Introdução a Microcontroladores

- Introdução a Microcontroladores
- Microcontroladores comuns

Introdução a Microcontroladores

O que são microcontroladores?

Os microcontroladores são dispositivos eletrônicos altamente integrados que combinam em um único chip um processador (CPU), memórias (RAM e ROM/Flash), periféricos de entrada e saída (I/O) e outros componentes necessários para controlar sistemas embarcados. Eles são projetados para realizar tarefas específicas em dispositivos eletrônicos, como controlar motores, sensores, displays e outros elementos de um sistema.

Devido à sua simplicidade e eficiência, os microcontroladores são amplamente utilizados em aplicações como eletrodomésticos, automação industrial, dispositivos médicos, sistemas automotivos, brinquedos eletrônicos, entre outros. Sua principal característica é a capacidade de operar de forma autônoma, sem a necessidade de componentes adicionais complexos.

Diferença entre Microcontroladores e Microprocessadores

Embora os termos "microcontrolador" e "microprocessador" sejam frequentemente confundidos, eles se referem a componentes distintos com finalidades diferentes:

Microprocessadores: Um microprocessador é essencialmente a unidade central de processamento (CPU) de um sistema. Ele é responsável por executar instruções e processar dados, mas depende de componentes externos, como memórias (RAM e ROM), controladores de entrada e saída e outros periféricos, para funcionar.

São comumente usados em sistemas de maior complexidade, como computadores pessoais, laptops e servidores, onde a capacidade de processamento e expansibilidade é essencial.

Microcontroladores: Um microcontrolador, por outro lado, é uma solução "tudo-em-um", que inclui a CPU, memórias, e periféricos integrados no mesmo chip. Essa integração permite o desenvolvimento de sistemas compactos, de baixo custo e baixa potência.

São ideais para aplicações embarcadas, onde o foco é realizar tarefas específicas com eficiência e simplicidade.

Exemplo Prático: Considere um termostato inteligente usado para controlar a temperatura de um ambiente. Um microcontrolador é a escolha ideal porque pode integrar sensores de temperatura, controles de relé para o sistema de aquecimento/resfriamento e comunicação com outros dispositivos, tudo em um único chip. Em contrapartida, um microprocessador seria mais adequado para uma aplicação como um computador de bordo em um automóvel, onde é necessária maior capacidade de processamento e a integração com diversos sistemas complexos.

Compreender essas diferenças é essencial para escolher o componente mais adequado para cada projeto, garantindo desempenho, custo-benefício e eficiência no uso dos recursos.

Tipos de Entradas e Saídas

Os microcontroladores, como o Arduino e o ESP32, são dispositivos versáteis capazes de interagir com o mundo externo por meio de pinos configuráveis como entradas ou saídas. Esses pinos são os principais canais de comunicação entre o microcontrolador e sensores, atuadores ou outros dispositivos eletrônicos.

Compreender os diferentes tipos de entradas e saídas é essencial para aproveitar todo o potencial de um microcontrolador. Saber como configurar e utilizar esses recursos é o primeiro passo para desenvolver projetos inovadores, desde sistemas simples de automação até aplicações mais complexas em IoT (Internet das Coisas). A seguir, vamos explorar os diferentes tipos de entradas e saídas que você encontrará em microcontroladores.

Entradas Digitais

As entradas digitais permitem que o microcontrolador detecte o estado lógico de um sinal:

- **Nível ALTO (HIGH):** Geralmente corresponde a uma tensão próxima ao valor da alimentação (ex.: 3.3V ou 5V).
- **Nível BAIXO (LOW):** Normalmente representa 0V ou GND.

Essas entradas são usadas para ler sinais de botões, interruptores e sensores digitais que operam com lógica binária. Exemplo de uso: Um botão pressionado conecta o pino a GND, mudando o estado de HIGH para LOW.

Entradas Analógicas

As entradas analógicas permitem medir sinais de tensão variáveis. Esses sinais são convertidos em valores digitais através de um conversor Analógico-Digital (ADC).

Resolução: O número de bits do ADC determina a precisão da medição. Por exemplo, um ADC de 10 bits em um Arduino gera valores entre 0 e 1023. O intervalo de tensão depende do

microcontrolador. No Arduino UNO, por exemplo, é de 0 a 5V.

Exemplo de uso: Leitura de um potenciômetro ou sensor de luz (LDR).

Saídas Digitais

As saídas digitais são usadas para controlar dispositivos que operam em dois estados: LIGADO (HIGH) ou DESLIGADO (LOW).

Podem acionar LEDs, relés, buzinas, entre outros dispositivos binários.

Muitas saídas podem fornecer corrente limitada (ex.: 20mA no Arduino UNO), necessitando de transistores ou módulos de driver para controlar dispositivos de maior potência.

Exemplo de uso: Acender um LED ou ligar/desligar um motor através de um relé.

Saídas Analógicas (PWM)

Embora a maioria dos microcontroladores não tenha saídas analógicas puras, eles podem simular sinais analógicos usando PWM (Pulse Width Modulation).

PWM: Uma técnica que controla a largura do pulso em uma série de sinais digitais para variar a potência ou tensão efetiva. Usado para controlar a intensidade de LEDs, a velocidade de motores ou criar sinais analógicos aproximados. Exemplo de uso: Ajustar o brilho de um LED ou controlar a velocidade de um motor.

Entradas e Saídas de Comunicação

Os microcontroladores possuem pinos dedicados para comunicação com outros dispositivos, incluindo:

- **UART (Serial):** Para comunicação serial, amplamente usada para debugar códigos ou comunicar com módulos externos, como módulos Bluetooth.
- **I2C:** Protocolo para conectar vários dispositivos com apenas dois pinos (SDA e SCL).
- **SPI:** Um protocolo mais rápido que I2C, mas que exige mais pinos (MISO, MOSI, SCLK e CS).

Exemplo de uso: Conectar um display OLED via I2C ou comunicar com um sensor IMU via SPI.

Entradas e Saídas Especiais

Pinos de interrupção: Permitem que o microcontrolador reaja imediatamente a um evento externo, ignorando o fluxo normal do programa.

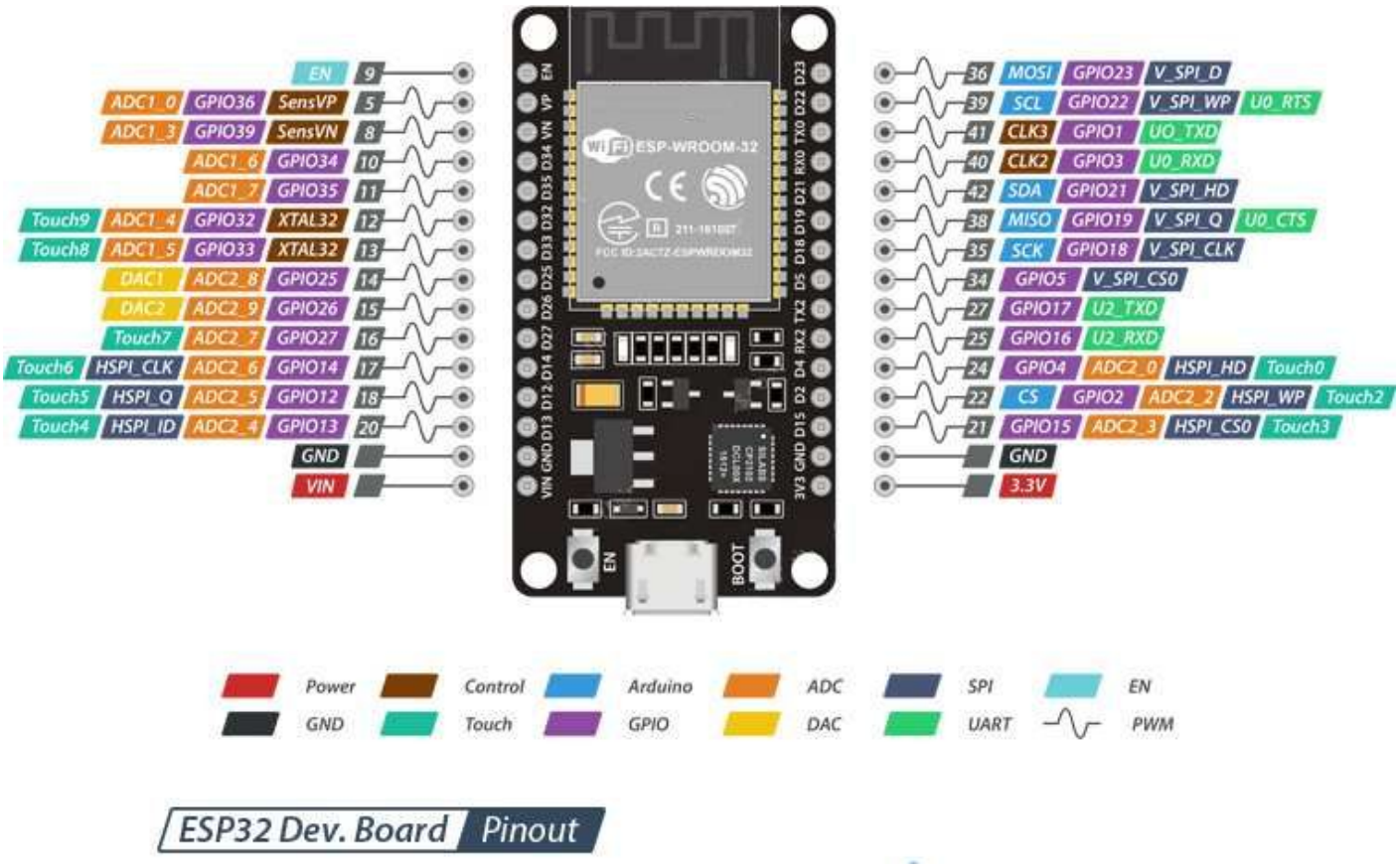
- **Entradas capacitivas (ESP32):** Usadas para criar botões touch sensíveis ao toque.
- **Conversores Digital-Analógico (DAC):** Disponíveis em alguns microcontroladores, como o ESP32, para gerar sinais analógicos reais.

Microcontroladores comuns

ESP32

Arquitetura

Abaixo temos uma imagem mostrando os tipos de conexões suportadas para cada porta da ESP32, as principais que serão usadas ao longo dessa apostila são: as entradas analógicas (ADC), digitais (GPIO) e as de comunicação I2C (GPIO21 E GPIO22).

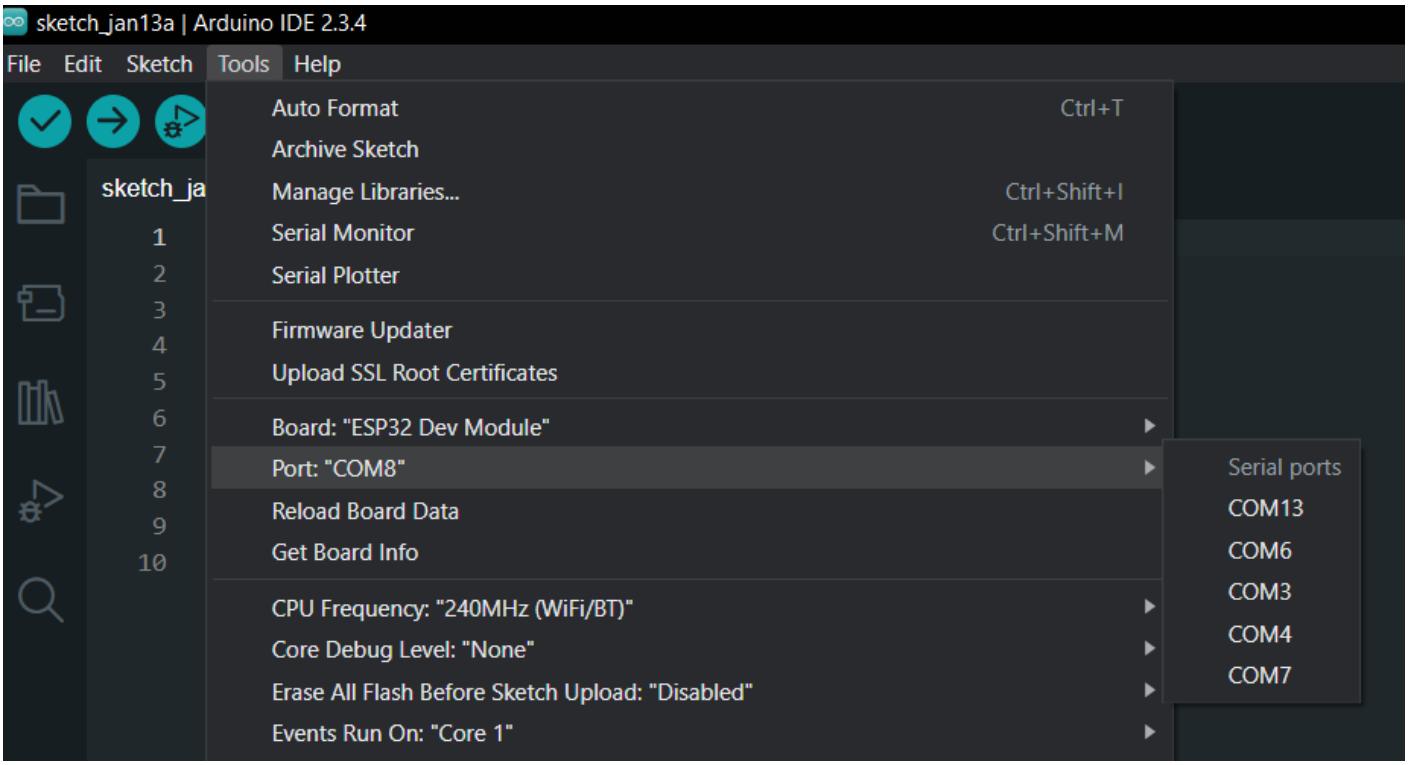


Guia de conexão

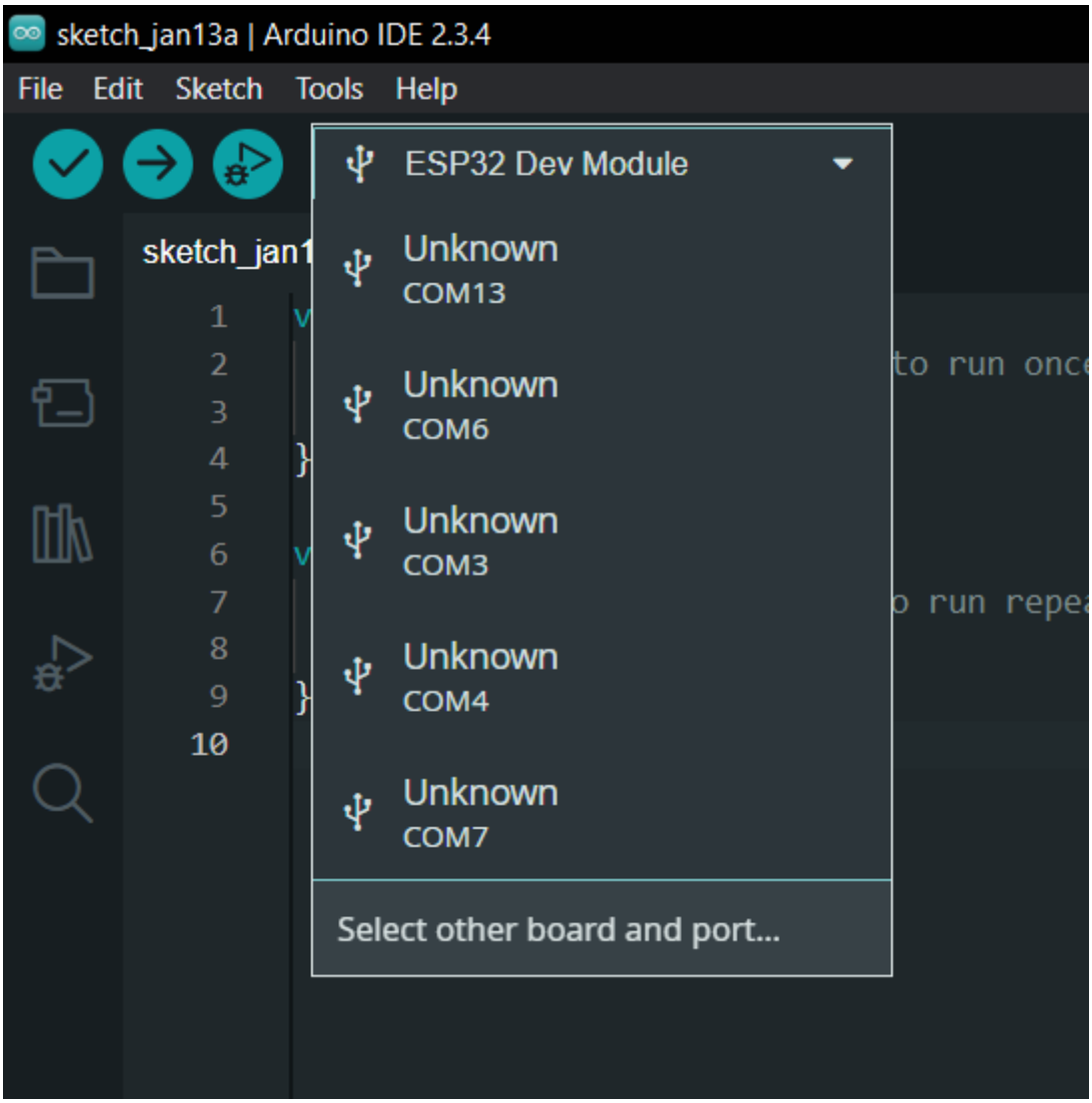
1. Baixar 2 drivers de USB para comunicação com a placa pelo Windows: CP210x e CH341SER. (Para o segundo driver, é necessário que a placa esteja conectada em uma

porta USB para funcionar)

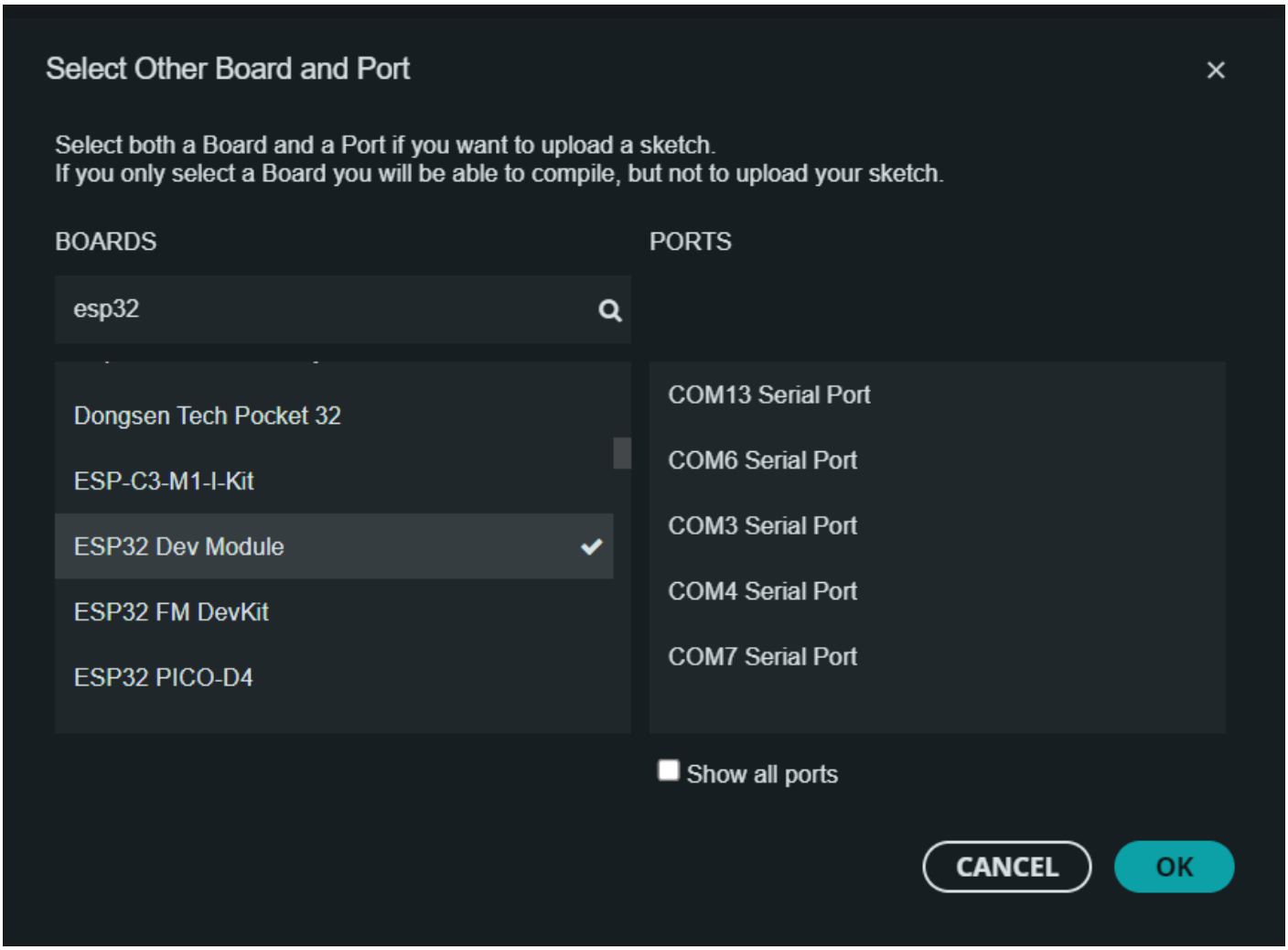
2. Com o aplicativo Arduino IDE (download na secção 1.1.1) será possível ver que, após a instalação dos drivers, uma nova porta COM aparecerá e ela é a que deve ser utilizada.
3. Para ver as portas disponíveis, abra o menu Tools $->$ Port: (figura \ref{fig: Econex1})



4. Escolher o modelo de placa que você está usando no menu de escolha de porta e placa (“Select other board and port...”): (figura \ref{fig: Econex2})



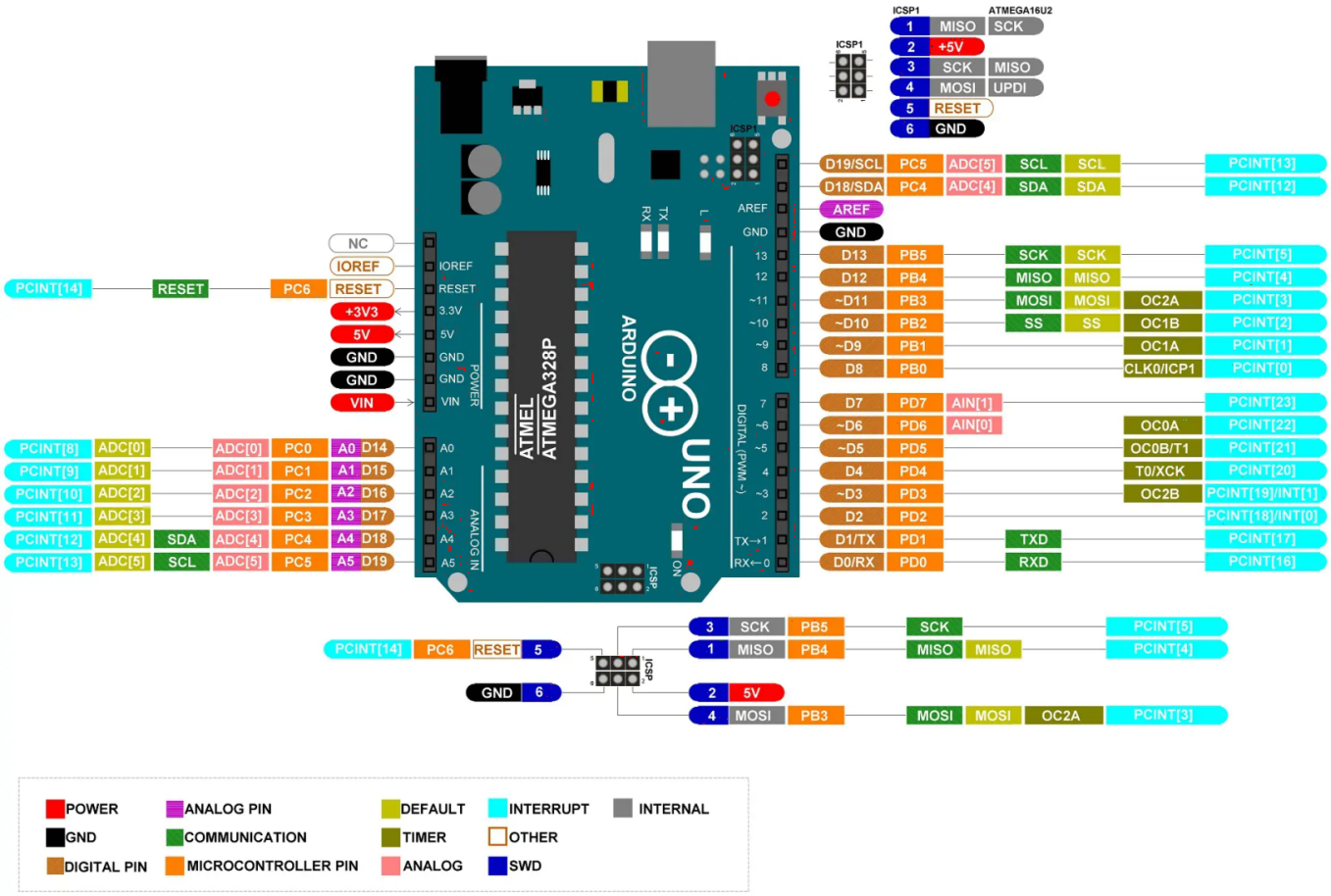
5. Com o menu aberto você irá pesquisar a placa que está usando e selecionar a porta COM que apareceu nova quando os drivers foram instalados: (figura \ref{fig: Econex3})



Arduino

Arquitetura

Abaixo temos uma imagem mostrando os tipos de conexões suportadas para cada porta dO aRDUINO, as principais que serão usadas ao longo dessa apostila são: as entradas analógicas (ANALOG PIN), digitais (DIGITAL PIN) e as de comunicação I2C (SCL e SDA).



Guia de Conexão

1. Fazer o download da interface Arduino IDE;
2. Com a IDE aberta, a placa Arduino que for conectada deve ser reconhecida imediatamente pelo programa que fará a conexão automática no sistema, pronta para uso.