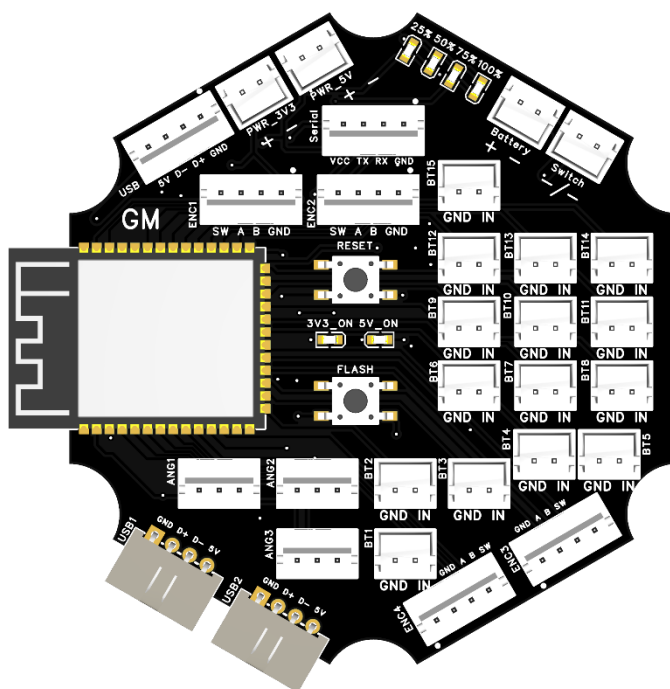




# GhiMa Generic Wheel Board

Manual de uso



# Sumário

Lista de figuras .....	3
Lista de tabelas .....	4
Sobre .....	5
Conexão ao PC via USB .....	7
Preparo do cabo .....	7
Conectando ao PC .....	8
Conexão ao <i>Software</i> .....	9
Funções do <i>Software</i> .....	10
Preparo da placa .....	12
VCC no conector Serial .....	12
Conexão dos <i>inputs</i> .....	12
Botão .....	12
Analógico .....	14
Encoders .....	16
Conexão dos <i>outputs</i> .....	18
LEDs .....	18
VoCore .....	20
Conexão com o SimHub .....	22
Configuração no <i>Software</i> .....	22
Adição no SimHub .....	23
Códigos personalizados .....	25
<i>Troubleshooting</i> .....	26
USB não reconhecido .....	26
Suporte .....	27

# Lista de figuras

Figura 1. Visão geral da placa.....	6
Figura 2. <i>Pinout</i> padrão de um conector USB A macho .....	7
Figura 3. Conector GX12 exemplo .....	7
Figura 4. Reconhecimento do <i>joystick</i> no Windows .....	8
Figura 5. Visão do <i>Software</i> em primeiro acesso.....	9
Figura 6. Visão do <i>Software</i> com a placa conectada.....	10
Figura 7. Vista traseira da placa com destaque na seleção de tensão.....	12
Figura 8. Exemplo 1 para botões.....	13
Figura 9. Exemplo 2 para botões.....	13
Figura 10. Configuração no <i>Software</i> para exemplo 1 e 2 dos botões .....	14
Figura 11. Exemplo 1 para analógicos .....	14
Figura 12. Configuração no <i>Software</i> para exemplo 1 do analógico .....	15
Figura 13. Exemplo 2 para analógicos .....	15
Figura 14. Configuração no <i>Software</i> para exemplo 2 do analógico .....	16
Figura 15. Exemplo 1 para <i>Encoders</i> .....	16
Figura 16. Configuração no <i>Software</i> para exemplo 1 dos <i>Encoders</i> .....	17
Figura 17. Exemplo 2 para <i>Encoders</i> .....	17
Figura 18. Configuração no <i>Software</i> para exemplo 2 dos <i>Encoders</i> .....	18
Figura 19. Exemplo 1 para LEDs.....	18
Figura 20. Configuração no <i>Software</i> para exemplo 1 dos LEDs .....	19
Figura 21. Exemplo 2 para LEDs.....	19
Figura 22. Configuração no <i>Software</i> para exemplo 2 dos LEDs .....	20
Figura 23 Exemplo para <i>VoCore</i> .....	20
Figura 24. Configuração no <i>Software</i> para exemplo da <i>VoCore</i> .....	21
Figura 25. Configuração no <i>Software</i> para conexão ao <i>SimHub</i> exemplo 1 .....	22
Figura 26. Configuração no <i>Software</i> para conexão ao <i>SimHub</i> exemplo 2 .....	23
Figura 27. Interface <i>SimHub</i> para adição de novo Device primeiro passo .....	23
Figura 28. Interface <i>SimHub</i> para adição de novo Device segundo passo .....	24
Figura 29. Placa adicionado com sucesso ao <i>SimHub</i> .....	24
Figura 30. Mensagem de erro apresentado pelo <i>Windows</i> .....	26

## Lista de tabelas

Tabela 1. Relação dos pinos entre ESP32-S3 e GGWB.....	25
--	----

# Sobre

A GGWB (Ghima Generic Wheel Board) é uma placa baseada em ESP32-S3 capaz de fornecer a solução definitiva para criação de volantes DIY. Com um formato favorável para volantes com furação 70mm, a mesma traz em sua composição as seguintes características:

- 32 entradas configuráveis
- Conexão ao SimHub via *Devices* para LEDs e *Dashboards*
- USB Joystick HID
- Bluetooth Joystick HID
- Saída 5V via conector
- Saída 3.3V via conector
- Funcionamento por bateria com gerenciamento completo
- Chip USB-Serial dedicado
- 2 entradas USBs via USB HUB interno para conexão de telas e periféricos

As 32 entradas podem ser configuradas das seguintes formas:

- Entrada digital (Botão)
- Entrada analógica (Eixos)
- Entrada Encoder (DT, CLK e SW)
- Saída para LED (Telemetria SimHub)

A estrutura de conectores da placa está distribuída da seguinte forma:

- 1 saída USB (5V, D-, D+ e GND)
- 1 Saída 3.3V (3.3V e GND)
- 1 Saída 5V (5V e GND)
- 1 entrada de bateria (VCC e GND)
- 1 entrada de Switch para bateria (IN e GND)
- 15 entradas de dois pinos (IN e GND)
- 3 entradas de 3 pinos (VCC, IN e GND)
- 4 entradas de 4 pinos (IN, IN, IN e GND)
- 1 entrada de 4 pinos com VCC alterável entre 5V e 3.3V (VCC, IN, IN e GND)
- 2 entradas USBs para periféricos (5V, D-, D+ e GND)

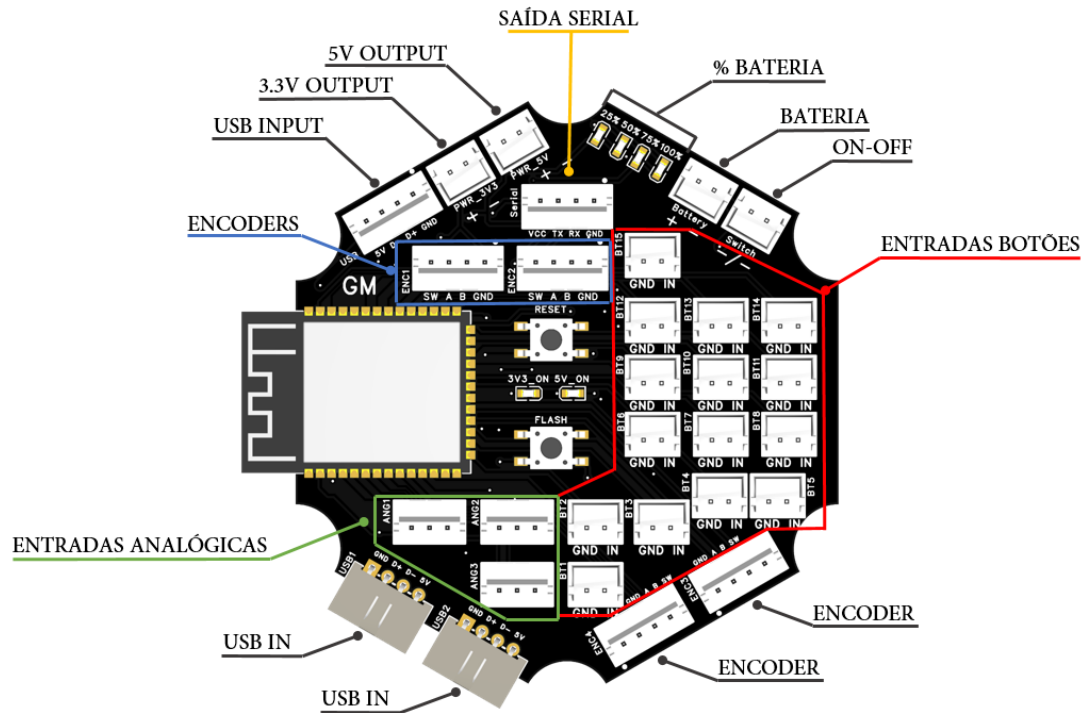


Figura 1. Visão geral da placa

**ATENÇÃO!!!** O conector de 4 pinos nomeado “Serial” está com o VCC ligado em padrão em 5V, cuidado ao usá-lo com as entradas da placa. A troca por 3,3V pode ser vista na seção referente a isto.

## Conexão ao PC via USB

A conexão deve ser feita pelo conector “USB” da placa, respeitando os pinos corretamente. Mesmo a placa tendo proteção contra polaridade invertida, é extremamente recomendado checar diversas vezes que tudo está OK, qualquer deslize pode causar danos irreversíveis a placa. O mais indicado a se fazer é passar um teste de continuidade da ponta do cabo USB até os conectores, pois há diversos relatos de cabos USBs com as cores erradas, e confiar nas cores pode ser um equívoco.

Para facilitar, esta é a pinagem de um conector USB tipo A:



Figura 2. Pinout padrão de um conector USB A macho

## Preparo do cabo

Para a conexão USB, usa-se o cabo de 4 pinos das cores preto, verde, branco e vermelho incluídos no pacote, **cuidado** para não usar os cabos invertidos devidamente marcados.

Para facilitar a conexão entre o volante e computador, recomenda-se o uso de conectores GX12 ou GX16 como exemplificado a seguir:



Figura 3. Conector GX12 exemplo

A ordem usada na criação do conector pode ser definida pelo usuário, porém recomenda-se seguir um padrão fortemente adotado no mundo do DIY, que pode ser visto na imagem a seguir (Créditos a *Pokorny Engineering* pela imagem):



## Conectando ao PC

Ao conectar no PC, as duas luzes centrais da placa devem acender, já as luzes de bateria podem apresentar comportamentos diferentes caso não haja uma bateria conectada, portanto não se preocupe com elas. O Windows deve emitir o som padrão de dispositivo conectado, e dois dispositivos devem aparecer, ambos visíveis no *Gerenciador de dispositivos*, eles são: Uma porta serial COM e um Joystick HID. O Joystick deve aparecer também em *Game controllers* sob o nome de *GhiMa Generic Wheel Board*.

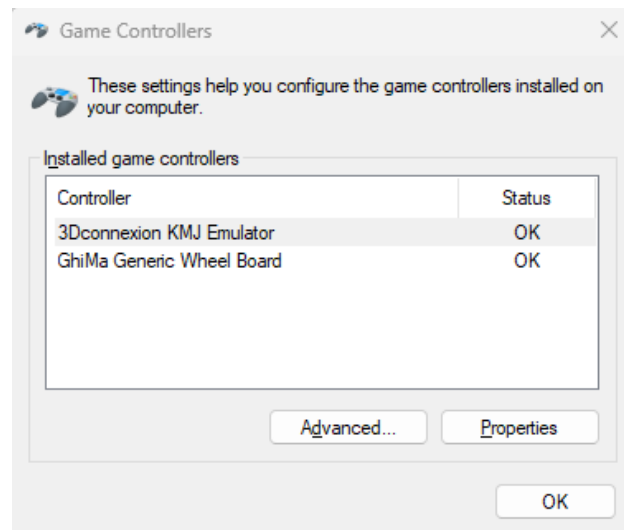


Figura 4. Reconhecimento do joystick no Windows



## Conexão ao *Software*

O software para configuração da placa pode ser encontrado na página do *GitHub* da placa ([github.com/julioghi/Generic-Wheel-Board/releases/](https://github.com/julioghi/Generic-Wheel-Board/releases/)) no arquivo *GGWB\_SW.zip*. Após descompactá-lo, o mesmo pode ser aberto com um duplo clique no arquivo *GGWB.exe*.

A visão do *Software* com a placa conectada ao computador é a seguinte:

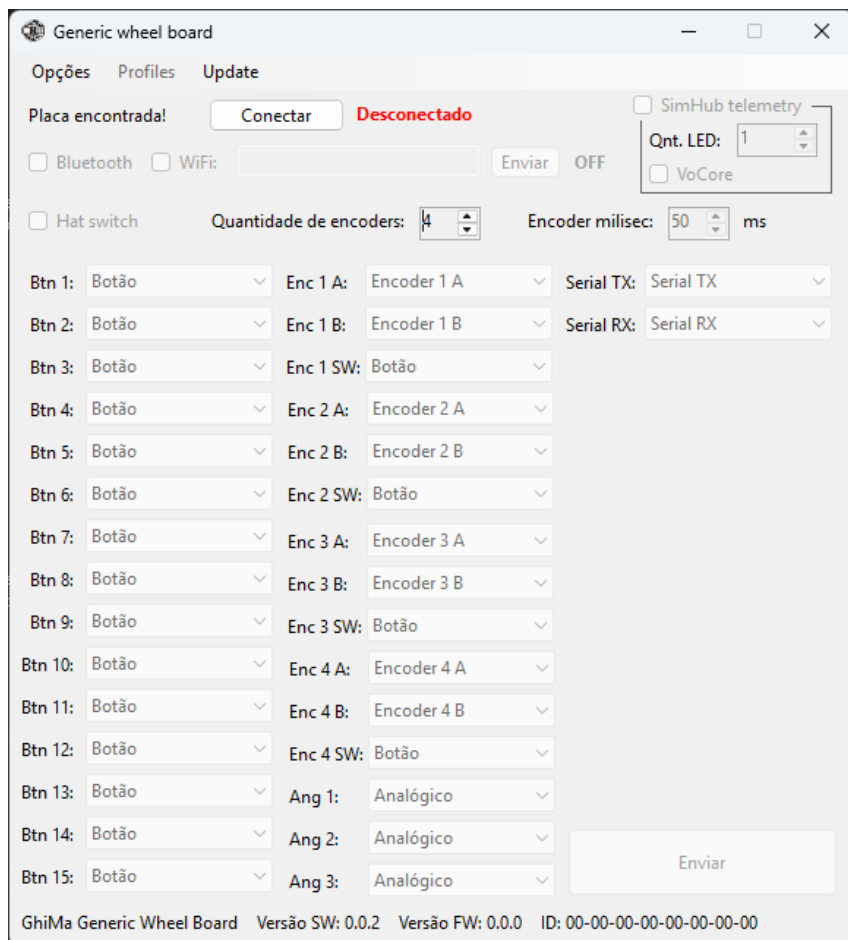


Figura 5. Visão do *Software* em primeiro acesso

Caso o *Software* esteja indicando que a placa não está sendo encontrada, certifique-se que a mesma esteja ligada no USB e esteja sendo detectada pelo Windows, conforme citado anteriormente. Se o SimHub estiver aberto, este pode estar causando conflito, recomenda-se fechá-lo. Caso o problema persista, entre em contato com o suporte.

Para fazer alterações na placa, clique em conectar para estabilizar a comunicação com a mesma. Caso tudo esteja OK, o status mudará para “CONECTADO” e as informações na área inferior serão atualizadas, como no exemplo a seguir:

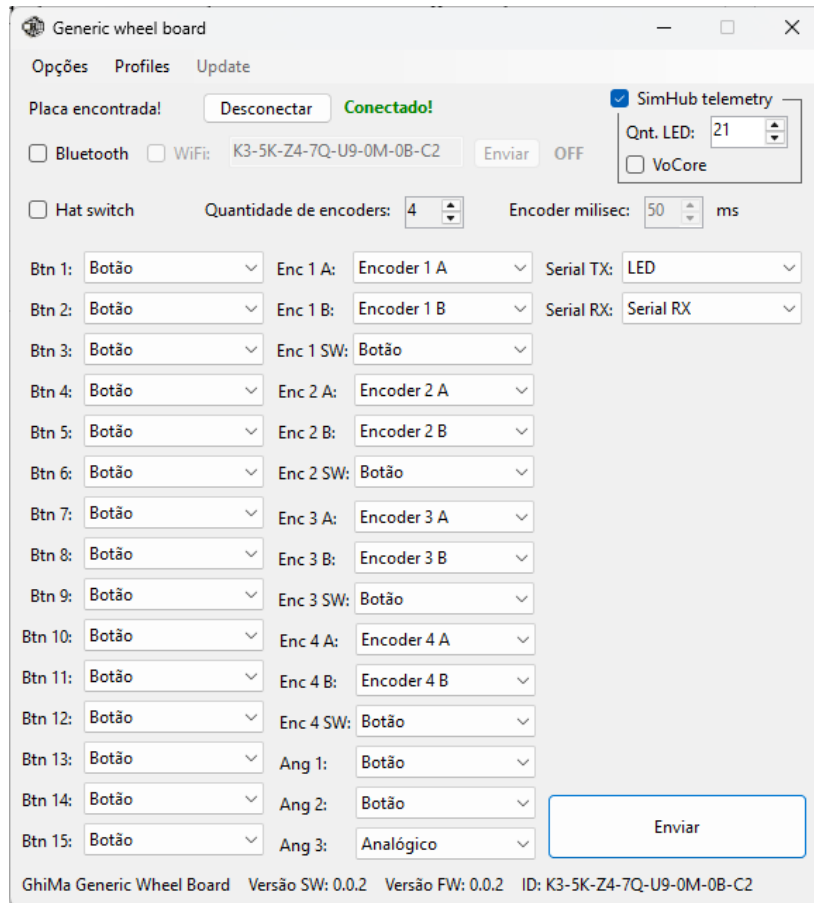


Figura 6. Visão do *Software* com a placa conectada

## Funções do *Software*

A seguir são listadas as opções disponíveis no *Software* para personalização da placa. Atente-se que algumas opções podem limitar outras.

Utilizar a placa em modo *Bluetooth* não permite comunicação com o *SimHub*, apenas com a implementação da conexão via *WiFi* trará esta possibilidade.

As funções no topo do *Software* são as seguintes:

- **Opções:**
  - **Porta COM:** Permite selecionar manualmente a porta COM a qual o *Software* irá se conectar. O uso será raro pois a necessidade disto indica algum erro de conexão da placa.
  - **Manual da placa:** Esta opção redireciona para o link do PDF deste manual.
  - **Software update:** Esta opção checará via *web* se tem um *update* para o software disponível
- **Profiles:**
  - **Open:** Permite abrir o arquivo “.xml” que contenha configurações previamente criadas pelo usuário.
  - **Save:** Permite ao usuário salvar as configurações feitas em um arquivo “.xml”, para *Backup* e futuro uso.
- **Update:** Esta opção abre uma tela para atualização da placa com upgrades de *firmware* disponíveis na página oficial do *GitHub*. Esta opção só estará disponível caso a placa esteja reconhecida e desconectada no *Software*.

As funções oferecidas pelo *Software* para modificação da placa são as seguintes:

- **Bluetooth:** Com esta opção ativada a placa irá emitir conexão via *bluetooth*, aparecendo como um *Joystick* HID quando estabelecida a conexão via *Bluetooth* no computador. Esta opção desativava o *Joystick* HID via USB.
- **SimHub telemetry:** Esta opção ativa e desativa o reconhecimento da placa no *SimHub* como um *Device*. Ao ativa-la é obrigatório indicar em qual saída estará conectado a fita de LED.
- **Qnt. LED:** Nesta caixa será indicada a quantidade de LEDs conectados a placa, esta informação será passada ao *SimHub* na primeira conexão.
- **VoCore:** Esta opção informa ou não ao *SimHub* se o *Device* tem uma tela incluída. Com ela desativada o *Device* será reconhecido apenas para controle de LEDs.
- **Hat Switch:** Esta opção ativa ou desativa a existência de um *Hat Switch* no *Joystick*, caso ativada, é obrigatório indicar em quais entradas estão localizadas as 4 funções do *Hat* (cima, baixo, esquerda e direita), a placa fará automaticamente a junção dos movimentos caso solicitado.
- **Quantidade de encoders:** Nesta caixa será indicado a quantidade de encoders que se deseja conectar a placa, conforme este valor aumenta ou diminui as opções são alteradas nos pinos. É obrigatório informar em quais entradas estarão o pino A e B de cada encoder, caso não informado o *Software* indicará erro ao tentar enviar.
- **Btn 1 ao Serial RX:** Nestas caixas serão selecionadas as funções de cada pino da placa, sendo as opções descritas a seguir:
  - **Botão:** Entrada digital de valor único, normalmente utilizada em *Push Buttons*.
  - **Analógico:** Entrada analógica com 12 *bits* de precisão (0~4095), normalmente utilizada com potenciômetros e sensores *Hall*. Esta função não está disponível em todas as entradas devido a limitação de Hardware do ESP32.
  - **LED:** Indica que esta conexão será utilizada como saída de LED de telemetria para o *SimHub*. Esta função só aparece caso *SimHub telemetry* esteja ativado.
  - **Encoder X A e Encoder X B:** Esta função indica que a entrada será utilizada em um dos dois pinos do Encoder, onde X representa o número do encoder.
- **Enviar:** Este botão envia as alterações para a placa e, caso não haja nenhum conflito, retorna com sucesso. Após o envio a placa se reiniciará sozinha para que as alterações surjam efeito, portanto, é esperado que o som de dispositivos do *Windows* seja executado. Caso haja algum conflito nas opções selecionadas, o software acusará o erro.

# Preparo da placa

## VCC no conector Serial

O VCC do Conector Serial está ligado aos 5V da placa por padrão, facilitando a conexão de fitas de LEDs, porém, caso seja necessário, há a possibilidade de alteração para 3,3V trocando a posição de um resistor na parte traseira da placa, indicado na imagem a seguir:

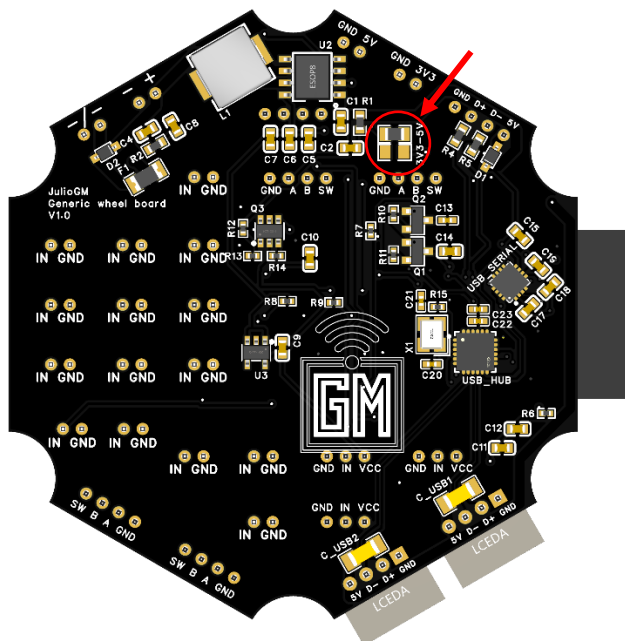


Figura 7. Vista traseira da placa com destaque na seleção de tensão

Esta alteração possui uma dificuldade de solda “média”, portanto só é recomendada ser feita caso possua uma boa experiência com ferro de solda e o equipamento apropriado.

## Conexão dos *inputs*

### Botão

A conexão do botão é um simples fechamento entre a entrada (IN) e o negativo (GND). As conexões já possuem *Pull-UP* internos, portanto não há necessidade de componentes extras. O fechamento entre IN e VCC (3,3V) não trará reação a placa. A placa NÃO aceita 5V nas entradas!

Um exemplo de conexão pode ser visto a seguir:

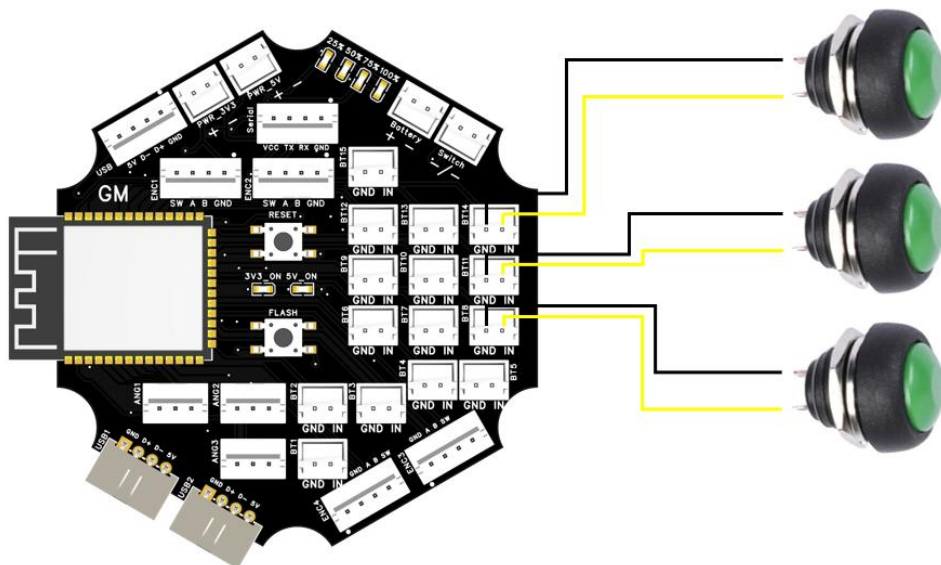


Figura 8. Exemplo 1 para botões

O GND dos conectores é o mesmo para a placa toda, portanto, para botões próximos, é recomendado que o GND seja compartilhado entre eles e apenas o fio de dados seja levado até a placa, como no exemplo a seguir:

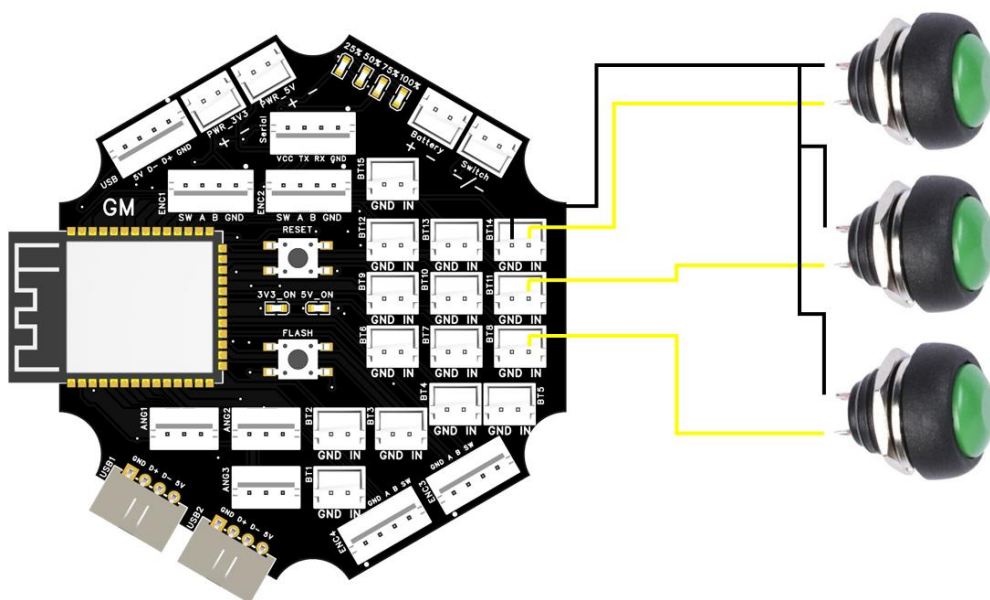


Figura 9. Exemplo 2 para botões

Todas as entradas aceitam botões, portanto pode ser utilizado também as entradas com marcação “A”, “B”, “SW”, “TX” e “RX”.

A configuração do software para a conexão mostrada anteriormente pode ser vista a seguir:

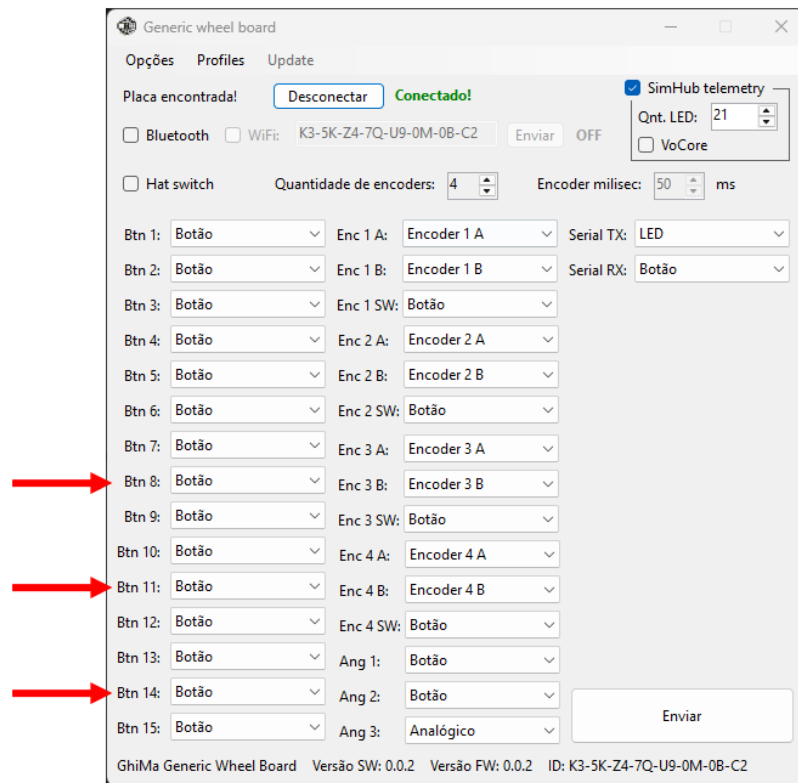


Figura 10. Configuração no *Software* para exemplo 1 e 2 dos botões

## Analogico

As conexões analógicas costumam usar 3 conexões da placa, o VCC, GND e IN. O ESP suporta apenas o range de 0~3.3V, portanto NÃO utilize sensores com range de 5V. A seguir pode ser visto um exemplo usando potenciômetro e *Hall*:

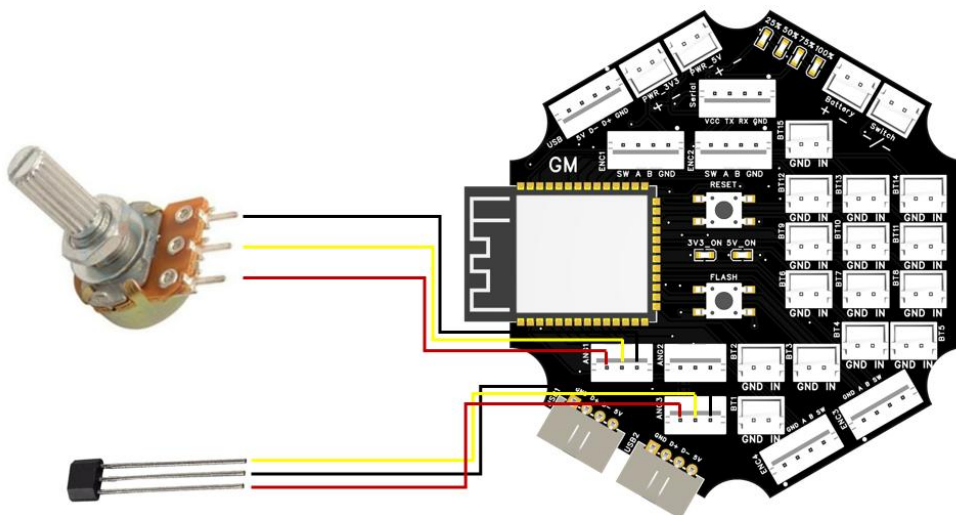


Figura 11. Exemplo 1 para analógicos

A imagem do sensor *Hall* é meramente ilustrativa, importante checar a pinagem do seu sensor antes de conectar.

A configuração do software para a conexão mostrada anteriormente pode ser vista a seguir:





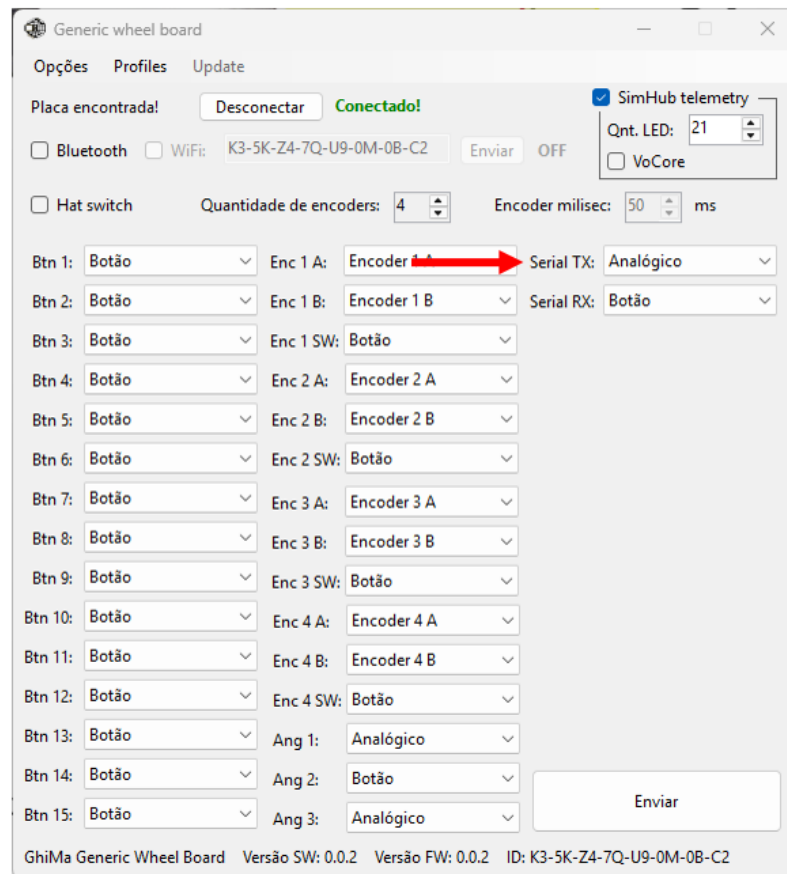


Figura 14. Configuração no *Software* para exemplo 2 do analógico

## Encoders

Encoders usam no mínimo duas conexões, podem ser chamadas de “A” e “B”, ou “DT” e “CLK”, uma terceira conexão pode ser utilizada para obter a *click* que a maioria dos Encoders oferecem. A seguir pode ser visto um exemplo utilizando um dos conectores favoráveis para encoders:

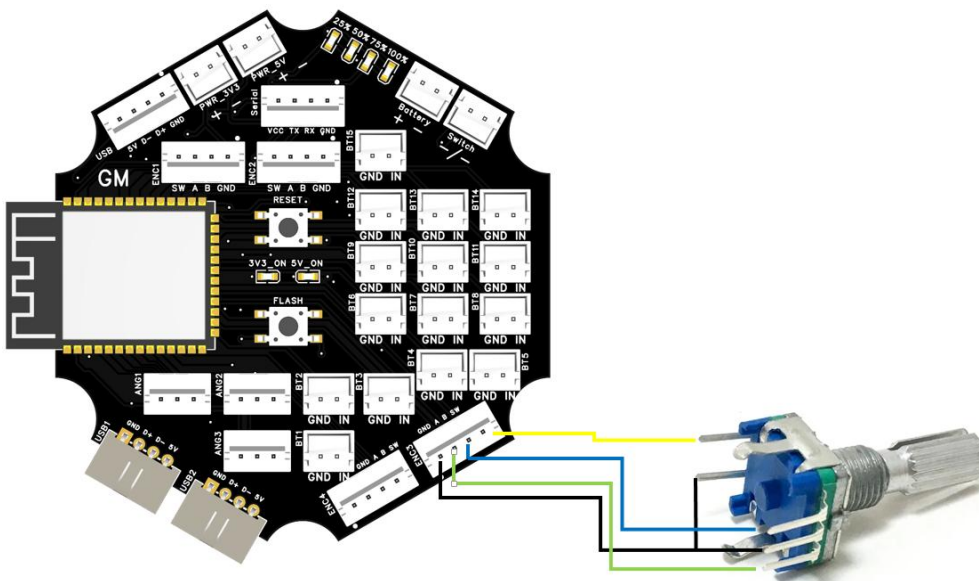


Figura 15. Exemplo 1 para *Encoders*

A cor azul é usada devido ao fundo branco da imagem, porém os cabos do kit vão com a cor branca ao invés do azul.



A configuração do software para a conexão mostrada anteriormente pode ser vista a seguir:

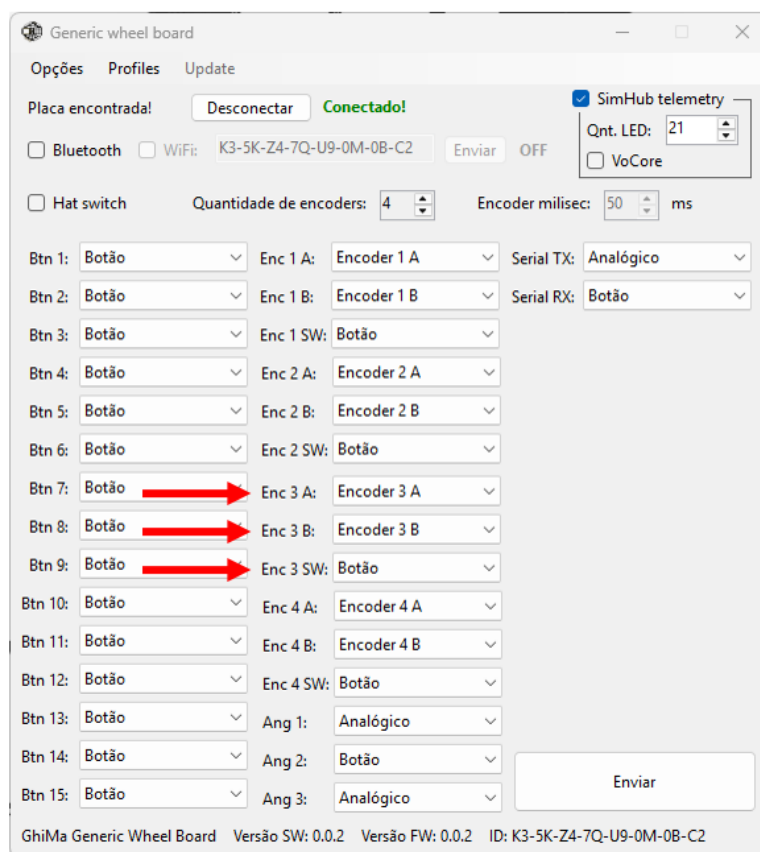


Figura 16. Configuração no *Software* para exemplo 1 dos *Encoders*

Como citado anteriormente, qualquer entrada pode ser utilizada com encoders, portanto, a seguir há um exemplo usando entradas aleatórias:

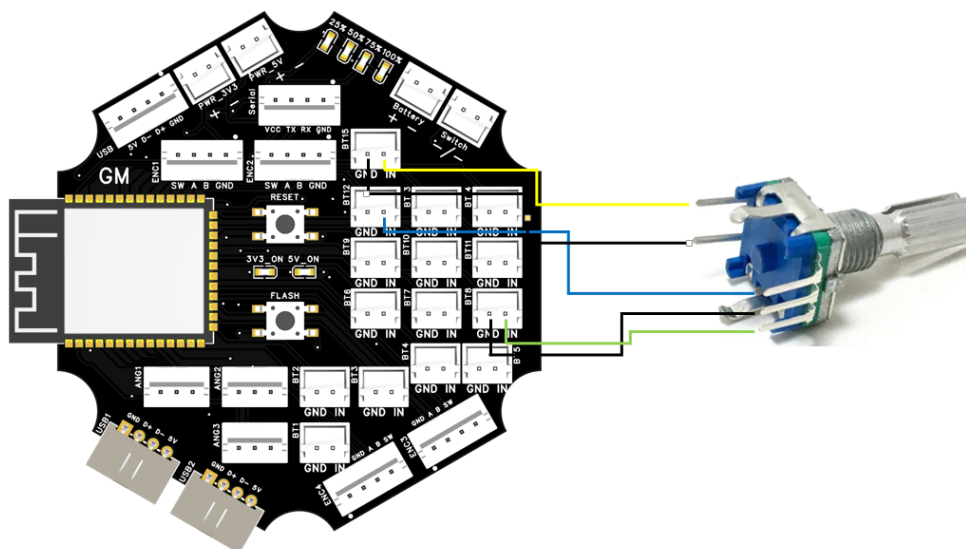


Figura 17. Exemplo 2 para *Encoders*

A configuração do software para a conexão mostrada anteriormente pode ser vista a seguir:

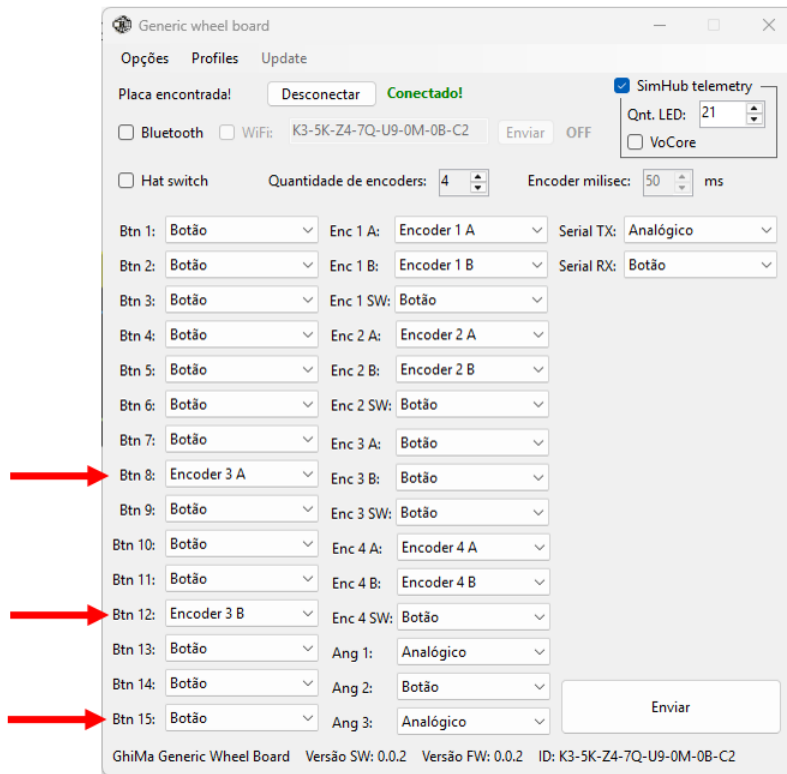


Figura 18. Configuração no Software para exemplo 2 dos Encoders

## Conexão dos outputs

### LEDs

A fita de LED endereçável (WS2812) pode ser conectada em qualquer conector da placa, porém recomenda-se utilizar o conector Serial por facilitar o acesso ao 5V e GND. Esta conexão pode ser vista a seguir:

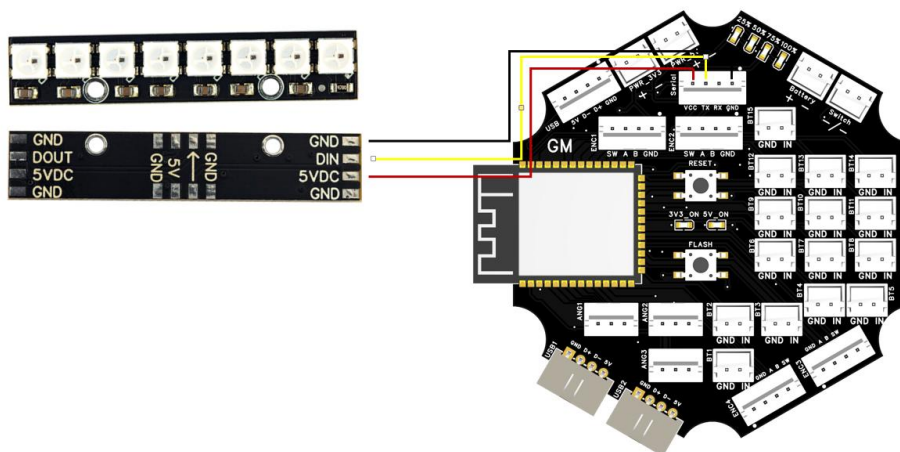


Figura 19. Exemplo 1 para LEDs

A configuração do software para a conexão mostrada anteriormente pode ser vista a seguir:





A cor azul é usada devido ao fundo branco da imagem, porém os cabos do kit vão com a cor branca ao invés do azul.

A configuração do software para a conexão mostrada anteriormente pode ser vista a seguir:

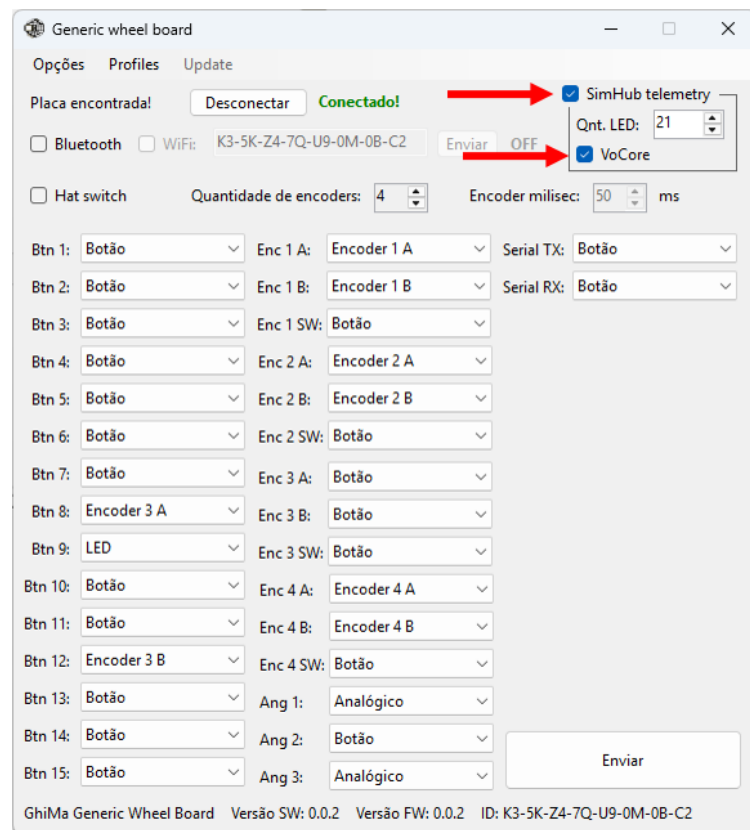


Figura 24. Configuração no *Software* para exemplo da *VoCore*

# Conexão com o SimHub

## Configuração no Software

Para que a placa seja reconhecida no SimHub, a mesma deve estar configurada para reagir como um *Device* ao ser requisitada. A placa pode funcionar como dois *Devices* diferentes, um apenas para LEDs e outro para LEDs e VoCore.

Caso deseje apenas usa-la com LEDs, basta marcar a opção de "SimHub telemetry", definir a quantidade de LEDs e escolher em qual pino o LED será conectado, como no exemplo a seguir:

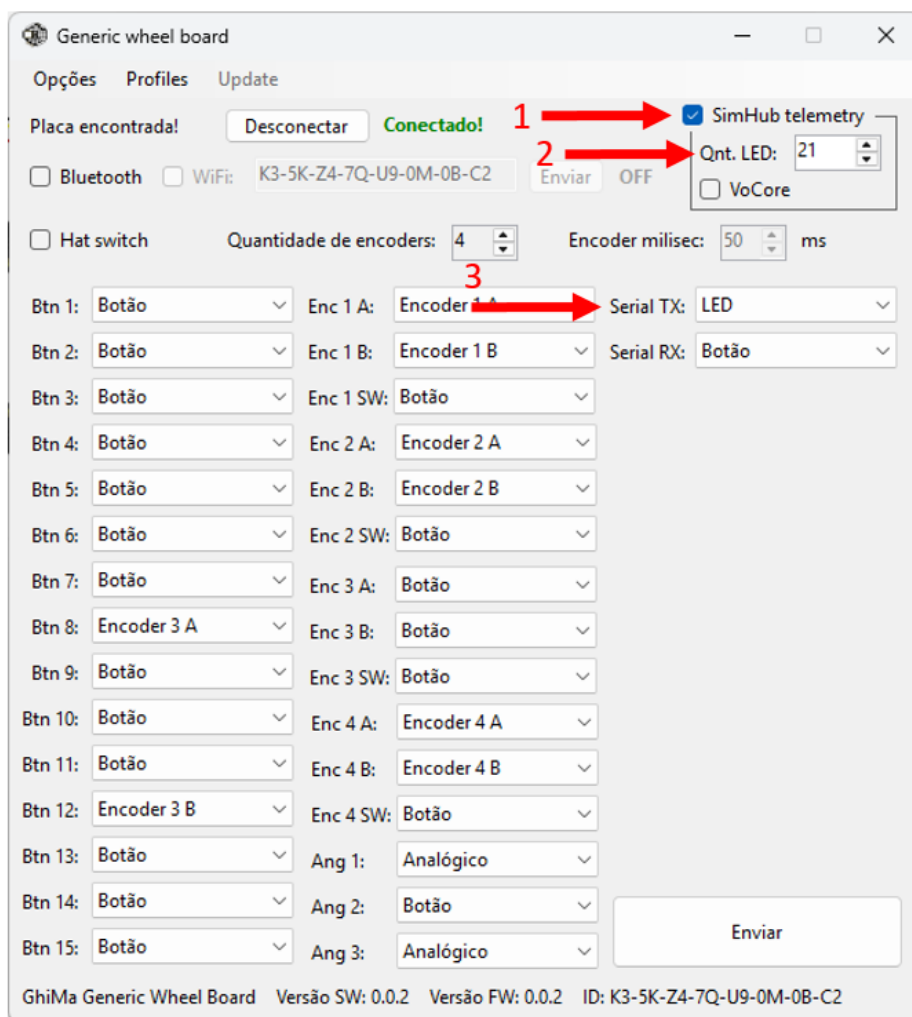


Figura 25. Configuração no Software para conexão ao SimHub exemplo 1

Caso deseje utilizar os LEDs e a Vocore, basta seguir os mesmos passos do LEDs e ativar a opção "VoCore", como no exemplo a seguir:

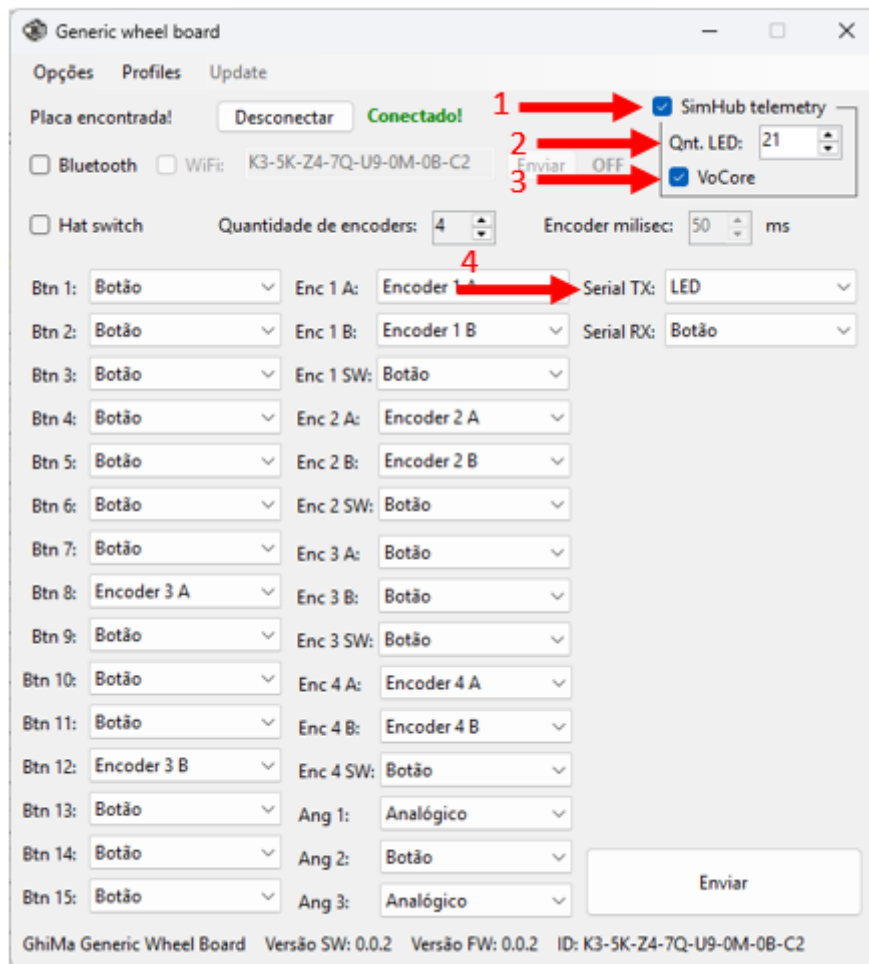


Figura 26. Configuração no *Software* para conexão ao *SimHub* exemplo 2

## Adição no SimHub

Para adicionar a placa ao SimHub, basta ir a aba “**Devices**”, clicar no “+” e em “add new device”, uma nova tela irá aparecer e a placa deve estar visível em “Devices found on your computer”, basta dar um duplo clique nela para que a mesma seja adicionada. A seguir pode ser visto um passo a passo:

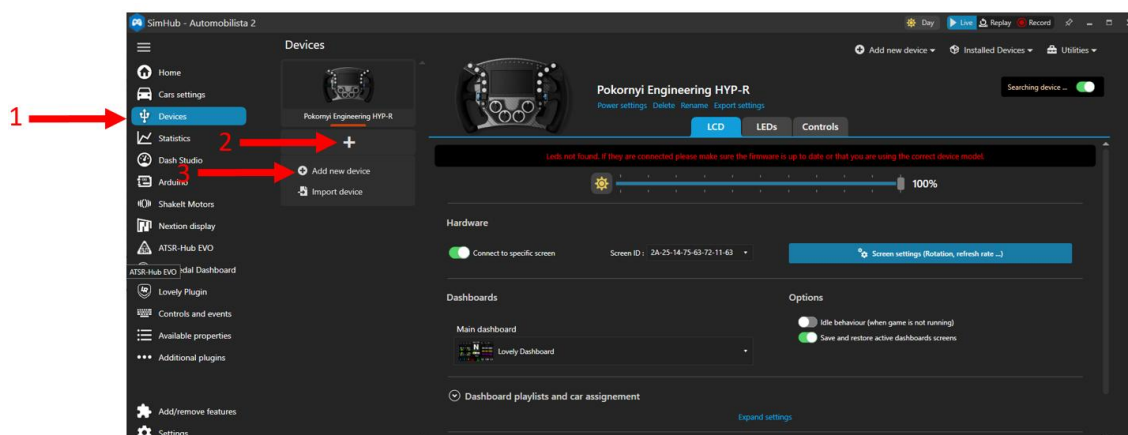


Figura 27. Interface *SimHub* para adição de novo Device primeiro passo



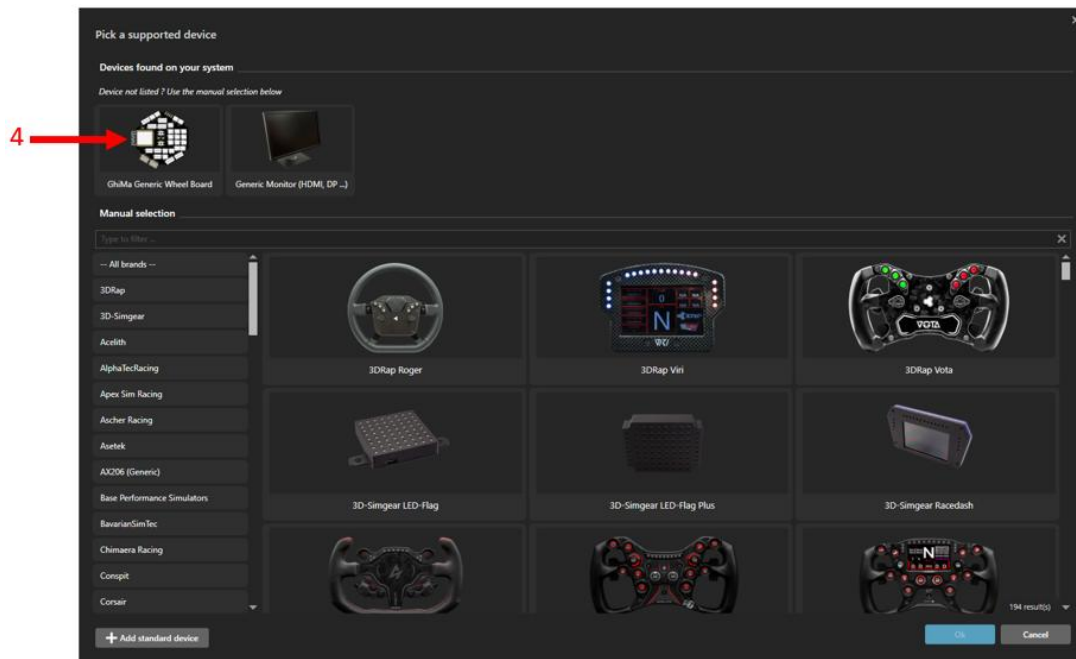


Figura 28. Interface *SimHub* para adição de novo Device segundo passo

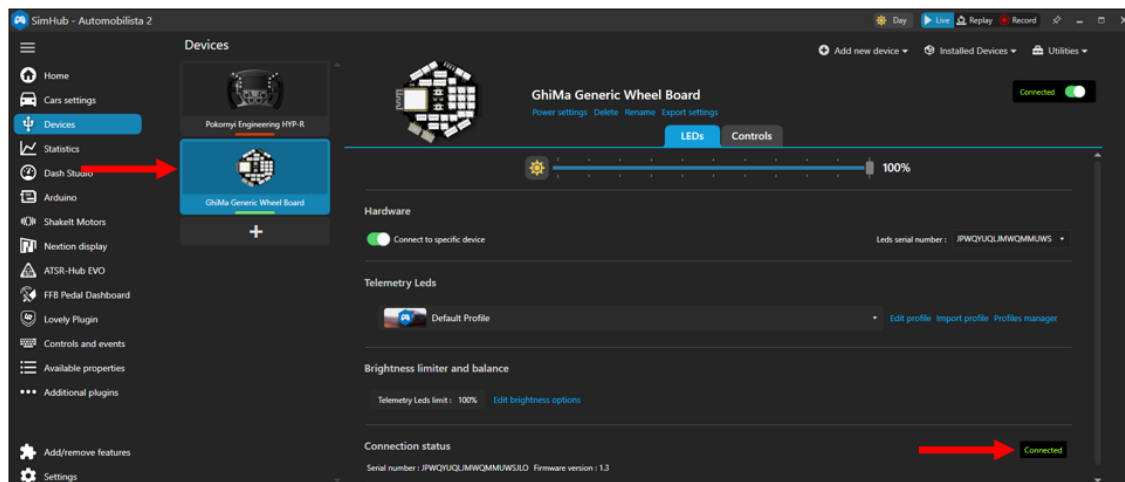


Figura 29. Placa adicionado com sucesso ao *SimHub*

**ATENÇÃO!** Qualquer alteração no *Software* que envolva a área do SimHub requer a exclusão do *Device* e a adição do mesmo novamente, uma vez que é gerado um novo ID de *Device* do SimHub.



## Códigos personalizados

A placa GGWB é basicamente um módulo baseado em ESP32-S3, portanto, sintá-se livre para criar e fazer upload de qualquer código para uso da placa, A GhiMa **não** oferece suporte para tais modificações, portanto erros e incompatibilidades precisam ser estudada e *debugadas* pelo criador.

O modelo do ESP32-S3 está listado a seguir:

- ESP32-S3
- 16MB Flash
- 2MB PS-RAM

A listagem dos pinos pode ser vista na tabela a seguir:

Tabela 1. Relação dos pinos entre ESP32-S3 e GGWB

Pino ESP32-S3	Pino GGWB
IO0	BT5
IO1	ENC1 SW
IO2	ENC1 B
IO3	BT2
IO4	ANG1
IO5	ANG3
IO6	ANG2
IO7	BT1
IO8	ENC4 SW
IO9	ENC3 A
IO10	ENC3 B
IO11	ENC3 SW
IO12	BT4
IO13	BT6
IO14	BT7
IO15	ENC4 A
IO16	ENC4 B
IO17	SERIAL TX
IO18	SERIAL RX
IO19	USB D+
IO20	USB D-
IO21	BT8
IO35	BT12
IO36	BT13
IO37	BT14
IO38	BT15
IO39	ENC2 A
IO40	ENC2 B
IO41	ENC2 SW
IO42	ENC1 B
IO45	BT11
IO46	BT3
IO47	BT9
IO48	BT10
TXD0 (PINO 37)	RX do serial dedicado (CH343P)
RXD0 (PINO 36)	TX do serial dedicado (CH343P)

# *Troubleshooting*

## USB não reconhecido

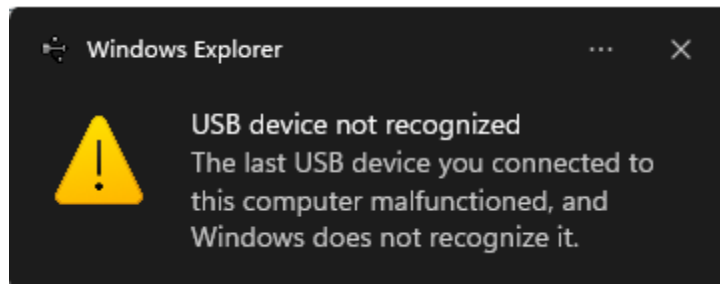


Figura 30. Mensagem de erro apresentado pelo *Windows*

A mensagem “dispositivo USB não reconhecido” do *Windows* pode aparecer devido aos seguintes erros:

- USB D+ ou D- não devidamente conectados
- USB D+ e D- trocados
- Cabo USB muito longo sem os devidos preparos (Discussão sobre em: <https://youtu.be/3HdcKPxUUXg>)

## Suporte

O *software*, *firmware* e manual estão em constante melhorias, qualquer dúvida ou erros encontrados podem ser reportados diretamente ao vendedor ou pelo e-mail a seguir:

[ghimasimracing@gmail.com](mailto:ghimasimracing@gmail.com)

## **Versões de documento**

- **Versão 1.0:** Documento base, incluindo o básico para uso da placa – 11/02/2026