ENN251 – Microcontroladores

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Proposta de projeto – Grupo 3

# Participantes

Carlos Eduardo Cardozo RA: 12.02500-3

Ilan Goldman RA: 13.00669-0

Lucas Deliberato RA: 13.00795-5

# Nome do Projeto

Beam Ball

# Descrição do Projeto

O objetivo do projeto será a criação de um controle de posicionamento de uma bola em cima de uma viga. Para tal, serão necessários os seguintes materiais:

* 2 bases de madeira;
* 1 viga de alumínio;
* 1 bola;
* 1 sensor ultrassônico (HC-SR04);
* 1 servo motor (MG 995R TowerPro);
* 1 kit de desenvolvimento Atmel SAN4N – EK2.

## Projeto Mecânico

A mecânica do projeto deve seguir o seguinte modelo (figura 1). A viga de alumínio principal, que vai suportar a bola, estará fixada em uma base de madeira vertical em uma de suas extremidades e conectada a uma das barras de alumínio móveis na outra. Essa barra de alumínio estará presa a outra barra semelhante, que por sua vez deve ficar acoplada ao motor cuja rotação se traduz na inclinação da viga principal.

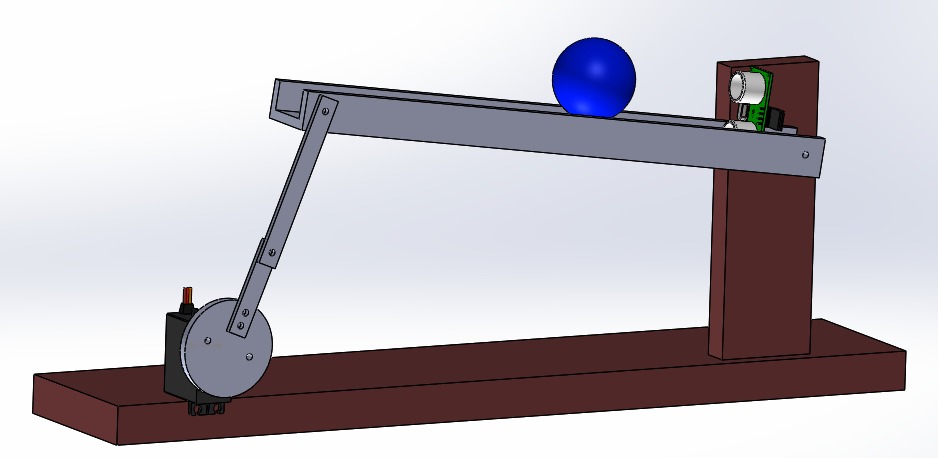


Figura 1: Modelo da construção mecânica do controlador de posicionamento

## Projeto Eletrônico

A parte eletrônica do projeto consiste no sensor ultrassônico, servo motor, o kit de desenvolvimento Atmel e um circuito auxiliar cuja função é interligar todos componentes eletrônicos e fornecer alimentação externa para o motor.

Na extremidade da viga fixa a base vertical estará fixo com um suporte o sensor ultrassônico (HC-SR04) cuja função é informar a distância real da bola em relação ao sensor. A informação do sensor será lida pelo microcontrolador presente no kit de desenvolvimento Atmel que em seguida realizará o processamento por uma malha de controle implementada. Uma vez completo o processamento, o microcontrolador envia um sinal de PWM correspondente para rotacionar o servo motor a fim de equilibrar a bola na viga sem que ela saia de seus limites.

## Periféricos usados no microcontrolador

|  |  |  |
| --- | --- | --- |
| Periférico | Quantidade | Função |
| UART | x1 | Exibir status do sistema e auxílio no desenvolvimento. |
|
| Botões | x2 | Gerar interrupção para melhorar o ajuste do controlador. |
|
| PWM | X1 | Fornecer uma onda quadrada de 50Hz para acionar o servo motor. |
|
| Timer | x2 | Cronometrar o tempo de resposta do sensor ultrassônico e o tempo da malha de controle. |
|

Figura 2: Tabela de periféricos usados

### PWM

A onda de PWM configurada precisa atender alguns requisitos:

* Possuir 50Hz de frequência.
* Ser quadrada.
* Output entre 5 V a 7.2 V de tensão.

Essas características são obrigatórias devido a escolha do servo motor usado no projeto que se não forem atendidas o motor não é acionado.

O servo motor MG 995R TowerPro fornece 180 graus de liberdade na rotação de seu eixo, para controlar essa amplitude alteramos o duty cycle do PWM.

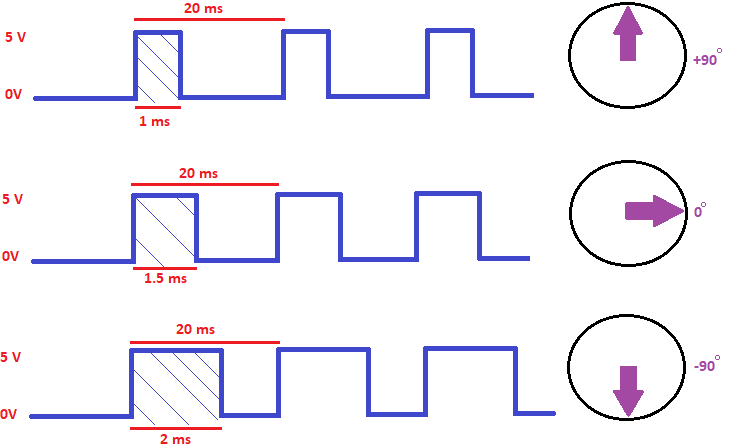


Figura 3: Sinais de PWM em relação a posição do motor

Note que possuímos uma pequena faixa para variar o duty cycle do sinal, por volta de 5% até 10% para varrer todas posições possíveis do motor. No projeto não serão necessários os 180 graus devido a limitações físicas portanto o duty será atualizado em pequenos incrementos.

### Timer

No projeto é configurado um timer para gerar um cronômetros que estoura a cada 10ms (Canal 0). Para entender o motivo desta ordem de grandeza é necessário compreender como funciona o sensor ultrassônico HC-SR04 usado no projeto.

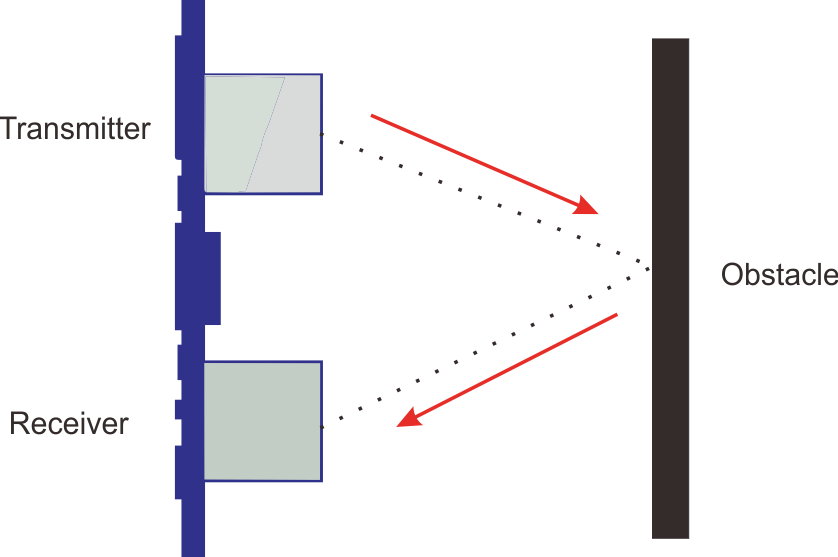


Figura 4: Funcionamento do sensor HC-SR04

O sensor é composto por duas partes um transmissor e um receptor ultrassônicos. O transmissor emite a onda de 40 KHz e o receptor é usado para medir a distância a partir da latência para captar o sinal e enviar uma resposta.

A sequência de funcionamento deste processo é seguinte:

1. O pino TRIGGER do sensor deve permanecer por pelo menos 10us em nível alto.
2. Feito isto o módulo automaticamente emite 8 ondas de 40 KHz e espera uma resposta do pino ECHO.
3. Se o sinal de resposta for recebido, por nível alto, o tempo em que este pino permaneceu em nível alto é proporcional à distância do obstáculo.

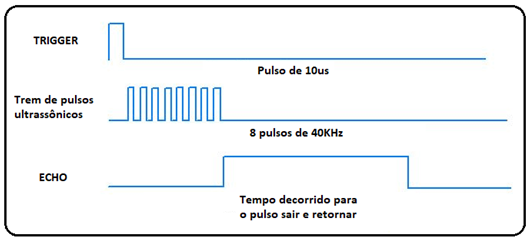


Figura 5: Diagrama de tempo dos pinos do HC-SR04

O contador da distância do objeto foi implementado dentro da interrupção do PIO ECHO. Isso foi necessário, pois a contagem do tempo em que o ECHO permanece em nível alto é critico e extremamente rápido, uma vez que a resolução do sensor é de 10us. No caso de ocorrer uma interrupção nesse período o tempo pode vir com erros.

O canal, configurado para 100ms, a cada interrupção verifica o status de seu Timer e executa uma função chamada *vReadSensor()* cujo encargo é realizar outra leitura do sensor.

# Diferenciais da Solução

O uso de um microcontrolador Atmel SAM4S fornece muitos recursos para realizar o processamento do sistema e realizar seu controle de maneira rápida e eficaz. Em um projeto de controle quanto mais rápido e maior a corretude que meu software for capaz de gerar, melhor será a atuação na malha de controle, projetos similares normalmente usam Arduino que em comparação possui menor potencial.

Outro ponto está no uso de interrupções com diferentes prioridades que asseguram um fluxo de execução para o microcontrolador atuar na malha de controle.

# Pontos Críticos

A parte mecânica de montagem do projeto não exige uma alta complexidade, de tal forma o principal foco está localizado no software. Dessa forma, podemos apontar como riscos os seguintes setores no desenvolvimento:

* Malha de Controle: Isso envolve o refinamento do controlador e modelo do sistema, o qual deve ser realizado na aula de Sistemas de Controle I
* A precisão dos sensores: Tempo de call-back e erro de posicionamento.
* A precisão do motor: Tempo de reação, entre o envio do sinal e o movimento final, e exatidão no valor da velocidade.

# Diagrama de Blocos Funcionais da Solução

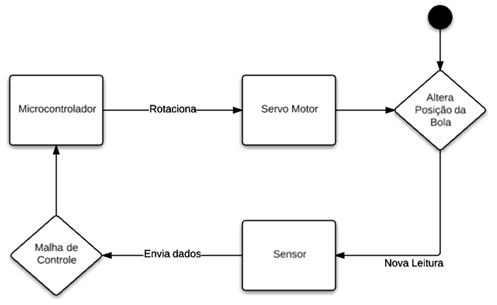


Figura 6: Diagrama de Blocos do Sistema

# Insumos Adicionais Necessários

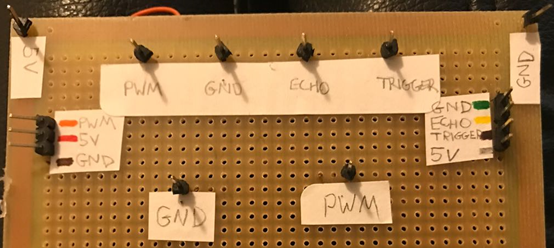
* Sensor ultrassônico HC - SR04



* Motor MG TowerPro 995



* Circuito auxiliar



* Atmel Studio



* Kit ATSAN3N – EK2

