

Controle de temperatura

Gerado por Doxygen 1.9.2

1 Índice dos Arquivos	1
1.1 Lista de Arquivos	1
2 Arquivos	3
2.1 Referência do Arquivo D:/BACKUP/Faculdade/16_Embarcados/Controlador_Temperatura/Controle- de-temperatura/Codigos/controle_temperatura_RTOS/Core/Src/FIRFilter.c	3
2.1.1 Descrição detalhada	3
2.1.2 Funções	3
2.1.2.1 FIRFilter_Init()	4
2.1.2.2 FIRFilter_Update()	4
2.2 FIRFilter.c	5
Índice Remissivo	7

Capítulo 1

Índice dos Arquivos

1.1 Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

D:/BACKUP/Faculdade/16_Embarcados/Controlador_Temperatura/Controle-de-temperatura/Codigos/controle↵
_temperatura_RTOS/Core/Src/[FIRFilter.c](#)
Implementação do filtro FIR 3

Capítulo 2

Arquivos

2.1 Referência do Arquivo D:/BACKUP/Faculdade/16_Embarcados/↵ Controlador_Temperatura/Controle-de-temperatura/↵ Codigos/control_e_temperatura_RTOS/Core/Src/FIRFilter.c

Implementação do filtro FIR.

```
#include "FIRFilter.h"
```

Funções

- void [FIRFilter_Init](#) (FIRFilter *fir)
- float [FIRFilter_Update](#) (FIRFilter *fir, float inp)

2.1.1 Descrição detalhada

Implementação do filtro FIR.

Autor

Arthur Damasceno

Atenção

Este código apresenta a implementação de filtro FIR com coeficientes calculados previamente

Definição no arquivo [FIRFilter.c](#).

2.1.2 Funções

2.1.2.1 FIRFilter_Init()

```
void FIRFilter_Init (
    FIRFilter * fir )
```

Definição na linha 19 do arquivo [FIRFilter.c](#).

```
00019         {
00020
00021     /* Reinicia o buffer do filtro */
00022     for(uint8_t n=0; n<FIR_FILTER_LENGTH;n++){
00023
00024         fir->buf[n] = 0.0f;
00025
00026     }
00027
00028     /* Reseta o index do buffer */
00029     fir->bufindex = 0;
00030
00031     /* Reseta a saída do filtro */
00032     fir->out = 0.0f;
00033 }
```

2.1.2.2 FIRFilter_Update()

```
float FIRFilter_Update (
    FIRFilter * fir,
    float inp )
```

Definição na linha 35 do arquivo [FIRFilter.c](#).

```
00035         {
00036
00037     /* Guarda a entrada no buffer */
00038     fir->buf[fir->bufindex] = inp;
00039
00040     /* Incrementa o index do buffer e reinicia se necessário */
00041     fir->bufindex++;
00042
00043     if(fir->bufindex == FIR_FILTER_LENGTH){
00044
00045         fir->bufindex = 0;
00046
00047     }
00048
00049     /* Calcula nova saída via convolução */
00050     fir->out = 0.0f;
00051
00052     uint8_t sumIndex = fir->bufindex;
00053
00054     for(uint8_t n=0; n<FIR_FILTER_LENGTH;n++){
00055
00056         /* Decrementa o index e reseta se necessary */
00057         if(sumIndex>0){
00058
00059             sumIndex--;
00060
00061         }else{
00062
00063             sumIndex = FIR_FILTER_LENGTH -1;
00064
00065         }
00066
00067         /* Multiplica resposta ao impulso com a entrada deslocada e soma a saída */
00068         fir->out += FIR_IMPULSE_RESPONSE[n] * fir->buf[sumIndex];
00069     }
00070
00071     /* Retorna saída filtrada */
00072     return fir->out;
00073
00074 }
```


2.2 FIRFilter.c

Vá para a documentação desse arquivo.

```
00001
00014 #include "FIRFilter.h"
00015
00016 /* Coeficientes do filtro */
00017 static float FIR_IMPULSE_RESPONSE[FIR_FILTER_LENGTH] = {0.02840647f, 0.23700821f, 0.46917063f,
00018 0.23700821f, 0.02840647f};
00019
00018 void FIRFilter_Init(FIRFilter *fir){
00020
00021     /* Reinicia o buffer do filtro */
00022     for(uint8_t n=0; n<FIR_FILTER_LENGTH;n++){
00023
00024         fir->buf[n] = 0.0f;
00025
00026     }
00027
00028     /* Reseta o index do buffer */
00029     fir->bufindex = 0;
00030
00031     /* Reseta a saída do filtro */
00032     fir->out = 0.0f;
00033 }
00034
00035 float FIRFilter_Update(FIRFilter *fir, float inp){
00036
00037     /* Guarda a entrada no buffer */
00038     fir->buf[fir->bufindex] = inp;
00039
00040     /* Incrementa o index do buffer e reinicia se necessário */
00041     fir->bufindex++;
00042
00043     if(fir->bufindex == FIR_FILTER_LENGTH){
00044
00045         fir->bufindex = 0;
00046
00047     }
00048
00049     /* Calcula nova saída via convolução */
00050     fir->out = 0.0f;
00051
00052     uint8_t sumIndex = fir->bufindex;
00053
00054     for(uint8_t n=0; n<FIR_FILTER_LENGTH;n++){
00055
00056         /* Decrementa o index e reseta se necessary */
00057         if(sumIndex>0){
00058
00059             sumIndex--;
00060
00061         }else{
00062
00063             sumIndex = FIR_FILTER_LENGTH -1;
00064
00065         }
00066
00067         /* Multiplica resposta ao impulso com a entrada deslocada e soma a saída */
00068         fir->out += FIR_IMPULSE_RESPONSE[n] * fir->buf[sumIndex];
00069     }
00070
00071     /* Retorna saída filtrada */
00072     return fir->out;
00073 }
00074 }
```


Índice Remissivo

D:/BACKUP/Faculdade/16_Embarcados/Controlador_Temperatura/Controle-
de-temperatura/Codigos/controle_temperatura_RTOS/Core/Src/FIRFilter.c,
[3](#), [5](#)

FIRFilter.c

FIRFilter_Init, [3](#)

FIRFilter_Update, [4](#)

FIRFilter_Init

FIRFilter.c, [3](#)

FIRFilter_Update

FIRFilter.c, [4](#)