INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS ELETRÔNICOS

ANÁLISE DOS SINAIS DE EEG COM HISTOGRAMA E MÉDIA MÓVEL

Trabalho de Conclusão da disciplina de equipamentos eletro médicos submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.  
Professor Orientador: Fernanda Isabel Marques Argoud. Dr.Eng.

BRUNO ENGELBERT

**FLORIANÓPOLIS, 2010**

# MICROCONTROLADOR

O microcontrolador utilizado para a simulação do histograma e da média móvel foi o ATMega328 da Atmel, no qual suas principais características são:

* 32KB de memória *Flash.*
* 2KB de SRAM.
* 1024 *Bytes* EEPROM.
* 20 MIPS trabalhando em 20MHz.

Para gerar o código binário da simulação na plataforma AVR foi utilizado o *toolchain* da AVR-libc fazendo a compilação cruzada do código na plataforma Linux.

Após gerar os binários com o *toolchain* foi utilizado o *software* Proteus para fazer as devidas simulações.

# HISTOGRAMA

O histograma é uma forma gráfica de apresentar quantas vezes um determinado valor é repetido.

A Figura 1 ilustra o fluxograma do algoritmo usado para montar o histograma. Primeiro é feito uma varredura no vetor das amostras do Eletro Encefalograma (EEG) procurando os pontos mínimos e máximos. Após encontrar estes valores calculamos o *range* que o histograma trabalhará, diminuindo o ponto máximo menos o ponto mínimo. Uma vez calculado o *range* nós dividimos este pelo número de patamares do histograma e obtemos qual faixa de pontos cada patamar irá representar. O número de patamar foi definido como sendo 64 para melhor visualização do gráfico.

Como o número de pontos amostrados era relativamente grande para caber na memória RAM do microcontrolador, o vetor de amostras foi definido na memória de programa do ATMega328 através de um atributo do compilador.

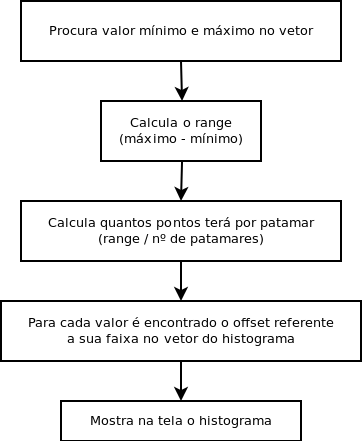


Figura - Fluxograma do algoritmo do histograma.

Abaixo segue o trecho do código responsável pelo cálculo do histograma. O tipo *histogram\_t* foi definido como sendo:

typedef struct \_histogram histogram\_t;

struct \_histogram {

int32\_t min; /\* valor mínimo do vetor. \*/

uint32\_t baseline; /\* faixa de representação de cada patamar \*/

uint32\_t n\_baselines; /\* número de patamares \*/

uint32\_t precision; /\* usado para ganhar mais precisão \*/

uint8\_t histogram[1024]; /\* vetor do histograma \*/

};

Com isso a função escrita em C fica:

void

histogram\_calc(histogram\_t \*t,

float \*samples,

size\_t n\_samples)

{

int32\_t max = 0;

uint32\_t offset = 0;

uint32\_t range = 0;

uint32\_t i;

/\* procura ponto máximo e mínimo \*/

for (i = 0; i < n\_samples; i++) {

max = MAX(max, pgm\_read\_float(&samples[i]) \* t->precision);

t->min = MIN(t->min, pgm\_read\_float(&samples[i]) \* t->precision);

}

/\* calcula o range \*/

range = max - t->min;

/\* calcula quantos pontos serão representados por cada patamar \*/

t->baseline = range / t->n\_baselines;

/\* incrementa em 1 o patamar do histograma referente a cada ponto amostrado \*/

for (i = 0; i < n\_samples; i++) {

offset = ((pgm\_read\_float(&samples[i]) \* t->precision) - t->min) / t->baseline;

if (offset < t->n\_baselines) {

t->histogram[offset]++;

}

}

}

## Simulação

Como já foi dito anteriormente o utilizamos o *software* Proteus para executar as simulações e uma particularidade do *firmware* desenvolvido é que o histograma é executado a partir de um comando enviado pela serial, que neste caso foi criado com nome de *histogram*.

A Figura 2 apresenta a saída do histograma efetuado a partir das amostras do arquivo *teste01.txt*.

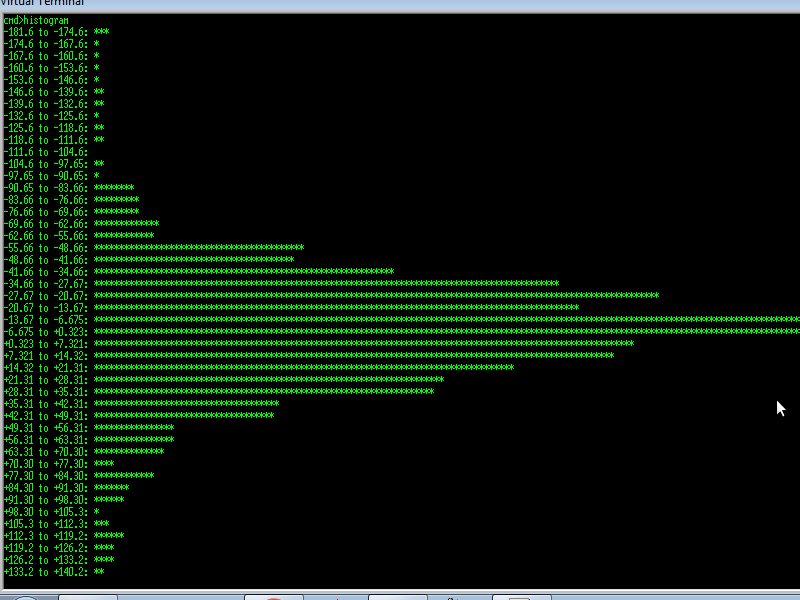


Figura - *Print* do histograma na serial do microcontrolador.

# MÉDIA MÓVEL