



Guia do instalador: Simova Track

▼ Introdução

▼ O que é o Simova Track:

É um dispositivo criado para coletar e transmitir dados de telemetria de veículos, se comunica via bluetooth (requer um celular para transmitir dados) e pode realizar apontamentos automáticos

Imagem da placa

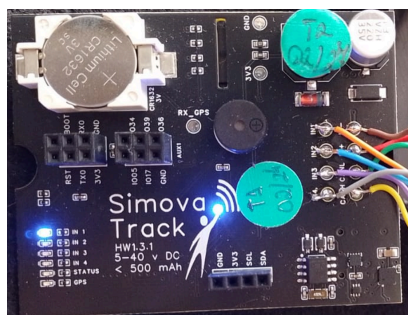


Imagem do case



▼ Principais aplicações:

- Registrar e transmitir dados de geolocalização completos (data e hora, localização, velocidade, direção, qualidade de sinal)
- Registrar e transmitir dados de telemetria (horímetros, odômetros) em veículos móveis (carros, caminhões, equipamentos pesados (linha amarela/verde) e equipamentos estacionários (ex geradores).

▼ Capacidades:

- Leitura de CAN J1939/OBD2 (não é compatível com K-Line ou J1850)
- Suporta "SimovaApps" via conexão BLE.
- 4 entradas digitais, leitura de tensão de bateria.
- Suporte a 7 fontes de Ignição e RPM, para uma configuração rápida

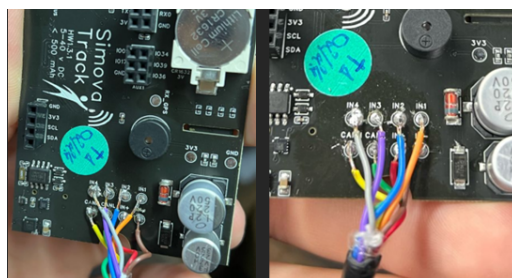
Fontes de Ignição e RPM:
REDE CAN (J1939 ou OBD2)

Entrada digital 1 (12-32v)
Entrada digital 2 (12-32v)
Entrada digital 3 (12-32v)
Entrada digital 4 (12-32v)
Tensão de Bateria (3.30 a 32.00v)
Fictício (1 ou 1234)

▼ Especificações Técnicas (SimovaTrack HW1.3.1)

▼ Cores dos Fios

Cor do fio	Onde solda	O que é
Marrom	IN-	Alimentação negativa
Vermelho	IN+	Alimentação positiva
Verde	CANL	CAN BUS LOW
Amarelo	CANH	CAN BUS HIGH
Laranja	IN1	Entrada 01
Azul	IN2	Entrada 02
Violeta/Roxo	IN3	Entrada 03
Cinza	IN4	Entrada 04



▼ Especificações Técnicas (Tensão, Proteção, Consumo...)

Parâmetro	Valor
Tensão de alimentação continua	7 – 32 VDC
Tensão de alimentação pico	36 VDC MAX
Proteção contra inversão de polaridade	-40v DC MAX
Proteção contra sobretensão	Zener 36v + Polyswitch
Fusível interno	1.5 A PPTC
Consumo médio em transmissão (celular conectado)	< 70 mAh @ 12v
Consumo em espera	< 70 mAh @ 12v
Descarga de bateria em espera (12v)	< 70 mAh @ 12v
Descarga de bateria em espera (24v)	< 35 mAh @ 24v
Descarga completa de bateria 12v 80A	80000mAh / 70mA = 47 dias
Descarga completa de bateria 24v 300A	300000 / 35mA = 357 dias

- A Faixa de tensão recomendada para leitura é de 12 a 32.00v MAX
 - O Uso em tensões entre 9-12v não acende o led, mas pode ser detectada pelo Track (não recomendado)
 - O uso de tensões acima de 32v podem queimar a fonte do Simova Track.
- Leitura de CAN J1939/OBD2 (não é compatível com K-Line ou J1850)
- É recomendado desligar o SimovaTrack (ou chave geral) caso o equipamento instalado fique por longos períodos desligados para evitar drenar a bateria; Note que os valores indicados na tabela acima são para

a descarga completa da bateria, em alguns veiculos (como carros), a partida pode ser bloqueada quando a tensão cai abaixo de ~11.5v

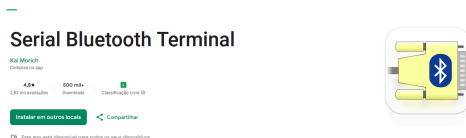
▼ Guias de instalação:

▼ Como instalar e configurar o aplicativo "Serial Terminal Bluetooth"

Aplicativo permite que o usuário envie comandos pré configurados que facilitam o uso.

▼ Baixe e instale o App Serial Bluetooth Terminal

- Serial Bluetooth Terminal:



https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&pcampaignid=web_share

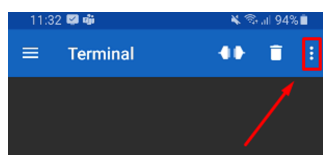
▼ Baixe o arquivo de configurações dos Botões

- Link de download:

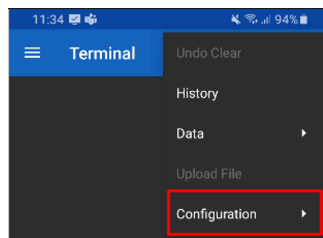
https://drive.google.com/file/d/1ULL46IEHzp965vBIKSMBYlBr_u7PRH83/view?usp=drive_link

▼ Configure o Aplicativo

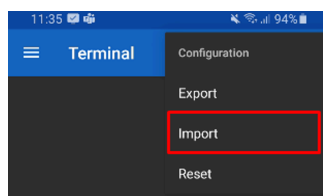
1. Abra o aplicativo e clique nos 3 pontinhos ("More Options").



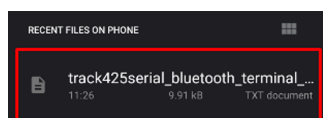
2. Clique em "Configurations"



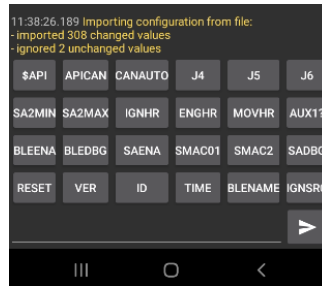
3. Clique em "Import"



4. Selecione o arquivo de configuração dos botões



5. Botões importados



▼ Como restaurar para o padrão de fábrica.

Sempre restaure o aparelho ao realizar uma nova instalação para evitar que dados antigos ou corrompidos afetem o funcionamento enviando o comando AT+RESET.



ATENÇÃO: Esse comando apagará **TODOS** os horímetros, odômetros e configurações do simova track, mas não apagará os arquivos de log de viagem ou de sistema. o aparelho vai reiniciar automaticamente em 5s

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

1. Como resetar o Simova Track.

Enviou: AT+RESET

Recebeu: AT+OK - Restaurando configuracoes de fabrica

▼ Como obter a versão de firmware atual:

Para saber qual versão do Simova Track estamos. (cada versão tem suas respectivas mudanças)

1. Envie o comando AT+VER

Enviou: AT+VER

Recebeu: AT+OK - Firmware ver: TRACK HW1.3 FW4.1.0 Partition label: ota_0 Partition size: 4194304

Entendendo a resposta:

- Versão de firmware: "TRACK HW1.3 FW4.1.0"
 - TRACK = nome do projeto
 - HW = versão do hardware suportado
 - FW = versão de firmware atual
- Partição atual: "OTA0"
 - Há mais de uma partição disponível para armazenar o programa, sendo alternada entre elas quando é feita uma atualização de software. permitindo reverter (rollback) caso necessário.
- Tamanho da partição: "4194304"
 - Valor em bytes da partição atual

▼ Como configurar uma nova ID

O ID tem a finalidade de autenticar o Simova Track com o equipamento do painel do cliente (onde veremos Trajeto e Telemetria).

1. Para obter a ID do dispositivo envie o comando "AT+ID" o comando "AT+ID=xxxx" (somente 4 caracteres)

```
# Consultar
Envio: AT+ID
Resposta: AT+ID=9999
# Configurar (parametro em id (numero))
Envio: AT+ID=5051
Resposta: AT+ID=OK - Changed from "9999" to "5051"
```

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Como configurar a hora (opcional)

O Simova Track atualiza a hora automaticamente a cada boot, quando há um sinal valido de GPS, para operações onde não há sinal de GPS é recomendado configurar a data e hora manualmente, note que o horário usado deve ser sempre GMT0

1. Para configurar a data e hora use `AT+DATETIME=`
 - a. Exemplo: Agora são 10:34h em São Paulo (GMT-3), devemos "somar +3" ao horário e configurar como
`AT+DATETIME=12/5/2024 13:34:00`
 - b. Exemplo: Agora são 12:10h em Mato Grosso do Sul (GMT-4), devemos "somar +4" ao horário e configurar como `AT+DATETIME=12/5/2024 16:10:00`

```
# Consultar
Enviou: AT+DATETIME
Recebeu: AT+DATETIME=12/5/2024 14:28:28

# Denifinir (formato: YYYYMMDDHHMMSS ou YYYY-MM-DD HH:MM:SS)
Enviou: AT+DATETIME=20240512143130
Recebeu: AT+OK: Set internal clock datetime to 2024-05-12 14:31:30
```

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Como configurar o nome do BLE

O nome do dispositivo BLE sempre será composto de `SERIAL_NOME`. Por padrão, o nome `serial_SMVTRACK`

1. Para consultar ou configurar, envie o comando AT+BLENAME
 - a. O limite de caracteres para o nome é 20
 - b. Utilize apenas caracteres entre a-z, A-Z e 0-9.
 - c. Não utilize caracteres especiais como `<>:\\"|?*` ou acentos `ãôâôí`

```
# Consultar
Envio: AT+BLENAME
Resposta: AT+BLENAME=100252_SMVTRACK

# Configurar
ENVIO: AT+BLENAME=01234567890123456789
Resposta: AT+BLENAME=OK - Changed from "100252_SMVTRACK" to "100252_01234567890123456789"
9" WILL RESTART IN 10s!!!
```

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Utilizando o Simova Angle com SimovaTrack

O Simova Angle é um sensor utilizado para calcular a diferença de ângulo entre ele o Track, a diferença de ângulo pode posteriormente ser utilizada através da API para realizar automações ou uso em programas de usuário.

▼ Preparando o Simova Angle

▼ Configuração dos Slots de transmissão

Para configuração dos Slots utilizamos o aplicativo BeaconXPro é utilizado para configurar o Simova Angle

(Link do App: [BeaconX Pro – Apps no Google Play](#).)



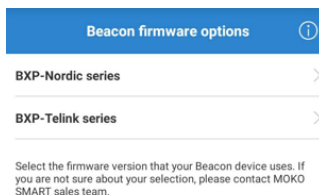
1. Ligue o Simova Angle encostando um ímã de neodímio no **símbolo vermelho**



2. Ligue o Simova Angle encostando um ímã de neodímio no símbolo vermelho

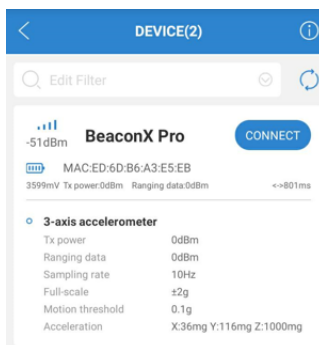


3. Instale e abra o App Clique em **"BXP-Nordic-series"**



4. Na tela ("Devices") procure o sensor a ser configurado

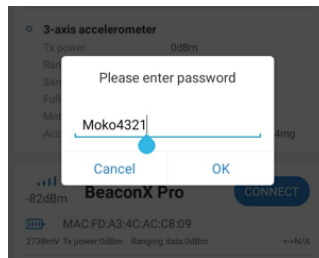
*Veja na Etiqueta do Simova Angle o Mac Address ou atrás do beacon estará escrito o Mac Address.
(Procure o no app MAC xx:xx:xx:xx:xx conecte no aparelho)*



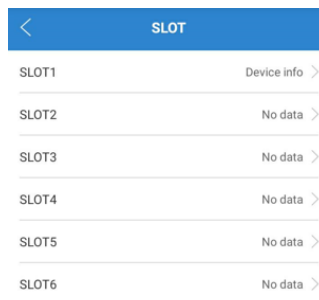
DICA: encoste o beacon na traseira do celular, a lista será ordenada via força do sinal, ou seja, o beacon mais próximo será o primeiro da lista.

▼ Configurações do SLOT1 e SLOT2

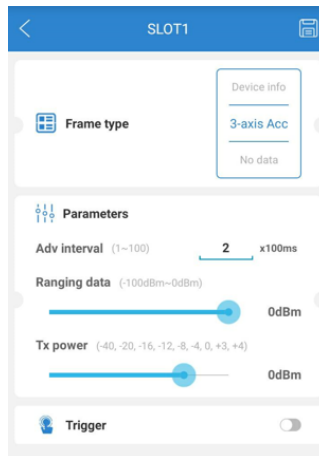
1. No menu, selecione "**Conectar**" e insira a senha "**Moko4321**".



2. Escolha o "SLOT1".



3. Configure os parâmetros conforme a tabela abaixo

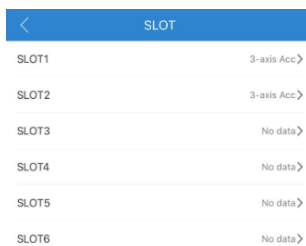


Parâmetro	Valor	O que é
Frame Type	3-axis Acc	3 eixos do Acelerômetro (x,y,z)
Adv Interval	2	Velocidade do Advertising/Dados Transmitidos
Ranging data	0dBm	Potência do sinal há 1m em dBm
Tx Power	0dBm	Potência do sinal transmitido

Repita as mesmas configurações para o SLOT2. Todos os outros slots devem estar selecionados como "No data".

▼ Configuração do sensor interno

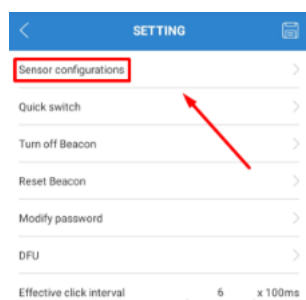
1. Após salvar, volte para tela de Slots



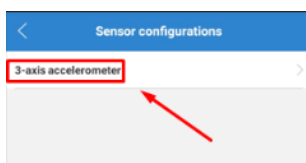
2. Clique em **"SETTING"**.



3. Clique em **"Sensor Configurations"**



4. Selecione **"3-axis accelerometer"**.



5. Configure os parâmetros conforme a imagem

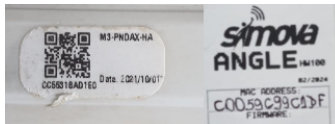


Parâmetro	Valor	O que é
Full Scale	+16g	Valor máximo e mínimo de aceleração da gravidade (afeta a resolução)
Sampling rate	100Hz	Frequência de atualização do acelerômetro interno (não reflete na velocidade de transmissão)

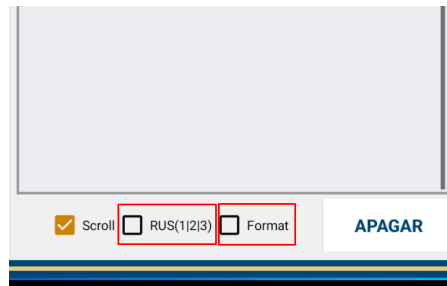
▼ Preparando o SimovaTrack

▼ Verificando MAC Address Simova Angle

Localize o MAC-ADDRESS na etiqueta do Código QR



Acesse o aplicativo da Simova e entre no terminal "MARTE CMD". Desligue os itens RUS(1|2|3) e Format antes de enviar os comandos (muito importante).

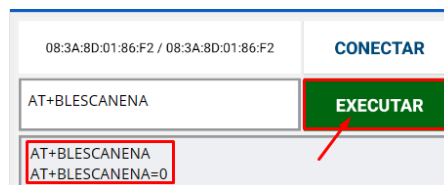


▼ Ative suporte ao BLE Scanner (necessário para obter pacotes BLE).

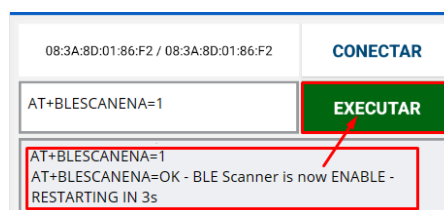
```
# Consultar
Envio: AT+BLESCANENA
Resposta: AT+BLESCANENA=0

# Configurar (1 = LIGADO, 0 = DESLIGADO)
Envio: AT+BLESCANENA=1
Resposta: AT+BLESCANENA=OK - BLE Scanner is now ENABLE - RESTARTING IN 3s
```

- Referencia Simova Apps - AT+BLESCANENA



- Referencia Simova Apps - AT+BLESCANENA=1



▼ Ative suporte ao Simova Angle

```
# Consultar
Envio: AT+SAENA
Resposta: AT+SAENA=0

# Configurar (1 = LIGADO, 0 = DESLIGADO)
Envio: AT+SAENA=1
Resposta: AT+SAENA=OK - Simova Angle support is now ENABLE - RESTARTING IN 3s
```

- Referencia Simova Apps - AT+SAENA

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAENA	EXECUTAR
AT+SAENA AT+SAENA=0	

- Referencia Simova Apps - AT+SAENA=1

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAENA=1	EXECUTAR
AT+SAENA=1 AT+SAENA=OK - Simova Angle support is now ENABLE - RESTARTING IN 3s	

▼ Verifique se o SimovaTrack está encontrando os beacons utilizando o comando `AT+BLEDGBGENA`

Consultar

Envio: `AT+BLEDGBGENA`

Resposta: `AT+BLEDGBGENA=0`

Para ligar o scanner

Envio: `AT+BLEDGBGENA=1`

Resposta: `AT+BLEDGBGENA=OK - BLE Debug is now ENABLE`

Para desligar o scanner

Envio: `AT+BLEDGBGENA=0`

Resposta: `AT+BLEDGBGENA=OK - BLE Debug is now DISABLE`

Resposta do escaneamento

Resposta: `AT+BLEDGBG MAC: 7D:63:7E:B1:04:A3, Last RSSI: -95, Package Count: 9, Last Seen: 480 ms ago`

Resposta: `AT+BLEDGBG MAC: CC:55:31:BA:D1:E0, Last RSSI: -78, Package Count: 44, Last Seen: 5 ms ago`

Resposta: `AT+BLEDGBG MAC: D4:9A:EB:01:6D:DC, Last RSSI: -84, Package Count: 43, Last Seen: 49 ms ago`

- Referencia Simova Apps - AT+BLEDGBGENA

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+BLEDGBGENA	EXECUTAR
AT+BLEDGBGENA AT+BLEDGBGENA=0	

- Referencia Simova Apps - AT+BLEDGBGENA=1

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+BLEDGBGENA=1	EXECUTAR
AT+BLEDGBGENA=1 AT+BLEDGBGENA=OK - BLE Debug is now ENABLE AT+BLEDGBG >MAC: FD:A3:4C:AC:C8:09, Last RSSI: -64, Package Count: 1, Last Seen: 4 ms ago AT+BLEDGBG >MAC: 6A:6B:E6:F7:DA:C6, Last RSSI: -98, Package Count: 15, Last Seen: 113 ms ago AT+BLEDGBG >MAC: FD:A3:4C:AC:C8:09, Last RSSI: -65, Package Count: 39, Last Seen: 19 ms ago	

▼ Configure o MAC_ADDRESS dos sensores (suportado 2 sensores) com os comandos `AT+SAMAC01=` e `AT+SAMAC02=`

```
# Consultar
Envio: AT+SAMAC01
Resposta: AT+SAMAC01=CC:55:31:BA:D1:E0

# Configurar mac 01
Envio: AT+SAMAC01=CC:55:31:BA:D1:E0
Resposta: AT+SAMAC01=OK - new mac is CC:55:31:BA:D1:E0

# Configurar mac 02 (OPCIONAL)
Envio: AT+SAMAC02=D4:9A:EB:01:6D:DC
Resposta: AT+SAMAC02=OK - new mac is D4:9A:EB:01:6D:DC
```

- Referencia Simova Apps - AT+SAMAC01

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAMAC01	EXECUTAR
AT+SAMAC01 AT+SAMAC01=ED:6D:B6:A3:E5:EB	

- Referencia Simova Apps - AT+SAMAC01=ED:6D:B6:A3:E5:EB

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAMAC01=ED:6D:B6:A3:E5:EB	EXECUTAR
AT+SAMAC01=ED:6D:B6:A3:E5:EB AT+SAMAC01=OK - new mac is CC:55:31:BA:D1:E0	

- Referencia Simova Apps - AT+SAMAC02=FD:A3:4C:AC:08:05

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAMAC02=FD:A3:4C:AC:08:05	EXECUTAR
AT+SAMAC02=FD:A3:4C:AC:08:05 AT+SAMAC02=OK - new mac is FD:A3:4C:AC:08:05	

▼ Utilize o `AT+API=ble` para conferir se os parâmetros `x,y,z` e `battery` foram lidos com sucesso, se todos os valores forem zero, confira se o sensor está ativo pelo aplicativo beaconxpro)

```
# Consultar
Envio: AT+API=ble
Resposta:
{
  "api.ble01": {
    "x": 0.00,
    "y": -0.64,
    "z": 12.16,
    "battery": 3599,
    "timestamp": 224948,
    "x_avg": 0.00,
    "y_avg": -0.43,
    "z_avg": 12.34,
    "sample_size": 20,
  }
}
```



```

    "timestamp_avg": 223668,
    "minRange": 20.01,
    "maxRange": 45.01,
    "isWorking": 1,
    "angle": 0.03
  },
  "api.ble02": {
    "x": 0.00,
    "y": 0.00,
    "z": 12.16,
    "battery": 3599,
    "timestamp": 225083,
    "x_avg": -0.03,
    "y_avg": -0.25,
    "z_avg": 12.72,
    "sample_size": 25,
    "timestamp_avg": 224808,
    "minRange": 0.51,
    "maxRange": 15.51,
    "isWorking": 1,
    "angle": 0.86
  },
}

```

- Fixe muito bem o sensor antes de prosseguir, a partir deste ponto é muito importante que o Simova Track e os Simova Angles utilizados já estejam colados e devidamente fixados.

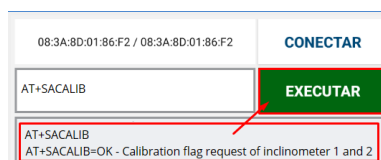
- ▼ Execute a calibração dos sensores enviando o comando `AT+SACALIB`

Calibrar

Envio: `AT+SACALIB`

Resposta: `AT+SACALIB=OK` - Calibration flag request of inclinometer 1 and 2

- Referencia Simova Apps - `AT+SACALIB`



- ▼ Consulte o angulo atual utilizando uma consulta unica `AT+SAANG01` e `AT+SAANG02` ou consulta automatica

`AT+SADBGENA=1`

Consulta unica

Envio: `AT+SAANG01`

Resposta: `AT+SAANG01= 0.10`

Consulta unica

Envio: `AT+SAANG02`

Resposta: `AT+SAANG02= 0.20`

Ativando consulta automatica

Envio: `AT+SADBGENA=1`

Resposta: `AT+SADBGENA=OK` - BLE Debug is now **ENABLE**

Resposta da consulta **automatica** (a cada 1s).

Resposta: AT+SADBG => api.ble01.angle: 100.40, api.ble02.angle: 97.14 | BAT 1: 3599mv, BAT 2: 3599 mv

- Referencia Simova Apps - AT+SAANG01

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAANG01	EXECUTAR
AT+SAANG01 AT+SAANG01= 0.09	

- Referencia Simova Apps - AT+SAANG02

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAANG02	EXECUTAR
AT+SAANG02 AT+SAANG02= 0.18	

- Referencia Simova Apps - AT+SADBGENA=1

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SADBGENA=1	EXECUTAR
AT+SADBGENA=1 AT+SADBGENA=OK - BLE Debug is now ENABLE AT+SADBG => api.ble01.angle: 90.40, api.ble02.angle: 67.12 BAT 1: 3599mv, BAT 2: 3599mv	

▼ Configure os ângulos mínimos e máximos para considerar o sensor "valido" utilizando os comandos

AT+SAANG01MIN e AT+SAANG01MAX para o sensor 1, e AT+SAANG02MIN e AT+SAANG02MAX para o sensor 2.

Note que os valores mínimos e máximos são definidos pelo instalador, e devem ser configurados com os sensores já fixados.

Consultando um valor ainda não configurado:

Envio: AT+SAANG01MIN

Resposta: AT+SAANG01MIN=ERROR Key ICL1_MIN does not exist

Consultando um valor já configurado:

Envio: AT+SAANG01MIN

Resposta: AT+SAANG01MIN=0.000000

Configurando um novo valor:

Envio: AT+SAANG01MIN=20.01

Resposta: AT+SAANG01MIN=OK - new min angle is 20.010000

Envio: AT+SAANG01MAX=40.99

Resposta: AT+SAANG01MAX=OK - new max angle is 40.990000

Envio: AT+SAANG02MIN=30

Resposta: AT+SAANG02MIN=OK - new min angle is 30.000000

Envio: AT+SAANG02MAX=60

Resposta: AT+SAANG02MAX=OK - new max angle is 60.000000

- Referencia Simova Apps - AT+SAANG01MIN=20.01

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAANG01MIN=31.00	EXECUTAR
<pre> AT+SAMAC01=ED:6D:B6:A3:E5:EB AT+OK - Sucessfully modified and saved new MAC Address ## ERROR SENDING [AT+SACALIB] ## ## ERROR SENDING [AT+SACALIB] ## ## ERROR SENDING [AT+SACALIB] ## AT+SAANG01_MIN=31.00 Command unknown: source: 5, data:AT+SAANG01_MIN=31.00 AT+SAANG01MIN=31.00 AT+OK - Sucessfully modified and saved new MIN ANGLE </pre>	

- Referencia Simova Apps - AT+SAANG01MAX=90

08:3A:8D:01:86:F2 / 08:3A:8D:01:86:F2	CONECTAR
AT+SAANG01MAX=75.00	EXECUTAR
<pre> ## ERROR SENDING [AT+SACALIB] ## ## ERROR SENDING [AT+SACALIB] ## ## ERROR SENDING [AT+SACALIB] ## AT+SAANG01_MIN=31.00 Command unknown: source: 5, data:AT+SAANG01_MIN=31.00 AT+SAANG01MIN=31.00 AT+OK - Sucessfully modified and saved new MIN ANGLE AT+SAANG01MAX=75.00 AT+OK - Sucessfully modified and saved new MAX ANGLE </pre>	

- A partir desse ponto o sensor já está pareado corretamente, e pode ser usado o comando AT+API=ble para consultar suas informações.
- Não se esqueça de ativar o `programa especifico` a ser utilizado (secção `Como selecionar um programa especifico (Experimental)`)

▼ Como configurar a REDE CAN

Antes de iniciar a instalação, confira se o equipamento está funcionando (ex ligue o carro ou caminhão e tenha certeza que aparecem dados no painel, por exemplo a RPM está aparecendo).

Por padrão o SimovaTrack é enviado com a rede CAN Habilitada e com o modo de escaneamento automático ativo, basta conecta-lo a rede CAN e o aparelho irá testar automaticamente as velocidades mais comuns e tentará validar uma conexão, Assim que validada, as configurações serão salvas na memória.

▼ Modo de uso CAN BUS

O formato do comando para AT+CAN=modo é o seguinte:

- ▼ **Modo:** O modo operacional do serviço CAN. Opções válidas são:

- ON:** Liga o serviço CAN.
- OFF:** Desliga o serviço CAN.
- SCAN:** Liga o serviço de SCANEAMENTO.
- INFO:** Mostra as informações atuais da CAN.

▼ Configurando CAN BUS

O formato do comando para AT+CANMODE= modo; velocidade; protocolo é o seguinte:

- ▼ **Modo:** O modo operacional do serviço CAN. Opções válidas são:

- NORMAL:** Modo de leitura e escrita.
- LISTEN_ONLY:** Modo de leitura somente.

- ▼ **Velocidade:** A velocidade do barramento CAN. Opções válidas são:

- Velocidades específicas:** 25000, 50000, 100000, 125000, 250000, 500000, 800000, 1000000.

- ▼ **Protocolo:** O protocolo usado pelo serviço CAN. Opções válidas são:

1. **J1939:** Usa o protocolo J1939.

`AT+CANMODE=LISTEN_ONLY;250000;J1939`

2. **OBD2:** Usa o protocolo OBD2.

`AT+CANMODE=NORMAL;250000;OBD2`

3. **NMEA2000:** Protocolo de troca de dados marítimos usando uma rede CAN.

`AT+CANMODE=NORMAL;250000;NMEA2000`

4. **CANOPEN:** Protocolo de CAN maior compatibilidade com veículos industriais.

`AT+CANMODE=NORMAL;250000;CANOPEN`

5. **DEVICENET:** Protocolo para controlar dispositivos industriais, como sensores e atuadores, em redes de automação.

`AT+CANMODE=NORMAL;250000;DEVICENET`

6. **SNIFUART:** Usa o protocolo Sniffer UART (**Cabo USB**). `AT+CANMODE=NORMAL;250000;SNIFUART`

7. **SNIFBLE:** Usa o protocolo Sniffer BLE (**Terminal BLE**). `AT+CANMODE=NORMAL;250000;SNIFBLE`

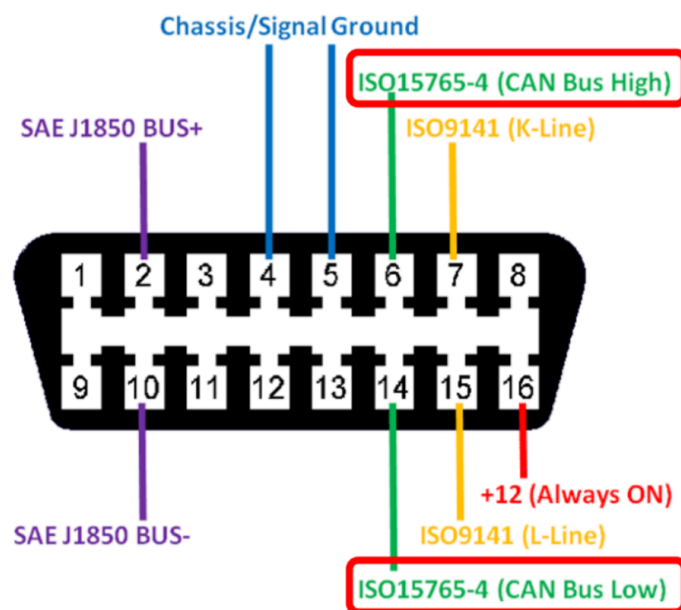
Saída do SNIFUART/SNIFBLE- Protocolo de engenharia reversa (captura de dados):

```
AT+CANSNIFFER=ID=0x0000010F;DLC=8;Data=FE 06 00 00 00 57 00 20;DLC Non-Comp=0;Extended=0;Flags=0;Reserved=0;RTR=0;Self=0;SS=0;
AT+CANSNIFFER=ID=0x00000260;DLC=8;Data=00 3A 30 00 00 69 04;DLC Non-Comp=0;Extended=0;Flags=0;Reserved=0;RTR=0;Self=0;SS=0;
AT+CANSNIFFER=ID=0x000002A0;DLC=8;Data=80 00 9E 00 00 00 00 00;DLC Non-Comp=0;Extended=0;Flags=0;Reserved=0;RTR=0;Self=0;SS=0;
AT+CANSNIFFER=ID=0x000000A0;DLC=8;Data=6F 80 91 00 00 26 02 00;DLC Non-Comp=0;Extended=0;Flags=0;Reserved=0;RTR=0;Self=0;SS=0;
AT+CANSNIFFER=ID=0x000000A1;DLC=8;Data=7E 81 00 00 3C 00 00 00;DLC Non-Comp=0;Extended=0;Flags=0;Reserved=0;RTR=0;Self=0;SS=0;
```

(Esse modo apenas exibe no dados, não decodifica)

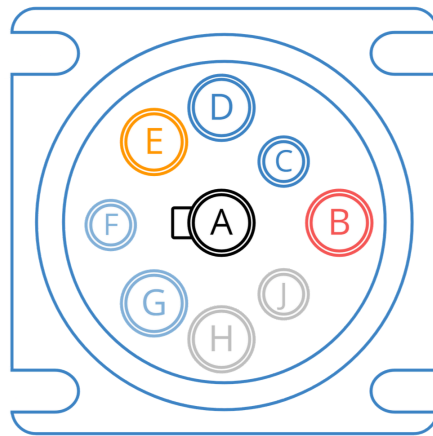
Curiosidade: Usado para detectar qual cabo da rede CAN esta transmitindo dados.

- ▼ Identificando CAN pelo conector OBD2 - pinos 6 e 14



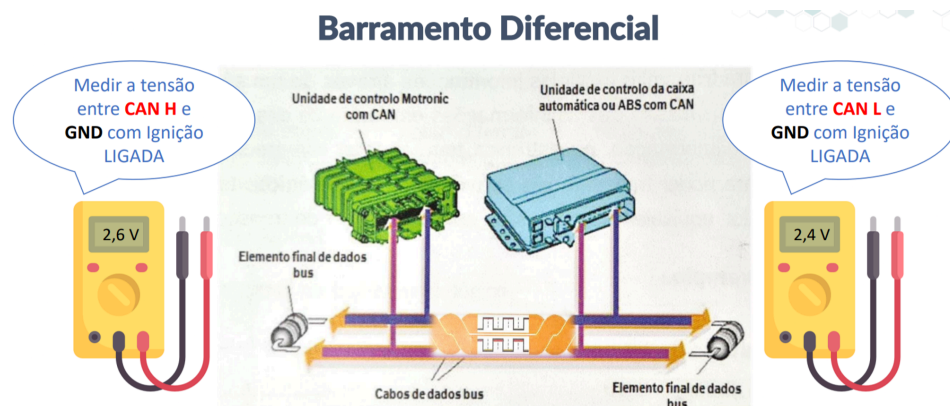
Localize os pinos 06 e 14 confira se há conector nesses pinos (se não houver, o veículo utiliza outro tipo de comunicação que não é suportado atualmente).

- ▼ Identificando CAN pelo conector J1939 - pinos C e D



- A Ground
 - B Battery power
 - C CAN 1 H
 - D CAN 1 L
 - E CAN shield
 - F J1708 (+) / CAN 2 H
 - G J1708 (-) / CAN 2 L
 - H OEM specific
 - J OEM specific
- Type 1 (black): CAN 1 = 250K
 Type 2 (green): CAN 1 = 500K

▼ Identificando CAN pela tensão dos pinos



Depois de identificar os fios candidados a CAN H e CAN L, ligue a ignição e utilize um multímetro na escala de tensão

1. fixe a ponteira NEGATIVA no chassi ou fio GND
2. Coloque a ponteira POSITIVA no fio/pino CAN H

Resultado ⇒ A tensão deverá ser de 2.6V

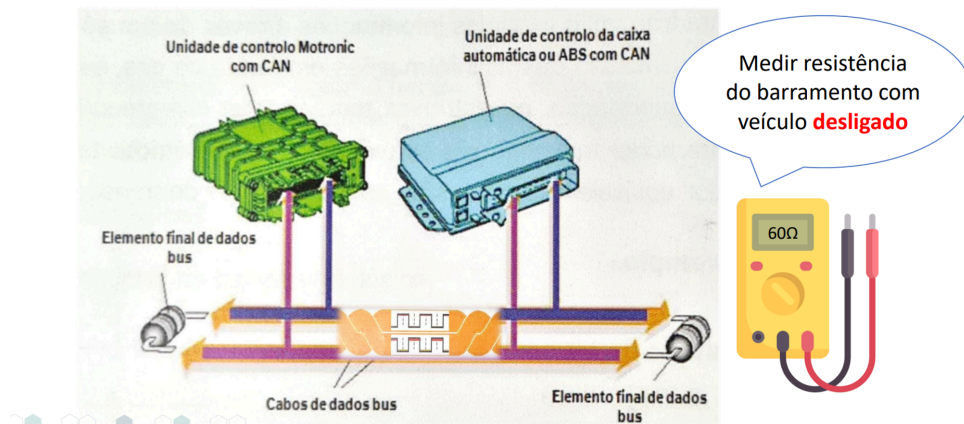
1. fixe a ponteira NEGATIVA no chassi ou fio GND
2. Coloque a ponteira POSITIVA no pino CAN L

Resultado ⇒ A tensão deverá ser de 2.4V

Caso não encontre essas tensões, o barramento não é CAN ou pode estar desconectado.

▼ Conferindo a terminação da rede CAN

Resistores nas extremidades



Importância da Terminação da Rede CAN

A terminação correta é importante para manter a integridade do sinal.

Para medir a terminação correta da rede CAN, que deve ser de aproximadamente 60 ohms, siga estes passos

1. Desligue a rede CAN

2. Use um Multímetro

- Ajuste o multímetro para medir resistência.
- Coloque as sondas do multímetro nos terminais do barramento CAN nas extremidades. Como existem dois resistores de 120 ohms, um em cada extremidade e eles estão em paralelo, o valor medido deve ser próximo de 60 ohms.

Como Ajustar a Terminação da Rede CAN

Se a medição mostrar que a terminação não está correta (próxima de 60 ohms), siga estes passos para ajustar:

- Verifique se os resistores de 120 ohms estão presentes em ambas as extremidades e se estão em boas condições.
- Se os resistores estiverem ausentes ou danificados, substitua por novos de 120 ohms em cada extremidade do barramento.
- Se o valor medido for diferente de 60 ohms, verifique a conexão e o posicionamento dos resistores.
- Após fazer os ajustes necessários, meça novamente a resistência para assegurar que está correta.

▼ Ativando o escaneamento automático (recomendado)

O Escaneamento automático irá testar automaticamente as velocidades mais comuns e tentará validar uma conexão. Assim que validada (enviar e receber uma resposta), as configurações serão salvas na memória

- Conecte-se ao terminal e envie o comando `AT+CAN=SCAN` (ele iniciará uma varredura na CAN para ver qual velocidade é a correta)
 - Varredura NÃO-Invasiva → Sem escrita (Apenas monitorando pacotes da rede para encontrar a velocidade correta).
 - Varredura Invasiva → Com escrita (Envios de mensagens/comandos para ver se a velocidade da rede CAN é compatível).

Como alterar a velocidade da rede can manualmente (não recomendado)

A lista abaixo contém as velocidades padrões em ordem de popularidade:

#	Velocidade	Comando	Uso comum:
1	500kbps	AT+CANMODE=LISTEN_ONLY;500000;J1939;	carros, caminhões, tratores

2	250kbps	AT+CANMODE=LISTEN_ONLY;250000;J1939;	carros, caminhões, tratores
3	125kbps	AT+CANMODE=LISTEN_ONLY;125000;J1939;	incomum
4	1Mbps	AT+CANMODE=LISTEN_ONLY;1000000;J1939;	incomum
5	62.5kbps	AT+CANMODE=LISTEN_ONLY;62500 ;J1939;	incomum
6	50kbps	AT+CANMODE=LISTEN_ONLY;50000 ;J1939;	incomum
7	20kbps	AT+CANMODE=LISTEN_ONLY;20000 ;J1939;	incomum
8	10kbps	AT+CANMODE=LISTEN_ONLY;10000 ;J1939;	incomum

Utilize o comando `AT+CANMODE` para obter o valor atual e configurar um novo valor, este parâmetro reinicia o aparelho após 10s para aplicar as alterações.

```
# Consultar
TX: AT+CANMODE
RX: AT+CANMODE=ON;250000;J1939;

# Configurar
TX: AT+CANMODE=ON;123321;J1939;
TX: AT+RESTART
RX: AT+CANMODE=OK - Changed from 250000 to 123321, will restart in 10s
```

▼ CAN INFO: Verificando o Funcionamento

1. Monitore o parâmetro `packages_per_second`
 - Se for "0", pode indicar baud rate incorreto ou problema na rede
2. Envie o comando `AT+CAN=INFO`
3. Conecte-se ao terminal serial (BLE ou físico)

Para verificar o funcionamento da rede CAN:

Informações do CANINFO:

- **Datetime:** 2024-12-30 05:27:15 (GMT 0)
- **State:** SCANNING

Estado	Descrição
0 - Stopped	Parado/Sem comunicação
1 - Running	Comunicação ativa
2 - Bus-Off	Falha na comunicação com CAN
3 - Recovering	Recuperando comunicação
4 - Unknown	Estado desconhecido/Desconectado

Contadores de Mensagens:

- Messages to TX: 0 (Mensagens transmitidas)
- Messages to RX: 0 (Mensagens recebidas)
- TX Error Counter: 0 (Falhas na transmissão)
- RX Error Counter: 0 (Falhas na recepção)
- TX Failed Count: 0 (Transmissões mal-sucedidas)
- RX Missed Count: 0 (Recepções perdidas)
- RX Overrun Count: 0 (Mensagens excedidas)
- Arbitration Lost: 0 (Mensagens não priorizadas)
- Bus Error Count: 0 (Erros no barramento)

Exemplos:

Simova track com cabo de rede can DESCONECTADO

```
{
  "timestamp": "2024-12-30 05:27:15",
  "uptime": "0d 00:01:52",
  "configuration": {
    "systemState": "SCANNING",
    "protocol": "Unknown",
    "speed": "125000",
    "mode": "NORMAL"
  },
  "tasks": {
    "busControl": {
      "state": "Blocked",
      "stackHWM": 1812
    },
    "dataProcessing": {
      "state": "Suspended",
      "stackHWM": 5216
    }
  },
  "memory": {
    "freeHeap": 84848,
    "minFreeEver": 82764,
    "fragmentation": 34
  },
  "busStatistics": {
    "state": "Running",
    "messages": {
      "tx": 0,
      "rx": 0
    },
    "errorCounters": {
      "tx": 0,
      "rx": 0
    },
    "errorMetrics": {
      "busErrors": 0,
      "failedTx": 0,
      "missedRx": 0,
      "rxOverrun": 0,
      "arbitrationLost": 0
    }
  }
}
```

A resposta do comando `AT+API=CANBUS` será similar a mensagem abaixo (com parâmetros preenchidos. parâmetros com o valor -1 não foram lidos pela rede.

```
{
  "api.sensors.can": {
    "serviceInfo": {
      "status": 1,
      "protocol": 0,
      "speed": 250000,
      "debug": 0,

```



```

        "debugTarget": 1
    },
    "bus": {
        "lastRxMs": 28108,
        "lastTxMs": 0,
        "lastProcessedJ1939": 0,
        "lastProcessedOBD2": 27430,
        "countProcessedJ1939": 0,
        "countProcessedOBD2": 20,
        "busStatus": 1
    },
    "j1939": {...},
    "obd2": {
        "rpm_PID0C": 870,
        "speedKmh_PID0D": 0,
        "fuelLevelPct_PID2F": 61.5686264,
        "runtimeSec_PID1F": 2691,
        ...
        "throttlePositionPct_PID11": 14.90196037,
        ...
    }
}
}

```

▼ Exemplo de velocidade errada

O veículo em teste usa 500000 e está configurado 250000

Comando enviado:

```
AT+CANMODE=ON;250000;AUTO;
```

```
[1323398][1][core_lib.cpp:300] update_sensor_task(): [SENSOR_TASK] RTC NOT synced to GPS
AT+CANMODE=OK - Mode set to ON, Speed set to 250000, Protocol set to AUTO
```

Exemplo de velocidade corrigida e rede identificada:

Comando enviado:

```
AT+CANMODE=ON;AUTO;AUTO;
```

```
AT+CANMODE=OK - Mode set to ON, Speed set to 250000, Protocol set to AUTO
AT+CANMODE=OK - Mode set to ON, Speed set to AUTO, Protocol set to AUTO
```

▼ PIDs Suportados (OBD2)

1. **OBD2_SPEED_PID (0x0D)** - Velocidade do veículo: `api.sensors.obd2.vehicle_speed`
2. **OBD2_RPM_PID (0x0C)** - RPM do motor: `api.sensors.obd2.engine_rpm`
3. **OBD2_TEMP_PID (0x05)** - Temperatura do líquido de arrefecimento do motor: `api.sensors.obd2.coolant_temp`
4. **OBD2_RUNTIME_PID (0x1F)** - Tempo de funcionamento do motor desde a última partida: `api.sensors.obd2.engine_runtime`
5. **OBD2_MAF_PID (0x10)** - Taxa de fluxo de massa de ar: `api.sensors.obd2.maf_rate`
6. **OBD2_THROTTLE_POSITION_PID (0x11)** - Posição do acelerador: `api.sensors.obd2.throttle_position`
7. **OBD2_ENGINE_HOURS_PID (0x09)** - Horas do motor (não existe, marcado como errado no código)
8. **OBD2_FUEL_FLOW_RATE_PID (0x5E)** - Taxa de fluxo de combustível: `api.sensors.obd2.fuel_flow_rate`

9. **OBD2_ENGINE_COOLANT_TEMPERATURE_PID (0x5C)** - Temperatura do líquido de arrefecimento do motor: `api.sensors.obd2.coolant_temp`
 10. **OBD2_AMBIENT_AIR_TEMPERATURE_PID (0x67)** - Temperatura do ar ambiente: `api.sensors.obd2.ambient_air_temp`
 11. **OBD2_FUEL_LEVEL_INPUT_PID (0x2F)** - Nível de combustível: `api.sensors.obd2.fuel_level`
 12. **OBD2_DISTANCE_TRAVELED_MIL_ON_PID (0x21)** - Distância percorrida com a luz de verificação do motor ligada: `api.sensors.obd2.distance_traveled_mil_on`
 13. **BD2_EVAP_SYSTEM_VAPOR_PRESSURE_PID (0x32)** - Pressão do vapor do sistema EVAP: `api.sensors.obd2.evap_vapor_pressure`
 14. **OBD2_BAROMETRIC_PRESSURE_PID (0x33)** Pressão barométrica: `api.sensors.obd2.barometric_pressure`
 15. **OBD2_OXYGEN_SENSOR1_VOLTAGE_PID (0x14)** Tensão do sensor de oxigênio 1: `api.sensors.obd2.o2_sensor1_voltage`
 16. **OBD2_OXYGEN_SENSOR2_VOLTAGE_PID (0x15)** Tensão do sensor de oxigênio 2: `api.sensors.obd2.o2_sensor2_voltage`
 17. **OBD2_OXYGEN_SENSOR3_VOLTAGE_PID (0x16)** Tensão do sensor de oxigênio 3: `api.sensors.obd2.o2_sensor3_voltage`
 18. **OBD2_OXYGEN_SENSOR4_VOLTAGE_PID (0x17)** Tensão do sensor de oxigênio 4: `api.sensors.obd2.o2_sensor4_voltage`
 19. **OBD2_FUEL_RAIL_PRESSURE_PID (0x22)** Pressão do trilho de combustível: `api.sensors.obd2.fuel_rail_pressure`
 20. **OBD2_VEHICLE_MILEAGE_PID (0x13)** Quilometragem do veículo: `api.sensors.obd2.vehicle_mileage`
- ▼ PGNs Suportados (J1939)
1. **PGN_61444 (0x18F004AA)** - RPM do motor
 - **SPN 190** - RPM do motor: `api.sensors.can.j1939_rpm`
 2. **PGN_65265 (0x18FEF1AA)** - Velocidade do veículo
 - **SPN 84** - Velocidade do veículo em km/h: `api.sensors.can.j1939_speed_kmh`
 3. **PGN_65262 (0x18FEEFAA)** - Temperatura do motor
 - **SPN 110** - Temperatura do motor em graus Celsius: `api.sensors.can.j1939_eng_temp_c`
 4. **PGN_65263 (0x18FEEFAA)** - Nível/Pressão dos fluidos do motor
 - **SPN 100** - Pressão do motor em kPa: `api.sensors.can.j1939_eng_pressure_kpa`
 5. **PGN_65253 (0x18FEE5AA)** - Horas de ignição
 - **SPN 247** - Timer de ignição em segundos: `api.sensors.can.j1939_eng_timer_s`
 6. **PGN_65217 (0x18FEC1AA)** - Parâmetros dinâmicos do motor
 - **SPN 917** - Timer do motor em segundos: `api.sensors.can.j1939_eng_timer_s`
 7. **PGN_65266 (0x18FEF2AA)** - Distância total do veículo e taxa de consumo de combustível
 - **SPN 183** - Distância total do veículo em metros e taxa de combustível em litros/hora: `api.sensors.can.j1939_total_vehicle_dist_m` e `api.sensors.can.j1939_fuel_rate_l`
 8. **PGN_65257 (0x18FEE9AA)** - Parâmetros estáticos do motor
 - **SPN 250** - Horas totais do motor: `api.sensors.can.j1939_total_fuel_used_l`

▼ Como configurar a fonte de IGNIÇÃO

A Fonte da Ignição onde será configurado a entrada digital.

1. Realize a configuração da fonte:

▼ Configuração da fonte:

Comando	Fonte	Saida
AT+IGNSRC=0	CAN J1939 ou OBD2	api.sensors.can. can_bus_status && (obd2_status j1939_status)
AT+IGNSRC=1	Entrada digital 1	api.sensors.gpio.gpio_1
AT+IGNSRC=2	Entrada digital 2	api.sensors.gpio.gpio_2
AT+IGNSRC=3	Entrada digital 3	api.sensors.gpio.gpio_1
AT+IGNSRC=4	Entrada digital 4	api.sensors.gpio.gpio_1
AT+IGNSRC=5	Tensão de Bateria	if(api.sensors.gpio.psu_voltage > api.telemetry.min_voltage_for_ignition)
AT+IGNSRC=6	Sempre ligado	1

Utilize o comando `AT+IGNSRC` para obter o valor atual e configurar um novo valor.

```
# Consultar
TX: AT+IGNSRC
RX: AT+IGNSRC=0
# Configurar
TX: AT+IGNSRC=0
RX: AT+IGNSRC=OK - Ignition source set to 0 - IGNITION_SRC_CAN - RESTARTING IN 10s
```

2. Para a fonte: `Tensão de Bateria` é necessário realizar a seguinte configuração:

▼ Configuração tensão de trabalho mínima (`AT+IGNVMIN`)

Utilize o comando `AT+IGNVMIN` para obter o valor atual e configurar um novo valor de tensão de trabalho mínima.

lembrete: a tensão mínima não é obrigatório ser 12.5, esse valor é uma média gerada da tensão da bateria desligada e ligada.

```
# Consultar
TX: AT+IGNVMIN
RX: AT+IGNVMIN=12.0
# Configurar
TX: AT+IGNVMIN=12.5
RX: AT+IGNVMIN=OK - Changed from 12.00 to 12.50
```

▼ Caso de dúvidas com a configuração da Tensão de Bateria

exemplo na aba "Exemplos de instalação", "Celta LS 2013-Bruno " <https://www.notion.so/Guia-do-instalador-Simova-Track-ff9e3b82bd5d4fe9b12f359e978a5b9f?pvs=4#cf22b31262d84fc0b54e9cbc727414f3>

▼ Como Inverter Logica de Entradas

Os comandos `AT+INVERTGPIO1`, `AT+INVERTGPIO2`, `AT+INVERTGPIO3` e `AT+INVERTGPIO4` são usados para inverter o comportamento das entradas.

O principal uso desses comandos é adequar o comportamento das entradas de sensores ou outros dispositivos, especialmente quando o sinal "**válido**" é considerado **desligado** em vez de ligado, ou seja, o estado ativo corresponde a um nível baixo de tensão (0V) em vez de um nível alto (24V).

▼ Exemplo:

Esses comandos são ideais para cenários onde há sensores NPN conectados às entradas digitais. Sensores NPN, que são comumente utilizados em automação industrial, mantêm uma tensão alta (como 24V) **quando inativos** e fornecem 0V (sinal baixo) **quando ativados**.

```
# Consultar
TX: AT+INVERTGPIO1
RX: AT+INVERTGPIO1=0
```

```
# Configurar
TX: AT+INVERTGPIO1=1
RX: AT+INVERTGPIO1=OK - Changed from 0 to 1
```

▼ Como configurar a fonte de RPM

RPM (Rotação por Minuto) frequência com que o motor gira, para configurar a fonte do RPM.

1. Realize a configuração da fonte:

▼ Configuração da fonte:

Comando	Fonte	Saida
AT+RPMSRC=0	CAN J1939 ou OBD2	<code>api.sensors.can.j1939_rpm</code> ou <code>api.sensors.can.j1939_rpm</code>
AT+RPMSRC=1	Entrada digital 1	<code>api.sensors.gpio.gpio_1</code>
AT+RPMSRC=2	Entrada digital 2	<code>api.sensors.gpio.gpio_2</code>
AT+RPMSRC=3	Entrada digital 3	<code>api.sensors.gpio.gpio_1</code>
AT+RPMSRC=4	Entrada digital 4	<code>api.sensors.gpio.gpio_1</code>
AT+RPMSRC=5	Tensão de Bateria	<code>if(api.sensors.gpio.psu_voltage > api.telemetry.min_voltage_for_rpm) =1235</code>
AT+RPMSRC=6	Fictício - Fixo	=1234

Utilize o comando `AT+RPMSRC` para obter o valor atual e configurar um novo valor.

```
# Consultar
TX: AT+RPMSRC
RX: AT+RPMSRC=0
# Configurar
TX: AT+RPMSRC=0
RX: AT+RPMSRC=OK - Engine source set to 0 - RPM_SRC_CAN - RESTARTING IN 10s
```

2. Para a fonte: `Tensão de Bateria` é necessário realizar a seguinte configuração:

▼ Configurando tensão de trabalho mínima (`AT+RPMVMIN`)

Utilize o comando `AT+RPMVMIN` para obter o valor atual e configurar um novo valor de tensão de trabalho mínima.

```
# Consultar
TX: AT+IGNVMIN
RX: AT+IGNVMIN=12.0
# Configurar
TX: AT+IGNVMIN=12.5
RX: AT+IGNVMIN=OK - Changed from 12.00 to 12.50
```

3. Para as fontes `Entrada digital 1` , `Entrada digital 2` , `Entrada digital 3` , `Entrada digital 4` , É necessário calibrar a RPM

▼ Configuração específica para RPM via Alternador

▼ Como localizar o fio do alternador

1. Procure pelo símbolo "w" do alternador

- a. Dica, localize o fio W com uma ponteira indicadora de polaridade (caneta de teste). o W irá oscilar entre positivo e negativo.



- b. Caso o fio não esteja exposto, pode ser necessário desmontar o alternador e extrair o fio direto do estator.

2. ligue o veículo e deixe o estabilizar na marcha lenta

▼ Como realizar e testar a conexão elétrica

1. Conecte o fio que vem do “w - alternador” na entrada digital escolhida anteriormente (1 a 4)
2. Monitore o led da entrada digital escolhida na seleção de fonte, ele deverá estar piscando quando o motor está ligado
 - a. Caso não esteja piscando, reveja a instalação com um multímetro.
3. Envie o comando `AT+API=gpio` e monitore o parâmetro `“pulse_counter”` que não pode estar em zero.

```
api.sensors.gpio.  
  gpio_1: 1  
  gpio_2: 1  
  gpio_3: 0  
  gpio_4: 0  
  psu_voltage: 13.47 V  
  pulse_counter: 280  
  rpm_multiplier: 1.00  
  rpm: 280.00
```

▼ Como calibrar a RPM

- Este comando é útil para ajustar a precisão das leituras de RPM em sistemas que dependem de se
- Ligue o veículo, desligue ar condicionado, aguarde pelo menos um minuto e anote o RPM exibido no painel do veículo (exemplo 860).

É muito importante aguardar um minuto, a função depende de 60s de amostragem.

- Utilize o comando `AT+CALIB_RPM` para obter o valor atual e configurar um novo valor

```
# Realizar calibração  
TX: AT+CALIB_RPM=1000  
RX: AT+OK: New rpm_multiplier set to 0.1433, expected avg rpm is: 1000  
  
TX: AT+API=gpio  
RX: api.sensors.gpio.  
  gpio_1: 0  
  gpio_2: 0  
  gpio_3: 0  
  gpio_4: 1  
  psu_voltage: 3.30 V  
  pulse_counter: 6  
  rpm_multiplier: 0.52  
  rpm: 1093.75
```

A função verifica se o parâmetro (exemplo 860) válido e maior que zero,
Em seguida, calcula o número atual de pulsos por segundo, que é derivado do contador de pulsos

- O novo multiplicador de RPM é calculado usando a fórmula:

```
novo_multiplicador = RPM_desejado / (pulsos_por_segundo * 60)
```

- No exemplo dado, se o sistema estiver contando 100 pulsos por segundo:

```
novo_multiplicador = 860 / (100 * 60) = 0.1433
```

- O multiplicador calculado é então atualizado na API.
- O valor é salvo no armazenamento não volátil (NVS) para persistência entre reinicializações.

▼ Conferindo a RPM

1. Envie o comando `AT+API=gpio` e monitore o parametro RPM:

```
Sensor GPIO:
GPIO 1: 1      ← Estado da entrada 01
GPIO 2: 0      ← Estado da entrada 02
GPIO 3: 0      ← Estado da entrada 03
GPIO 4: 0      ← Estado da entrada 04
PSU Voltage: 3.116667 V ← Tensão de alimentação da placa
Pulsed RPM: 2   ← contador de pulsos (é reiniciado a cada 1s)
RPM: 1075.000000 ← rpm calculada
RPM Multiplier: 3.583333 ← multiplicador utilizado
```

▼ Lista de comandos relacionados

- `AT+GPORPM` → Responde o multiplicador de RPM atual
- `AT+GPORPM=12.345` → Configura um novo multiplicador de RPM
- `AT+CALIB_RPM` → Responde o multiplicador de RPM atual
- `AT+CALIB_RPM=1600` → Utiliza o valor da RPM do painel (ex 1600) para calcular e salvar o multiplicador automaticamente - Recomenda-se usar apenas esse comando
- `AT+API=gpio` → Responde os dados da API relacionados

```
Sensor GPIO:
GPIO 1: 1      ← Estado da entrada 01
GPIO 2: 0      ← Estado da entrada 02
GPIO 3: 0      ← Estado da entrada 03
GPIO 4: 0      ← Estado da entrada 04
PSU Voltage: 3.116667 V ← Tensão de alimentação da placa
Pulsed RPM: 2   ← contador de pulsos (é reiniciado a cada 1s)
RPM: 1075.000000 ← rpm calculada
RPM Multiplier: 3.583333 ← multiplicador utilizado
```

▼ Como consultar e configurar Horímetros, Hodômetros e contadores

O Simova track possui os seguintes valores que podem ser utilizados na telemetria:

▼ Horímetro de ignição ligada (cumulativo)

Utilize o comando `AT+IGNHR` para consultar e configurar

```
# Consultar
TX: AT+IGNHR
RX: AT+IGNHR=7607s | 2.11hrs
# Configurar (parametro em horas)
TX: AT+IGNHR=12.34
RX: AT+IGNHR=OK - Changed from 7605s | 2.11hrs to 44424s | 12.34hrs
```

Pode ser localizado na API em `api.telemetry.ignition_timer_seconds`

- Valor em segundos, para converter em horas divida por 3600
- Variável de alta precisão, capaz de armazenar até 1.2 milhões de horas ou 136.10 anos com precisão de 1s. Não suporta números negativos!

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Horímetro de motor ligado (cumulativo)

Utilize o comando `AT+ENGHR` para consultar e configurar

```
# Consultar
TX: AT+ENGHR
RX: AT+ENGHR=5868s | 1.63hrs
# Configurar (parametro em horas)
TX: AT+ENGHR=12.34
RX: AT+ENGHR=OK - Changed from 5868s | 1.63hrs to 44424s | 12.34hrs
```

Pode ser localizado na API em `api.telemetry.engine_timer_seconds`

- Valor em segundos, para converter em horas divida por 3600
- Variável de alta precisão, capaz de armazenar até 1.2 milhões de horas ou 136.10 anos com precisão de 1s. Não suporta números negativos!

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Horímetro de movimento (cumulativo)

Utilize o comando `AT+MOVHR` para consultar e configurar

```
# Consultar
TX: AT+MOVHR
RX: AT+MOVHR=7607s | 2.11hrs
# Configurar (parametro em horas)
TX: AT+MOVHR=12.34
RX: AT+MOVHR=OK - Changed from 7605s | 2.11hrs to 44424s | 12.34hrs
```

Pode ser localizado na API em `api.telemetry.movement_timer_seconds`

- Valor em segundos, para converter em horas divida por 3600
- Variável de alta precisão, capaz de armazenar até 1.2 milhões de horas ou 136.10 anos com precisão de 1s. Não suporta números negativos!

É atualizado automaticamente baseado no Horímetro de GPS (`api.telemetry.gps_trip_distance_meters`)

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Contador Auxiliar 01

Utilize o comando `>QCT03` e `>SCT03` para consultar e configurar

- Também conhecido como CT03 no protocolo XVM (virlocs 6 e 8)
- pode ser localizado na API em `api.telemetry.aux1_timer_seconds`
 - Variável de alta precisão, capaz de armazenar 4294967295 valores diferentes. Não suporta números negativos! Pode ser usada para contar horas, metros, pulsos, etc.

```
# Consultar
TX: >QCT03
RX: >RCT03 0
# Configurar (parametro em segundos)
TX: >SCT03 1234
RX: >RCT03 1234
```

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Contador Auxiliar 02

- Também conhecido como CT10 no protocolo XVM (virlocs 6 e 8)

- pode ser localizado na API em `api.telemetry.aux2_timer_seconds`
 - Variável de alta precisão, capaz de armazenar 4294967295 valores diferentes. Não suporta números negativos! Pode ser usada para contar horas, metros, pulsos, etc.

```
# Consultar
TX: >QCT10
RX: >RCT10 0

# Configurar (parametro em segundos)
TX: >SCT10 1234
RX: >RCT10 1234
```

Por segurança, esse comando gera um registro permanente em memória, contendo data, hora e localização GPS e informações pertinentes.

▼ Como consultar a API

Todos os dados do SimovaTrack podem ser monitorados através da API, que pode ser acessada através dos seguintes comandos:

```
# O comando AT+API retorna todos os dados disponiveis
AT+API

# O comando abaixo diz quais parametros de filtros podemos usar
AT+API=?

# Os comandos abaixo filtram apenas uma seção da mensagem
AT+API=device_info
AT+API=temp
AT+API=compass
AT+API=gps
AT+API=clock
AT+API=accelerometer
AT+API=gpio
AT+API=ble
AT+API=telemetry
AT+API=canbus
AT+API=comms

# O comando AT+APIS funciona de forma identica, mas entrega os dados de forma
# compactada, ideal para processamento por computadores
```

▼ Resposta do comando `AT+API`

```
{
  "api.device_info": {
    "boot_counter": 4,
    "device_connected": false,
    "device_connected_mtu": 0,
    "sdcard_working": true,
    "can_bus_status": false,
    "firmware_version": "TRACK HW1.3.0 FW4.4.0",
    "uart0_baudrate": 115200,
    "short_firmware_version": 440,
    "mac_address": 100100,
    "current_program": 0,
    "ble_scanner_enable": false,
```



```

    "ble_scanner_debug": false,
    "simova_angle_enable": false,
    "simova_angle_debug": false,
    "ota_initialize": 0,
    "sattelite_s1c_enable": 0,
    "sattelite_s1c_debug": 0
  },
  "api.sensors.temperature": {
    "temp_celsius": 22.375
  },
  "api.sensors.gps": {
    "altitude": 0.3,
    "speedKmh": 0.666719973,
    "latitude": -23.0000,
    "longitude": -40.0000,
    "link": "https://www.google.com/maps?q=-20.0000,-40.0000",
    "course": 95.87000275,
    "satellites": 7,
    "day": 4,
    "month": 6,
    "year": 2024,
    "hour": 13,
    "minute": 32,
    "second": 9,
    "age": 253,
    "hdop": 1.03,
    "trip_distance_meters": 734,
    "total_distance_meters": 0,
    "last_valid_fix": 0,
    "is_moving": false
  },
  "api.sensors.clock": {
    "day": 4,
    "month": 6,
    "year": 2024,
    "hour": 13,
    "minute": 32,
    "second": 9
  },
  "api.sensors.accelerometer": {
    "x": -0.170000002,
    "y": -9.520000458,
    "z": -2.690000057,
    "roll": 105.7783279,
    "pitch": -0.98449105,
    "x_avg": -0.167999998,
    "y_avg": -9.531999588,
    "z_avg": -2.720000029,
    "roll_avg": 105.9263382,
    "pitch_avg": -0.970974088,
    "sample_count": 10,
    "timestamp": 1746927,
    "is_moving": false,
    "move_time_pct": 0,
    "move_sample_count": 0,
    "last_eval_time": 1735127,
    "eval_sample_size": 100
  },

```

```

"api.sensors.gpio": {
  "gpio_1": false,
  "gpio_2": false,
  "gpio_3": false,
  "gpio_4": false,
  "psu_voltage": 3.329166651,
  "pulse_counter": 0,
  "pulse_counter_30s": 0,
  "rpm": 0,
  "rpm_multiplier": 1
},
"api.ble01": {
  "x": 0,
  "y": 0,
  "z": 0,
  "battery": 0,
  "timestamp": 0,
  "x_avg": 0,
  "y_avg": 0,
  "z_avg": 0,
  "sample_size": 0,
  "timestamp_avg": 0
},
"api.ble02": {
  "x": 0,
  "y": 0,
  "z": 0,
  "battery": 0,
  "timestamp": 0,
  "x_avg": 0,
  "y_avg": 0,
  "z_avg": 0,
  "sample_size": 0,
  "timestamp_avg": 0
},
"api.telemetry": {
  "source_for_ignition": 0,
  "source_for_rpm": 0,
  "min_voltage_for_ignition": 12.80000019,
  "min_voltage_for_rpm": 12.80000019,
  "engine_status": 0,
  "ignition_status": 0,
  "rpm": 0,
  "speed_kmh": 0.870440006,
  "aux1_timer_seconds": 0,
  "aux2_timer_seconds": 0,
  "ignition_timer_seconds": 0,
  "engine_timer_seconds": 0,
  "movement_timer_seconds": 0,
  "gps_variation_degrees_per_sec": 0,
  "gps_trip_distance_meters": 734,
  "gps_total_distance_meters": 0,
  "working_state": 0,
  "voltage_module": 0,
  "phnXID": 0,
  "seqdb_appointment_ct21": 0
},
"api.sensors.can": {

```

```

"packages_per_second": 0,
"age_ms": 0,
"j1939_rpm": 0,
"j1939_speed_kmh": 0,
"j1939_eng_temp_c": 0,
"j1939_eng_pressure_kpa": 0,
"j1939_ign_timer_s": 0,
"j1939_eng_timer_s": 0,
"j1939_total_vehicle_dist_m": 0,
"j1939_total_engine_h": 0,
"j1939_fuel_rate_l": 0,
"j1939_total_fuel_used_l": 0,
"obd2_rpm": 0,
"obd2_speed_kmh": 0,
"obd2_fuel_level_pct": 0,
"obd2_runtime_sec": 0,
"obd2_dist_mil_on_km": 0,
"obd2_evap_sys_pressure_pa": 0,
"obd2_barometric_pressure_kpa": 0,
"obd2_oxygen_sensor_1_volt": 0,
"obd2_oxygen_sensor_2_volt": 0,
"obd2_oxygen_sensor_3_volt": 0,
"obd2_oxygen_sensor_4_volt": 0,
"obd2_fuel_rail_pressure_kpa": 0,
"obd2_maf_air_flow_rate_gsec": 0,
"obd2_throttle_position_pct": 0,
"obd2_vehicle_mileage_km": 0,
"obd2_engine_hours_hr": 0,
"obd2_fuel_flow_rate_lhr": 0,
"obd2_engine_coolant_temp_c": 0,
"obd2_intake_air_temp_c": 0
},
"comms": {
"xvm_msg_counter": 6,
"virloc_id": "9999",
"ble_name": "100100_SMVTRACK",
"ble_packages_per_second": 0,
"ble_package_counter": 0,
"last_logtx_requested": 0,
"last_logtx_completed": 0
},
"AT+API": {
"filters": [
"device_info",
"temp",
"gps",
"clock",
"accelerometer",
"gpio",
"ble",
"telemetry",
"canbus",
"comms"
]
}
}

```

▼ Resposta do comando **AT+APIS**

```
{
  "api.device_info": {
    "boot_counter": 4,
    "device_connected": false,
    "device_connected_mtu": 0,
    "sdcard_working": true,
    "can_bus_status": false,
    "firmware_version": "TRACK HW1.3.0 FW4.4.0",
    "uart0_baudrate": 115200,
    "short_firmware_version": 440,
    "mac_address": 100100,
    "current_program": 0,
    "ble_scanner_enable": false,
    "ble_scanner_debug": false,
    "simova_angle_enable": false,
    "simova_angle_debug": false,
    "ota_initialize": 0,
    "satellite_s1c_enable": 0,
    "satellite_s1c_debug": 0,
    "api.sensors.temperature": {
      "temp_celsius": 22.375
    },
    "api.sensors.gps": {
      "altitude": 0.4,
      "speedKmh": 1.500120044,
      "latitude": -23.000000,
      "longitude": -45.000000,
      "link": "https://www.google.com/maps?q=-23.0000,-45.0000",
      "course": 258.25,
      "satellites": 8,
      "day": 4,
      "month": 6,
      "year": 2024,
      "hour": 13,
      "minute": 32,
      "second": 31,
      "age": 252,
      "hdop": 0.9,
      "trip_distance_meters": 734,
      "total_distance_meters": 0,
      "last_valid_fix": 0,
      "is_moving": false
    },
    "api.sensors.clock": {
      "day": 4,
      "month": 6,
      "year": 2024,
      "hour": 13,
      "minute": 32,
      "second": 31
    },
    "api.sensors.accelerometer": {
      "x": -0.170000002,
      "y": -9.539999962,
      "z": -2.730000019,
      "roll": 105.9691925,
      "pitch": -0.981497586,
      "x_avg": -0.180000007,
      "y_avg": -9.539999962,
      "z_avg": -2.719999987,
      "roll_avg": 105.8803329,
      "pitch_avg": -1.039680243,
      "sample_count": 10,
      "timestamp": 1769636,
      "is_moving": false,
      "move_time_pct": 0,
      "move_sample_count": 0,
      "last_eval_time": 1755127,
      "eval_sample_size": 100
    },
    "api.sensors.gpio": {
      "gpio_1": false,
      "gpio_2": false,
      "gpio_3": false,
      "gpio_4": false,
      "psu_voltage": 3.357500076,
      "pulse_counter": 0,
      "pulse_counter_30s": 0,
      "rpm": 0,
      "rpm_multiplier": 1
    },
    "api.ble01": {
      "x": 0,
      "y": 0,
      "z": 0,
      "battery": 0,
      "timestamp": 0,
      "x_avg": 0,
      "y_avg": 0,
      "z_avg": 0,
      "sample_size": 0,
      "timestamp_avg": 0
    },
    "api.ble02": {
      "x": 0,
      "y": 0,
      "z": 0,
      "battery": 0,
      "timestamp": 0,
      "x_avg": 0,
      "y_avg": 0,
      "z_avg": 0,
      "sample_size": 0,
      "timestamp_avg": 0
    },
    "api.telemetry": {
      "source_for_ignition": 0,
      "source_for_rpm": 0,
      "min_voltage_for_ignition": 12.80000019,
      "min_voltage_for_rpm": 12.80000019,
      "engine_status": 0,
      "ignition_status": 0,
      "rpm": 0,
      "speed_kmh": 1.500120044,
      "aux1_timer_seconds": 0,
      "aux2_timer_seconds": 0,
      "ignition_timer_seconds": 0,
      "engine_timer_seconds": 0,
      "movement_timer_seconds": 0,
      "gps_variation_degrees_per_sec": 0,
      "gps_trip_distance_meters": 734,
      "gps_total_distance_meters": 0,
      "working_state": 0,
      "voltage_module": 0,
      "phnXID": 0,
      "seqdb_apointment_ct21": 0
    },
    "api.sensors.can": {
      "packages_per_second": 0,
      "age_ms": 0,
      "j1939_rpm": 0,
      "j1939_speed_kmh": 0,
      "j1939_eng_temp_c": 0,
      "j1939_eng_pressure_kpa": 0,
      "j1939_ign_timer_s": 0,
      "j1939_eng_timer_s": 0,
      "j1939_total_vehicle_dist_m": 0,
      "j1939_total_engine_h": 0,
      "j1939_fuel_rate_l": 0,
      "j1939_total_fuel_used_l": 0,
      "obd2_rpm": 0,
      "obd2_speed_kmh": 0,
      "obd2_fuel_level_pct": 0,
      "obd2_runtime_sec": 0,
      "obd2_dist_mil_on_km": 0,
      "obd2_evap_sys_pressure_pa": 0,
      "obd2_barometric_pressure_kpa": 0,
      "obd2_oxygen_sensor_1_volt": 0,
      "obd2_oxygen_sensor_2_volt": 0,
      "obd2_oxygen_sensor_3_volt": 0,
      "obd2_oxygen_sensor_4_volt": 0,
      "obd2_fuel_rail_pressure_kpa": 0,
      "obd2_maf_air_flow_rate_gsec": 0,
      "obd2_throttle_position_pct": 0,
      "obd2_vehicle_mileage_km": 0,
      "obd2_engine_hours_hr": 0,
      "obd2_fuel_flow_rate_lhr": 0,
      "obd2_engine_coolant_temp_c": 0,
      "obd2_intake_air_temp_c": 0
    },
    "comms": {
      "xvm_msg_counter": 6,
      "virloc_id": "9999",
      "ble_name": "100100_S MVTRACK",
      "ble_packages_per_second": 0,
      "ble_package_counter": 0,
      "last_logtx_requested": 0,
      "last_logtx_completed": 0
    },
    "AT+API": {
      "filters": [
        "device_info",
        "temp",
        "gps",
        "clock",
        "accelerometer",
        "gpio",
        "ble",
        "telemetry",
        "canbus",
        "comms"
      ]
    }
  }
}
```

▼ Como Atualizar o Simova Track Via WI-FI (OTA).

A Função OTA (Over The Air) é uma função para atualizar o firmware do SimovaTrack utilizando uma pagina web criada pelo proprio aparelho. É necessário possuir baixado o firmware a ser atualizado em seu dispositivo (ex: "firmware 4.4.0").

Observações:

- É muito importante gravar apenas firmwares compatíveis com o SimovaTrack, **o uso de firmware errado poderá destruir o aparelho de forma irreversível.**
- Salve o firmware em uma pasta de fácil acesso (ex downloads)
- Durante o modo OTA, o simova track fará **4 beeps a cada 10s**, indicando que está no modo OTA.
- Quando um aparelho se conecta no WiFi, é tocado um som ascendente, quando desconecta, um som descendente
- Caso nenhum aparelho se conecte ao WiFi em até **5min**, o aparelho reiniciará automaticamente (**seguido de 5 longos beeps**) e retornara ao modo de operação normal
- Após iniciar a atualização de Firmware, não será possível enviar comandos via BLE.

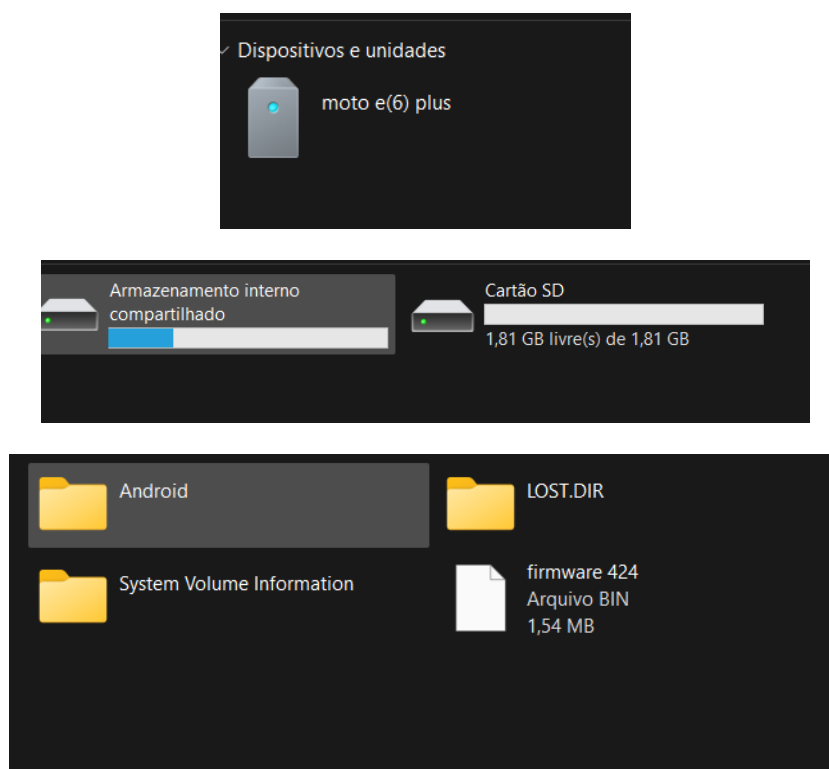
Passo a passo:

1. Como Transferir seu arquivo de firmware:

▼ Transferindo o firmware via cartão de memória

Há dois meios de passar do computador para o cartão de memória:

1. Retire o cartão de memória do celular e com um adaptador micro SD conecte em seu computador (a necessidade do adaptador micro SD varia de computador para computador).
 - > Formate o cartão SD se for necessário
 - > crie uma pasta dentro do cartão SD
 - > coloque a nova versão de firmware, quando voltar o cartão de memória para o celular, acessará o firmware por > arquivos > Cartão SD > e o nome da pasta que adicionou.
2. Com o cartão SD dentro do telefone, conecte o aparelho no computador via usb. Logo após acesse as configurações do dispositivo no explorador de arquivos.
 - > Selecione o celular
 - > Logo Após Cartão SD
 - > copie o arquivo e cole dentro do Cartão SDComo no Exemplo abaixo:

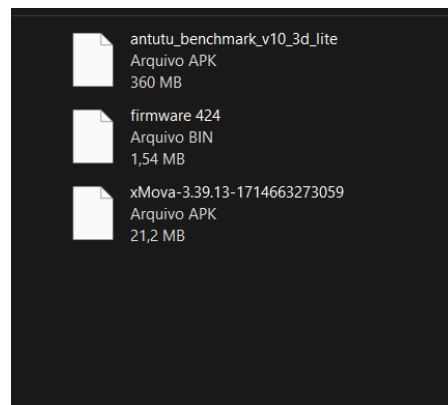
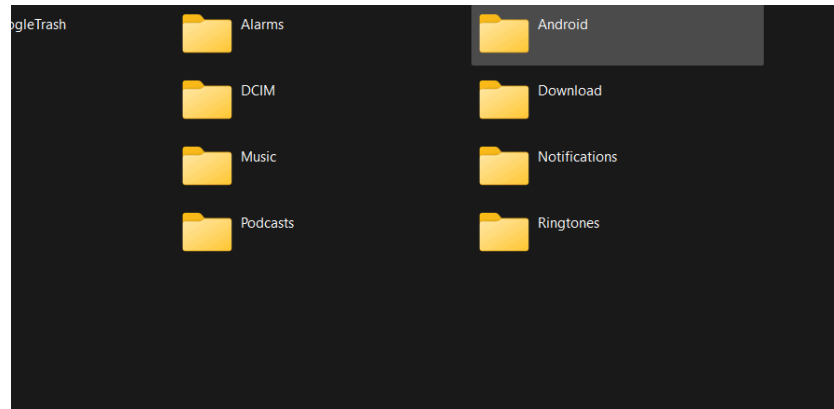
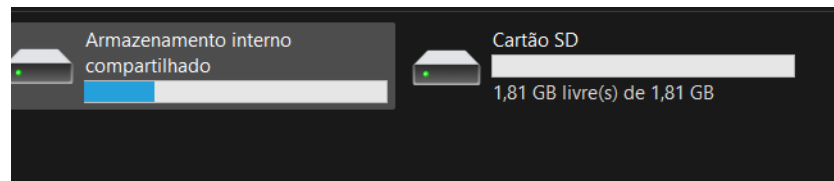


▼ Fazendo download diretamente do aparelho

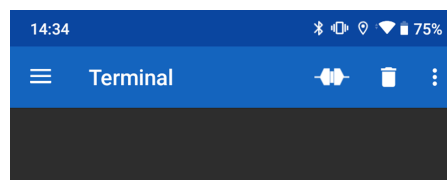
1. Caso tenha recebido o arquivo via whatsapp ou teams. basta mover o arquivo para uma pasta de fácil acesso (ex Downloads)

▼ transferindo arquivo via usb

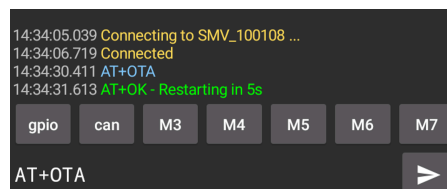
1. conecte o aparelho no computador via usb. Logo após acesse as configurações do dispositivo no explorador de arquivos.
 - > Selecione o celular
 - > copie o arquivo e cole dentro da pasta que for de mais fácil acesso (no exemplo abaixo foi adicionado na pasta downloads).Como no Exemplo abaixo:



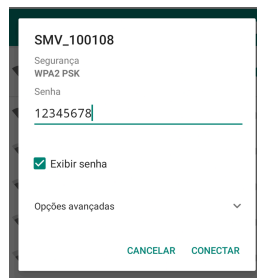
2. Conecte ao Simova Track com o terminal serial bluetooth



2. Envie o comando AT+OTA e aguarde a reinicialização do Simova Track, o mesmo fará um som diferente (com 4 beeps) indicando o modo OTA a cada 10s, e continuara fazendo este som até que saia do modo OTA.



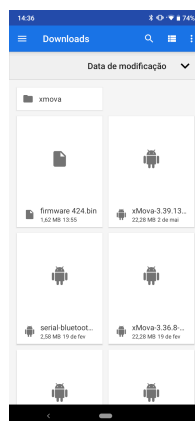
3. Abra as configurações de WI-FI e digite a senha padrão 12345678



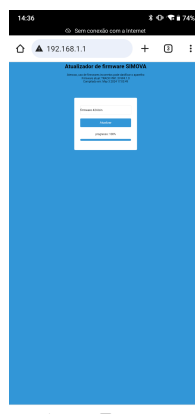
4. abra a página de configuração no navegador 192.168.1.1



5. selecione o firmware desejado.



6. Aguarde a conclusão, o SimovaTrack irá reiniciar automaticamente.



7. Para conferir a atualização, envie o comando AT+API e verifique a versão do firmware, deverá ser a mesma do arquivo recebido (parâmetro "firmware_version").

```
{
  "api.device_info": {
    "boot_counter": 4,
    "device_connected": false,
    "device_connected_mtu": 0,
    "sdcard_working": true,
    "can_bus_status": false,
    "firmware_version": "TRACK HW1.3.0 FW4.4.0",
    "uart0_baudrate": 115200,
    "short_firmware_version": 440,
  }
}
```

▼ Como selecionar um programa de sensoriamento

USRPROGRAM é uma forma de o usuário escolher como o Simova Track vai monitorar certos comportamentos em diferentes máquinas.

Programas específicos são uma função experimental!

1. Use o comando `AT+USRPROGRAM=1` para escolher um novo programa

```
enviou: AT+USRPROGRAM=1
recebeu: AT+USRPROGRAM=OK - User logic program set to 1 - HELLO_WORLD - RESTARTING IN 10s
```

2. Confira a lista de programas disponíveis

Programas disponíveis atualmente:

```
NO_PROGRAM = 0,      // Padrão, nenhum programa especial inicializado
HELLO_WORLD = 1,     // Demonstração de como usar a API e programas de usuário.
PLANTER_MS01 = 50,   // Usado em planter para contar produtividade (CT03) e tempo de trabalho (CT10), usa entradas 2,3,4
MOTONIVELADORA_MS01 = 60 // Usado em moto niveladora para atualizar WS baseado em duas entradas (3 e 4)
MOTONIVELADORA_MS02 = 61 // Usado em moto niveladora para atualizar WS baseado em dois sensores SimovaAngle
```

3. Como entender as lógicas

Dica para entender os códigos:

Utilizamos os símbolos:

```
&& para indicar (situação1) E (situação2) = ação
|| para indicar (situação1) OU (situação2) = ação
```

Utilizamos as abreviações:

```
WS = workingstate (api.telemetry.working_state)
SA = SimovaAngle (api.ble01 e api.ble02)
IGN = Ignição (api.telemetry.ignition_status)
ENG = Ignição (api.telemetry.engine_status)
RPM = Rotação por minuto (api.telemetry.rpm)
```

Resumo de lógica dos programas

▼ 0 - **NO_PROGRAM** → Nenhum programa

- Programa por padrão, nenhum programa especial é inicializado.

Comando:

- AT+USRPROGRAM=0

▼ 1 - **HELLO_WORLD** → Programa do Hello World

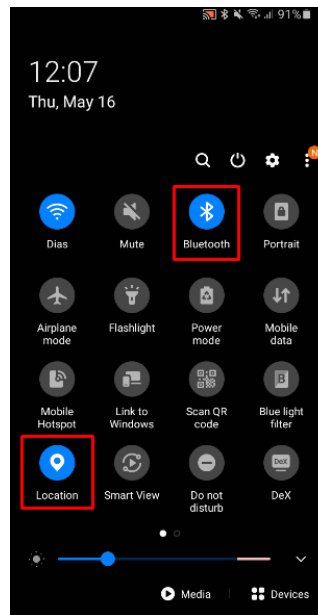
- Programa funciona hoje como um looping de mensagens de "Hello World", e se conectar via BLE o usuário consegue ver a entradas ativas no Simova Track.

▼ Como se conectar via BLE no Simova Track

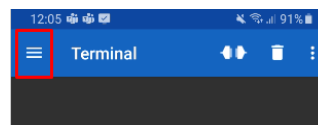
- Serial Terminal Bluetooth

1. Ative o Bluetooth e Localização

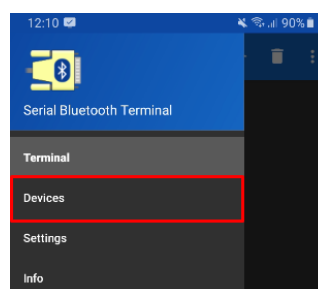
Exemplo do meu aparelho (samsung)



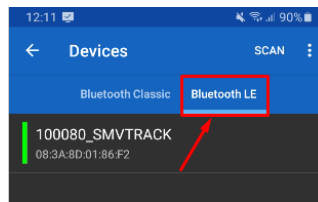
2. clique no botão (hamburger)



3. clique em "Devices"



4. Selecione "Bluetooth LE" e clique no **xxxxx_SMVTRACK**



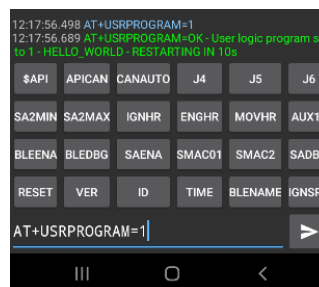
(caso não apareça o simova track, clique em "SCAN" para procurar).

5. Após conectar ao Simova Track ira aparecer essa mensagem no aplicativo

```
12:13:52.040 Connecting to 100080_SMVTRACK ...
12:13:53.035 Connected
12:13:53.120 PROGRAM_LOOP: Hello world BLE - GPIO: 1:0, 2:0, 3:0, 4:0
```

▼ Como utilizar a função de ver entradas ativas (AT+USRPROGRAM=1)

1. Mande o comando AT+USRPROGRAM=1

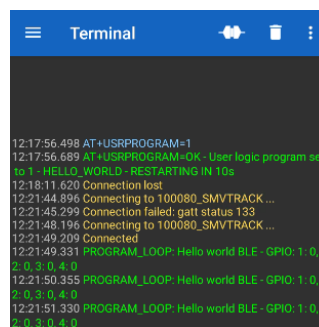


Após mandar esse comando o Simova Track irá reiniciar em 10 segundos, reiniciando conecte-se novamente (**IMPORTANTE**)

2. Após se conectar novamente (botão "Connect")



3. Função ativada



Já estará mostrando as entradas ativas

4. Descrição das entradas

Descrição:

PROGRAM_LOOP: Hello world BLE → Looping de mensagem somente via Bluetooth

GPIO:

1 → Entrada 1 (1:0 → ZERO significa que não esta ativa, 1:1 → UM significa que esta ativa)

2 → Entrada 2 (2:0 → ZERO significa que não esta ativa, 2:1 → UM significa que esta ativa)

3 → Entrada 3 (3:0 → ZERO significa que não esta ativa, 3:1 → UM significa que esta a

tiva)
 4 → Entrada 2 (4:0 → ZERO significa que não esta ativa, 4:1 → UM significa que esta a tiva)
 Exemplo acima:
 PROGRAM_LOOP: Hello world BLE - GPIO: 1:0, 2:0, 3:0, 4:0
 Explicação:
 Todas entradas estão ZERO ou seja desativadas.

- AT+USRPROGRAM=1

▼ 50 - **PLANTER_MS01** → Programa para contar produtividade e Tempo (entradas 2,3,4)

- Programa para monitorar produtividade e Tempo, usam 2 auxiliares de tempo (entradas) para monitorar.

Exemplo de uso

- PLANTER: Monitora a produtividade e o tempo durante a operação de plantio.

Como Funciona

- O programa monitora a produtividade e o tempo durante a operação de plantio usando sensores conectados às entradas 2, 3 e 4.
- Utiliza dois auxiliares de tempo para monitorar a produtividade e o tempo decorrido.

Comando:

- AT+USRPROGRAM=50

▼ 60 - **DOISIMPLEMENTOS_DIGITAL** → Usado em moto niveladora para atualizar WS baseado em duas entradas (3 e 4)

- O programa verifica o funcionamento da lâmina e do escarificador na moto niveladora, usando sensores para determinar o modo de operação.

Comando

- AT+USRPROGRAM=60

Exemplo de uso

- MOTONIVELADORA: Monitora o funcionamento da lâmina e do escarificador para determinar o modo de operação

Como funciona

- O programa monitora o funcionamento da lâmina e do escarificador usando sensores conectados às entradas 3 e 4.
- Atualiza o Working State (WS) com base no estado dos sensores e na lógica especificada.

Lógica

- WS 0 = Ambos os sensores estão desligados
- WS 1 = Apenas o sensor da lâmina está ativado
- WS 2 = Apenas o sensor do escarificador está ativado
- WS 3 = Ambos os sensores estão ativados

Lógica

- AT+USRPROGRAM=60

▼ 61 - **MOTONIVELADORA_MS02** → Programa indicado para uso em sensoriamento de um ou dois implementos por ângulo de inclinação.

Exemplo de uso

- MOTONIVELADORA: Monitorar implementos de lâmina e escarificador (dentro do angulo minimo e maximo) com saída de resultado em working-state
- CAMINHÃO BASCULANTE: Monitorar a caçamba para identificar alto/baixo, com saída através de working state.

Como funciona:

- Atua o WS de acordo o angulo atual do sensor, em relação aos angulos maximos e minimos permitidos.
 - IGN = Fonte de ignição selecionada | ON = ligado, OFF = desligado
 - SA01 = SimovaAngle01 | ON = dentro do angulo, OFF = fora do angulo ou não conectado
 - SA02 = SimovaAngle02 | ON = dentro do angulo, OFF = fora do angulo ou não conectado
- Lógica
 - WS 0 = (IGN OFF) || (SA01 OFF) && (SA02 OFF)
 - WS 1 = (IGN ON) && (SA01 ON) && (SA02 OFF)
 - WS 2 = (IGN ON) && (SA01 OFF) && (SA02 ON)
 - WS 3 = (IGN ON) && (SA01 ON) && (SA02 ON)

Comando:

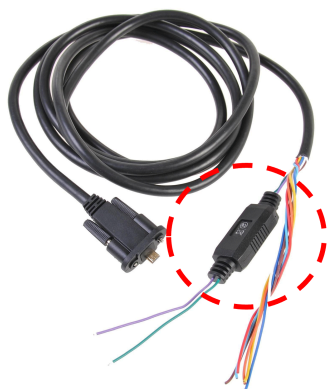
- AT+USRPROGRAM=61

▼ Como ativar suporte a antena satelital Smartone C

A SmartOneC é uma antena compacta que pode ser utilizada para transmitir pequenas mensagens.

▼ Como ligar a antena

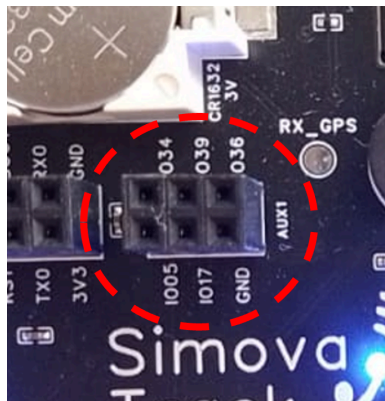
A Antena é alimentada por 5 Volts, mas muitos dos cabos já vem com um conversor incluso, caso o conversor (marcado pelo circulo vermelho) não esteja presente, você precisa instalar uma fonte 5V (como por exemplo um carregador usb simova).



Conecte os fios da antena com muito cuidado.

Cor	Função	Onde conectar
Marrom	Setup/Handshake	NEGATIVO
Vermelho	RX	AUX1 IO17
Laranja	TX	AUX1 IO05
Azul claro	Contato seco 2	N/C
Branco	Contato seco 1	N/C
Amarelo	Soft power down	N/C
Verde	Negativo	Negativo

Azul	Negativo	N/C
Violeta	Positivo 5V	com conversor: POSITIVO sem conversor: CONVERSOR



Referência dos pino IO05 e IO07

▼ Como ativar suporte no firmware

1. Use o comando `AT+S1CENABLE=x` para escolher um modo de operação

```
# Desligando
enviou: AT+S1CENABLE
recebeu: AT+S1CENABLE=0
# Ligando
enviou: TX: AT+S1CENABLE=1
recebeu: AT+S1CENABLE=OK
```

▼ Como enviar Payload do Tipo RAW (definido pelo usuário)

1. Use o comando `AT+S1CRAW=dataHex` para enviar imediatamente o evento de número não pré-programado

```
# Enviando
enviou: AT+S1CRAW=000102030405060708
recebeu: AT+S1CRAW=OK
```

▼ Como enviar Evento por Gatilhos/Tempo

1. Use o comando `AT+S1CEVENTEDIT=event_number,enabled,trigByCondition,trigByTime,timedInterval` para enviar Evento por Condição/Tempo

OBS: Gatilhos são por exemplo: event 06 = Motor Ligou (gatilho será quando motor ligar por exemplo)

```
# Enviando
enviou: AT+S1CEVENTEDIT=14,1,0,0,0
recebeu: AT+S1CEVENTEDIT=OK,Event 14 updated successfully
```

▼ Como enviar Evento imediato

1. Use o comando `AT+S1CEVT=event` para enviar o evento agora

```
# Enviando
enviou: AT+S1CEVT=14
recebeu: AT+S1CEVT=OK
```

▼ Eventos Programados - Protocolo de Mensagem de Evento

A S1C permite transmitir até 9 bytes, sendo 1 byte para identificação do evento e 8 bytes de carga útil. O **primeiro byte** é sempre a identificação que contém o evento e o tipo de mensagem.

▼ Exemplo de Mensagem

0x1A655176CD776300A9

- **1A** = Evento e tipo: 00011010 → [000110 evento 06] + [10 → tipo: 2]

▼ Eventos Disponíveis

Evento	Parâmetros	Condições de Disparo	Descrição
Evento 00	N/A	N/A	Reservado para uso futuro
Evento 01	N/A	N/A	Reservado para uso futuro
Evento 02	N/A	N/A	Reservado para uso futuro
Evento 03	N/A	N/A	Reservado para uso futuro
Evento 04	N/A	N/A	Reservado para uso futuro
Evento 05	N/A	N/A	Reservado para uso futuro
Evento 06	Latitude, Longitude, Horímetro	if (motorLigouAgora) && (ultimaTransmissao > 5min)	Motor Ligou (Localização + horímetro)
Evento 07	Latitude, Longitude, Horímetro	if (motorDesligouAgora) && (ultimaTransmissao > 5min)	Motor Desligou (Localização + horímetro)
Evento 08	Latitude, Longitude, Horímetro	if (motorLigado) && (ultimaTransmissao > 30min)	Idle com motor ligado
Evento 09	Latitude, Longitude, Horímetro	if (motorDesligado) && (ultimaTransmissao > 60min)	Idle com motor desligado
Evento 14	Latitude, Longitude, Apontamento	if (AtividadeAgora != ultimaAtividade) && (ultimaTransmissao > 2min)	Localização e Atividade Atual (SimovaApps)

▼ Mais Detalhes (Todos Eventos)

Evento 06: Motor Ligou (Localização + horímetro do motor ligado)

- **Condições de disparo:**
 - if (motorLigouAgora) && (ultimaTransmissao > 5min)

- **Payload Exemplo:**

0x1A655176CD776300A9

- **1A** = Evento e tipo: 00011010 → [000110 evento 06] + [10 → tipo: 2]
- **655176** = Latitude: (23.6001)
- **CD7763** = Longitude: (-45.34557)
- **00A9** = Horímetro: (169h)

Evento 07: Motor Desligou (Localização + horímetro do motor desligado)

- **Condições de disparo:**
 - if (motorDesligouAgora) && (ultimaTransmissao > 5min)

- **Payload Exemplo:**

0x1E655176CD776300A9

- **1E** = Evento e tipo: 00011110 → [000111 evento 07] + [10 → tipo: 2]
- **655176** = Latitude: (23.6001)
- **CD7763** = Longitude: (-45.34557)

- **00A9** = Horímetro: (169h)

Evento 08: Idle com Motor Ligado (*Localização + horímetro do motor ligado*)

- **Condições de disparo:**

- `if (motorLigado) && (ultimaTransmissao > 30min)`

- **Payload Exemplo:**

```
0x22655176CD776300A9
```

- **22** = Evento e tipo: `00100010` → [001000 evento 08] + [10 → tipo: 2]
- **655176** = Latitude: (23.6001)
- **CD7763** = Longitude: (-45.34557)
- **00A9** = Horímetro: (169h)

Evento 09: Idle com Motor Desligado (*Localização + horímetro do motor desligado*)

- **Condições de disparo:**

- `if (motorDesligado) && (ultimaTransmissao > 60min)`

- **Payload Exemplo:**

```
0x26655176CD776300A9
```

- **26** = Evento e tipo: `00100110` → [001001 evento 09] + [10 → tipo: 2]
- **655176** = Latitude: (23.6001)
- **CD7763** = Longitude: (-45.34557)
- **00A9** = Horímetro: (169h)

Evento 14: Localização e Atividade Atual (*Requer conexão com SimovaApps*)

- **Condições de disparo:**

- `if (AtividadeAgora != ultimaAtividade) && (ultimaTransmissao > 2min)`

- **Payload Exemplo:**

```
0x3A655176CD776300A9
```

- **3A** = Evento e tipo: `00111010` → [001110 evento 14] + [10 → tipo: 2]
- **655176** = Latitude: (23.6001)
- **CD7763** = Longitude: (-45.34557)
- **00E2** = Apontamento: (226)

Observações:

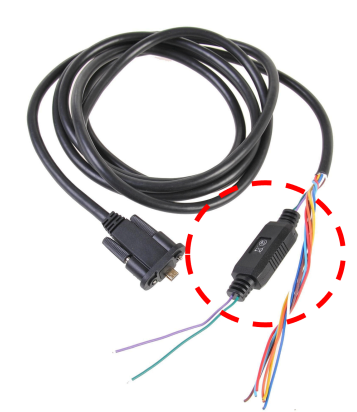
- A formatação dos eventos segue um padrão de 9 bytes.
- Os primeiros dois bits do primeiro byte indicam o **tipo de mensagem** e os seis bits restantes indicam o **código do evento**.
- Cada evento é acompanhado de um **payload** que carrega informações como **localização**, **horímetro** e outros dados relevantes.

▼ Como ativar suporte a antena satelital STX3 (**futuro**)

A STX3 é uma antena compacta e que pode ser utilizada para transmitir pequenas mensagens.

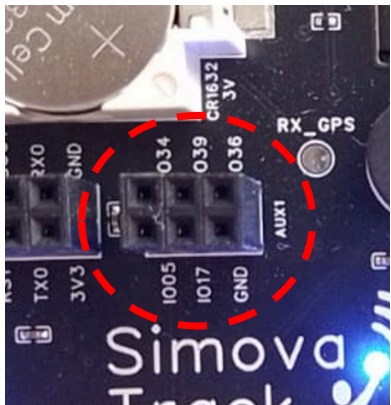
- ▼ Como ligar a antena (**OBSOLETO- PRECISA ATUALIZAR**)

A Antena é alimentada por 5 Volts, mas muitos dos cabos já vem com um conversor incluso, caso o conversor (marcado pelo circulo vermelho) não esteja presente, você precisa instalar uma fonte 5V (como por exemplo um carregador usb simova).



Conecte os fios da antena com muito cuidado.

Cor	Função	Onde conectar
Marrom	Setup/Handshake	NEGATIVO
Vermelho	RX	AUX1 IO17
Laranja	TX	AUX1 IO05
Azul claro	Contato seco 2	N/C
Branco	Contato seco 1	N/C
Amarelo	Soft power down	N/C
Verde	Negativo	Negativo
Azul	Negativo	N/C
Violeta	Positivo 5V	com conversor: POSITIVO sem conversor: CONVERSOR



Referência dos pino IO05 e IO07

▼ Como ativar suporte no firmware (OBSOLETO- PRECISA ATUALIZAR)

1. Use o comando `AT+SATS1CENA=x` para escolher um modo de operação

```
# Desligando
enviou: AT+SATS1CENA=0
recebeu: AT+SATS1CENA=OK - RESTARTING IN 10s

# Ligando
```



```
enviou: AT+SATS1CENA=4
recebeu: AT+SATS1CENA=OK - RESTARTING IN 10s
```

2. Confira a lista de programas disponíveis

Programs and Corresponding Messages

The user can change the program by sending `AT+SATS1CENA=x`, where `x` is the program number. Below is a brief explanation of what each value means and what the program will transmit:

1. **Program 1**

- **Description:** Continuously retrieves device information to check if the device is working properly, can be used to debug.
- **Messages requested:**
 - `get\_esn(esn)`: Retrieves the Electronic Serial Number (ESN).
 - `get\_version(major, minor, patch)`: Retrieves the firmware version.
- **Messages Transmitted:**
 - none
- **Interval for Next Call:** 5 seconds

2. **Program 2**

- **Description:** Sends a raw SMV message with vehicle and sensor data.
- **Messages Transmitted:**
 - `send\_raw\_smv\_message\_0()`: Sends a message containing:
 - Byte 0: `type\_and\_id`
 - Byte 1: `packed states`
 - Byte 2: `seqDbAppointments (high byte)`
 - Byte 3: `seqDbAppointments (low byte)`
 - Byte 4: `horimeter (high byte)`
 - Byte 5: `horimeter (mid byte)`
 - Byte 6: `horimeter (low nibble) / powerSupply (high nibble)`
 - Byte 7: `powerSupply (low byte)`
 - Byte 8: `vehicleSpeed`
- **Interval for Next Call:** 20 minutes

3. **Program 3**

- **Description:** Sends a raw SMV message with telemetry and sensor data.
- **Messages Transmitted:**
 - `send\_raw\_smv\_message\_11()`: Sends a message containing:
 - Byte 0: `type\_and\_id`
 - Byte 1: `packed states`
 - Byte 2: `Telemetry RPM`
 - Byte 3: `Telemetry Speed`
 - Byte 4: `Device Temperature`
 - Byte 5: `Device Voltage`
 - Byte 6: `Working State`
 - Byte 7: `SeqDB CT21 (high byte)`
 - Byte 8: `SeqDB CT21 (low byte)`
- **Interval for Next Call:** 20 minutes

4. **Program 4**

- **Description:** Sends a raw SMV message with geolocation data.
- **Messages Transmitted:**
 - `send\_raw\_smv\_message\_12()`: Sends a message containing:
 - Byte 0: `type\_and\_id`
 - Byte 1: `latitude (high byte)`
 - Byte 2: `latitude (mid byte)`
 - Byte 3: `latitude (low byte)`

- Byte 4: 'longitude (high byte)'
- Byte 5: 'longitude (mid byte)'
- Byte 6: 'longitude (low byte)'
- Byte 7: 'altitude (high byte)'
- Byte 8: 'altitude (low byte)'
- **Interval for Next Call:** 20 minutes

5. **Program 5**

- **Description:** Sends a raw SMV message with activity data.
- **Messages Transmitted:**
 - 'send_raw_smv_message_13()': Sends a message containing:
 - Byte 0: 'type_and_id'
 - Byte 1: 'large_horimeter - ignition (high byte)'
 - Byte 2: 'large_horimeter - ignition (mid byte)'
 - Byte 3: 'large_horimeter - ignition (low byte)'
 - Byte 4: 'large_horimeter - engine (high byte)'
 - Byte 5: 'large_horimeter - engine (mid byte)'
 - Byte 6: 'large_horimeter - engine (low byte)'
 - Byte 7: 'RFU (high byte)'
 - Byte 8: 'RFU (low byte)'
- **Interval for Next Call:** 20 minutes

6. **Program 6**

- **Description:** Sends a raw SMV message with geolocation and sequence database data.
- **Messages Transmitted:**
 - 'send_raw_smv_message_14()': Sends a message containing:
 - Byte 0: 'type_and_id'
 - Byte 1: 'latitude (high byte)'
 - Byte 2: 'latitude (mid byte)'
 - Byte 3: 'latitude (low byte)'
 - Byte 4: 'longitude (high byte)'
 - Byte 5: 'longitude (mid byte)'
 - Byte 6: 'longitude (low byte)'
 - Byte 7: 'SeqDB CT21 (high byte)'
 - Byte 8: 'SeqDB CT21 (low byte)'
- **Interval for Next Call:** 20 minutes

7. **Program 7**

- **Description:** Sends a raw SMV message with device information.
- **Messages Transmitted:**
 - 'send_raw_smv_message_20()': Sends a message containing:
 - Byte 0: 'type_and_id'
 - Byte 1: 'track_uuid (part 1)'
 - Byte 2: 'track_uuid (part 2)'
 - Byte 3: 'track_uuid (part 3)'
 - Byte 4: 'track_uuid (part 4)'
 - Byte 5: 'track_uuid (part 5)'
 - Byte 6: 'track_uuid (part 6)'
 - Byte 7: 'device_fw_ver (high byte)'
 - Byte 8: 'device_fw_ver (low byte)'
- **Interval for Next Call:** 20 minutes

▼ Como ativar sensoriamento

O SimovaTrack suporta sensoriamento de diversos equipamentos através dos dados da API (entradas, acelerometro, CAN, SimovaAngle, etc). Para selecionar um programa, utilize o comando `AT+USRPGRAM=xx` é uma forma de o usuário escolher como o Simova Track vai monitorar certos comportamentos em diferentes máquinas.

1. Use o comando `AT+USRPGRAM=1` para escolher um novo programa

enviou: AT+USRPROGRAM=1

recebeu: AT+USRPROGRAM=OK - User logic program set to 1 - HELLO_WORLD - RESTARTING IN 10s

2. Como entender as lógicas

Dica para entender os códigos:

Utilizamos os símbolos:

&& para indicar (situação1) E (situação2) = ação

|| para indicar (situação1) OU (situação2) = ação

Utilizamos as abreviações:

WS = workingstate (api.telemetry.working_state)

SA = SimovaAngle (api.ble01 e api.ble02)

IGN = Ignição (api.telemetry.ignition_status)

ENG = Ignição (api.telemetry.engine_status)

RPM = Rotação por minuto (api.telemetry.rpm)

Resumo de lógica dos programas

▼ **AT+USRPROGRAM=0** → Nenhum programa

▼ **AT+USRPROGRAM=61** - **MOTONIVELADORA ou BASCULANTE COM SimovaAngle** → Monitorar o ângulo do sensor e a velocidade para determinar o estado de trabalho (working_state).

Atua o working_state de acordo com o ângulo atual dos sensores, em relação aos ângulos máximos e mínimos permitidos.

- **IGN:** Fonte de ignição selecionada | **ON** = ligado, **OFF** = desligado.
 - **SA01:** SimovaAngle01 | **ON** = dentro do ângulo, **OFF** = fora do ângulo ou não conectado.
 - **SA02:** SimovaAngle02 | **ON** = dentro do ângulo, **OFF** = fora do ângulo ou não conectado.
 - **Lógica:**
 - WS 0 = (IGN OFF) || (SA01 OFF) && (SA02 OFF).
 - WS 1 = (IGN ON) && (SA01 ON) && (SA02 OFF).
 - WS 2 = (IGN ON) && (SA01 OFF) && (SA02 ON).
 - WS 3 = (IGN ON) && (SA01 ON) && (SA02 ON)
 - Configuração do sensor:
 - Configure o MAC dos Simova Angle (caso não tenha configurado)
 - Leia atentamente o guia de instalação do SimovaAngle e garanta que o mesmo esteja funcionando e sendo lido na API.
 - Fixe o SimovaTrack e o SimovaAngle e envie o comando AT+SACALIB
 - Configure o ângulo mínimo e máximo usando **AT+SAANG01MIN=xx.xx** | **AT+SAANG01MAX=xx.xx** | **AT+SAANG02MIN=xx.xx** | **AT+SAANG02MAX=xx.xx**
 - Confira o ângulo atual usando **AT+SADBGENA=1**
Lembrando que utilizando **AT+SADBGENA=1** se ativa o debug e **AT+SADBGENA=0** para desativar.
 - ▼ **AT+USRPROGRAM=100** - **MOTONIVELADORA** → Monitorar a presença do sensor e a velocidade para determinar o estado de trabalho (working_state).
- Atua o working_state de acordo com a presença do sensor e a velocidade do veículo.
- **Sensor de Presença:** Indica se o implemento está ativo (entrada 02 - fio azul).
 - **Velocidade:** Velocidade do veículo em km/h.

Lógica:

- WS 0 = Sensor inativo ou velocidade ≤ 1 km/h (após timeout produtivo de 59 segundos).
- WS 2 = Sensor ativo e velocidade > 1 km/h.

▼ **AT+USRPROGRAM=101** - BASCULANTE → Monitorar a presença do sensor e a velocidade para determinar o estado de trabalho (working_state).

Atua o working_state de acordo com o estado do motor e a presença do sensor.

- **Sensor de Presença:** Indica se a caçamba está baixa (entrada 02 - fio azul).
- **Motor:** Estado do motor (ligado/desligado).
- **Velocidade:** Velocidade do veículo em km/h.

Lógica:

- WS 0 = Motor desligado.
- WS 0 = Motor ligado e caçamba baixa (sensor ativo) com velocidade ≤ 1 km/h.
- WS 2 = Motor ligado e caçamba baixa (sensor ativo) com velocidade > 1 km/h.

▼ **AT+USRPROGRAM=102** - CAMINHAO PIPA → Monitorar botões e válvula de retorno para determinar o estado de trabalho (working_state).

Atua o working_state de acordo com a presença dos botões e a válvula de retorno, além da velocidade do veículo.

- **Botão 01:** Indica se o botão 01 está pressionado (entrada 03).
- **Botão 02:** Indica se o botão 02 está pressionado (entrada 04).
- **Válvula de Retorno:** Indica se a válvula de retorno está ativa (entrada 02).
- **Velocidade:** Velocidade do veículo em km/h.

• **Lógica:**

- WS 0 = Nenhuma condição atendida ou após timeout produtivo de 59 segundos.
- WS 1 = Botão 01 ou Botão 02 pressionado e velocidade ≥ 1 km/h.
- WS 2 = Válvula de retorno ativa e velocidade < 1 km/h.

▼ **AT+USRPROGRAM=103** - TRATOR DE ESTEIRA → Monitorar o acelerômetro e a velocidade para determinar o estado de trabalho (working_state).

Atua o working_state de acordo com os dados do acelerômetro e a velocidade do veículo.

- **Acelerômetro:** Monitora os eixos X, Y e Z para detectar movimento.
- **Velocidade:** Velocidade do veículo em km/h.

• **Lógica:**

- WS 0 = Nenhuma condição atendida ou após timeout produtivo de 59 segundos.
- WS 1 = Motor ligado e acelerômetro detecta movimento.
- WS 3 = Motor ligado e (acelerômetro detecta movimento ou velocidade $>$ velocidade mínima).

• **Configuração de sensibilidade:**

Não se esqueça de configurar a sensibilidade!!!

- AT+USR103TRESHMIN=0.4 → Força em metros por segundo quadrado
- AT+USR103SPEEDMIN=10 → Velocidade limite para WS 3

▼ **AT+USRPROGRAM=104** - PLANTER → Monitorar o sensor de presença, estado do motor e velocidade do veículo para determinar o estado de trabalho (working_state).

Monitora somente o Plantio, se a maquina está ligada trabalhando e plantando

Verificar Sensor e Velocidade:

- WS 0 = Motor estiver desligado, definir working_state como 0.

- WS 1 = Se o motor estiver ligado e uma das entradas (2,3,4) estiverem ativas (working_state = 1). Podendo uma das entrada monitorar o braço da maquina e uma monitorar o plantio, e uma para monitorar a irrigação das plantas.

▼ **AT+USRPROGRAM=105** - CEMAK ROBOPLANT → Manual de Operação v4

 CEMAK ROBOPLANT - Manual de Operação v4

REQUISITOS CRÍTICOS

- **CRÍTICO:** A detecção de MOTOR LIGADO deve estar corretamente configurada
- **CRÍTICO:** A detecção de IGNIÇÃO deve estar corretamente configurada
- O programa não funcionará corretamente se estas configurações estiverem erradas!

Instalação

1. Conexões dos Sensores

- Fio AZUL → GPIO2 (sensor de plantio 1)
- Fio ROXO → GPIO3 (sensor de plantio 2)
- Ambos os fios devem ser conectados aos sensores de plantio da máquina

2. Ativação

- Envie o comando: **AT+USRPROGRAM=105**
- Aguarde a mensagem "Sistema CEMAK ROBOPLANT iniciado"

Configurações do Sistema

- **Timeout de inatividade: 1 minuto** (corrigido!)
- Debounce dos sensores: 2 segundos
- Marco de contagem: 50 árvores
- Backup automático: 60 segundos

Funcionamento

Sistema de Contagem

- Conta plantio quando sensores mudam de TRUE para FALSE
- Requer:
 1. Motor ligado
 2. Ignição ligada
 3. 2 segundos entre plantios (debounce)

Estados e Contadores

1. Estado de Trabalho (Working State)

- 1 = Em operação
- 0 = Parado
- **Desativa após 1 minuto sem plantio** (corrigido!)
- Desativa imediatamente se motor ou ignição desligarem

2. Contador de Árvores (aux1_timer_seconds)

- Conta total de árvores plantadas
- Incrementa com cada plantio válido
- Salvo a cada 60 segundos

3. Contador de Tempo Produtivo (aux2_timer_seconds)

- Registra tempo efetivo de trabalho
- Conta apenas durante working state = 1

- Para após 1 minuto sem plantio
- Salvo a cada 60 segundos

Sistema de Avisos

- Ignição ON: "Sistema CEMAK ROBOPLANT iniciado"
- Motor ON: "Motor ligado. Total de árvores plantadas: X"
- Motor OFF: "Motor desligado. Total de árvores plantadas: X"
- Marco: "Atingido marco de X árvores plantadas"

Paradas do Sistema

1. Motor desligado (parada imediata)
2. 1 minuto sem plantio (timeout)
3. Ignição desligada (parada imediata)

Sistema de Backup

- Backup automático a cada 60 segundos
- Recuperação automática ao religar
- Armazena:
 - Total de árvores plantadas
 - Tempo total produtivo

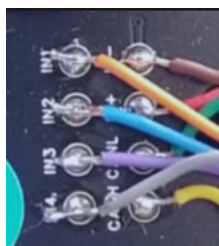
Verificação de Funcionamento

1. Ligue a ignição → mensagem inicial
2. Ligue o motor → mensagem com total
3. Faça um plantio → working state = 1
4. Pare por 1 minuto → working state = 0
5. Desligue motor → mensagem com total

▼ Técnicas de sensoriamento

▼ Conhecendo as entradas do Simova Track

▼ Características técnicas



O Simova Track possui 4 entradas, com as seguintes características:

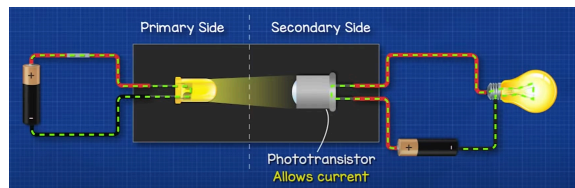
- Tensão de entrada 09 - 36v
- Fusível resetável.
- Resistor limitador.
- Proteção contra sobretensão.
- opto-acoplador infravermelho.

Em situações normais, cada entrada ativa irá drenar do fio sensoriado:

- 1.09mA em 24 V

- 0.50mA em 12 V

▼ Importância do Opto Acoplador



O opto acoplador é um componente que protege o sinal sensoriado e o processador ao inibir o contato elétrico direto, funciona da seguinte forma:

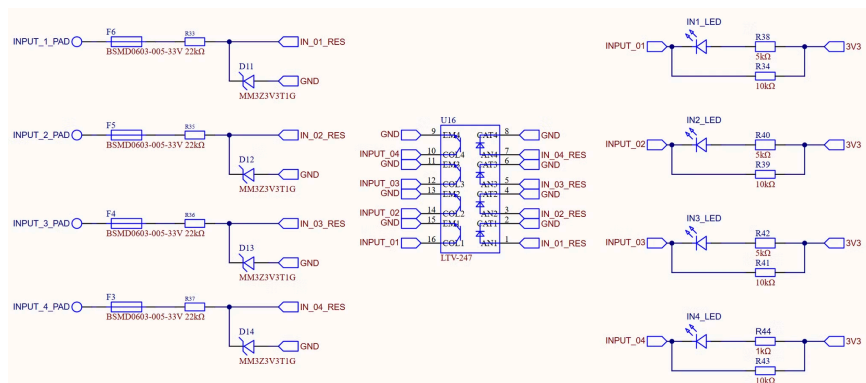
1. Um sinal elétrico acende um LED dentro do opto acoplador.
2. A luz do LED atravessa um espaço vazio.
3. A luz atinge um fototransistor, que conduz corrente elétrica.
4. O LED e o fototransistor estão eletricamente isolados, protegendo contra choques e interferências.

Benefícios:

- **Proteção:** Isola circuitos de alta e baixa tensão.
- **Redução de Ruído:** Minimiza interferências eletromagnéticas.
- **Segurança:** Essencial em aplicações críticas, protege o sinal sensoriado em caso de panes ou curto-circuitos, garantindo que não haverá interferência em hipótese alguma.

▼ Circuito elétrico e referencias técnicas

Abaixo temos o circuito elétrico do SimovaTrack HW 1.1.0



▼ Como escolher um Sensor PNP ou NPN?

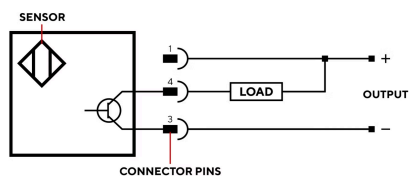
Escolha entre NPN e PNP

- **NPN:** (INCOMPATIVEL) requer o uso de um relê para inversão de polaridade para uso com o Simova Track
- **PNP:** (COMPATIVEL) Pode ser conectado diretamente ao Simova Track.

NPN (Sinking - NEGATIVO):

- Quando ativado, o sensor conecta a saída ao terra (0V).
- A corrente flui do dispositivo de controle para o sensor.
- **Corrente:** Coletor (C) → Emissor (E)
- Exemplo de conexão:

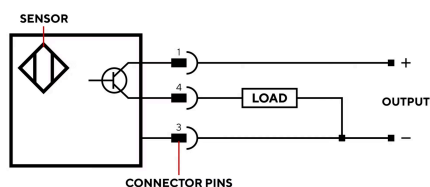
NPN 3-WIRE STANDARD



PNP (Sourcing - POSITIVO):

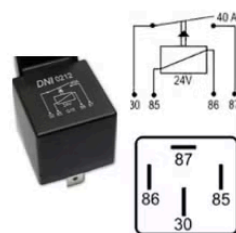
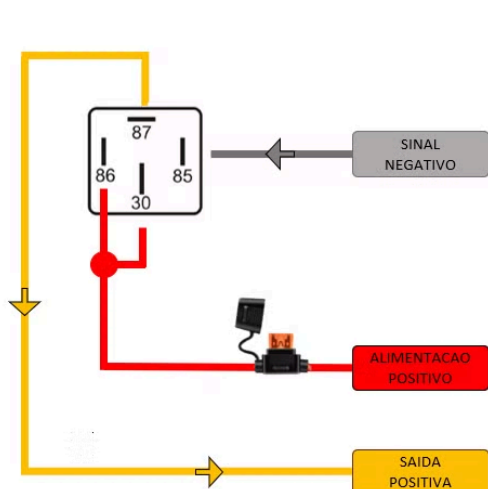
- Quando ativado, o sensor conecta a saída à tensão positiva (Vcc).
- A corrente flui do sensor para o dispositivo de controle.
- **Corrente:** Emissor (E) → Coletor (C)
- Exemplo de conexão:

PNP 3-WIRE STANDARD



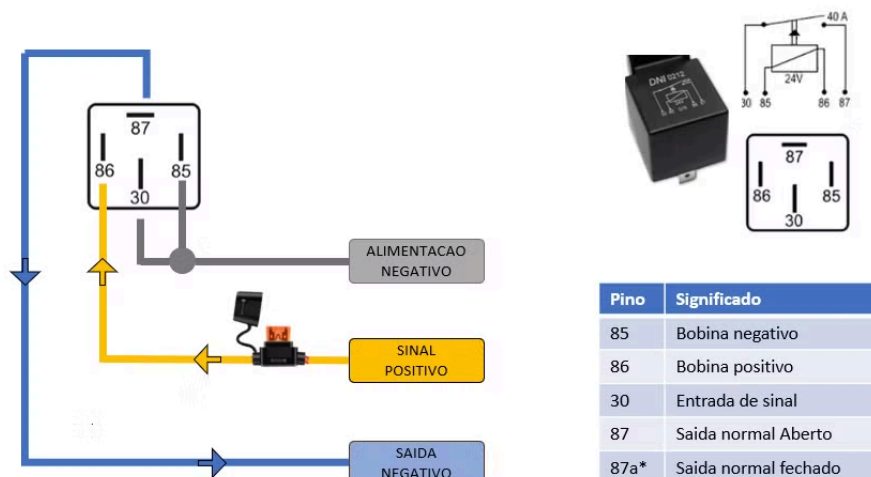
▼ Como inverter o sinal caso necessário:

- ▼ Inversão de sinal (-) para (+) com relê auxiliar



Pino	Significado
85	Bobina negativo
86	Bobina positivo
30	Entrada de sinal
87	Saída normal Aberto
87a*	Saída normal fechado

- ▼ Inversão de sinal (+) para (-) com relê auxiliar

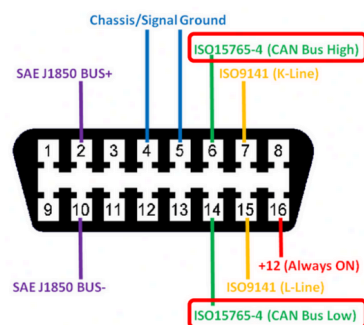


▼ Exemplos de instalação

▼ como montar cabo obd2

1. Instalação do Simova Track em carro sem a rede CAN:

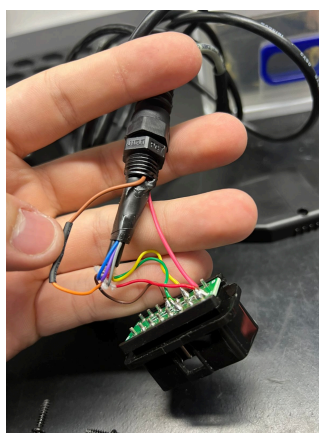
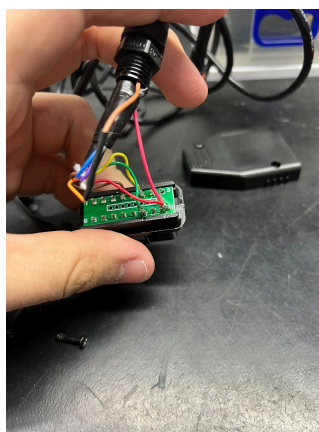
Realize a conexão de um botão no conector OBD2 do Simova Track, como no exemplo abaixo:



cor	pino OBD2
vermelho	16
verde	14
amarelo	6
marrom	4

Se for necessário realizar a ligação de um botão (ex simulando pós chave):

cor	ligação
vermelho(botão)	pino 16(OBD2)
marrom(botão)	laranja(track)



▼ HB20 2015 - OBD2 CAN

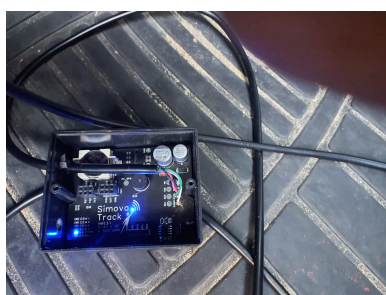
1. Instalação do Simova Track em carro com a rede CAN:
2. Localize a porta OBD2 em seu veículo:



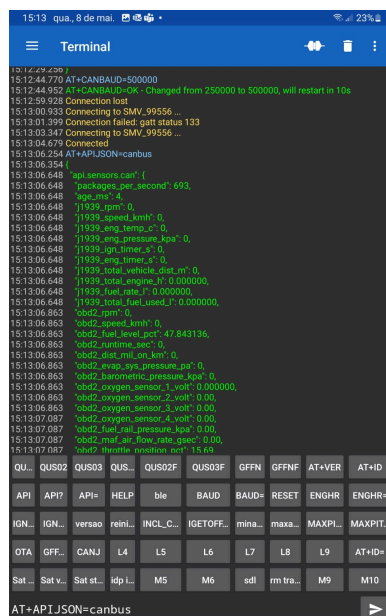
3. No caso do Hb20 está localizado nessa região próximo a embreagem:



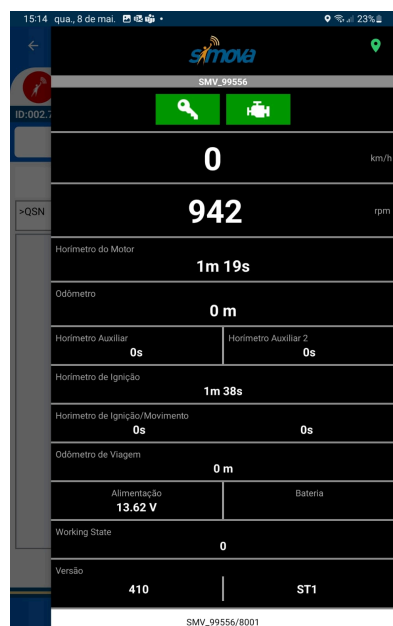
4. Ao conectar o Simova track a porta OBD2 ele deve acender suas luzes(ligar):



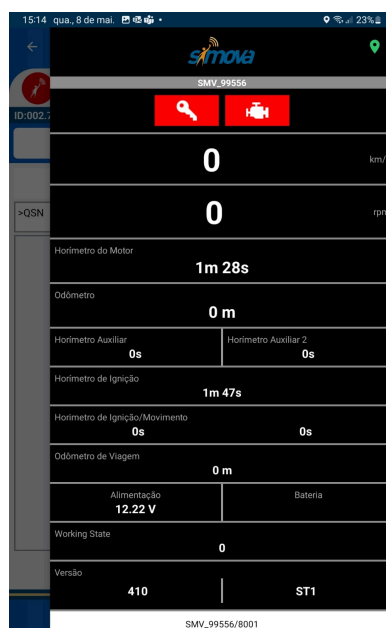
5. Feito isso realize os comandos no Simova Track, como no exemplo abaixo:



- Guia do instalador: Simova Track



7. exemplo de quando o motor está desligado.



▼ Celta LS 2013 - Ignição/RPM via tensão de bateria

O Celta LS 2013 não tem OBD2 CAN, apenas OBD2 K line, usamos o conector OBD2 para facilitar a instalação e evitar o corte de fios.

1. Localize a porta OBD2 de seu veículo e instale o Simova Track:

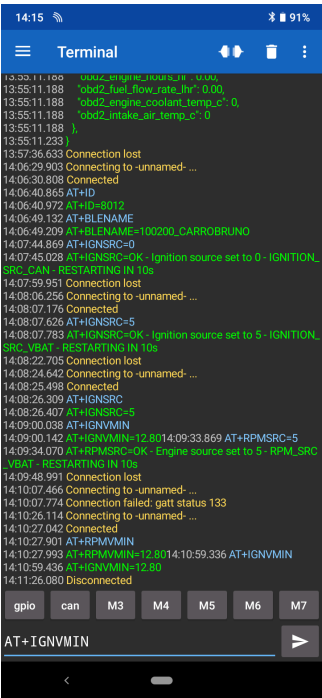
No caso do celta, é na caixa de fusíveis, próximo ao puxador de abertura do capô, como no exemplo abaixo:



2. Configure o Simova Track seguindo os comandos abaixo (mas lembre-se de trocar os valores dos parâmetros para seu uso real).

```
AT+RESET  
AT+ID=1234  
AT+BLENOME=exemplo  
AT+IGNSRC=5  
AT+RPMSRC=5  
AT+IGNVMIN=12.5  
AT+RPMVMIN=12.5  
* para mais detalhes sobre o funcionamento desses comandos  
* verifique o guia na seção "Guias de Instalação"
```

3. Exemplo de configuração





Motor e ignição desligados



motor e ignição ligados

▼ Celta LS 2013 - Ignição/RPM via entrada digital

1. Localize a porta OBD2 de seu veículo:

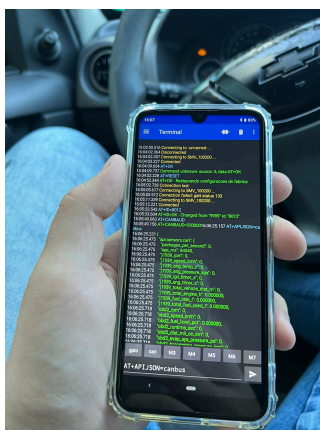
No caso do celta, é na caixa de fusíveis, próximo ao puxador de abertura do capô, como no exemplo abaixo:

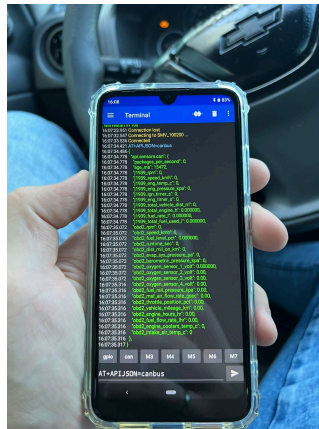
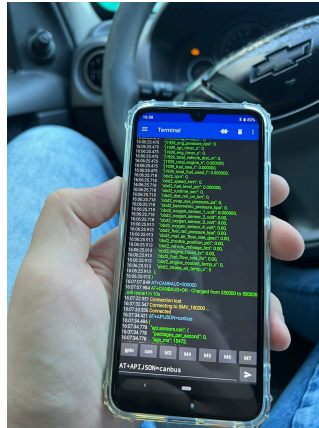


2. Nessa Etapa, é executado os comandos, como no **EXEMPLO** abaixo:

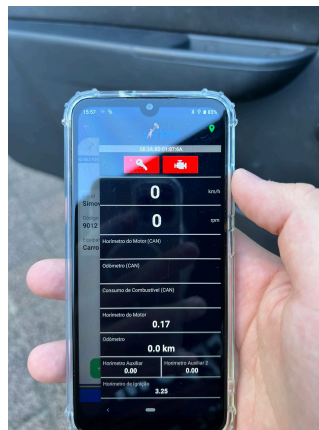
```
AT+RESET
AT+ID=1234
AT+BLENAM=exemplo
AT+IGNSRC=1
AT+RPMSRC=1
```

- * para mais detalhes sobre o funcionamento desses comandos
- * verifique o guia na seção "Guias de Instalação"
- * lembrando que não há necessidade de ser a entrada 1
- * E pode ser realizado com outra coloração do fio

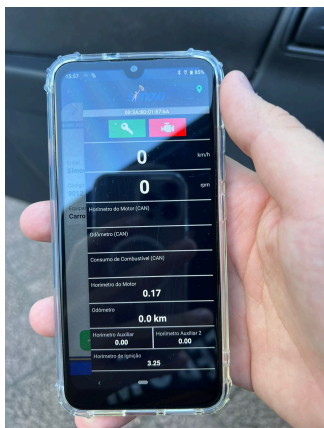




3. antes do acionamento do botão, ao conectar com o Track a tela aparecerá como no exemplo:



4. Após o acionamento do botão, a tela, estará assim:
Com a ignição ligada, assim possibilitando a marcação da telemetria no Simova Track



5. Feito Isso, o botão terá que ser acionado toda vez que a ignição/motor do carro for ligado. Para assim conseguir se obter a telemetria e a velocidade do veículo.

▼ Revisões

2024-05-01 - Italo, Gustavo, Bruno - Document created

2024-05-09 - Italo, Gustavo, Bruno - Added support for the command AT+BLENAME, including comms, BLE and API changes, Added information about Simova Angle, about CAN debug, API Debug, and more

2024-06-04 - Italo - OTA section updated, AT+API section updated to match new API structure, gramatical review