

IV SEMANA DO CONHECIMENTO

COMPARTILHANDO E FORTALECENDO
REDES DE SABERES

6 A 10 DE NOVEMBRO DE 2017



Marque a opção do tipo de trabalho que está inscrevendo:

Resumo

Relato de Caso

SINTETIZADOR DIGITAL DE ONDAS SIMPLES UTILIZANDO UM CONVERSOR DIGITAL-ANALÓGICO POR SOFTWARE EM MICROCONTROLADORES AVR.

AUTOR PRINCIPAL: Angelo Elias Dalzotto

ORIENTADOR: Adriano Canabarro Teixeira

UNIVERSIDADE: Universidade de Passo Fundo

INTRODUÇÃO:

Os microcontroladores AVR, acrônimo para “*Alf and Vegard's RISC processor*” apesar de nunca confirmado pelo fabricante (LANGBRIDGE, 2015), são populares por terem várias funcionalidades e custo reduzido, mas realmente tornaram-se conhecidos mundialmente por equipar a plataforma de prototipagem Arduino. Uma das suas limitações é a falta de um *Digital-to-Analog Converter* (DAC), que teria como objetivo produzir tensões analógicas nas saídas do microcontrolador. Comprar um DAC para ser acoplado ao AVR, muitas vezes, custa caro e possui várias limitações. Para casos específicos, nos quais é necessário gerar sinais e ondas simples, foi feito um DAC por *software* auxiliado por componentes eletrônicos que custam poucos centavos. O motivo que inspirou a criação dessa solução foi possibilitar a construção de um teremim (instrumento musical eletrônico que gera sons a partir do movimento das mãos) digital como parte de pesquisa de doutorado do Programa de Pós-Graduação em Educação (PPGEdu) da UPF.

DESENVOLVIMENTO:

Muitos programadores ou *hobbyistas* (pessoas que programam ou desenvolvem soluções eletrônicas por *hobby*) iniciantes utilizam o ambiente de desenvolvimento integrado - *Integrated Development Environment* (IDE) - Arduino para fazer seus códigos. Ele possui algumas limitações quanto ao controle do que se pode fazer com o *hardware* do microcontrolador justificado pela necessidade de ser mais rápido para desenvolver as soluções para o mercado (MYKLEBUST, 1996). Para escapar dessas limitações, o *firmware* foi escrito em linguagem C e compilado de forma nativa. Dessa forma, foi utilizado o recurso de geração de pulsos de largura modulada - *Pulse Width Modulation* (PWM) - para criar ondas quadradas em uma frequência definida, podendo escolher o nível de tensão média de saída. Foi possível levar essa frequência

IV SEMANA DO CONHECIMENTO

COMPARTILHANDO E FORTALECENDO
REDES DE SABERES

6 A 10 DE NOVEMBRO DE 2017



ao máximo alcançável de 62KHz por meio da configuração dos registradores de controle, não permitindo a divisão da frequência de incremento do contador, limitando o seu valor máximo em 255 e configurando o seu comparador ligado a uma porta de saída. O PWM foi configurado para estar em nível alto até o registrador atingir o valor de comparação e após isso o nível é baixo, até finalmente ocorrer o *overflow* e retornar ao valor alto. Também foi configurado o registrador para salvar em memória quantos microssegundos se passaram desde o início do procedimento. Com essa informação é possível saber em que nível de tensão encontra-se um sinal que varia com o tempo. Como teste de funcionamento, foi gerada uma onda senoidal de frequência de 500Hz, centro em 2,5V e pico máximo em 5V - nível lógico do AVR. Calculando continuamente teremos o valor de tensão desejado no instante, mas que não é o desejado para ser comparado no PWM para gerar a tensão média. Para isso, também é preciso equacionar um valor de comparação inteiro a ser transferido para o registrador. Esse valor será variado seguindo as equações, atualizando a cada poucos microssegundos, um tempo que mesmo assim garante que outras ações não sejam bloqueadas pelos cálculos, podendo o microcontrolador realizar outras tarefas até a próxima atualização do PWM. A saída da porta digital do Arduino é uma onda quadrada de frequência fixa, mas que varia seu ciclo de trabalho, também alterando a tensão média por ciclo. Para tornar essa onda da forma que foi desejada, foi utilizado um amplificador operacional funcionando como filtro ativo passa-baixa na frequência de 2KHz, conforme esquemático (Figura 1), que foi suficiente para cortar as componentes quadradas da onda inicial e deixar passar somente o nível médio variável, caracterizado pela equação presente no código. A forma de onda e frequência de saída podem ser definidos pelo equacionamento. Para garantir o funcionamento, porém, deve-se estar atento à frequência de corte do amplificador operacional, alterando os valores dos componentes. Também pode ser aumentada a resolução do PWM para 9 bits ou 10 bits, ao custo da frequência máxima de saída desejada.

CONSIDERAÇÕES FINAIS:

Foi possível utilizar os recursos nativos do microcontrolador para criar uma funcionalidade extra que não é uma proposta do AVR. Apesar de obter um resultado muito satisfatório, é importante afirmar que essa solução pode não ser útil quando aplicada para tarefas que exigem maior precisão. Uma das tarefas que pode realizar com perfeição é a sintetização de sons simples.

REFERÊNCIAS:

LANGBRIDGE, J. A. Arduino Sketches: Tools and Techniques for Programming Wizardry. Wiley, 2015. 429 p.

IV SEMANA DO CONHECIMENTO

COMPARTILHANDO E FORTALECENDO REDES DE SABERES

6 A 10 DE NOVEMBRO DE 2017



MYKLEBUST, G. The AVR microcontroller and C compiler co-design. In: 3rd European Microprocessor and Microcontroller Seminar. 1996, Heathrow, UK. Proceedings p. 164–170.

ANEXOS:

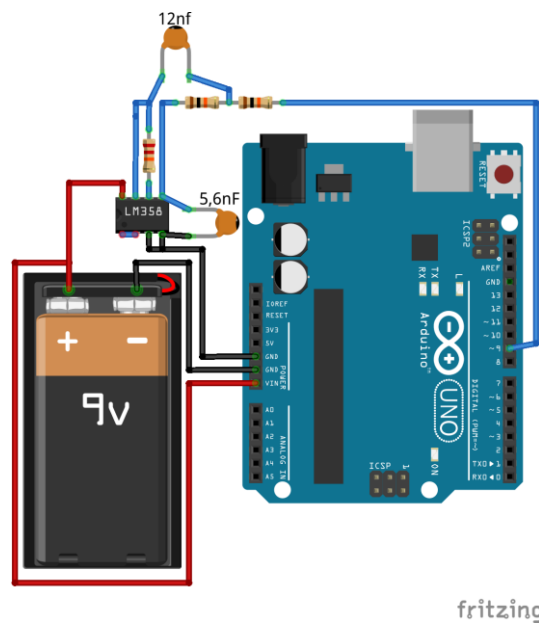


Figura 1. Esquemático do projeto.