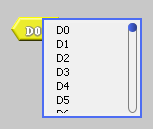
Manual Ardublock – Pinos

* Pino Digital

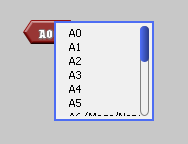
Figura 1 – Blocos representando os pinos digitais



O pino digital serve para referenciar qual a porta digital do arduino estamos usando para conectar um sensor ou atuador. E podemos escolher desde a porta D0 até a porta D13 para o arduino uno ou arduinos similares, por exemplo para arduino Mega, a quantidade de pinos digitais aumentada e essa mesma quantidade tem que ser verificada para melhor utilização. Mostraremos exemplos deste pino implementando seu funcionamento com outros blocos.

* Pino Analógico

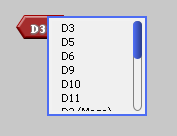
Figura 2 – Blocos representando os pinos analógicos



No arduino uno, as entradas e saídas analógicas estão localizadas do pino A0 até o pino A5, e servem para ler ou enviar valores especificados, isso significa que este irá mapear tensões entra 0 e a tensão de funcionamento que pode ser 5V, ou 3.3V e é representado por valores inteiros entre 0 e 1023. No arduino uno, por exemplo isso permite uma resolução entre leituras de: 5V / 1024 unidades, sendo assim 0.0049 volts (4.9mV) por unidade. Então variando a aumentando o valor de unidades, aumentará a tensão de saída na porta analógica.

* Pino Digital PWM

Figura 3 –Blocos representando os pinos pwm



Os pinos pwm são alguns dos mesmos pinos digitais citados anteriormente, começando pelo D3 e continuando para o D5, D6, D9, D10 e D11. E sua diferença é que podemos usar esses pinos tanto como digitais e acionando sua saída para ligado ou desligado, como também podemos controlar a tensão de saída desses pinos, parecido com a mesma utilização dos pinos analógicos. E a diferença em relação aos dois, é que no pino analógico podemos variar seu valor de 0 até 1023 e já nos pwm podemos variar seu valor de 0 até 255, sendo em 0 = 0 volts, e 255 = 5 volts. Veremos exemplos sobre este pino mais para frente com a utilização de outros blocos em conjunto.

* define pino digital



Figura 4.A – Bloco define pino digital



Figura 4.B – LED verde

Pode ser compara os blocos de LED da figura 4.B com o bloco define pino digital da figura 4.A. os dois fazem a mesma função, a diferença é que o bloco LED fica de uma forma mais interativa para as crianças aprenderem.

Este bloco define uma saída como ALTA (HIGH) ou BAIXA (LOW), e é usado para deixar um dos pinos da AttoBox com 5 volts ou com 0 volts. Neste caso da imagem estamos declarando que nosso pino 13 é uma saída e seu estado vai ser HIGH, que é equivalente a porta estar transmitindo 5 volts para um atuador, como por exemplo o LED, que irá ligar quando estiverem nesse formato e irá desligar quando estiverem em LOW onde representa que a porta está em 0 volts.

Exemplo: Na figura 5 logo abaixo, podemos ver um exemplo simples, onde é possível colocar o bloco dentro do loop e mandar o código para a AttoBox, conectando fisicamente o LED na nossa placa, iremos observar que o LED irá ficar ligado por 1 segundo e após isso irá desligar, esperando mais 1 segundo ele retorna para o estado de ligado e fica nesse loop até a placa ser desligada.



Figura 5 – Exemplo de aplicação do bloco “define pino digital”

* inverte pino digital



Figura 6

Este bloco é fácil e simples de ser usado, ele simplesmente identifica qual a porta de saída que desejamos usar e inverte seu estado, se o pino está HIGH, ele transforma para LOW, e vice-versa.

Exemplo: Vamos usar o exemplo do pino digital 13 onde fica 1 segundo ligado e 1 segundo desligado em um loop infinito, é o mesmo funcionamento que o exemplo anterior, a diferença que aqui usamos primeiro o bloco para definir o pino digital e depois apenas invertemos o seu estado, neste exemplo podemos observar que quando definimos o pino digital 13, arrumamos seu estado como ALTO, que quer dizer ligado, e após se passar 1 segundo, invertemos o pino D13, então agora seu estado passa para BAIXO e permanece por 1 segundo novamente e assim repete o ciclo continuamente.



Figura 7

* pino digital



Figura 8

Neste bloco escolhemos um pino digital, mas um pino de entrada, definimos qual pino vai ser usado para que nossa AttoBox esteja recebendo um valor, através de um botão ou qualquer outro sensor que mande um sinal para a nossa placa.

Exemplo: Na figura 9 usamos outros blocos que já foram comentados anteriormente, então já sabemos basicamente seu funcionamento, e agora só deixaremos de uma forma mais fácil o entendimento do bloco da Figura 8. Reparamos que no encaixe ‘teste’ do bloco se/senão, colocamos um bloco de igualdade e no primeiro encaixe tempos o bloco da figura 8 onde definimos o pino digital D2 como entrada e comparamos ele com o segundo encaixe que tem um bloco escrito “ALTO”, isso quer dizer que quando o pino D2 estiver em um estado de ALTO onde ele está recebendo uma tensão de 5 volts ele irá executar a programação prevista conforme o bloco se/senão foi descrito no manual de controle.



Figura 9

* entrada pullup



Figura 10 – entrada pullup

O bloco entrada pullup refere-se a um resistor integrado no chip Atmega da placa, e pode ser acessado através da programação. Esse bloco efetivamente inverte o comportamento do modo Input, onde HIGH significa que o sensor está desligado e LOW significa que o sensor está ligado.

Exemplo: Na figura 11 podemos observar que arrumamos uma entrada no pino D6 e fazendo a comparação de quando for estado baixo, o led conectado na porta 13 irá acender, e quando o estado do pino D6 for alto, o led irá apagar, mas como a entrada pullup inverte seu estado, quando um botão não estiver pressionado seu estado será ALTO, e quando for pressionado muda para BAIXO, sendo ao contrário do bloco de entrada que vimos anteriormente.



Figura 11 – Exemplo da Aplicação do Bloco Pullup

* pino analógico



Figura 12 – Bloco pino analógico

Este bloco define uma porta analógica como entrada, seu funcionamento foi explicado quando falamos do pino analógico da figura 2.

Exemplo: para visualizar melhor seu funcionamento, podemos conectar na porta A0 da AttoBox um potenciômetro, e conectarmos o pino analógico no bloco ‘imprime’ conforme a figura 13, e abrindo o serial monitor do Ardublock, podemos mexer no potenciômetro e ver a variação de seus valores entre 0 até 1023, isso é referente a tensão de 0 volts até 5 volts.



Figura 13 – Exemplo de aplicação do bloco pino analógico

* define pino analógico



Figura 14 – Bloco define pino analógico

Também pode ser chamado de Controle PWM, permite que através da pulsação da energia fornecida no pino digital, tenhamos valores intermediários de tensão entre 0 e 5 volts, o que permite controlar a velocidade dos motores ou o brilho emitido por um LED. Você tem a possibilidade no Ardublock de escolher valores entre 0 e 255 que correspondem ao intervalo de voltagem de 0 a 5V. Os pinos ou portas digitais que permitem este controle são: D3, D5, D6, D9, D10 e D11.

Exemplo: Na figura 15.a usamos apenas o bloco de controle do PWM e um led conectado na porta D3 para demonstrarmos o seu funcionamento, arrumamos o valor de 255 em um dos encaixes do bloco e enviamos para o AttoBox o programa, iremos ver nosso LED brilhar com a maior intensidade possível. Agora mudamos o valor de 255 para 10 conforme a figura 15.b e enviamos o programa novamente para o arduino, podemos observar que o brilho do led diminui bastante, com isso podemos controlar seu brilho apenas alterando o valor deste bloco.

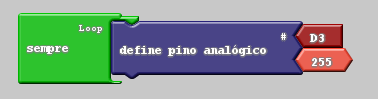


Figura 15.a – Exemplo da aplicação do Bloco analógico



Figura 15.b – Exemplo da aplicação do Bloco analógico

* Som e Bloco Sem Som



Figura 16 – Bloco Som



Figura 17 – Bloco sem Som

Aqui vamos falar de dois blocos para que depois o exemplo fique mais completo, o bloco som da figura 16 gera ondas quadradas na frequência especificada em um pino. Quando conectamos um alto falante, um buzzer neste pino, iremos escutar tons gerados pela frequência escolhida pelo usuário.

Já o bloco sem som da figura 17 faz a interrupção do bloco anterior, deixando um buzzer por exemplo sem tom até ser requisitado o bloco som novamente.

Exemplo: Analisando a figura 18 podemos ver que primeiro arrumamos o bloco ‘Som’ e depois um bloco de delay de 1 segundo, após isso um bloco ‘sem som’ e outro bloco delay, e o que esse programa irá fazer? Devemos conectar na porta 8 da nossa AttoBox o atuador Buzzer, e com este programa ele primeiro irá ficar 1 segundo com o buzzer ligado na frequência 440 gerando um determinado tom, e após isso o bloco sem som desliga esse buzzer por mais 1 segundo e retorna com o loop infinitamente até a placa ser desligada.



Figura 18 – Exemplo da aplicação dos blocos ‘Som’ e ‘Sem Som’