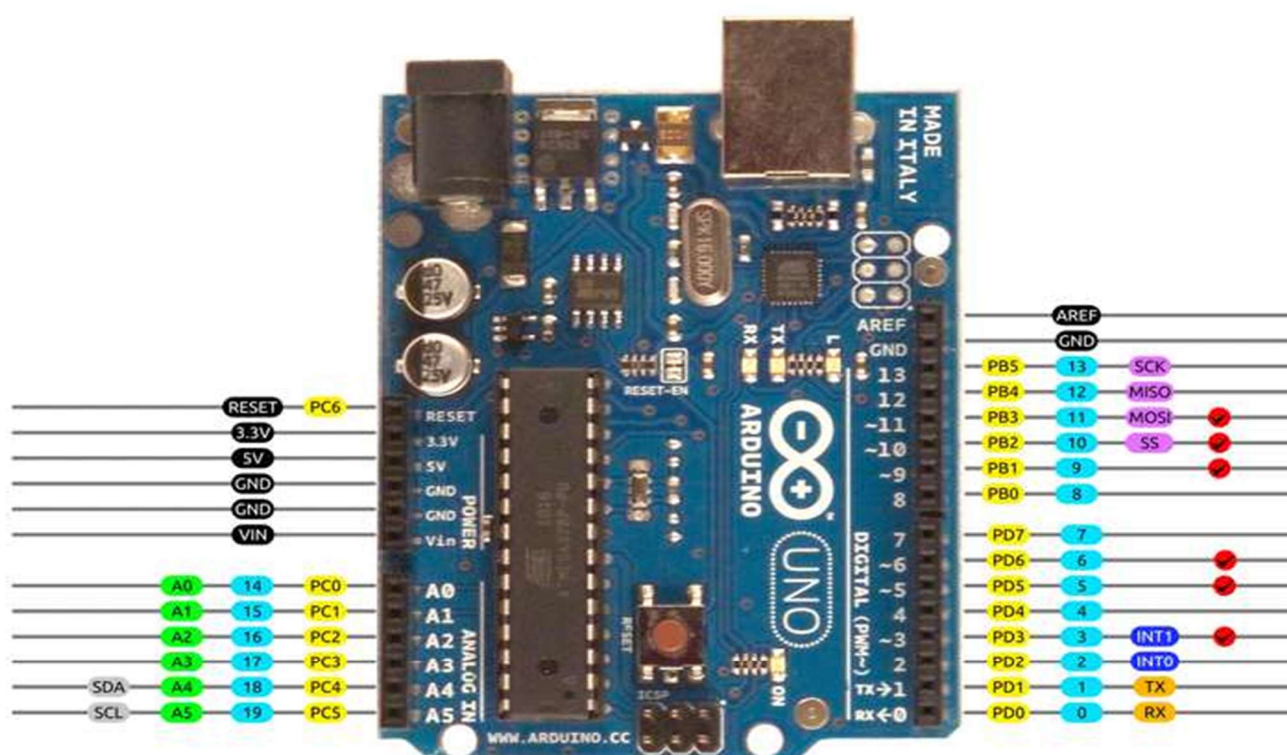


Portas do Arduino

Como visto até aqui, os Arduinos possuem uma série de portas diferentes, cada uma dedicada a um determinado fim e em números diferentes. Ao planejar um projeto, conhecer essas portas é fundamental para dimensionar nosso circuito, verificar a oferta das portas necessárias e, principalmente, saber fazer o melhor uso possível de cada uma. Nesta disciplina, utilizaremos o Arduino Uno R3 e por isso analisar a sua pinagem (Pinout) é fundamental. Então, vamos analisar o esquemático abaixo e a partir dele falar um pouco sobre as principais portas.



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

CC BY SA 2014 by Bouni
Photo by Arduino.cc

Portas Digitais

As portas digitais do Arduino, como o próprio nome demonstra, trabalham com sinal digital, tanto para leitura quanto para escrita. Resumidamente, elas têm a capacidade de trabalhar com uso do sistema binário, onde um bit 0 é a porta não energizada e o bit 1 é a porta energizada (no caso do Arduino Uno, com uma tensão aproximada de 5 V). No caso da porta atuando como uma entrada (recebendo sinais de sensores e outros dispositivos),

de tempos em tempos é realizada uma leitura dessa tensão e a presença ou ausência da mesma é convertida em um conjunto de 0 e 1. De forma análoga, o mesmo ocorre quando as portas digitais funcionam com portas de saída, escrevendo na porta 0 ou 1 (ou seja, sem tensão ou com tensão) de tempos em tempos. Estas portas são indicadas para trabalhar com sensores e atuadores que também utilizem sinal digital, mas também como portas de controle, ligando e desligando dispositivos externos (seja um led, seja um relé). Na leitura de dados (de um sensor, por exemplo) é possível receber diretamente o valor exato da grandeza medida de forma binária.

Portas PWM

As portas PWM (Pulse Width Modulation) são portas digitais especiais com a capacidade de transformar sinais digitais em modularizações de sinal que permitam, na prática, regular a saída modificando a sua tensão (e consequentemente a sua corrente). Imaginando um led ligado a uma porta digital normal, o máximo que poderemos fazer com ele é ligá-lo ou desligá-lo. Porém, se essa porta digital for uma PWM, podemos regular a sua saída e, deste modo, alterar a luminosidade do led, fazendo que o mesmo brilhe mais forte ou mais fraco.

Portas analógicas

Diferentemente das portas digitais, as portas analógicas trabalham com sinais que podem variar entre 0 e 5 V, assumindo diversos valores diferentes. Assim, um sensor analógico é especialmente desenvolvido para, ao medir a grandeza ou o fenômeno desejado, variar a tensão emitida. Por exemplo, um sensor de temperatura analógico que mede temperaturas entre 0 e 100 graus celsius, é construído para ao medir 0°C (o primeiro valor possível) gerar uma tensão de 0 V e ao medir uma temperatura de 100°C (o último valor possível) gerar uma tensão de 5 V. Qualquer temperatura acima de 0°C e abaixo de 100°C, irá gerar uma tensão diferente (acima de 0 V e abaixo de 5 V). Estes sensores são geralmente mais baratos e rápidos que os digitais (por não necessitam de conversões binárias para devolver o valor final pronto) mas, por isso mesmo, têm uma maior dificuldade no uso, já que sempre necessitamos fazer um mapeamento entre a tensão medida e o quanto ela representa dentro da grandeza medida.

Portas de alimentação

O Arduino Uno pode ser alimentado de 3 formas diferentes: pela sua porta USB (com uma entrada de 5V), pelo seu conector de alimentação (permitindo carregadores ou pilhas com uma tensão recomendada entre 7V e 12V) e pela sua porta Vin (soldando uma fonte ou bateria com entrada também entre 7V e 12V). Alimentada a placa, ela garante uma saída de 5V nas suas portas digitais e de até 5 V nas suas portas analógicas. Porém, para isso, teríamos que configurar as portas (via código) como portas de saída e escrever na sua saída que estão em estado alto (como se fosse um estado “ligado”). Para simplificar então a alimentação de sensores, atuadores e outros dispositivos externos, o Arduino Uno oferece algumas portas com tensão fixa e que estão sempre ativas: uma com tensão de 5V, outra com tensão de 3.3V. Além disso, o Vin, caso não utilizado para alimentar o Arduino, entregará uma tensão igual à tensão de alimentação da placa (se o Arduino está alimentado por uma fonte de 12V no seu conector, o Vin oferecerá uma tensão de saída fixa também de 12V, por exemplo). Além disso, temos 2 portas GND que são o terra (negativo).

Porta Reset

A porta reset serve para, quando acionada, resetar a nossa placa, apagando o código carregado nela pelo bootloader e, com isso, interromper qualquer execução que estiver ocorrendo no momento. Podemos também fazer isso simplesmente pressionando o botão reset presente na placa (que está ligado à mesma porta). Mas é interessante esse acionamento programaticamente para situações extremas. Por exemplo, imagine que detectamos um comportamento inesperado de nosso sistema ou na execução de nosso código que pode trazer riscos à segurança do usuário. Então caso, não consigamos parar essa execução de outra forma (em um loop travado, por exemplo), resetamos a placa e forçamos a interrupção de qualquer execução.

Porta RX e TX

Os pinos digitais 0 e 1 do Arduino Uno, são dedicados às portas RX e TX. Elas são portas de comunicação serial direta, capazes de enviar comandos e informações (TX = Transmitter) ou de receber comandos e informações (RX = Receiver) que são interpretadas pela placa. Isso permite desde a programação do Arduino por outro Arduino (como o Pro Micro ou o LilyPad que não possuem entrada USB para carregar os códigos através do

bootloader), de um sensor ou atuador pelo Arduino (configurar os parâmetros de uma rede WiFi em um módulo ESP8266) ou mesmo para que Arduinos possam trocar informações durante o funcionamento (mensagens, valores de variáveis, medições, etc).

Portas de Interrupção

Os pinos digitais 2 e 3 do Arduino Uno, são dedicados às portas de interrupção. Como o Arduino utiliza microcontroladores com apenas um núcleo, temos um monoprocessamento. Isso na prática fará com que quando entrarmos em uma tarefa muito pesada (um laço que executa várias vezes, um método que demora muito para ser executado), enquanto essa tarefa não terminar, ficamos “travados” na linha de código. Agora imagine um sistema crítico que realiza tarefas pesadas mas, caso um sensor detecte uma situação de perigo, seja necessário interromper a execução. Se este sensor estiver ligado a uma das portas de interrupção, ele poderá ser lido em paralelo, de forma independente, mesmo que uma tarefa pesada esteja sendo realizada e a execução de nosso código esteja parada. Assim, conseguimos implementar uma camada de segurança ao nosso sistema.

Portas I2C (SDA e SCL)

Os pinos analógicos A4 e A5 do Arduino Uno, são dedicados às portas SDA (Serial Data) e SCL (Serial Clock), utilizadas para permitir o uso do protocolo de comunicação I2C. O I2C (Inter-Integrated Circuit), é na verdade uma comunicação serial, porém com uma característica muito interessante: trabalha com endereçamento. Isso permite que possamos ligar quantos sensores e atuadores I2C quisermos na mesma porta, pois na hora de ler um valor ou enviar um valor, o endereço deles (que deve ser único) será utilizado para auxiliar no direcionamento dos pacotes.

Portas SPI

De forma análoga ao I2C, as portas SPI (Serial Peripheral Interface) são uma comunicação serial com endereçamento mas para a ligação entre placas ou entre a placa e periféricos externos. No Arduino Uno, são implementadas nas portas digitais 10, 11, 12 e 13.

Portas AVR

As portas AVR nada mais são do que as portas que podem ser controladas via programação no Arduino. Incluem todas as portas digitais, todas as portas analógicas, além da porta reset.