



Gerador de formas de onda

Discentes:

Hugo Aparecido Moraes Oliveira	11311EBI032
Otávio Yamanaka Marin	11311EBI002
Taíssa Oliveira Ferreira	11011EBI015

Prof. Sérgio Ricardo de Jesus Oliveira

Uberlândia/MG, Novembro de 2015

Sumário

1.	INTRODUÇÃO	3
2.	OBJETIVO	3
3.	MATERIAIS E MÉTODOS	4
4.	PROCEDIMENTO.....	4
5.	RESULTADOS E DISCUSSÃO	7

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 controle 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.

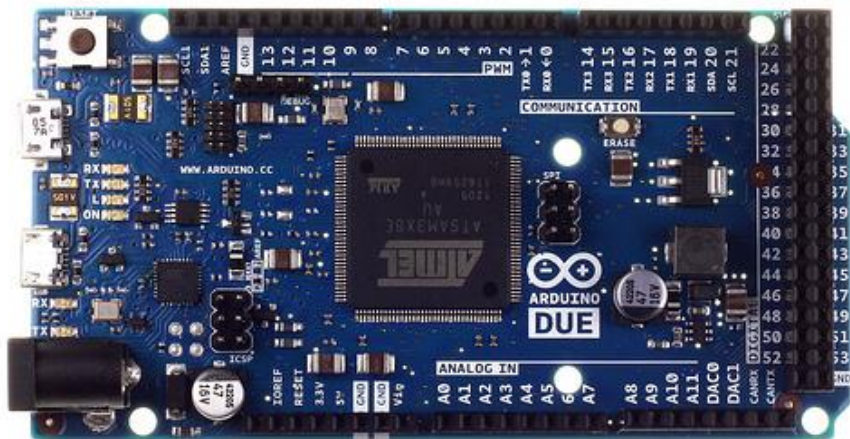


Figura 1 - Arduino DUE

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 etc., com diversas frequências, de acordo com o período da onda analógica em questão.

2. OBJETIVO

Este relatório apresenta como objetivo mostrar o funcionamento da porta analógica do Arduino DUE, que possui um conversor D/A, e produzir nesta saída formas de onda analógicas do tipo quadrada, triangular e senoidal.

3. MATERIAIS E MÉTODOS

Os materiais utilizados para a realização do experimento foram:

- Arduíno DUE;
- 1 Potenciômetro de 10k Ω ;
- Osciloscópio;
- 3 botões;
- 3 resistores de 1k Ω ;
- Fios conectores.

4. PROCEDIMIENTO

Inicialmente foi realizada a montagem do circuito eletrônico com a utilização dos elementos citados acima. O esquema eletrônico montado com o Arduino é mostrado na figura a seguir.

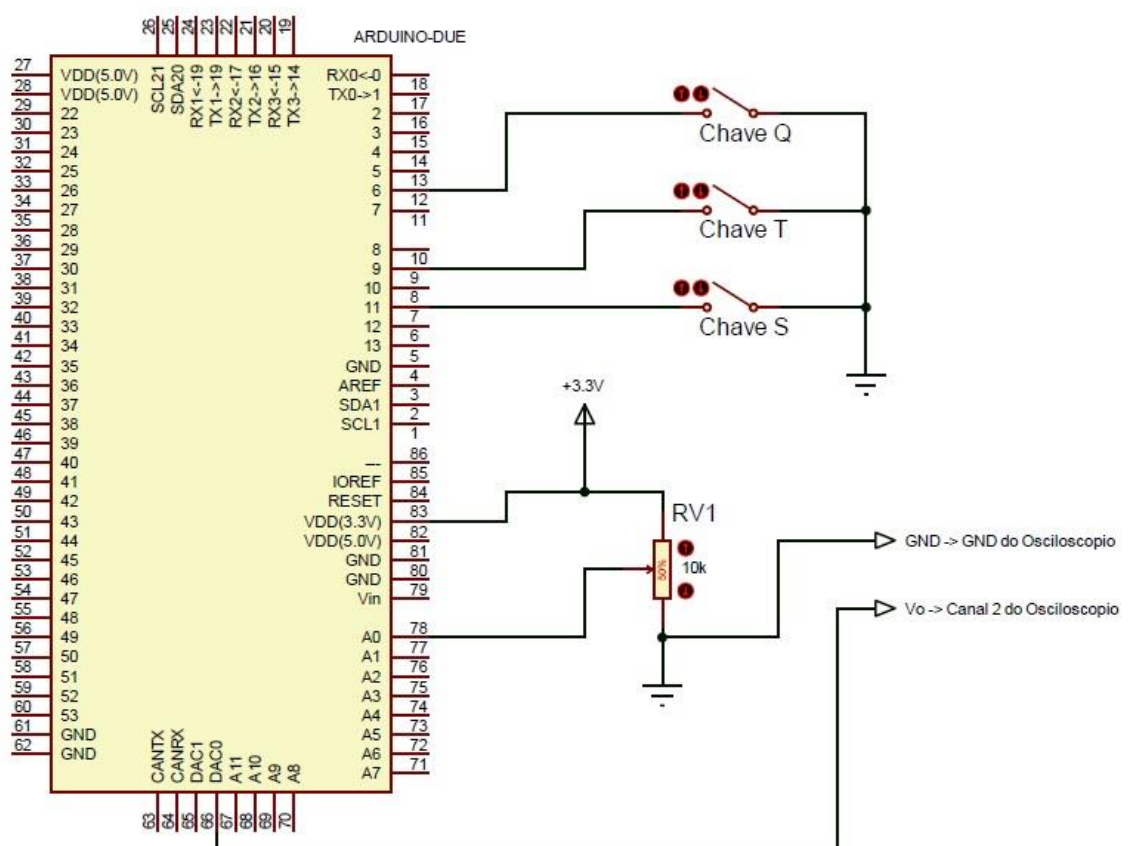


Figura 2 - Esquema eletrônico

Em seguida, foi elaborado um software para gerar na saída V0 uma onda Quadrada, Triangular e Senoidal com frequência de 10 Hz [100ms].

A onda Quadrada é gerada quando a chave Q estiver pressionada, a Triangular quando a chave T estiver pressionada e a Senoidal quando a chave S estiver pressionada. Quando nenhuma das chaves estiver pressionada, nenhuma onda é gerada.

O código do software em questão é mostrado em seguida.

```
int Q = 12;
int T = 9;
int S = 7;
int Pot = 0;
int DAC0 = DAC0;
int y;
int i;
int j;
int k;

void setup() {
  pinMode(Q, INPUT);
  pinMode(T, INPUT);
  pinMode(S, INPUT);
  pinMode(DAC0, OUTPUT);
}

void loop() {

  if(!digitalRead(Q)) //Onda quadrada
  {
    analogWrite(DAC0, 0);
    delay(50);
    analogWrite(DAC0, 4095);
    delay(50);
  }

  if(!digitalRead(T)) //Onda triangular
  {
    for(i=0; i<4096; i++)
    {
      analogWrite(DAC0,i);
      delayMicroseconds(12);
    }
    for(j=4095; j>=0; j--)
    {
      analogWrite(DAC0,j);
      delayMicroseconds(12);
    }
  }
}
```

```

if(!digitalRead(S)) //Onda senoidal
{
    for(k=0; k<=360; k++)
    {
        y = (sin(k)+1)*(4095/2);
        analogWrite(DAC0, y);
        delayMicroseconds(277);
    }
}

```

Com o auxílio do osciloscópio é possível observar as formas de ondas geradas.

Após, o software foi alterado de tal forma que, a partir do potenciômetro, seja possível alterar a frequência da onda gerada. O código deste software é mostrado em seguida.

```

int Q = 6;
int T = 9;
int S = 11;
int Pot = 0;
int DAC0 = DAC0;
int Val = 1;
int y;
#define Pot 0

void setup() {
    pinMode(Q, INPUT);
    pinMode(T, INPUT);
    pinMode(S, INPUT);
    pinMode(DAC0, OUTPUT);
}

void loop() {

    Val = analogRead(Pot);
    Val = map(Val,0,1023,1,1000);

    if(!digitalRead(Q)) //Onda quadrada
    {
        analogWrite(DAC0, 0);
        delay(Val);
        analogWrite(DAC0, 4095);
        delay(Val);
    }

    if(!digitalRead(T)) //Onda triangular

```

```

{
    for(int i=0; i<4096; i++)
    {
        analogWrite(DAC0,i);
        delayMicroseconds(Val);
    }
    for(int j=4095; j>=0; j--)
    {
        analogWrite(DAC0,j);
        delayMicroseconds(Val);
    }
}

if(!digitalRead(S)) //Onda senoidal
{
    for(int k=0; k<=360; k++)
    {
        y = (sin(k)+1)*(4095/2);
        analogWrite(DAC0,y);
        delayMicroseconds(Val);
    }
}
}

```

Com o auxílio do osciloscópio é possível observar as formas de ondas geradas.

5. RESULTADOS E DISCUSSÃO

Primeiramente, foram geradas ondas com frequência de 10Hz (100ms). Para tal feito, foi necessário o cálculo do *delay*, que nada mais é do que o tempo de espera entre a geração de um valor a outro da função que gerará a onda.

Para a onda quadrada o *delay* encontrado foi de 50 ms, pois a onda de um período completo será metade desse período no máximo valor (4096) e outra metade no mínimo (0).

Para a onda triangular, o tempo de espera foi de 12 microssegundos. Esse valor foi obtido, pois a onda triangular é basicamente duas retas, uma crescente e outra decrescente, alternando entre si. Ou seja, metade do tempo a reta subirá e na outra metade descenderá, sendo assim, em um tempo de 50 ms o *software* pegará 4096 valores. Portanto, esse tempo deverá ser dividido por 4096 e assim será obtido 12 microssegundos.

Para a onda senoidal foi utilizado a função `sin()`, que gerará valores de -1 a 1, dependendo dos valores do seno dos ângulos de 0° a 360°. Como não é desejado que o gráfico tenha parte positiva, o sinal será deslocado de 1 para cima, para fazer isso somou-se 1 ao seno, fazendo-o variar, agora, de 0 a 2. Como os valores necessários para se mandar pela saída DAC0 tem que variar de 0 a 4096, multiplica-se o seno obtido pela metade de 4096, o que vai fazer o sinal variar de 0 a 4096. Dessa forma, para se obter o *delay* é necessário dividir o tempo de 100ms por 360, que é a quantidade de ângulos que serão utilizados para obter os valores do seno.

Na segunda parte o código mal foi alterado, o que realmente mudou foi a adição de um potenciômetro. Este, será o responsável pela mudança de frequência no sinal desejado. Para que isso ocorra, ele deverá modificar os *delays* obtidos primeiramente, sendo que quanto maiores, menor a frequência do sinal.

Os tempos de espera, então, foram multiplicados por valores de 1 a 20, assim, o potenciômetro poderia aumentar e posteriormente diminuir a frequência do sinal.