**  title**(**'Sinal'**);**  xlabel**(**'Tempo [s]'**);**  grid on**;**  grid minor on**;**  #Setando Grafico 12  subplot**(**2**,** 1**,** 2**,** 'parent'**,** grafico1**);**  g12 **=** plot**([**1**],** 'marker'**,** 'none'**);**  title**(**'Sinal (FFT)'**);**  xlabel**(**'Frequencia [Hz]'**);**  grid on**;**  grid minor on**;**  % Conteiner plot 2  grafico2 **=** uipanel**(**MainFrm**,** ...  'title'**,** 'Sinal entrada'**,** ...  'units'**,** 'normalized'**,** ...  'position'**,** **[**0.5**,** 0.5**,** 0.5**,** 0.5**]);**  #Setando Grafico 21  subplot**(**2**,** 1**,** 1**,** 'parent'**,** grafico2**);**  g21 **=** plot**([**1**],** 'marker'**,** 'none'**);**  title**(**'Sinal'**);**  xlabel**(**'Tempo [s]'**);**  grid on**;**  grid minor on**;**  #Setando Grafico 22  subplot**(**2**,** 1**,** 2**,** 'parent'**,** grafico2**);**  g22 **=** plot**([**1**],** 'marker'**,** 'none'**);**  title**(**'Sinal (FFT)'**);**  xlabel**(**'Frequencia [Hz]'**);**  grid on**;**  grid minor on**;**  % Conteiner teclado  teclado **=** uipanel**(**MainFrm**,** ...  'units'**,** 'normalized'**,** ...  'position'**,** **[**0**,** 0**,** 0.3**,** 0.5**]);**  #Codigo ascii do caractere**|**indice**|**cor  teclas **=** **[**  49 1 **[**0.90980 0.90980 0.90588**]**  50 2 **[**0.90980 0.90980 0.90588**]**  51 3 **[**0.90980 0.90980 0.90588**]**  65 12 **[**242**/**255 165**/**255 165**/**255**]**  52 4 **[**0.90980 0.90980 0.90588**]**  53 5 **[**0.90980 0.90980 0.90588**]**  54 6 **[**0.90980 0.90980 0.90588**]**  66 13 **[**242**/**255 165**/**255 165**/**255**]**  55 7 **[**0.90980 0.90980 0.90588**]**  56 8 **[**0.90980 0.90980 0.90588**]**  57 9 **[**0.90980 0.90980 0.90588**]**  67 14 **[**242**/**255 165**/**255 165**/**255**]**  42 10 **[**153**/**255 190**/**255 255**/**255**]**  48 0 **[**0.90980 0.90980 0.90588**]**  35 11 **[**153**/**255 190**/**255 255**/**255**]**  68 15 **[**242**/**255 165**/**255 165**/**255**]**  **];**  **for** i **=** 1**:**1**:**16**,**  d **=** i**/**4**;**  m **=** ceil**(**d**)-**1**;**  uicontrol**(**teclado**,** ...  'style'**,** 'pushbutton'**,** ...  'backgroundcolor'**,** **[**teclas**(**i**,** 3**)** teclas**(**i**,** 4**)** teclas**(**i**,** 5**)],** ...  'string'**,** char**(**teclas**(**i**,** 1**)),** ...  'units'**,** 'normalized'**,** ...  'position'**,** **[**d**-**m**-**0.25**,** 0.75**-(**m**\***0.25**),** 0.25**,** 0.25**],** ...  'callback'**,** **{@**aperta**,** teclas**(**i**,** 2**),** txt\_frm**,** g11**,** g12**,** g21**,** g22**});**  **endfor**  **set(**txt\_frm**,** 'string'**,** 'URASim'**);** #Indica sistema pronto  % Limpeza  clear teclas t i m d**;** #Limpa variaveis de construção do teclado  clear MainFrm grafico1 grafico2 texto**;** # Limpa variaveis de construção iu |

serviço.m

|  |
| --- |
| **function** lambda **=** servico**(**sinal**,** fs**)**  **global** historico **=** '@'**;**  raiz **=** 'roteiro/'**;**  tecla **=** detecta\_tecla**(**sinal**,** fs**,** 0.3**);**  **if** **(**historico **==** '@'**)** #Primeira execução  lambda **=** ler\_arquivo**([**raiz '\_start.txt'**]);**  historico **=** **[**''**];**  **elseif** **(**historico **==** '$'**)** #Fim do programa  lambda **=** ler\_arquivo**([**raiz '\_end.txt'**]);**  **else**  **if** **(**exist**([**raiz **[**historico char**(**tecla**)]** '.txt'**])** **==** 0**)**  # Caso o arquivo não exista (Opção invalida)  lambda **=** ler\_arquivo**([**raiz '\_invalid.txt'**]);**  **if** **(**length**(**historico**)** **==** 0**)** # Opção invalida no inicio, historico vazio  lambda **=** **[**lambda ler\_arquivo**([**raiz '\_start.txt'**])];**  **else**  lambda **=** **[**lambda ler\_arquivo**([**raiz historico '.txt'**])];**  **endif**  **else**  tipo **=** ler\_arquivo**([**raiz historico char**(**tecla**)** '.txt'**],** true**);**  **if** strcmpi**(**tipo**,** 'end'**)** #Ultima opcção  lambda **=** ler\_arquivo**([**raiz historico char**(**tecla**)** '.txt'**]);**  historico **=** '$'**;**  **elseif** strcmpi**(**tipo**,** 'return'**)** #Retorno  historico**(**end**)** **=** **[];**  lambda **=** ler\_arquivo**([**raiz historico '.txt'**]);**  **else**  historico **=** **[**historico char**(**tecla**)];**  lambda **=** ler\_arquivo**([**raiz historico '.txt'**]);**  **endif**  **endif**  **endif**  lambda **=** **[**char**(**tecla**)** '-' char**(**10**)** lambda**];**  **endfunction** |

detecta\_tecla.m

|  |
| --- |
| **function** lambda **=** detecta\_tecla**(**sinal**,** fs**,** sensibilidade**)**  fh **=** **[**  1209  1336  1477  1633  **];**  fl **=** **[**  697  770  852  941  **];**  tecla1 **=** **[**  '10001000'  '01001000'  '00101000'  '10000100'  '01000100'  '00100100'  '10000010'  '01000010'  '00100010'  '01000001'  '10000001'  '00100001'  '00011000'  '00010100'  '00010010'  '00010001'  **];**  tecla2 **=** **[**  49  50  51  52  53  54  55  56  57  48  120  104  65  66  67  68  **];**  codigo **=** **[**''**];**  **for** f **=** fh**',**  codigo **=** **[**codigo int2str**(**filtro\_detecta\_faixa**(**sinal**,** f**-**50**,** f**+**50**,** 50**,** fs**,** sensibilidade**))];**  endfor  **for** f **=** fl**'**  codigo **=** **[**codigo int2str**(**filtro\_detecta\_faixa**(**sinal**,** f**-**50**,** f**+**50**,** 50**,** fs**,** sensibilidade**))];**  **endfor**  #disp(index(tecla1, codigo));  lambda **=** tecla2**(**index**(**tecla1**,** codigo**)** **==** 1**);**  **endfunction**  **function** lambda **=** filtro\_detecta\_faixa**(**sinal**,** f\_1**,** f\_2**,** f\_d**,** f\_s**,** sensibilidade**)**  h **=** passa\_faixa**(**f\_1**,** f\_2**,** f\_d**,** f\_s**);**  s\_f **=** conv**(**sinal**,** h**);** #Sinal filtrado  **if** **(**mean**(**abs**(**s\_f**))>=**sensibilidade**)**  lambda **=** true**;**  **else**  lambda **=** false**;**  **endif**  **endfunction**  **function** lambda **=** passa\_faixa**(**f\_1**,** f\_2**,** f\_d**,** f\_s**)**  lambda **=** conv**(**passa\_baixa**(**f\_d**,** f\_2**,** f\_s**),** passa\_alta**(**f\_d**,** f\_1**,** f\_s**));**  **endfunction**  **function** lambda **=** passa\_baixa**(**deltaf**,** fc**,** fs**)**  # Trecho para arredondar para o numero impar  n **=** ceil**(**fs**/**deltaf**);**  **if** **(**rem**(**n**,** 2**)** **==** 0**)**  n **=** n **+** 1**;**  **endif**  x **=** **-((**n**-**1**)/**2**):**1**:((**n**-**1**)/**2**);**  x**(**x**==**0**)** **=** 10e-10**;**  lambda **=** sin**(**x**\***2**\***pi**\***fc**/**fs**)./(**pi**\***x**);**  **endfunction**  **function** lambda **=** passa\_alta**(**deltaf**,** fc**,** fs**)**  h **=** passa\_baixa**(**deltaf**,** fc**,** fs**);**  N **=** numel**(**h**);**  n**=-(**N**-**1**)/**2**:(**N**-**1**)/**2**;**  impulso **=** zeros**(**size**(**n**));**  impulso**(**n**==**0**)** **=** 1**;**  lambda **=** impulso **-** h**;**  **endfunction** |

dtmf\_gen.m

|  |
| --- |
| **function** lambda **=** dtmf\_gen**(**tecla**,** fs**)**  lista **=** **[**  2 4  1 1  2 1  3 1  1 2  2 2  3 2  1 3  2 3  3 3  1 4  3 4  4 1  4 2  4 3  4 4  **];**  fg **=** **[**1209 1336 1477 1633**];**  fb **=** **[**697 770 852 941**];**  #fs = 8000;  t **=** 0**:**1**/**fs**:**0.3**;**  lambda **=** **(**sin**(**2**\***pi**\***fg**(**lista**(**tecla **+** 1**,** 1**)).\***t**)** **+** sin**(**2**\***pi**\***fb**(**lista**(**tecla **+** 1**,** 2**)).\***t**))/**2**;**  #lambda = sin(2\*pi\*697.\*t) + sin(2\*pi\*1477.\*t);  **endfunction**  % Observação  % As teclas vão de 0 a 11, sendo:  # de 0 a 9 numéricas normais  # 10 -> \*  # 11 -> #  # 12 a 15 -> A a D |

ler\_arquivo.m

|  |
| --- |
| % Se o arquivo não existir retorna 0  % Variavel tipo determina se retorna apenas a primeira linha (Sem o primeiro  %caractere) ou se retorna o resto do texto  **function** lambda **=** ler\_arquivo**(**caminho**,** tipo**=**false**)**  **if** **(**exist**(**caminho**)==**0**)**  lambda **=** 0**;**  **else**  temp **=** fileread**(**caminho**);**  #lambda = temp;  i **=** strfind**(**temp**,** char**([**13 10**]))(**1**);**  **if** tipo  lambda **=** temp**(**2**:**i**-**1**);**  **else**  lambda **=** temp**(**i **+** 2**:**end**);**  **endif**  **endif**  **endfunction** |

Simulação





