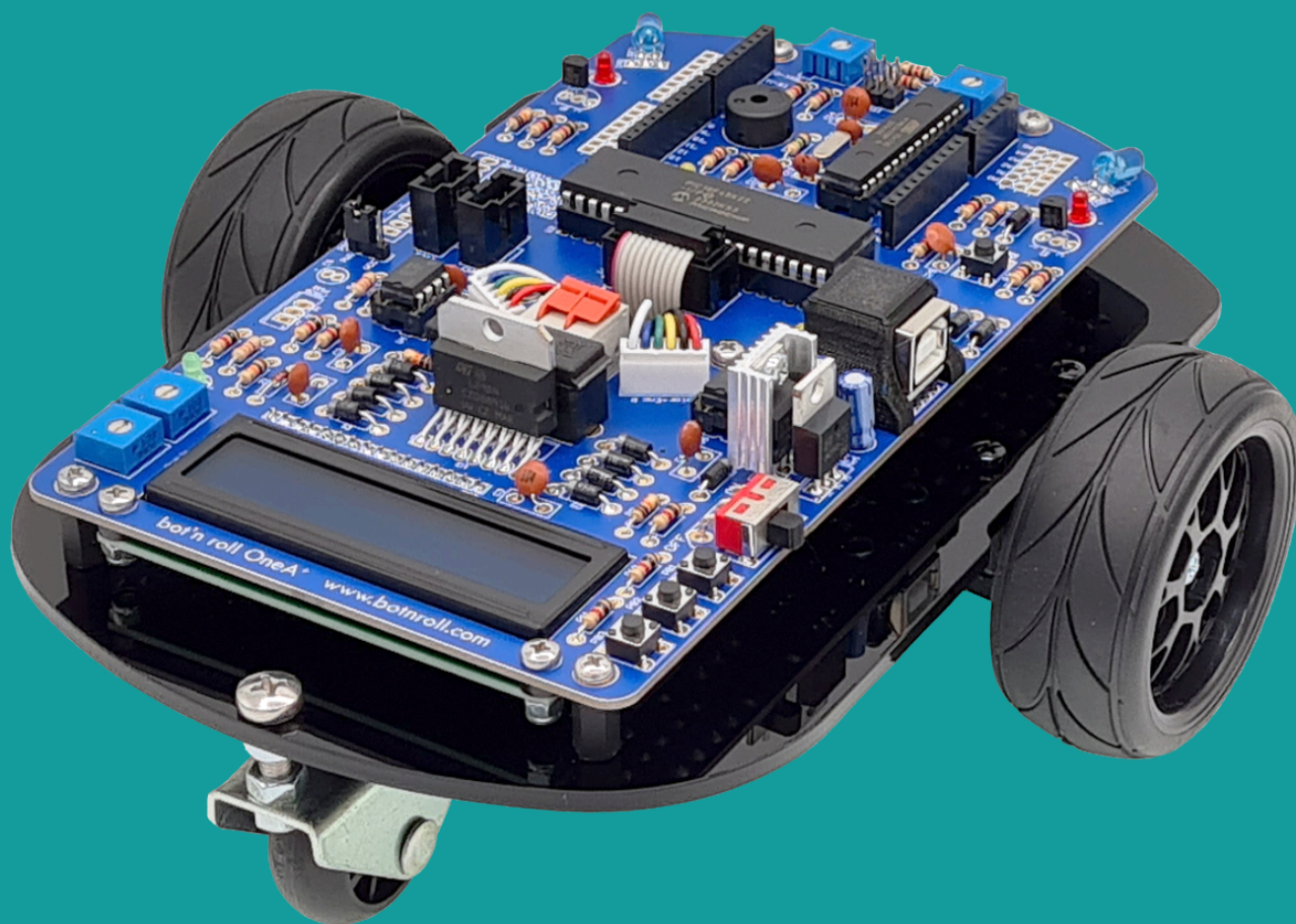


bot'n roll ONE A⁺

build your own robot



manual de software
Arduino[®]



CONTEÚDO

1. Introdução.....	2
1. Programar o Bot'n Roll ONE A+.....	2
2. Biblioteca BnrOneAPlus para Arduino.....	3
3. Arduino IDE.....	4
4. Linguagem de Programação C.....	5
Anexo A: Instalação do VCP Driver do Conversor USB-Série (RS232).....	6
Anexo B: Ambiente de Programação Arduino 2.0.....	6
B.1 Instalação do Arduino IDE.....	6
B.2 Instalação da Biblioteca BnrOneAPlus para Arduino.....	6
B.3 Configuração da Comunicação com o Robô.....	7
B.4 Carregar um Programa para o Bot'n Roll ONE A+.....	8

Revisão do Documento: 2 de Março de 2025

1. INTRODUÇÃO

O Bot'n Roll ONE A+ é programado usando linguagem C com o ambiente de programação Arduino IDE. O microcontrolador ATmega328 presente no robô possui o *bootloader* do Arduino Uno, logo o robô é programado como se de um Arduino Uno se tratasse.

O robô possui um segundo microcontrolador, um PIC18F45K22 fornecido, pré-programado, com *software* desenvolvido pela botnroll.com. No Bot'n Roll ONE A+, este microcontrolador funciona como um dispositivo escravo, “*slave*”, que executa as ordens de comando do “*master*” ATmega328.

Os dois microcontroladores do Bot'n Roll ONE A+ comunicam entre si através do barramento SPI “*Serial Peripheral Interface*”. Os microcontroladores trocam informação de uma forma coordenada e bem definida. Para isso foi desenvolvido um protocolo de transferência de dados entre o *master* e o *slave*. O *master* utiliza uma lista de comandos que correspondem a ordens de controlo e cada comando gera uma resposta por parte do *slave*. A listagem de comandos e o modo como os dados são transmitidos entre *master* e *slave* estão definidos na biblioteca BnrOneAPlus.

A biblioteca para Arduino “BnrOneAPlus” permite ao utilizador controlar o robô de uma forma simples e para isso basta que use corretamente os comandos da biblioteca no Arduino IDE. Estes comandos estão listados e explicados neste manual.

Embora ambos os microcontroladores possam ser programados em linguagem C pelo utilizador, somente o ATmega328 com *bootloader* Arduino é programado no dia a dia com recurso à biblioteca “BnrOneAPlus”.

O PIC18F45K22 pode ser programado em linguagem C usando o ambiente de programação MPLABX IDE e o compilador XC8 da *Microchip* ou outro *software* compatível. No entanto, isto deve ser feito somente por utilizadores avançados, pois programar o PIC18F45K22 para incluir uma nova funcionalidade requer que se atualize também a biblioteca “BnrOneAPlus” para que o Arduino consegue utilizar a nova funcionalidade. Contacta a botnroll.com se gostarias de ver uma nova funcionalidade implementada no teu Bot'n Roll ONE A+!

1. PROGRAMAR O BOT'N ROLL ONE A+

Para programar o Bot'n Roll ONE A+ é necessário que tenhas o teu computador preparado com todas as ferramentas necessárias, ou seja:

- [VCP driver instalado, o driver para porta USB do Bot'n Roll ONE A+ \(Anexo A\);](#)
- [Arduino IDE instalado \(Anexo B1\);](#)
- [Biblioteca “BnrOneAPlus” instalada no Arduino IDE \(Anexo B2\).](#)

Para informações detalhadas sobre a instalação dos itens acima referidos consulta os anexos A e B no final deste manual.

A linguagem C++ é também uma ferramenta necessária para a programação do Bot'n Roll ONE A+. Se ainda não estás muito à vontade com a linguagem C++, tens os exemplos da biblioteca que são um bom guia para te iniciar neste mundo da programação. Consulta também as apresentações da RoboParty sobre programação em C, e claro, na internet existem milhares de páginas que explicam a linguagem C++.

2. BIBLIOTECA *BnrOneAPlus* PARA ARDUINO

Uma biblioteca é um conjunto de código “pré-fabricado” que podes inserir e utilizar no teu programa. Para usares a biblioteca “**BnrOneAPlus**” basta que a incluas no teu código:

```
#include<BnrOneAPlus.h>
```

e que crie uma instância para a classe `BnrOneAPlus`:

```
BnrOneAPlus one;
```

A partir daqui tens acesso a todas as funções da biblioteca precedidas pela instância que definiste, ou seja: `one.função_da_biblioteca()`;

Uma biblioteca é normalmente criada para manipulação de dados ou de *hardware* e tem sempre, no mínimo, dois ficheiros mas no caso do Arduino há ainda um ficheiro adicional com a extensão “.txt”.

- Um ficheiro com a extensão “.h” (“*header*”) que contém a listagem de todas as funções, comandos e definições da biblioteca;
- Um ficheiro com a extensão “.cpp” (“*c++ source*”) com a codificação de todas as funções apresentadas no ficheiro *header*.
- Um ficheiro **keywords.txt** que permite ao Arduino IDE identificar as funções da biblioteca e apresentá-las com uma coloração diferente do resto do código.

A biblioteca “*BnrOneAPlus*” foi criada para a manipulação do *hardware* associado ao PIC18F45K22 e permite ao Arduino interagir com ele através do barramento de comunicação SPI. O Arduino tem acesso a todo o *hardware* e funcionalidades definidas na biblioteca e no *software* do PIC18F45K22. A biblioteca “*BnrOneAPlus*” e o *software* do PIC18F45K22 foram feitos “um para o outro” e qualquer alteração num deles requer que se ajuste o outro também.

A biblioteca encontra-se disponível online em:

- <https://github.com/botnroll/BnrOneAPlus>

A sua respetiva documentação e descrição das funções disponíveis encontra-se em:

- https://botnroll.github.io/BnrOneAPlus/class_bnr_one_a_plus.html.

3. ARDUINO IDE

O ambiente de desenvolvimento Arduino contém um editor de texto para escrever o código, uma área de mensagens, uma consola de texto, uma barra de ferramentas com as funções mais importantes e ainda uma série de menus. Este efetua a ligação ao *hardware* Arduino do Bot'n Roll ONE A+ para transferir o código e comunicar com o robô.







Um programa para Arduino tem o nome de “*sketch*”, é escrito no editor de texto e guardado com a extensão “.ino” no seu computador.

A área de mensagens apresenta informação sobre a gravação e exportação dos programas e também apresenta os erros.

A consola apresenta mensagens de texto com informação detalhada sobre os erros e outra informação.

No canto inferior direito da janela é apresentada informação sobre a placa a ser programada e a porta série em utilização.

Na barra de ferramentas existem botões e estas são as suas funções:

-  **Verify:** Verificar a existência de erros no código.
-  **Upload:** Compilar o código e enviar para o Arduino.
-  **New:** Criar um *sketch*.
-  **Open:** Abrir um *sketch* guardado no computador.
-  **Save:** Guardar o *sketch*.
-  **Serial Monitor:** Abrir a monitorização da porta série.

O *serial monitor* permite visualizar dados enviados do Arduino para o computador e também permite o envio de dados do computador para o Arduino. É muito útil na programação, pois consegue imprimir aqui texto e o valor das variáveis e assim efetuar o “*debug*” (procura de erros) do teu programa. Quando abres o *serial monitor*, o teu programa no Arduino reinicia.

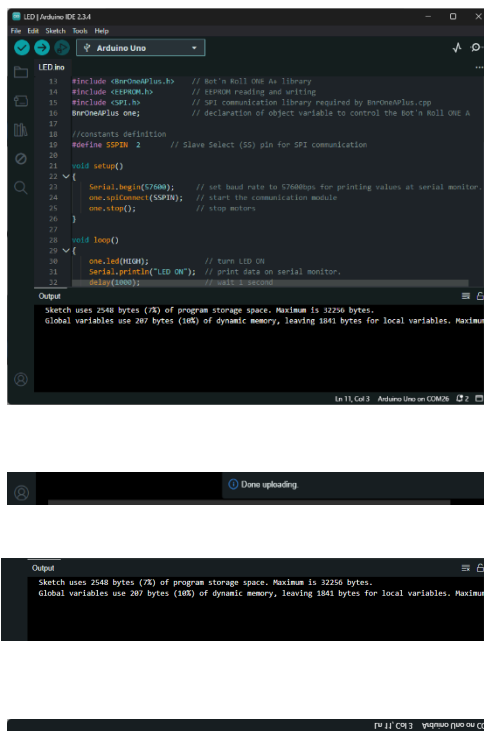


Fig. 1: Módulos do Arduino IDE

4. LINGUAGEM DE PROGRAMAÇÃO C

A linguagem C foi desenvolvida em 1972 por Dennis Ritchie nos laboratórios Bell em Nova Jersey. Surgiu com o intuito de ser uma linguagem poderosa e rápida para ser utilizada no sistema operativo Unix que Dennis desenvolvia. Ao longo do tempo foi melhorada e atualizada, mostrando-se muito robusta e fiável e passou a ser utilizada também por outros sistemas operativos como o Windows, MacOs e Linux. Está em constante evolução desde que surgiu a primeira versão conhecida como “K&R C”. Em 1989 surgiu a primeira especificação como padrão pelo instituto norte-americano de padrões, o “ANSI C”. Em 1990 o “ISO C” pela Organização Internacional para a Padronização. Em 1999 surgiu o *standard* “C99” e desde então seguiram-se revisões periódicas de modo a tornar a linguagem mais poderosa e mais adaptada aos tempos modernos.

Adiciona aqui talvez um subtítulo: “Linguagem C no Bot'n Roll ONE A+”

Todos os programas para Arduino têm duas rotinas, ou funções, que são obrigatórias. A rotina de configuração, “*setup()*”, é executada somente uma vez no arranque do seu programa. Aqui deve ser colocado todo o código necessário para inicializar variáveis, configurar pinos de entrada e saída, configurar comunicação SPI, Série, I2C, enfim, todas as configurações necessárias.

Depois da configuração, o teu programa entra na rotina “*loop()*” e lá permanece indefinidamente. O termo *loop* significa **ciclo** e neste caso é um ciclo infinito, pois quando o programa atinge o final do ciclo, volta ao início e começa tudo de novo! É aqui que escreves o teu programa e crias a inteligência para o teu robô!

A programação em C++ propriamente dita, não está explicada neste manual, mas remetemos para os exemplos da biblioteca “BnrOneAPlus” e do Arduino em geral. Todo o código está devidamente comentado e terás que experimentar e testar para perceberes como funciona. Deixamos-te aqui algumas dicas:

- Cria programas novos a partir dos exemplos básicos. Experimenta juntar 3 ou 4 funcionalidades do robô no mesmo programa a partir dos exemplos básicos!
- Um programa raramente funciona à primeira! Não desanimes, procura o problema e resolve-o!
- Insere código gradualmente e testa com frequência para ver se tudo acontece como esperado.
- Para um programa ficar bom vais demorar mais tempo a testá-lo do que a escrevê-lo!
- Usa as ferramentas de *debug* como o LED, o serial monitor ou o LCD para imprimir o valor das variáveis e verificar se o programa passa numa determinada zona do código.
- Programar é como praticar um desporto novo, no início é doloroso, pois não tens a condição física necessária, não sabes as regras e andas um pouco perdido. Nos treinos praticas, aprendes e melhoras em todos os aspetos. Como resultado do teu trabalho, acabas por fazer parte da equipa principal!

O Bot'n Roll ONE A+ permite a interação com um leque de *hardware* muito vasto. Existem extras, vulgarmente chamados “*shields*” para Arduino para fazeres praticamente tudo o que imaginas e são compatíveis com o Bot'n Roll ONE A+! Todos os *shields* possuem bibliotecas para te auxiliar na sua utilização e integração e a tua imaginação é o limite!

ANEXO A: INSTALAÇÃO DO VCP DRIVER DO CONVERSOR USB-SÉRIE (RS232)

O driver permite que o sistema operativo do teu computador comunique com o Bot'n Roll ONE A+.

Para instalar o driver visita à página de suporte do Bot'nRoll ONE A+ (<http://botnroll.com/one-a-plus/>) e faz *download* clicando em “VCP Driver - Windows” ou “VCP Driver - Mac OS X” de acordo com o teu sistema operativo. Assim que terminar o *download*, descompacta o ficheiro com a extensão “.zip” e executa a aplicação.

Sempre que ligares o robô ao computador usando o cabo USB, é criada uma porta COM virtual (VCP) pela qual é efetuada a comunicação entre o Bot'n Roll ONE A+ e o PC. A aplicação para a programação do robô usa esta porta para comunicar com o Bot'n Roll ONE A+ e desta forma transferir os programas para o robô.

O conversor USB-Série utilizado no Bot'n Roll ONE A+ é um **PoUSB12** da *PoLabs* usa o dispositivo **Bridge CP2102** da *Silicon Labs*.

ANEXO B: AMBIENTE DE PROGRAMAÇÃO ARDUINO 2.0

O *software* utilizado para a programação do robô é o Arduino IDE 2.0. Esta aplicação é necessária para fazer a edição dos programas em linguagem C++. Serve também para transferir os teus programas para o Bot'n Roll ONE A+.

B.1 INSTALAÇÃO DO ARDUINO IDE

Para a instalação do Arduino IDE, visita a página de suporte do Bot'n Roll ONE A+ <http://botnroll.com/one-a-plus/>. Na secção “*Software | Drivers*” clica em “**Arduino IDE**” para instalar de acordo com o teu sistema operativo.

Assim que o *download* terminar, descompacta o ficheiro com a extensão “.zip” e coloca a pasta extraída numa diretoria do teu computador a teu gosto.

Esta pasta contém várias subpastas e ficheiros, entre eles a aplicação “**arduino.exe**”, o executável que arranca o Arduino IDE.

B.2 INSTALAÇÃO DA BIBLIOTECA *BnrOneAPlus* PARA ARDUINO

As bibliotecas são as suas ferramentas de trabalho em programação. A biblioteca “**BnrOneAPlus**” desenvolvida pela **botnroll.com** para o Arduino IDE possui todos os comandos necessários para o controlo do robô. Esta biblioteca deve ser instalada no Arduino IDE.

Para instalares a biblioteca no [Arduino™ IDE 2.0](#) tens que clicar no separador “**Sketch**” --> “**Include Library**” --> “**Manage Libraries...**”, procurar “**BnrOneAPlus**” e instalar a biblioteca clicando no botão “**Install**”.

Para instalares a biblioteca no [Arduino™ IDE 1.X](#) tem que clicar no separador “**Sketch**” --> “**Include Library**” --> “**Add .ZIP Library...**” seleciona o ficheiro [BnrOneAPlus.zip](#) e a biblioteca é instalada automaticamente. Fecha e abre novamente o Arduino™ IDE para teres a biblioteca funcional! Os procedimentos para a instalação do VCP driver e da biblioteca *BnrOneAPlus* estão também descritos no Manual de Montagem.

B.3 CONFIGURAÇÃO DA COMUNICAÇÃO COM O ROBÔ

Antes de efetuares este passo, certifica-te que instalaste o VCP driver corretamente ([ver ANEXO A](#)). Conecta o Bot'n Roll ONE A+ ao computador usando o cabo USB fornecido. Neste momento, será atribuída automaticamente uma porta COM para a comunicação com o robô.

Abre o Arduino IDE e na barra no topo da janela, encontrarás um menu onde é possível escolher a Placa e a Porta. Selecciona a placa “**Arduino Uno**”. O **Bot'n Roll ONE A+** será programado como se de um Arduino Uno se tratasse.

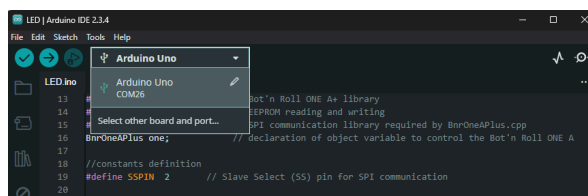


Fig. Seleccionar a placa a programar

Caso este não apareça instantaneamente, podes clicar em “**Select other board and port...**” e irá aparecer o menu seguinte. Escolhe a placa “**Arduino Uno**” e a porta COM correspondente para programar o **Bot'n Roll ONE A+**.

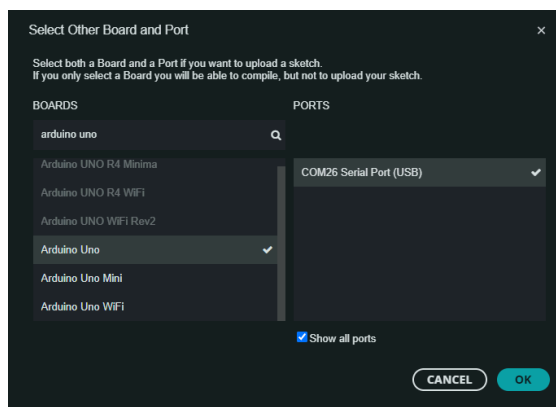


Fig. Seleccionar a Porta Série

Se nenhuma porta COM estiver disponível, o mais certo é não teres instalado corretamente o VCP driver do conversor USB-Série.

Abre o gestor de dispositivos do Windows e procura o item com a designação “Portas (COM e LPT)”. Expandindo este item, verás todas as portas COM atribuídas.

“**Silicon Labs CP210x USB to UART Bridge (COM21)**” é a designação que identifica a porta de ligação ao Bot'n Roll ONE A+. (No exemplo da figura foi atribuída a porta **COM21**.)

Caso não apareça o item “**Silicon Labs CP210x USB to UART Bridge**” terás que instalar corretamente o VCP driver.

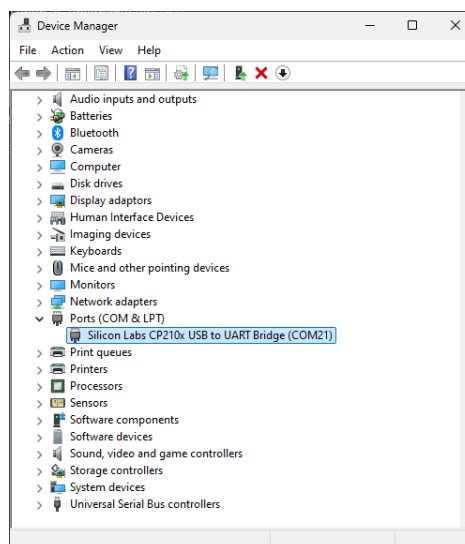


Fig. Portas COM no Gestor de Dispositivos

B.4 CARREGAR UM PROGRAMA PARA O BOT'N ROLL ONE A+

No ambiente de programação Arduino encontra vários programas de exemplo que pode carregar para o robô.

Clica em **"File -> Examples -> 01.Basics -> Blink"** e aparece uma nova janela com o código deste exemplo.

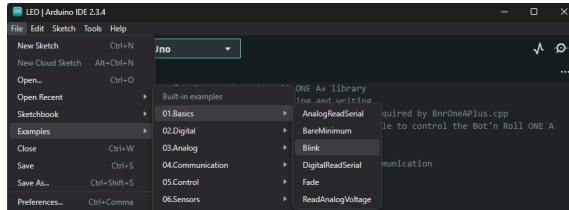


Fig. Carregar um programa de exemplo

Clica em **"File -> Upload"** ou carrega no símbolo com a seta para o lado direito para enviar o programa para o robô. Assim que o *upload* terminar deverás ver o LED amarelo a piscar a cada segundo!

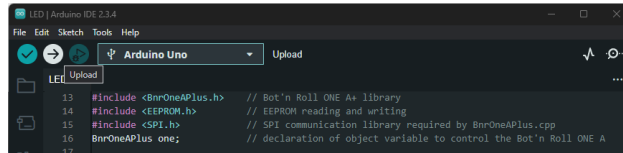


Fig. Enviar o programa para o robô

Clicando em **"File -> Examples -> BnrOneAPIus->..."** encontra todos os programas de exemplo fornecidos pela botnroll.com especificamente para o Bot'n Roll ONE A+.

Em **"File -> Examples -> BnrOneAPIus -> Basic ->..."** estão os programas básicos cuja finalidade é testar todo o *hardware* do robô. Deverás estudar e compreender bem estes pequenos programas!

Em **"File -> Examples -> BnrOneAPIus -> Advanced -> ..."** estão programas mais avançados que só deverás estudar quando perceberes os mais simples.

Em **"File -> Examples -> BnrOneAPIus -> Extra -> ..."** estão os programas relacionados com os componentes extra que expandem o teu Bot'n Roll ONE A+.

Em **"File -> Examples -> BnrOneAPIus -> Fun Challenge -> ..."** estão os programas relacionados com o Fun Challenge.

Em **"File -> Examples -> BnrOneAPIus -> Calibrate -> ..."** estão os programas relacionados com calibração do teu Bot'n Roll ONE A+.

Em **"File -> Examples -> BnrOneAPIus -> CrazyRace -> ..."** estão os programas relacionados com a prova CrazyRace.

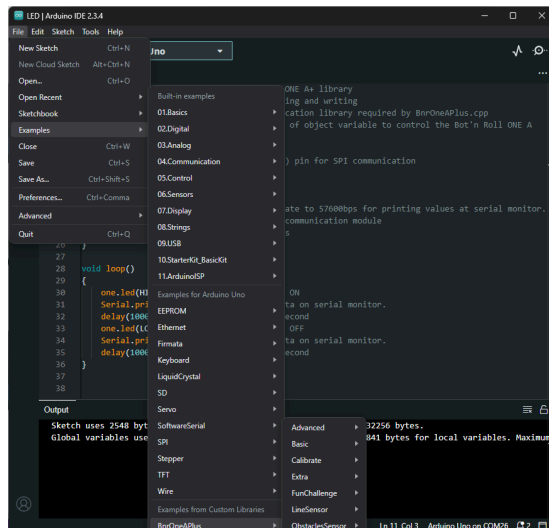


Fig. 7: Programas da biblioteca BnrOneAPIus

Em **"File -> Examples -> BnrOneAPIus -> Obstacles Sensor -> ..."** são os programas relacionados com os sensores de Obstáculos.

Em **"File -> Examples -> BnrOneAPIus -> Line Sensor -> ..."** são os programas relacionados com o sensor de Seguimento de Linha.