UNIVERSIDADE FEDERAL DE UBERLÂNDIA - UFU

FACULDADE DE ENGENHARIA ELÉTRICA – FEELT

SINAIS E SISTEMAS EM ENGENHARIA BIOMÉDICA

Gerador de Formas de Onda

Alunos:

1. Ítalo Gustavo Sampaio Fernandes - 11511EBI004
2. Nathalia Rodrigues - 11411EBI018
3. Paulo Camargos Silva - 11611EBI023

Prof. Sérgio Ricardo de Jesus Oliveira

Uberlândia, **08** de **novembro** de 2017

1 – Introdução

.O Arduino DUE possui 54 pinos digitais de entrada ou saída, onde 12 desses podem ser utilizados para controle PWM aumentando a capacidade de control PWM se comparada a Arduino UNO que possui apenas 6 saídas PWM. Os conversores analógicos-digital (ADC) dessa placa são rápidos e podem operar em até 1 MHz de Frequência de amostragem. Essa nova placa possui 12 entradas de ADC, com 12 bits de resolução.

Além disso, a placa possui dois conversores digital-analógicos (DAC) de 12 bits, o que permite mais versatilidade a diferentes aplicações. É possível então, ter uma saída com um valor de tensão entre 0 e 3,3V, com 4096 diferentes valores, nesse intervalo. Pela existência destes conversores D/A, é possível gerar sinais senoidais, triangulares, quadradas, dente-de-serra e até mesmo emular sinais biomédicos, com diversas frequências e amplitudes.

2 – Materiais e Métodos

Os materiais utilizados neste experimento foram:

- Arduino;
- Osciloscópio;
- Potenciômetro com valor de 1 a 50kΩ.

2.1 – Esquema do circuito

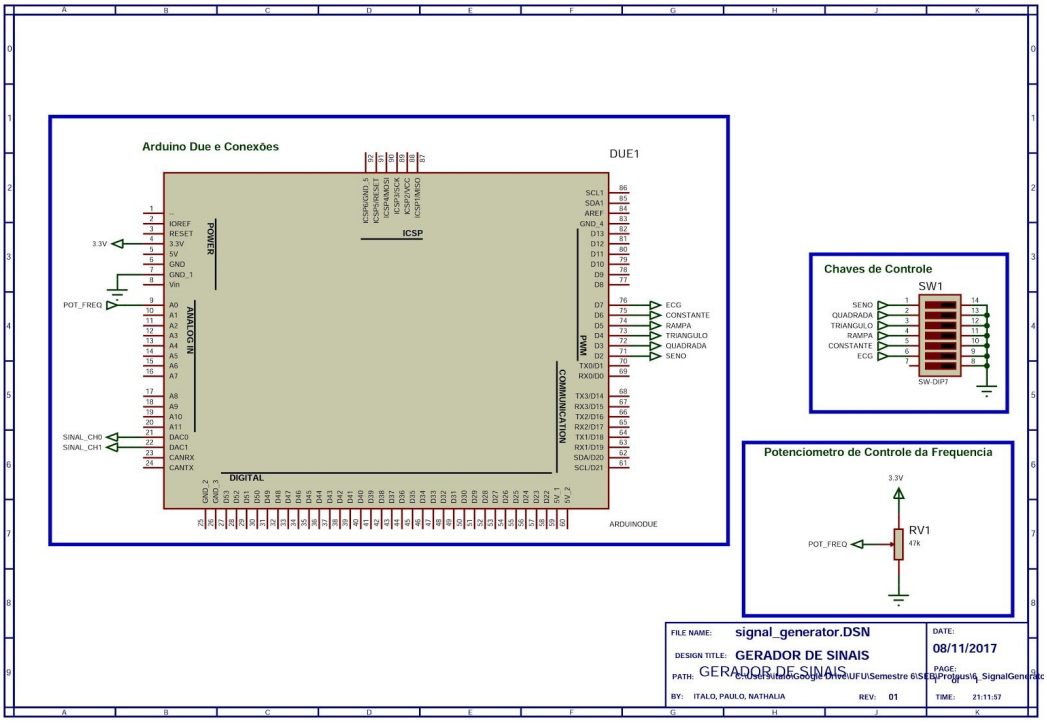

Figura 1 - Esquema do circuito. (Dísponivel em: https://goo.gl/AAMyX7)

2.2 – Código

O código embarcado criado pode ser dividido em 2 etapas, inicialização e loop. Na inicialização é instanciado um objeto da classe *SignalGenerator* (que será explicado adiante), para a configuração deste é utilizada a função "*analogWriteResolution(int resolution)*" onde a resolução do DA é configurada para 12 bits, ou seja valores de 0 a 4095. São configurados os pinos 2 ao 7 como entradas com resistor de Pull-Up, estes são usados para seleção do tipo de onda, e é iniciada a

interface de comunicação UART (*universal asynchronous receiver-transmitter)* com uma taxa de 115200 bits por segundo, através da qual os valores de sinal gerados também são enviados, permitindo que sejam plotados pela ferramenta Serial Plotter (disponível na versão 1.8 da IDE do arduino).

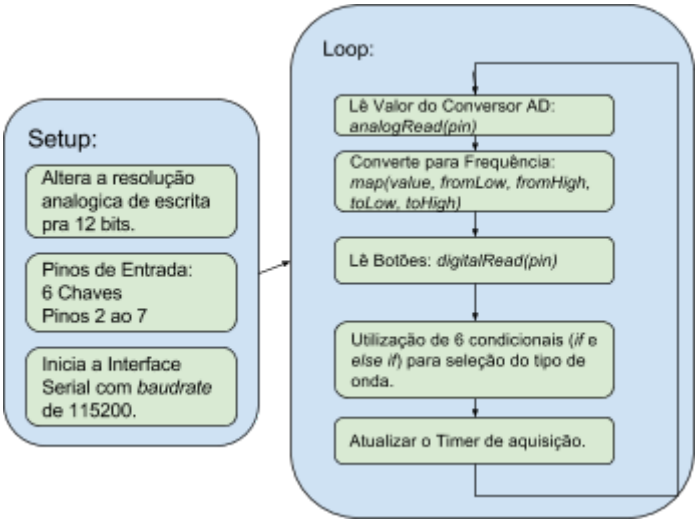

Figura 2 - Diagrama de código.

Para o devido funcionamento do código as rotinas de geração de sinais foram encapsuladas em uma classe chamada SignalGenerator, esta classe possui um Timer interno (implementado através da função "*micros()"*) que através dele faz a chamada continua do método "*generate_value()*". O método "*generate_value()*" chama o método privado "getnext()" (responsável por calcular o valor instantâneo do sinal), após calculado são utilizadas as funções *"Serial.println()"* e "*analogWrite(pino, valor)*" para envio do valor para a porta serial e para o conversor DA. O método "*getnext()"* trata cada tipo de onda de forma diferente:

- Caso uma onda **Senoidal ou ECG:** Obtém o próximo valor armazenado em um vetor de dados salvo na memória de programa do microcontrolador (com 500 amostras), após isso incrementa o index atual (ponteiro que aponta para o valor do vetor a ser mostrado em cada chamada).
- Caso uma onda **quadrada**: Verifica se o index atual aponta para um número maior do que a metade do total de posições, retornando 1 caso verdadeiro e -1 caso falso.
- Caso uma onda de **Rampa**: Retorna o valor normalizado do index atual, ou seja um valor de 0 a 1, onde 1 representa que o index aponta para a máxima posição.
- Caso **Triangular**: Retorna o valor do index atual normalizado pela metade do total de posições, caso passe dessa metade o valor passa a ser decrementado, formando assim uma onda triangular.
- Caso **Constante**: Retorna sempre o valor 1 (um).

Após o cálculo do valor para cada tipo de onda este é multiplicado pela amplitude e somado a um offset, neste programa a amplitude sempre foi de 80% do valor máximo da porta DAC (80% de 4095, no caso) e o offset foi da metade deste valor (2048).

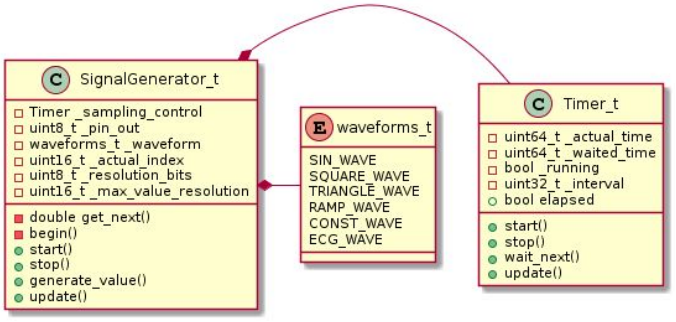

Figura 3 - Diagrama de Classes UML do código implementado no Arduino.

2.3 – Gráficos

A partir da ativação utilizando um botão, pode-se escolher a forma de que será gerada no pino DAC0. Todas as formas de onda foram visualizadas no osciloscópio. As Figuras 4, 5, 6 e 7 exibem as formas de onda geradas, exibidas com a ferramenta *serial plotter*.

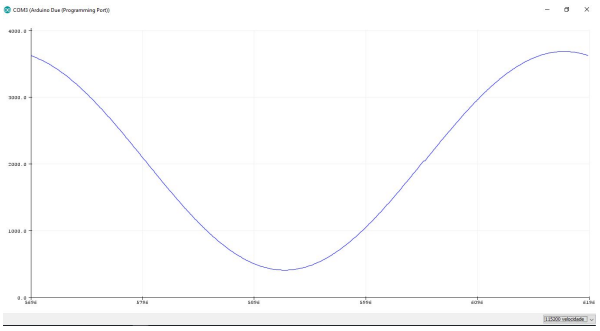

Figura 3 - Senoide

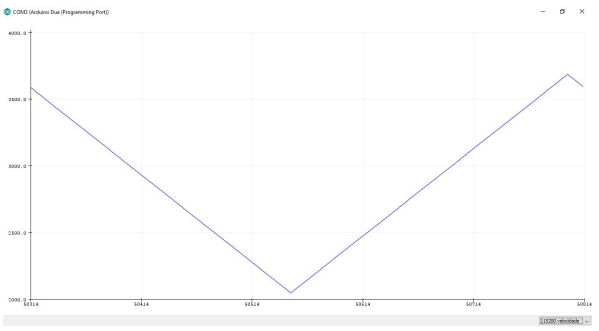

Figura 5 - Triangular

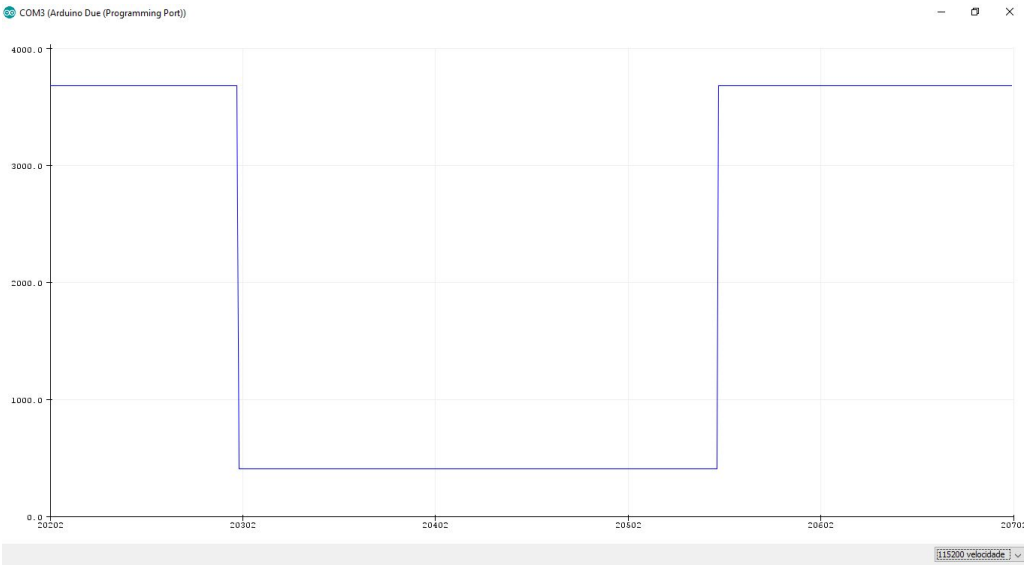

Figura 6 - Quadrada

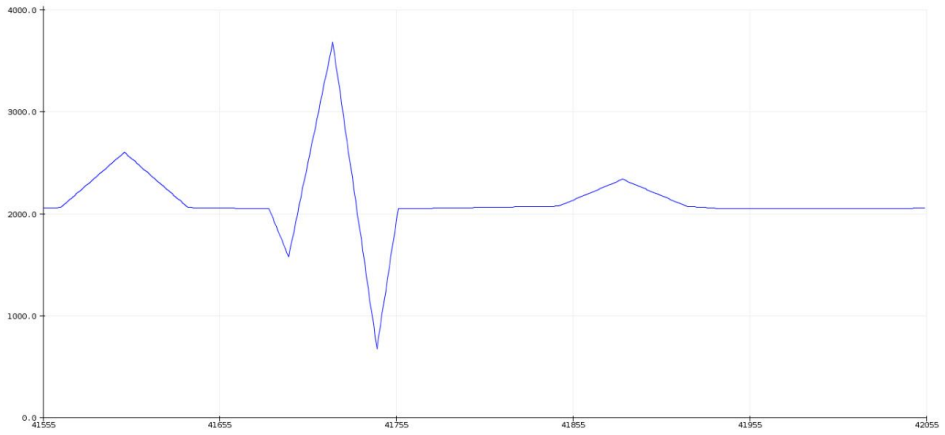

Figura 7 - ECG padrão

3 – Resultados e discussões

Como resultado obtivemos 6 tipos de onda diferentes (quadrada, senoidal, rampa, constante, ecg e triangular) de acordo com o botão acionado. Com o utilização de 500 pontos, obtivemos uma boa resolução do sinal ao longo do tempo, sendo que cada forma de onda foi conferida no osciloscópio.

Foi criado também uma matriz com valores semelhantes a um sinal captado de um eletrocardiograma (ECG) utilizando o Matlab. Este sinal foi plotado no *serial plotter* da IDE e também foi visualizado na tela do osciloscópio. Esta técnica mostra a ampla possibilidade de geração de sinais que é possível utilizando as portas DAC0 e DAC1.

4 – Conclusão

A partir do experimento montado e testado em laboratório foi possível visualizar e entender a utilização das portas Digital-Analógico (DAC0 e DAC1) gerando valores analógicos. Utilizando um osciloscópio foi possível visualizar a fidelidade do sinal, alterando a frequência com um potenciômetro.

O conversor DAC do Arduino possibilita a geração de sinais com diversos tipos de onda. Isto é útil, tendo em vista a grande quantidade de sinais biomédicos, que podem ser simulados e testados utilizando o próprio Arduino.

5 – Referências

1. https://www.arduino.cc/en/Tutorial/DueSimpleWaveformGenerator
2. https://www.arduino.cc/reference/en/language/functions/trigonometry/sin/
3. https://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardDue

Anexo I - CÓDIGO

Todo o código deste experimento se encontra no link do GitHub apresentado abaixo:

https://github.com/italogfernandes/SEB/blob/master/projeto_6_signal_generator/src/main.ino