

Desenvolvimento de software para Placa de Contagem em plataforma Arduino.

Kevin Nogueira Moreira*, Pedro Henrique Silva Oliveira, Prof. Dr. Cristiano de Mello Gallep.

Resumo

Neste trabalho foram desenvolvidos softwares de linguagem C e de Java, para a interface gráfica de usuário (GUI). Os supracitados foram usados para configurar e operar um sistema eletrônico de baixo custo desenvolvido usando um microcontrolador Arduino, com o objetivo de substituir a placa de contagem de pulsos previamente utilizada, que devido ao seu alto custo, dificulta o acesso a pesquisas na área de biofotônica. Esta placa, conectada a um computador para o seu manuseio, conta os pulsos de um tubo fotomultiplicador (PMT), que faz a medição da emissão espontânea de biofotons em sementes germinando em uma câmara escura no Laboratório de Fotônica Aplicada (LaFA).

Palavras-chave:

Arduino, Biofotons, Software.

Introdução

O código criado em linguagem C funciona como um preset do sistema eletrônico utilizado, é nele que estão os comandos que irão ditar como o micro controlador opera e realiza a contagem de pulsos em si, que acontece através das rotinas de interrupção do micro controlador Arduino [1].

O software em java por sua vez, foi desenvolvido na plataforma Processing para servir como uma interface gráfica de usuário (GUI) em que o indivíduo irá, usando um computador, configurar o modo e período para a aquisição de dados, ou seja, o intervalo de tempo de um loop de contagem e a quantidade de loops em dado experimento. Dessa forma será possível acompanhar em tempo real a contagem de pulsos através de um gráfico "loops x tempo".

Resultados e Discussão

O Arduino foi configurado em C através de seu Ambiente de Desenvolvimento Integrado (IDE). A contagem de pulsos é feita utilizando um contador que opera na borda de descida de uma rotina de interrupção (ISR) de um dos pinos analógicos do circuito que é conectado ao tubo fotomultiplicador (PMT). A cada ciclo de contagem, este é enviado serialmente pelo cabo USB ao computador, onde através do monitor serial do próprio IDE, pode-se observar a contagem.

No ambiente Processing, o código java utilizado se comunica com o Arduino e recebe os valores de contagem serialmente. É neste software que foi criado a interface gráfica com botões no qual o usuário pode definir o número de loops e o intervalo de tempo de cada loop em milissegundos, além de um gráfico em tempo real para facilitar a visualização dos dados adquiridos e acompanhar o andamento do experimento.

Os dados também são salvos em um formato .txt ou .csv, a critério do usuário, de modo que este pode criar uma biblioteca de dados, assim como fazer um melhor tratamento deste em outro software de sua escolha.

```
Trecho_ISR $
ISR(INT0_vect) {
  unsigned int temp;
  contagem++; // Incrementa variável de contagem (contagem inicializa como contagem =1)
  conta[0]= contagem % 10 + 0x30; // Dígito 0 do valor de contagem
  temp= contagem/10;
  conta[1]= temp % 10 + 0x30; // Dígito 1 do valor de contagem
  temp= temp/10;
  conta[2]= temp % 10 + 0x30; // Dígito 2 do valor de contagem
  temp= temp/10;
  conta[3]= temp % 10 + 0x30; // Dígito 3 do valor de contagem
  temp= temp/10;
  conta[4]= temp + 0x30; // Dígito 4 do valor de contagem
}
// ISR do TC1 por overflow
ISR(TIMER1_OVF_vect){
  unsigned int valor;
  TCNT1=-2500; // Valor inicial de contagem do TC1 -- TCNT1 eh o registrador "Timer/Counter".
  time_ms=; // VALOR EM MS DO GATETIME ESCOLHIDO, RECARREGADO TODO LOOP PARA FAZER A CONTAGEM
  if(time_ms==0) {
    time_ms=temp; // Recarrega valor de gatetime
    UDRO=conta[4]; // Dígito 4 do valor de contagem
    while(! (UCSRA & (1<<UDRE0)));
    UDRO=conta[3]; // Envia dígito 3 do valor de contagem
    while(! (UCSRA & (1<<UDRE0)));
    UDRO=conta[2]; // Envia dígito 2 do valor de contagem
    while(! (UCSRA & (1<<UDRE0)));
    UDRO=conta[1]; // Envia dígito 1 do valor de contagem
    while(! (UCSRA & (1<<UDRE0)));
    UDRO=conta[0]; // Envia dígito 0 do valor de contagem
  }
  else
    return;
}
```

Figura 1. Rotina de interrupção do código C.

Conclusões

Os códigos, por serem criados em linguagens de fácil acesso, podem também ser alterados para cumprirem com quaisquer novas necessidades que o usuário queira, sendo assim compatíveis para uso em novos experimentos e pesquisas no campo de biofotônica.

Agradecimentos



1 BERTOIGNA, Eduardo G. Projeto, construção e aplicações de câmara escura portátil para medidas de Bioluminescência ultra-fraca. Tese de Doutorado. Campinas-SP 2013.