



# Relatório do laboratório 09

**Maria Eduarda Pedroso**

**Matrícula: 210669**

**Sistemas Digitais (SICO5A)**

### Objetivos

Construir um conversor A/D e D/A.

Utilizando o gerador de funções construir um sinal analógico de que o Arduino seja capaz de ler. Projetar o sistema de comunicação. A fonte é um sinal analógico, proveniente do gerador de funções, e a saída é a reconstrução do sinal analógico em um monitor. A comunicação entre os arduinos deve ser digital.

### Materiais e equipamentos

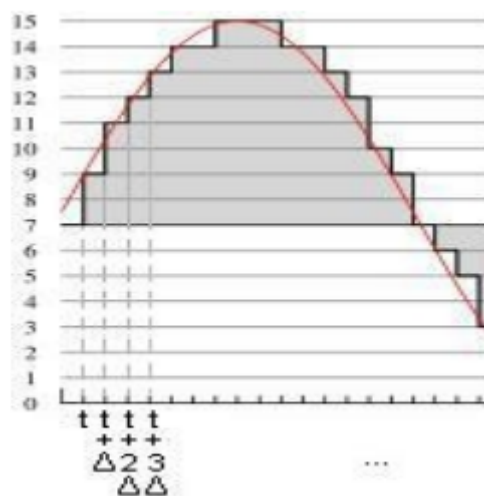
Os materiais utilizados foram todos códigos e funções no simulador TinkerCad, sendo estes explicados sua utilidade no relatório.

### Fundamentação teórica

Qualquer tipo de informação que necessite ser armazenada em um sistema computacional ou transmitida em um canal de transmissão, necessita passar por um tratamento de sinal, ou seja um processo chamado de codificação de fonte. A grande vantagem da utilização de um sistema assim é a redução de entropia do sinal.

O PCM é um tipo de codificador de forma de onda, que procura reproduzir o sinal amostra por amostra.

Figura 1 – Exemplo de amostragem com intervalos de  $\Delta$  e exemplo de quantização com 15 níveis de intervalo.



A modulação PCM de um sinal analógico acontece de forma simples: o sinal é amostrado em intervalos de tempo iguais, o sinal é quantizado em alguns níveis finitos de amplitude, finalmente o sinal é codificado. O método de codificação da forma de onda PCM está definido nas especificações internacionais CCITT G.711, e AT&T 43801.

A amostragem deve respeitar a frequência de Nyquist,  $f_a = 2 \cdot f_M$ . Primeiramente adota-se um intervalo fixo de tempo entre uma amostra e outra. Tanto o modulador quanto o demodulador tem os passos de amostragem já definidos pelo sistema, no caso do PCM de telefonia o equipamento “enxerga” uma banda de 4000Hz que corresponde aproximadamente a banda que o sinal de voz ocupa. A frequência de amostragem é de 8000Hz (8000 amostras por segundo ou ainda uma amostra a cada 125us), respeitando a frequência de Nyquist.

No sistema telefônico o intervalo entre uma amostra e outra de um canal é utilizado para agregar outras amostras de outros canais diferentes.

O processo de quantização pode ser visto como o mapeamento do sinal, a partir do domínio contínuo para um número finito de níveis de saída. Como o sinal analógico pode ter uma amplitude teoricamente infinita, mas na realidade com um valor máximo de amplitude, divide-se este intervalo de valores possíveis em alguns níveis de amplitude de saída.

No PCM temos 256 níveis de quantização, como o PCM utiliza a codificação binária, cada palavra-código contém 8 bits ( $2^8 = 256$ ). Temos 8 bits para representar cada amostra.

Por causa da limitação da representação do sinal existirá sempre a presença de um ruído de quantização, no sinal quantizado.

O quantizador uniforme tem um passo de quantização constante o intervalo entre os níveis de quantização.

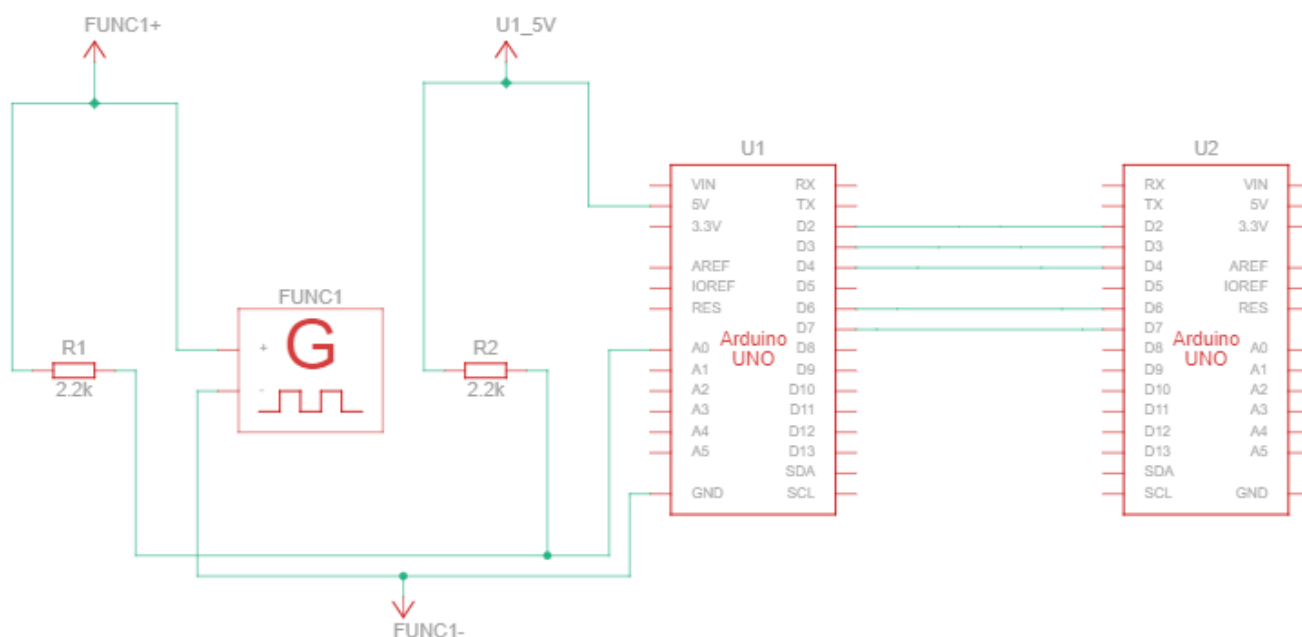
Atualmente o PCM é utilizado em diversas aplicações do nosso cotidiano: do sistema telefônico, ao vídeo digital passando por diversos formatos de áudio digital usado em computadores. No sistema de DVD, por exemplo, utiliza algumas variantes compactadas do PCM, pois o PCM em tem uma elevada taxa de bits.

## **Códigos circuito e análise**

O circuito foi realizado para a implementação do conversor analógico/digital, digital/analógico, para isso utilizamos de dois arduinos códigos estão abaixo e uma fonte geradora de função setada em uma senoide. Como mostra o esquema do circuito abaixo:

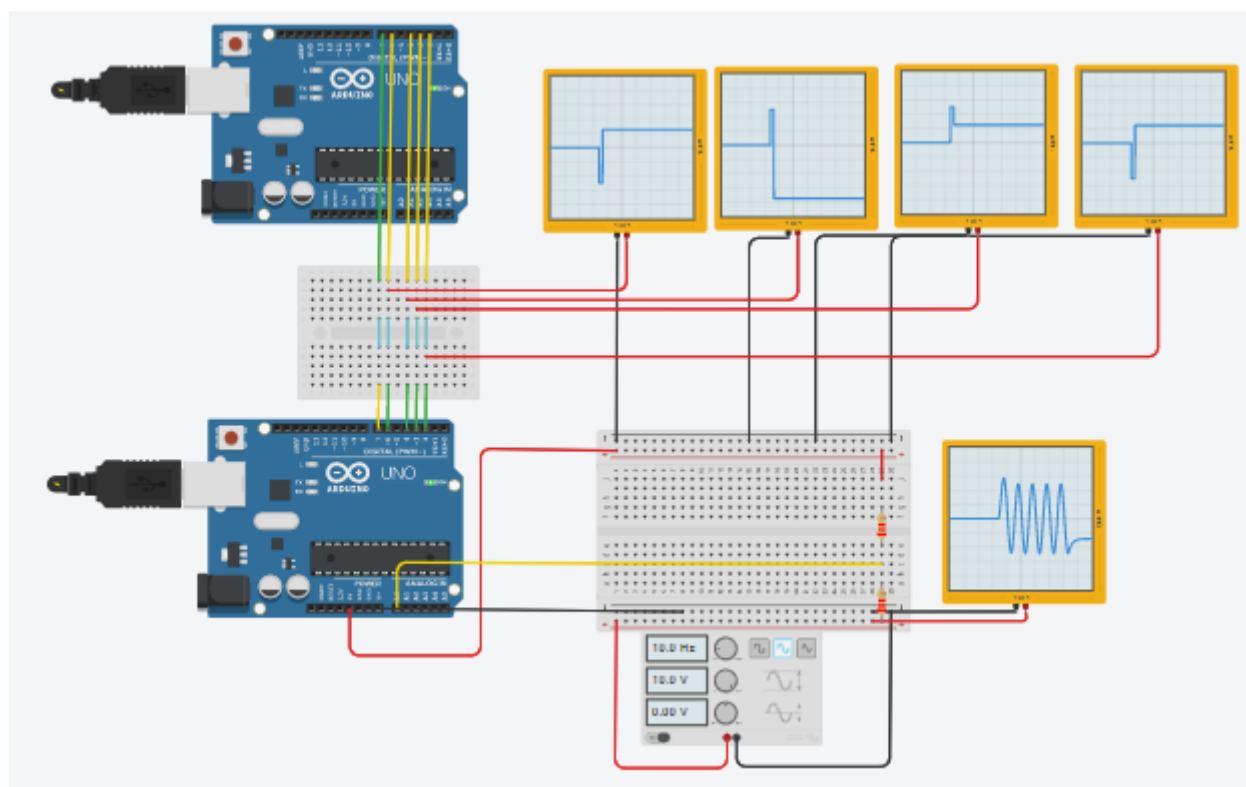
Para manter o sincronismo foi utilizado da porta 7, visto que no software foi o único jeito no qual funcionou

Imagem 1: Esquema do circuito do conversor AD/DA.



Fonte: Autoria própria(2022).

Imagem 2: Circuito do conversor AD/DA.



Fonte: Autoria própria(2022).

Código DA

```
// DA
//
void setup()
{
  pinMode(2, INPUT);
  pinMode(3, INPUT);
}
```

```

pinMode(4, INPUT);
pinMode(6, INPUT);
pinMode(7, OUTPUT);
Serial.begin(57600);
digitalWrite(7, HIGH);
}

int transmitido;
int convertido;
int sinc1 = 0;
int sinc2 = 1;
int sincAux;
int aux[3];

void loop()
{
    sincAux = digitalRead(6);

    sinc1 = sincAux;
    if (sinc2 == 1) {
        sinc2 = 0;
        digitalWrite(6, LOW);
    } else {
        sinc2 = 1;
        digitalWrite(6, HIGH);
    }

    if (sinc1 != sincAux) {

        sinc1 = sincAux;

        aux[0] = digitalRead(2);
        aux[1] = digitalRead(3);
        aux[2] = digitalRead(4);

        transmitido = (aux[0] * 4) + (aux[1] * 2) + (aux[2]);
        convertido = map(transmitido, 0, 7, 0, 1023);
        Serial.println(convertido, DEC);

        delay(25);
    }
} // loop

```

Código AD:

```

// AD
//
void setup()
{
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
}

```

```
pinMode(6, OUTPUT);
pinMode(7, INPUT);

Serial.begin(57600);
digitalWrite(6, HIGH);
}

int leitura;
int conversao;
int sinc1 = 1;
int sinc2 = 0;
int sincAux;

void loop()
{
    leitura = analogRead(A0);
    conversao = map(leitura, 0, 1023, 0, 7);

    sincAux = digitalRead(7);

    if (sinc2 != sincAux) {
        sinc2 = digitalRead(7);

        switch (conversao) {
            case 0://000
                digitalWrite(2, LOW);
                digitalWrite(3, LOW);
                digitalWrite(4, LOW);
                break;
            case 1://001
                digitalWrite(2, LOW);
                digitalWrite(3, LOW);
                digitalWrite(4, HIGH);
                break;
            case 2://010
                digitalWrite(2, LOW);
                digitalWrite(3, HIGH);
                digitalWrite(4, LOW);
                break;
            case 3://011
                digitalWrite(2, LOW);
                digitalWrite(3, HIGH);
                digitalWrite(4, HIGH);
                break;
            case 4://100
                digitalWrite(2, HIGH);
                digitalWrite(3, LOW);
                digitalWrite(4, LOW);
                break;
            case 5://101
                digitalWrite(2, HIGH);
                digitalWrite(3, LOW);
                digitalWrite(4, HIGH);
                break;
```

```
case 6://110
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  break;
case 7://111
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  break;
default://NADA
  Serial.println("ERRO");
}
if (sinc1 == 1) {
  sinc1 = 0;
  digitalWrite(6, LOW);
} else {
  sinc1 = 1;
  digitalWrite(6, HIGH);
}

delay(25);
}
}
```

**Maria Eduarda Pedroso**

## **Resultados e Conclusão**

Com essa prática, aprendemos como criar código e fazer CONV AD/DA com auxílio do arduino. Muitos dos tópicos desta prática estão relacionados ao aprendizado teórico ministrado pelo professor em sala de aula, pois não há cálculo e sim análise, acredito que essa atividade seja crucial para melhor compreensão do tema.