**INTRODUÇÃO AO ARDUINO.**

Funções abordadas: **int, pinMode, OUTPUT, INPUT,digitalWrite, digitalRead, HIGH, LOW, delay.**

**Int;** Usado para criar uma varável.

**EX: int VAL1;** Foi usado para criar a variável VAL1, que poderá ser usada em algum momento do programa, todas as variáveis precisam ser criadas previamente!

**pinMode:** Usado para definir o *modo* do pino a ser usado.(Que pode ser OUTPUT ou INPUT, “entrada” ou “saída” de dados)

**EX: pinMode (10, OUTPUT);** onde o número 10, indica a porta que vai ser *setada*(configurada)como porta de saída de dados(OUTPUT).

**OUTPUT:** Comando usado para definir como “**saída de dados**” uma porta. (Um pino) ou seja, se aquele pino vai mandar energia e não receber energia.

**EX: pinMode (10, OUTPUT);** neste caso, atribuímos ao pino 10 a função de emissão de dados, onde tal pino irá acender o LED que se conectará a ele.

**INPUT:** Comando usado para definir como “**entrada de dados**” uma porta. (Um pino) ou seja, se aquele pino vai receber energia. No caso de um botão, para ler os dados (ligado e desligado)

**EX: pinMode (10, INPUT);** neste caso, atribuímos ao pino 10 a função de recepção de dados, onde tal pino irá intender o nível logico como LOW ou HIGH, assim alterando outra variável.

**INPUT\_PULLUP:** Comando usado para “setar” um pino como porta de entrada de dados, porem com o adivento do **resistor de PULLDOWN**. Evitando colocar um resistor de PULLDOWN no circuito.

**EX: pinMode (10, INPUT\_PULLUP);** neste caso, atribuímos ao pino 10 a função de recepção de dados, com o resistor de PULLDOWN, (trazendo para baixo) onde tal pino irá intender o nível logico como LOW ou HIGH, assim alterando outra variável.

**digitalWrite:** usado para atribuir um valor a uma variável. (Escrever um valor) geralmente ***HIGH*** ou ***LOW.***

**EX: digitalWrite (10);** Foi atrbuido a porta 10 o nível logico ALTO.

**digitalRead:** usado para **ler** um valor em uma porta. (LER um valor) geralmente ***HIGH*** ou ***LOW.***

**EX: digitalRead (10);** Ser álido o valor na porta 10

**HIGH:** nível logico de alto.

**EX: digitalWrite (10, HIGH);** Foi atrbuido a porta 10 o nível logico **ALTO**.

**LOW**: nível logico de baixo.

**EX: digitalWrite (10, LOW);** Foi atrbuido a porta 10 o nível logico **BAIXO**.

**delay**: Como o Arduino faz o looping 16 mil vezes por segundo, (processador de 16 MHetz, em alguns momentos será necessário atrasar o tempo para se notar o efeito da variável. (Como a de piscar um led)

**EX: delay(1000);** foi atribuído o delay de 1000 mili segundo para o período.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Linha de programação – Tela de inicio\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**void setup () {**

//Linha programável. Onde se receberá as linhas de programação. **Entre {Area De Programaçâo} Onde é executado apenas uma vez, quando liga o Arduino ou resetado. Nessa faze, devemos setar os parâmetros de início, como valor 0 para display de contagem, onde sempre se inicia de 0.**

**}**

**void loop() {**

//Linha programável. Onde se receberá as linhas de programação. **Entre {Area De Programaçâo} Onde o programa fica instalado, e rodando o tempo todo. (Loooping)**

**}**

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

EXEMPLO DE CÓDIGO: CÓDIGO ALTERNAR O PICAR DE UM LED USANO UM PUSHBUTON

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

// área também programável, usada para definir ou atribuir variáveis globais, aqueles que existirá em todos os campos de programação do Arduino.

**void setup () {** // início da área programável de “void setup”

// Put your setup code here, to run once:

**pinMode (10, OUTPUT);** // Atribuímos o o parâmetro de SAIDA na porta 10.

**pinMode (9, OUTPUT);** // Atribuímos o o parâmetro de SAIDA na porta 9.

**pinMode (7, INPUT\_PULLUP);** // Atribuímos o o parâmetro de ENTRADA, com o Resistor de PULLDOWD na porta 7.

// fim da área programável de “void setup”

**}**

**void loop() {** // início da área programável de “voip loop”

// Put your main code here, to run repeteadly:

**int SB;** // Criei a variável SB. (Estado do botão)

**SB = digitalRead (7);** // Atribui o “endereço” do botão na placa física do Arduino: pino 7, onde com a **digitalRead**, o arduino vai ler e o estado do pino 7 e gravar na variável **SB (ALTO ou BAIXO)**.

**digitalWrite (10, SB);** // Nível logico da porta 10, será ESCRITO de acordo com o estado do botão **SB, (ALTO ou BAIXO)**

**digitalWrite (9, !SB);** // Nível logico da porta 9, será ESCRITO ao CONTRARIO (Devido ao ponto de Exclamação “!”)de acordo com o estado do botão **SB, (ALTO ou BAIXO)**

**delay(1000);** // delay de 1000 mili segundos até a próxima ação.

// Fim da área programável de “voip loop” após isso irá voltar ao início e recomeçar o Loop, \*sempre devemos pensar de forma Cíclica\*

**}**

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