



Manual Speeduino

23/04/21

Conteúdo

Introdução	6
Carregando o firmware do Speeduino	7
Visão geral.....	7
Instalação - SpeedyLoader	7
Instalação - Compilando Manualmente	8
Conectando ao Tuner Studio	11
Baixando Tuner Studio	12
Configurando seu projeto	12
Configurando as propriedades do projeto TunerStudio	15
Guia Configurações	17
Guia Dispositivos CAN	19
Guia de fiação de alto nível	20
Fiação do injetor	20
Visão geral.....	20
Injetores Suportados	21
Layouts	21
Fiação de Ignição	25
Visão geral.....	25
Faísca Desperdiçada	26
Sequencial (COPs).....	27
Distribuidor.....	28
Fiação do sensor analógico	29
Constantes do motor	30
Visão geral.....	30
Configuração	31
Características do injetor	33
Visão geral.....	33
Definições	34

Manual Speeduino	23/04/21
Configuração do gatilho	35
Visão geral.....	35
Configurações de gatilho	36
Encontrando o dente nº 1 e o ângulo de gatilho.....	37
Densidade IAT	37
Visão geral.....	37
Curva de Exemplo	38
Tabela de combustível (VE)	38
Configuração.....	39
Tabela de Combustível Secundário	40
Enriquecimento de aceleração (AE)	42
Teoria.....	42
AFR / O2 (combustível de circuito fechado)	45
Definições.....	45
Limiters	47
Definições.....	48
Combustível flex	48
Visão geral.....	48
Hardware.....	49
Tuning.....	50
Injeção faseada	51
Visão geral.....	51
Configuração.....	51
Configurações do Spark	53
Visão geral.....	53
Definições.....	54
Dwell Control	57
Visão geral.....	57
Definições.....	58
Correção de tensão.....	59
Mudanças de tempo baseadas na temperatura	60
Exemplo	60

Manual Speeduino	23/04/21
Visão geral.....	60
Definições.....	61
Enriquecimento de aquecimento	62
Visão geral.....	62
Definições.....	63
Controle de marcha lenta	64
Visão geral.....	64
Tipos de válvula ociosa compatíveis	64
Controle de malha fechada.....	73
Controle de avanço ocioso	73
Definições.....	74
Termoventilador	75
Definições	76
Controle de lançamento e deslocamento plano	76
Configurar.....	77
Bomba de combustível	78
Definições.....	79
Boost Control	79
Definições.....	80
Tabela de destino.....	81
Controle Nitroso	81
Configurações de ativação.....	82
Configurações do palco	83
Calibração do Sensor	83
Sensor MAP	84
Sensores de temperatura de admissão e refrigerante	85
Sensor de oxigênio.....	88
Sensor de posição do acelerador	89
Configuração IO Auxiliar	90
Como usar	90
Padrões de gatilho suportados	92

Manual Speeduino	23/04/21
Falta de padrão de dente.....	92
Dente faltando (velocidade do Cam)	95
Roda Dupla.....	98
Distribuidor Básico	100
Sinal de gatilho.....	101
GM 7X.....	101
Padrão 4G63	101
Visão geral.....	103
Sinal de gatilho.....	104
Visão geral.....	104
Sinal de gatilho.....	104
Harley Evo	104
Visão geral.....	104
Miata 99-05	105
Decodificador não 360.....	106
Visão geral.....	107
Formulários	107
Configuração do Tuner Studio	107
Ajuste de tempo	107
Padrão de gatilho	107
Subaru 36-2-2-2	108
Placa V0.4	109
Visão geral.....	109
Recursos da placa	109
Formato físico	110
Montagem do Conselho.....	112
Configuração da placa	114
Revisões do conselho	116
Placa V0.3	116
Visão geral.....	116
Recursos da placa	117
Formato físico	117
Área proto	118
Montagem do Conselho.....	118
Configuração da placa	119
Revisões do conselho	122

Introdução

Este manual cobre o hardware (sensores, fiação, etc.), a configuração do software e os elementos de ajuste relacionados à operação de uma unidade Speeduino. Ao começar com o Speeduino, principalmente se for a primeira vez que instala e configura um sistema de gerenciamento do motor, este manual ajudará a entender as capacidades do Speeduino e como ele deve ser instalado, tanto em termos de hardware quanto de software / -firmware.

Embora este documento auxilie no fornecimento de informações relacionadas à configuração do Speeduino, ele não cobre o ajuste avançado do motor, estratégias de combustível / ignição, etc. Como acontece com qualquer alteração no gerenciamento do motor, a possibilidade de danos ao hardware é muito real caso um sistema seja configurado incorretamente.

Recebendo Começado

Em termos de começar com o Speeduino, pode ajudar a entender os vários componentes que fazem o sistema:

1. **Uma placa Speeduino** - Este é o músculo da ECU Speeduino e contém todos os drivers e circuitos IO. Esta pode ser uma das placas genéricas (como a v0.4) ou uma placa PNP para um específico modelo de carro
2. **Um Arduino** - Este é o cérebro do Speeduino e contém o processador, a memória e o armazenamento. Ele se conecta à placa Speeduino para fazer a interface com a fiação do veículo. Normalmente um ArduinoMega 2560.
3. **Firmware** - Este é o software do sistema que roda na placa Arduino e aciona sua operação. Novo firmware é lançado regularmente com atualizações, melhorias de desempenho e correções de bugs.

Como ponto de partida, geralmente é recomendado carregar primeiro o firmware para o seu Arduino e conectá-lo ao software de ajuste (Tuner Studio) antes de passar para a montagem do hardware ou fiação, etc. A instalação e configuração do software no Speeduino podem ser concluídas sem o necessário hardware adicional estar presente (além do próprio arduino) e isso permite a exploração do software e das opções disponíveis antes de um desembolso de fundos significativo ou um investimento significativo de tempo.

Mais detalhes sobre requisitos de hardware e recursos específicos de versão podem ser encontrados na página Primeiros passos

Sobre este manual

Como um projeto de código aberto, esta documentação está crescendo continuamente e isso significa que você pode encontrar lacunas na documentação onde poucas informações são fornecidas atualmente. Por favor, não hesite em postar no fórum se houver algo faltando de que você precisa criticamente (ou mesmo não criticamente).

Além disso, se você quiser contribuir com a documentação do Speeduino, adoraríamos ouvir sua opinião! O método preferido para solicitar acesso ao wiki é via [Folga](#)

Carregando o firmware do Speeduino

Visão geral

O firmware Speeduino é o código que alimenta o hardware e deve ser instalado em sua placa antes de usar a ECU. Novos lançamentos de firmware são feitos regularmente (aproximadamente a cada 2 meses) que trazem novos recursos, correções de bugs e melhorias de desempenho, portanto, manter-se atualizado é altamente recomendado.

Com o objetivo de máxima simplicidade em mente, o processo de compilar e instalar o firmware é razoavelmente simples. A maioria dos usuários usará o método SpeedyLoader para instalar o firmware

Instalação - SpeedyLoader

O método mais simples (e recomendado) de instalar o firmware Speeduino em um Arduino Mega 2560 padrão é com o utilitário SpeedyLoader. O SpeedyLoader se encarrega de baixar o firmware e instalá-lo em um Arduino sem a necessidade de compilar manualmente qualquer parte do código. Você pode escolher o firmware mais recente que foi lançado ou selecionar um dos mais antigos, se preferir. O SpeedyLoader também baixará o arquivo INI e, opcionalmente, uma melodia base para o firmware que você escolher, para que possa ser carregado em seu projeto TunerStudio.

- **Janelas:** 32 bits / 64 bits
- **Mac:** SpeedyLoader.dmg
- **Linux:** SpeedyLoader.AppImage (Vontade precisa ser tornado executável após o download)
 - O Linux requer que as bibliotecas libusb sejam instaladas. EG se no Debian /
 - Ubuntu:`sudo apt-get install libusb-1.0-0 libusb-0.1-4:i386`
- **Raspberry Pi** SpeedyLoader.AppImage

- Framboesa Os usuários Pi / Raspbian podem instalar as bibliotecas necessárias com: `sudo apt-get install libusb-1.0-0 libusb-0.1-4`

Assim que o firmware estiver instalado na placa, consulte Conectando-se ao TunerStudio para obter mais detalhes sobre como configurar o TunerStudio

Instalação - Compilando Manualmente

Observe que a compilação manual do firmware NÃO é necessária para instalar o Speeduino, o método mais fácil (e recomendado para a maioria dos usuários) é usar o SpeedyLoader conforme descrito acima. {.is-warning}

Se você deseja compilar o firmware por conta própria ou fazer qualquer alteração no código, o código-fonte de ambos os lançamentos e a versão de desenvolvimento atual estão disponíveis gratuitamente.

Requisitos

- Um PC Windows, Mac ou Linux
- Um dos seguintes:
 - O IDE Arduino. Versão mínima atual necessária é 1.6.7, embora uma versão mais recente seja recomendado.
 - PlatformIO. Pode ser baixado [de aqui](#)
- Uma cópia da base de código mais recente do Speeduino. Veja abaixo.
- Uma cópia de TunerStudio para testar se o firmware foi carregado com sucesso

Baixando o firmware

Existem dois métodos para obter o firmware Speeduino:

1. Quedas regulares de código estável são produzidas e feitas como versões no Github. Estes podem ser encontrados no: [Lançamentos](#)
2. Se você quiser o código mais recente e melhor (e ocasionalmente o mais esquisito), o repositório git pode ser clonado e atualizado. [Veraqui](#)

Compilando o firmware

- Inicie o IDE, selecione Arquivo > Abrir, navegue até o local onde você baixou o Speeduino e abra o arquivo `speeduino.ino`.

- Defina o tipo de placa: Ferramentas> Placa> Arduino Mega 2560 ou Mega ADK (esta é a única placa suportada atualmente)
- Clique no ícone Verificar no canto superior esquerdo (parece uma marca)

Neste ponto, você deve ter um firmware compilado! Se você teve um problema durante a compilação, consulte a seção Solução de problemas abaixo.

Este vídeo percorre todo o processo de instalação do firmware em seu Arduino a partir do zero:

Opcional (mas recomendado) Existe uma opção disponível para alterar o nível de otimização do compilador, que pode melhorar. Por padrão, o IDE usa a opção de compilação -Os, que se concentra na produção de pequenos binários. Como o tamanho do código Speeduino não é um problema, mas a velocidade é uma consideração, alterá-lo para -O3 produz melhores resultados (aproximadamente 20% mais rápido, com um tamanho de esboço 40% maior). Para fazer isso, você precisa editar o arquivo platform.txt:

- Certifique-se de que o IDE do Arduino não esteja em execução
- Abra o arquivo platform.txt que está nos seguintes locais:
 - No Windows: c:\Arquivos de programas\Arduino\hardware\arduino\avr
 - No Mac: /Aplicativos/Arduino/Conteúdo/Recursos/Java/hardware/arduino/avr/
 - No Linux:
- Sobre as 3 entradas a seguir, altere o Os para O3:
 - compiler.c.flags
 - compiler.c.elf.flags
 - compiler.cpp.flags
- Salve o arquivo e reinicie o Arduino IDE

Observação: Isso NÃO é necessário se estiver usando PlatformIO, a otimização acima é aplicada automaticamente lá

Instalando

Depois de compilar o firmware com sucesso, a instalação na placa é trivial.

- Conecte seu Mega 2560 a uma porta USB livre
- Se você está executando uma versão mais antiga do Windows e esta é a primeira vez que usa um Arduino, pode ser necessário instalar drivers para o chip serial do Arduino (USB-UART ou "chip adaptador USB").

A maioria As placas oficiais e muitas versões não oficiais usam o ATMega16U2 ou 8U2, enquanto muitas das placas clone Mega2560 utilizam o CH340G IC. Ambos os tipos funcionam bem. Os chips seriais geralmente podem ser identificado pela aparência:

- **ATMega16U** - Tem um IC quadrado perto do conector USB - os drivers estão incluídos no Windows 7 ou superior, MacOS e Linux.
- **WCH CH340G** - Tem um IC retangular perto do conector USB - usa drivers "CH341" de WCH para Windows
 - Os drivers CH340 / CH341 originais do WCH para outros sistemas (Mac, Linux, Android, etc) podem ser encontrado aqui.
- Em Arduino IDE; selecione o Mega2560: Ferramentas> Placa
- Selecione a porta serial do seu sistema para fazer o upload: Ferramentas> Porta Serial
- Clique no botão Upload de o canto superior esquerdo

Versões de firmware mais antigas

Se necessário, versões de firmware mais antigas e detalhes podem ser encontrados em Firmware_History

Verificando Firmware

O firmware agora está carregado em sua placa e você agora pode passar para Conectando-se ao Tuner- Estúdio.

Opcionalmente, você pode realizar uma verificação manual do firmware usando o Serial Monitor do IDE do Arduino. Isso pode ser iniciado selecionando 'Monitor serial' no menu Ferramentas.

Na janela que aparece, digite um "S" maiúsculo (sem aspas) e pressione Enter. O Mega deve responder com o ano e mês da versão do código instalada (xxxx.xx):

```
1 Speeduino 03.07.2017
```

NOTA: Certifique-se de que a taxa de transmissão esteja definida para 115200

Você também pode inserir "?" para obter uma lista de consultas do seu Mega.

Solução de problemas

Placa Arduino incorreta selecionado Se você vir os seguintes erros (ou semelhantes) ao tentar compilar o firmware e as soluções:

```

1 Agendador.eu não: 317: 7: erro: ''OCR4A não foi declarado neste escopo
2 Agendador.eu não: 323: 8: erro: ''TIMSK5 não foi declarado neste escopo
3 Agendador.eu não: 323: 25: erro: ''OCIE4A não foi declarado neste

```

Você pode ter selecionado o tipo errado de placa Arduino. Defina o tipo de placa selecionando Ferramentas> Placa > *Arduino Mega 2560 ou Mega ADK*

Todo o projeto Speeduino não é aberto O seguinte pode ocorrer se você abriu apenas o arquivo speeduino.ino em vez de todo o projeto.

speeduino.eu não: 27: 21: erro fatal: globais.h: Não existe tal arquivo ou diretório

Certifique-se de que todos os arquivos estejam contidos no mesmo diretório, selecione Arquivo-> Abrir e encontre o arquivo speeduino.ino. Se você abriu o projeto corretamente, deve ter várias guias ao longo do principal:

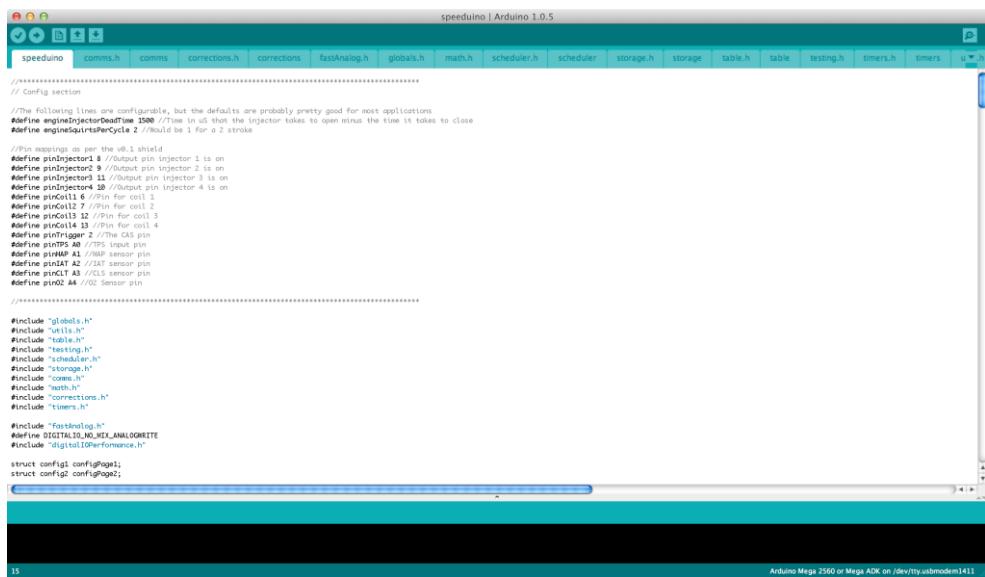


Figura 1: speeduinoIDE.png

Se você vir apenas um único arquivo ou um pequeno número de arquivos, então não abriu o projeto inteiro.

Conectando ao Tuner Studio

Tuner Studio é o software de interface de ajuste usado pelo Speeduino. Ele roda em Windows, Mac e Linux e oferece recursos de configuração, ajuste e registro.

Depois de ter o firmware compilado e carregado em seu Arduino, você está pronto para configurar o Tuner Studio para configurá-lo e monitorá-lo. Se você ainda não compilou e carregou o firmware, consulte a página [Instalando o Firmware](#).

Baixando Tuner Studio

Se ainda não o fez, pegue uma cópia do Tuner Studio em [EFI Analytics Tuner Studio](#) está disponível para Windows, Mac e Linux e pode ser executado na maioria dos PCs, pois os requisitos de sistema são bastante baixos.

A versão mínima atual do TunerStudio necessária é 3.0.7, mas a versão mais recente é geralmente recomendada.

Se você achar que o Tuner Studio é útil, considere pagar por uma licença. Este é um programa fantástico de um único desenvolvedor que rivaliza com o melhor software de ajuste do mundo, vale a pena o dinheiro gasto.

Contexto seu projeto

Crie um novo projeto

Ao iniciar o TunerStudio pela primeira vez, você precisará configurar um novo projeto que contém as configurações, ajuste, registros, etc. Na tela de inicialização, selecione 'Criar novo projeto'

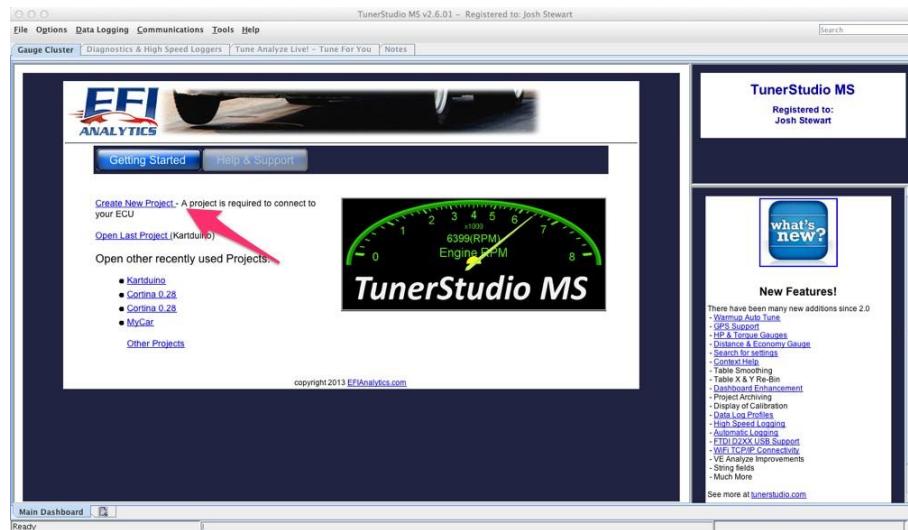


Figura 2: TS_1.png

Dê a você um nome para o projeto e selecione o diretório no qual deseja que o projeto seja armazenado. O Tuner Studio então requer um arquivo de definição de firmware para se comunicar com o arduino. Marque o 'Outro /Botão Navegar '.

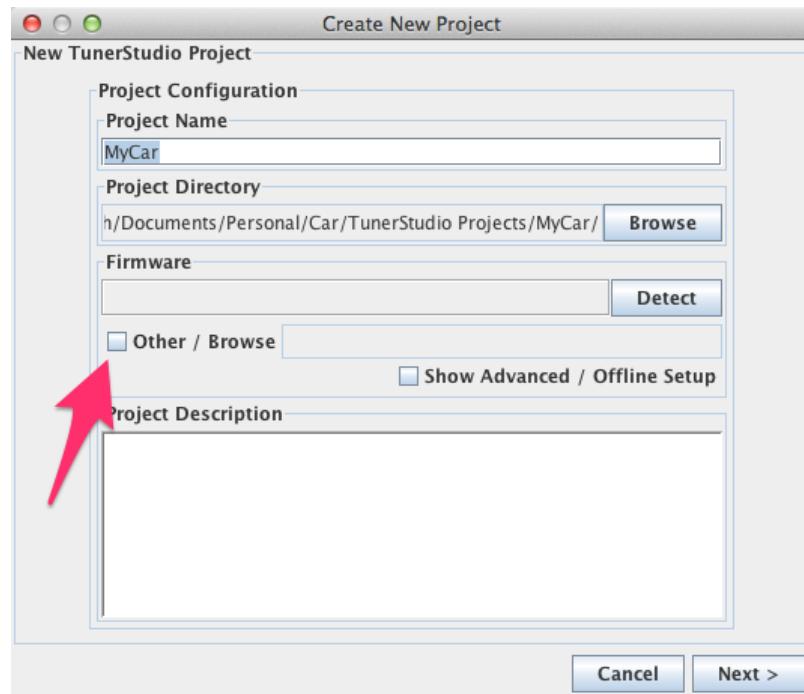


Figura 3: TS_2.png

Em seguida, navegue até o diretório de origem do Speeduino, insira a subpasta de referência e selecione o arquivo speeduino.ini

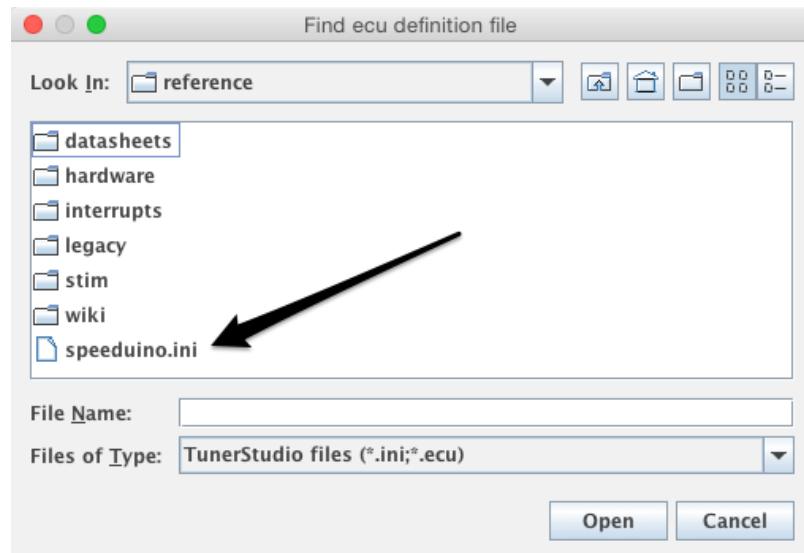


Figura 4: TS_3.png

Opções de configuração

Consulte a página de opções de configuração do projeto TunerStudio para isso

Configurações de comunicação

Selecione suas opções de comunicação. O nome exato da porta dependerá de qual sistema operacional você está executando e será o mesmo que no IDE do Arduino. A taxa de transmissão deve ser 115200.

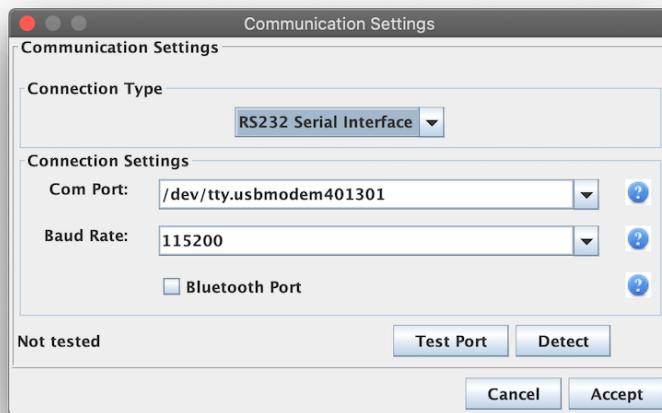


Figura 5: Configurações de comunicação

Observação: O [Detectar](#) e [Porta de teste](#) opções requerem Tuner Studio versão 3.0.60 ou superior para funcionar corretamente {.is-info}

Carregar melodia base

Depois que o projeto for criado, você precisará carregar em uma melodia base para garantir que todos os valores sejam pelo menos um pouco lógicos. Deixar de fazer isso pode levar a questões e valores muito estranhos em sua música.

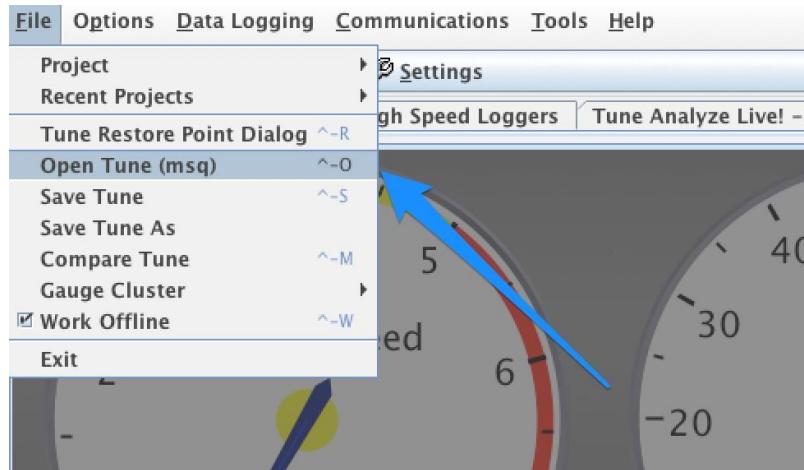


Figura 6: TS_6.png

No diretório de referência do Speeduino, você encontrará o arquivo de tune base a ser aberto:

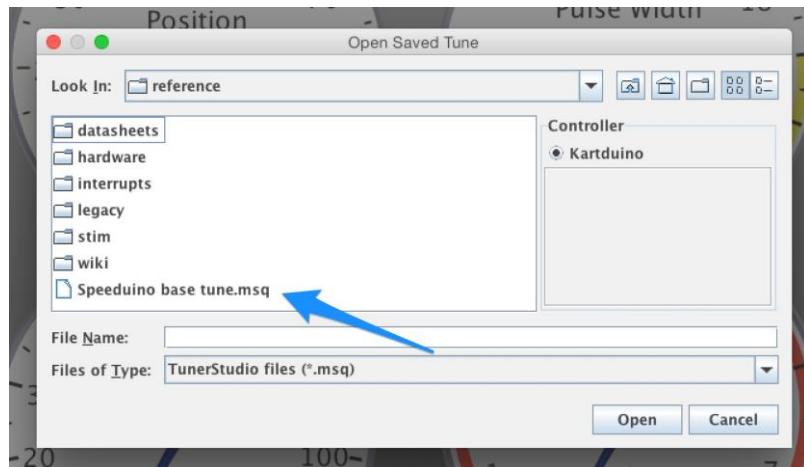


Figura 7: TS_7.png

E é isso! O Tuner Studio agora deve tentar se conectar ao Arduino e mostrar uma tela em tempo realdo ECU.

Configurando as propriedades do projeto TunerStudio

A opção de menu para a página de propriedades do projeto pode ser encontrada aqui



Figura 8: TS_9.png

Uma vez aberta, esta página será vista.

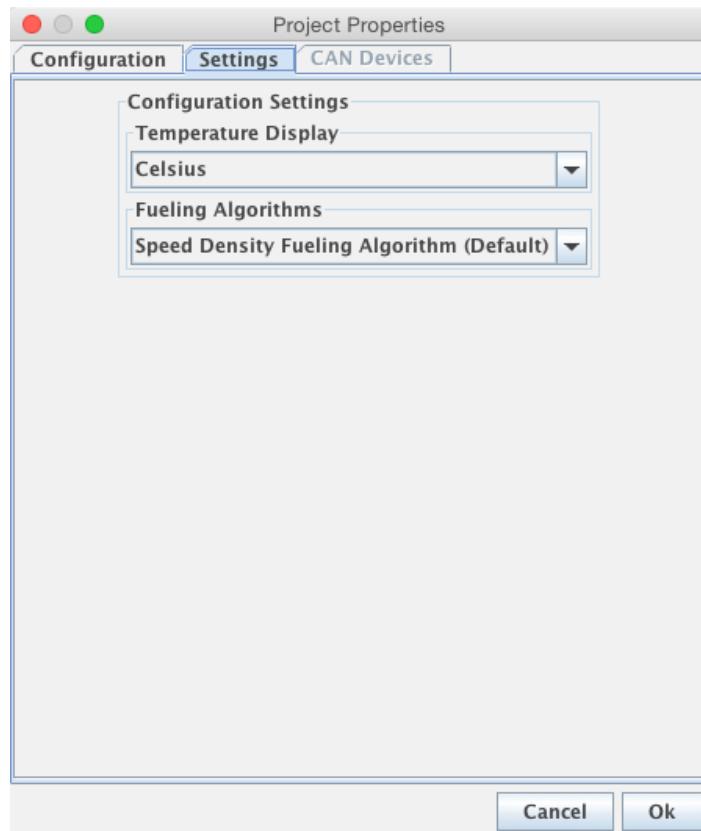


Figura 9: TS_4.png

Guia Configurações

A guia Configurações não afeta a melodia diretamente, mas muda a maneira como algumas coisas são exibidas no Tuner Studio. Alguns menus estão ocultos por padrão, por razões de segurança ou porque estão ainda em desenvolvimento e podem ser ativados aqui.

Lambda Display

Isso muda se os motivos do sensor de oxigênio são mostrados em AFR (padrão) ou Lambda.

Display de temperatura

A seleção de temperatura muda todos os valores de graus dentro do TunerStudio.

- Fahrenheit (padrão)

- Celsius

Alterar este valor não altera os valores afinados de forma alguma, apenas em qual escala os valores estão dispersos jogado em {.is-info}

Habilitar teste de hardware

A caixa de diálogo de teste de hardware permite que você ligue e desligue manualmente as saídas de ignição e injeção para testar se os circuitos estão funcionando. Isso pode ser perigoso se as saídas estiverem conectadas ao hardware e, portanto, esta caixa de diálogo deve ser ativada explicitamente.

Ligue SOMENTE quando a ECU não está conectado a um veículo {.is-perigo}

Se ativado, uma guia adicional aparecerá na página de ajuste

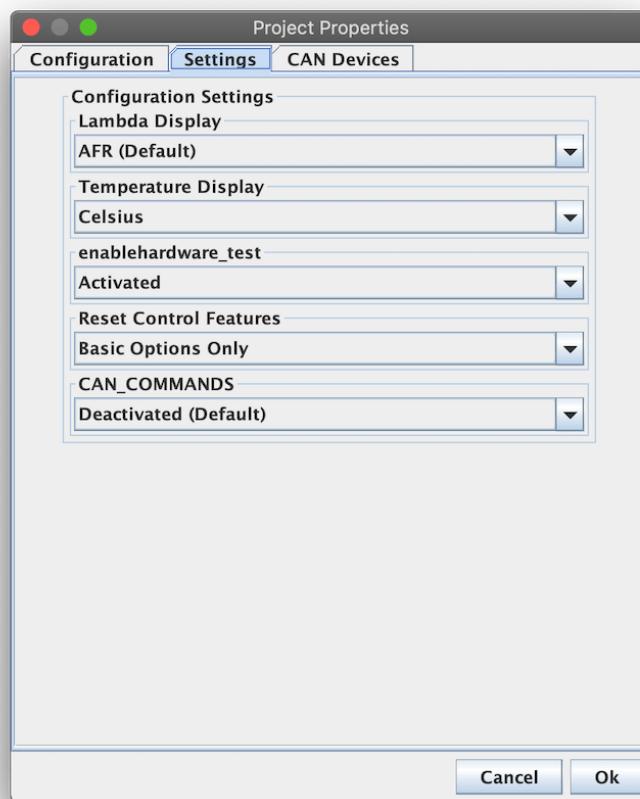


Figura 10: Configurações do projeto

Redefinir recursos de controle

Um carregador de boot específico opcional do Speeduino está disponível, com diferentes métodos de controle da reinicialização automática. A grande maioria dos usuários deve deixar isso no padrão 'Opções básicas apenas'

Guia Dispositivos CAN

As opções CAN estão atualmente em desenvolvimento, mas as configurações estão disponíveis nesta guia para teste se você suportaram hardware.

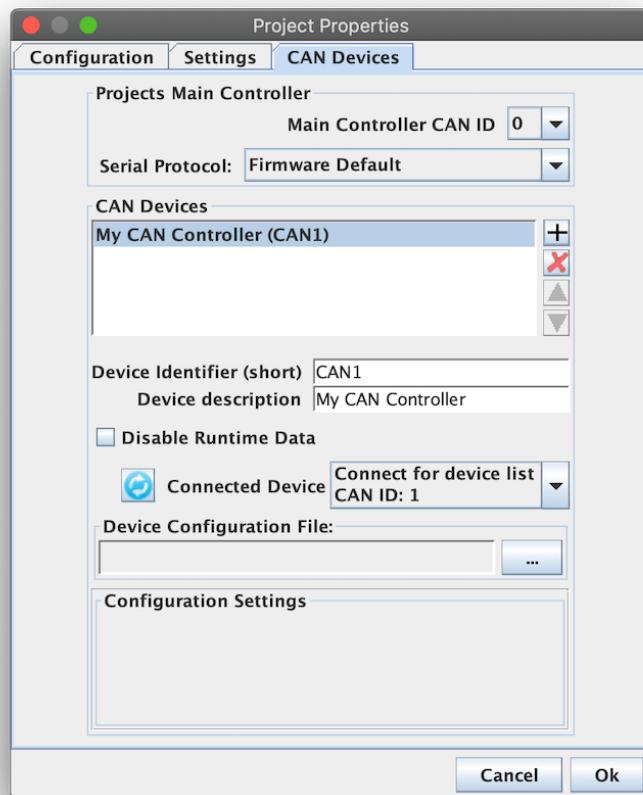


Figura 11: Dispositivos CAN

A configuração de dispositivos CAN não é compatível no momento

Guia de fiação de alto nível

O Speeduino pode ser configurado de várias maneiras, dependendo do motor, sensores, ignição e hardware de combustível usados. Por esta razão, é impossível fornecer um único diagrama que cubra todos os cenários, no entanto, o abaixo é fornecido como um guia de alto nível que pode ser usado como um ponto de partida.

Consulte a página Requisitos de hardware para requisitos específicos e exceções à imagem abaixo.

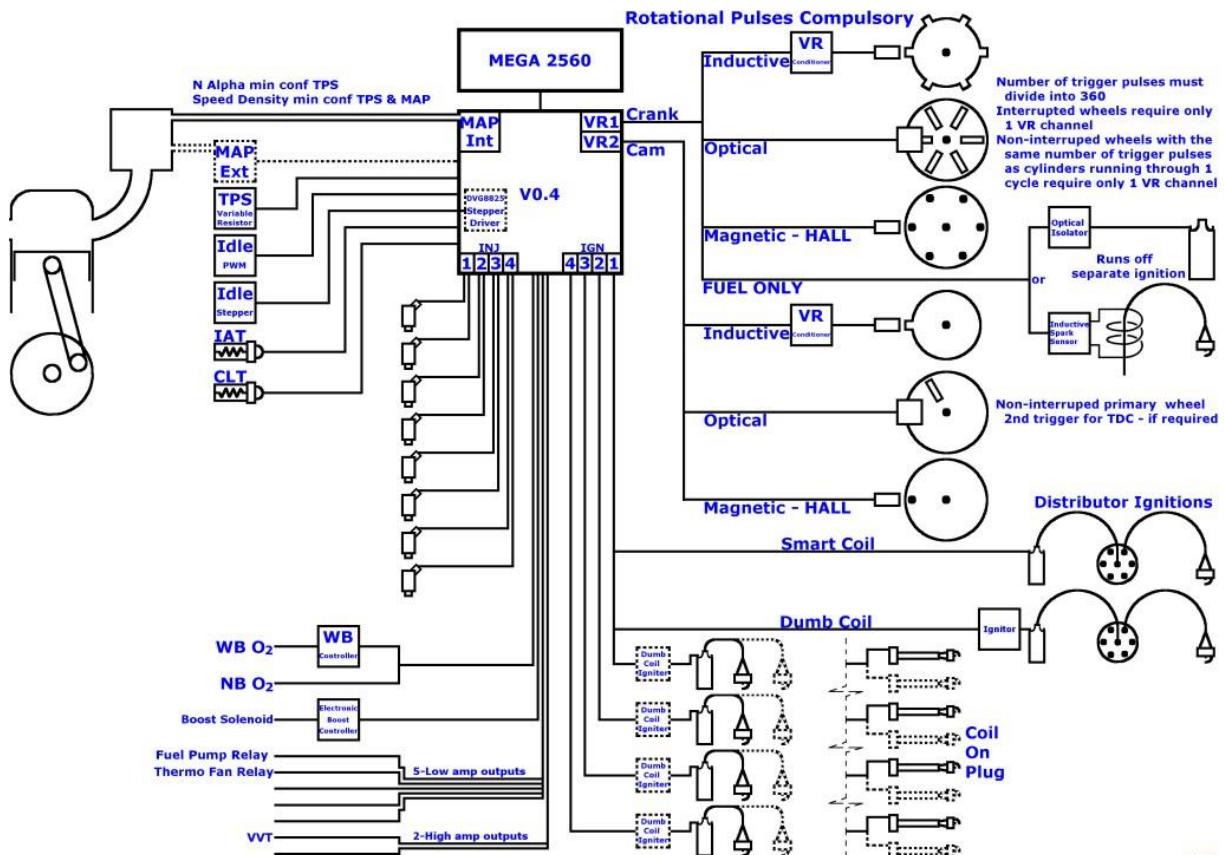


Figura 12: wiring_overview.png

Fiação do injetor

Visão geral

O Speeduino contém 4 circuitos de controle de injetores e é capaz de suportar até 8 injetores (e ders) com estes.

Injetores Suportados

O Speeduino suporta injetores High-Z (também conhecidos como 'alta impedância' ou 'saturados') nativamente. Os injetores Low-Z são suportados com a adição de resistores conectados em série com os fios de sinal. Os injetores High-Z são normalmente aqueles com uma resistência superior a 8 Ohms.

Se forem usados injetores "Low-Z" ("pico e retenção" ou controlados por PWM) com impedância mais baixa, a fiação exigirá resistores em série em cada injetor para evitar danos à placa com corrente excessiva. A classificação de ohms e watt do resistor pode ser calculada pela Lei de Ohm ou usar uma calculadora da Internet página como a Calculadora do resistor injetor Speeduino.

Layouts

Existem várias maneiras de conectar os injetores, dependendo da sua configuração e preferência.

1, 2 e 3 injetores

Para essas configurações, cada injetor é conectado em sua própria saída da placa Speeduino.

4 injetores

Para 4 cilindros / injetores, existem 2 maneiras de conectá-los ao Speeduino:

Método 1 (emparelhado) O método padrão é o mesmo usado para configurações de 6 ou 8 cilindros, onde 2 injetores são conectados a cada canal do injetor. Nesta configuração, apenas 2 canais injetores serão usados. Os injetores emparelhados devem ter seus centros mortos superiores (TDC) 360 graus de manivelaseparado.

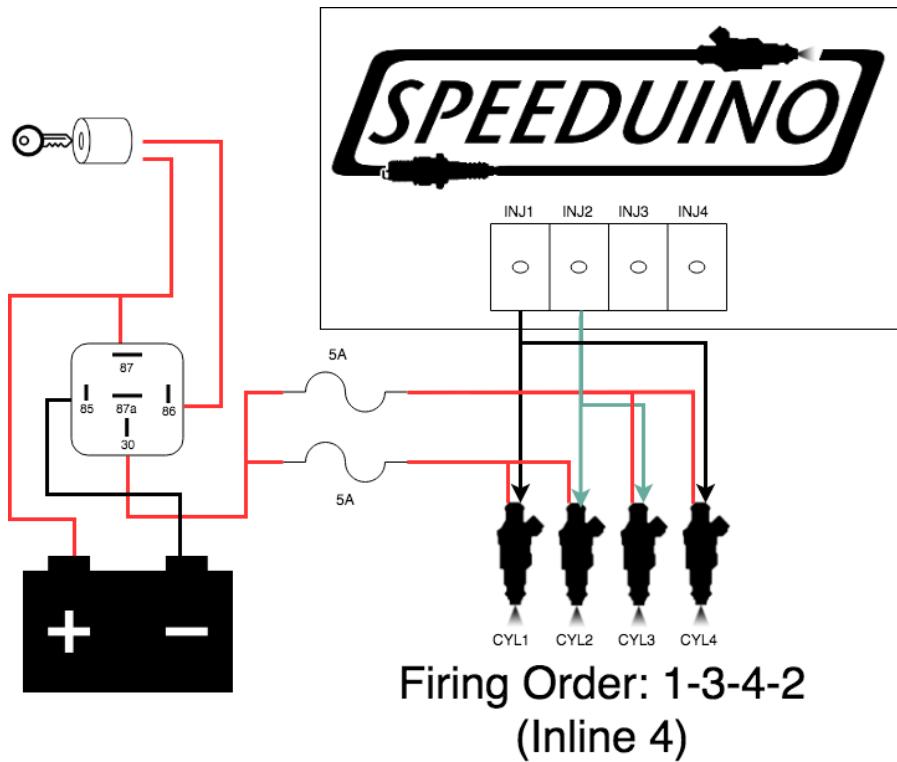


Figura 13: inj_4Cyl_semi-seq.png

Método 2 (sequencial completo) Este método está disponível apenas em aplicações de 4 cilindros / 4 injetores e permite conectar 1 injetor por canal. Os canais do injetor sempre disparam em ordem numérica (ou seja, 1, 2, 3, 4), portanto, seus injetores devem ser conectados para levar em consideração a ordem de disparo. Dentro do Tuner Studio, esta opção pode ser habilitada selecionando:

Definições -> Constantes do motor -> Sincronização do injetor -> Sequencial

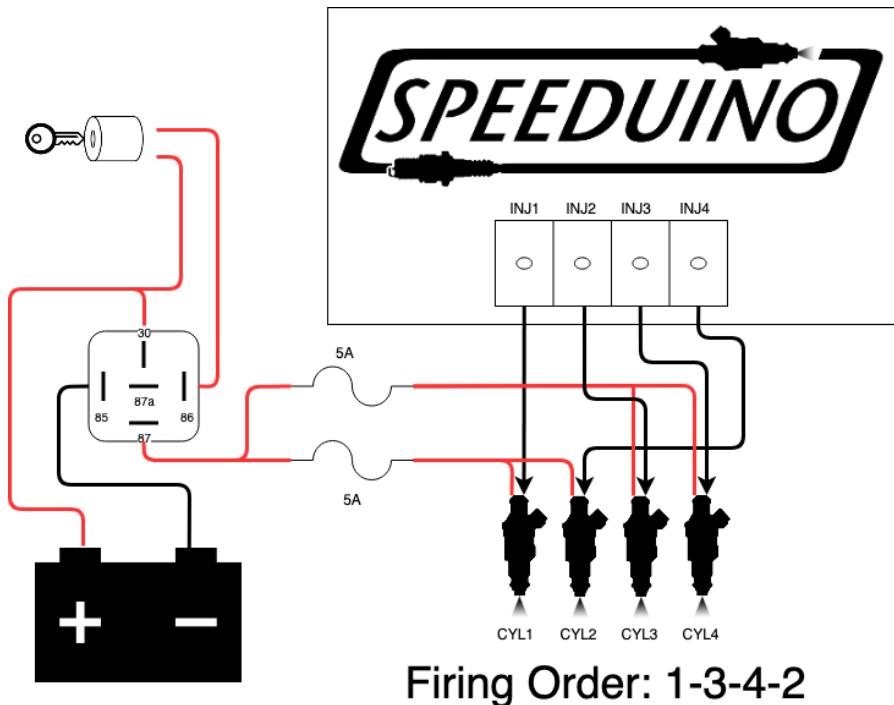


Figura 14: inj_4Cyl_seq.png

Observação: O uso de abastecimento sequencial exige que um sinal de came compatível seja usado além da manivela. Se nenhum sinal de came for fornecido quando a opção sequencial for selecionada, o sistema nãosync {.is-warning}

5 injetores

5 as configurações do cilindro devem ser conectadas para usar todas as 4 saídas do injetor com 2 injetores compartilhando a saída # 3. Para a ordem de disparo típica de 5 cilindros em linha (1-2-4-5-3), os injetores 4 e 3 seriam unidos emsaída do jeto 3.

Mais de 5 injetores

Para configurações com mais de 4 injetores, o número de saídas usadas será igual a metade do número de injetores.

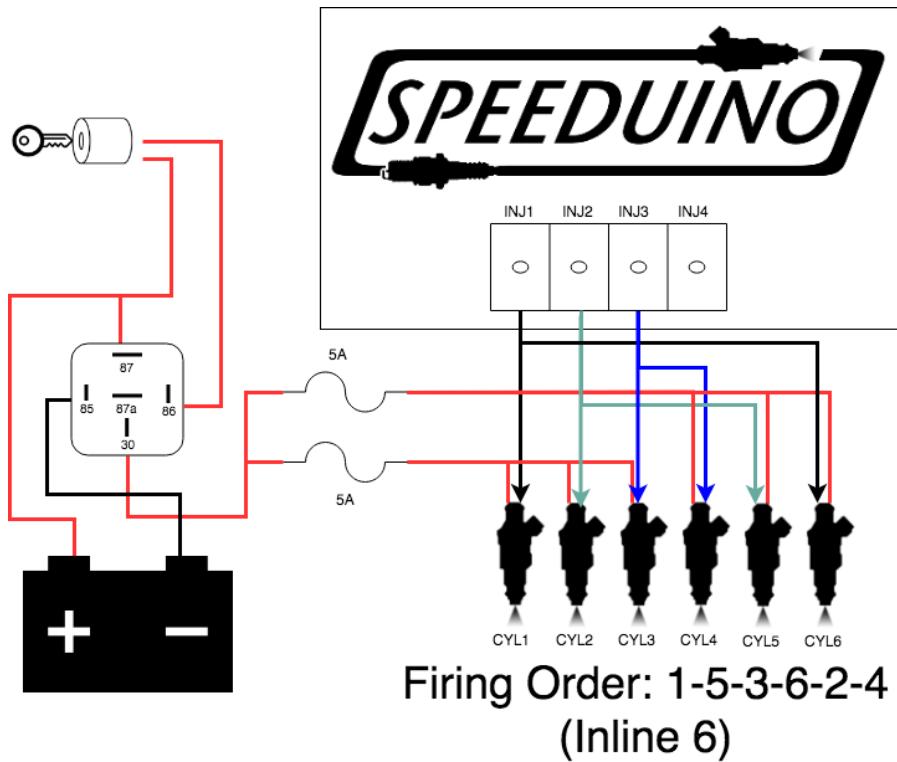


Figura 15: inj_6Cyl_semi-seq.png

6 Cilindro Para um V6 com uma ordem de disparo de (1,4,2,5,3,6), os injetores serão conectados em 3 grupos de (1,5) e (4,3), e (2,6) como estes os cilindros estão separados por 360 graus da manivela.

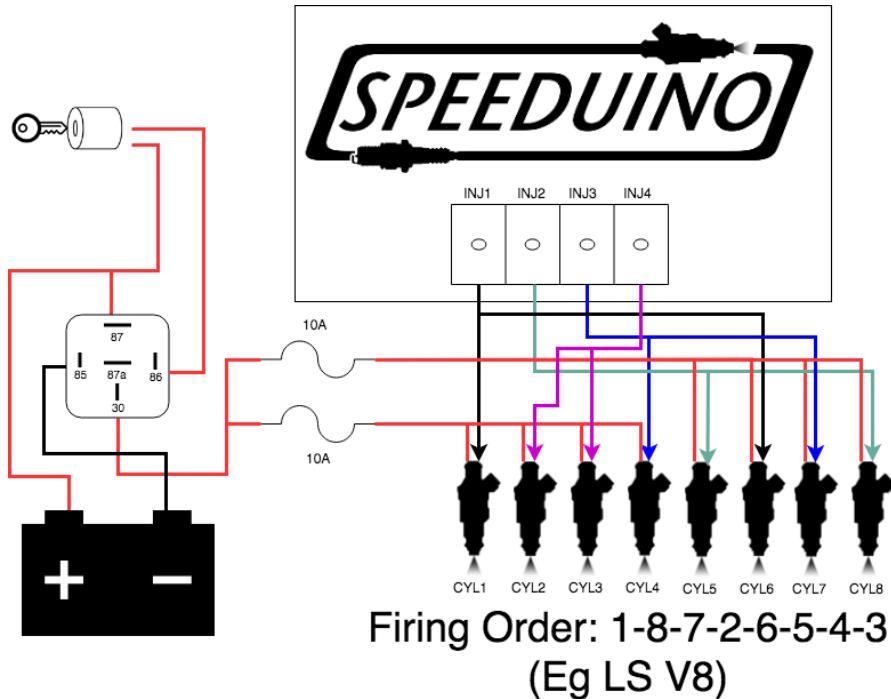


Figura 16: inj_8Cyl_semi-seq.png

8 cilindros Em linha com o acima, esta configuração requer que cada saída do injetor seja conectada a 2 injetores. Os injetores devem ser agrupados em pares opostos, ou seja, cilindros cujo Top DeadOs centros estão separados por 360 graus.

Fiação de Ignição

Visão geral

A configuração da saída de ignição pode ser uma das áreas mais difíceis da fiação da ECU e uma das que costuma causar mais confusão. Uma grande parte dessa complexidade vem do grande número de diferentes tipos de ignição que estão disponíveis, com mudanças significativas no hardware usado no finalAnos 80 e ao longo dos anos 90 em comparação com designs mais recentes.

Embora este guia não cubra todos os estilos de ignição e ferragens, ele cobre os cenários mais comuns. Geralmente, é recomendado (onde possível) usar estilos mais novos de hardware de ignição (normalmente 'smart' Coil-on-Plug ou Coil-Near-Plug) em vez de utilizar módulos de ignição separados.

Faísca Desperdiçada

A faísca desperdiçada é um meio comum de controlar a faísca que requer apenas metade do número de saídas de ignição do que os cilindros, com 2 cilindros acoplados a cada saída. EG: * Motor de 4 cilindros requer 2 saídas de ignição * Motor de 6 cilindros requer 3 saídas de ignição * Motor de 8 cilindros requer 4 saídas de ignição

Wasted Spark tem a vantagem de não requerer nenhum sinal de came ou entrada, pois não precisa saber a fase do motor. Isso é possível disparando as saídas de ignição uma vez por revolução e emparelhando essa saída para 2 cilindros que estão ambos no PMS (com um cilindro no curso de compressão e o outro ligado escape)

Ao usar faísca desperdiçada, é fundamental que os pares corretos de bobinas e / ou velas de ignição sejam unidos.

Existem muitos pacotes de bobina de ignição desperdiçada de pólo duplo disponíveis com e sem ignitores integrados. Ambos são adequados para uso com Speeduino, mas o uso de bobinas com dispositivos de ignição embutidos é recomendado

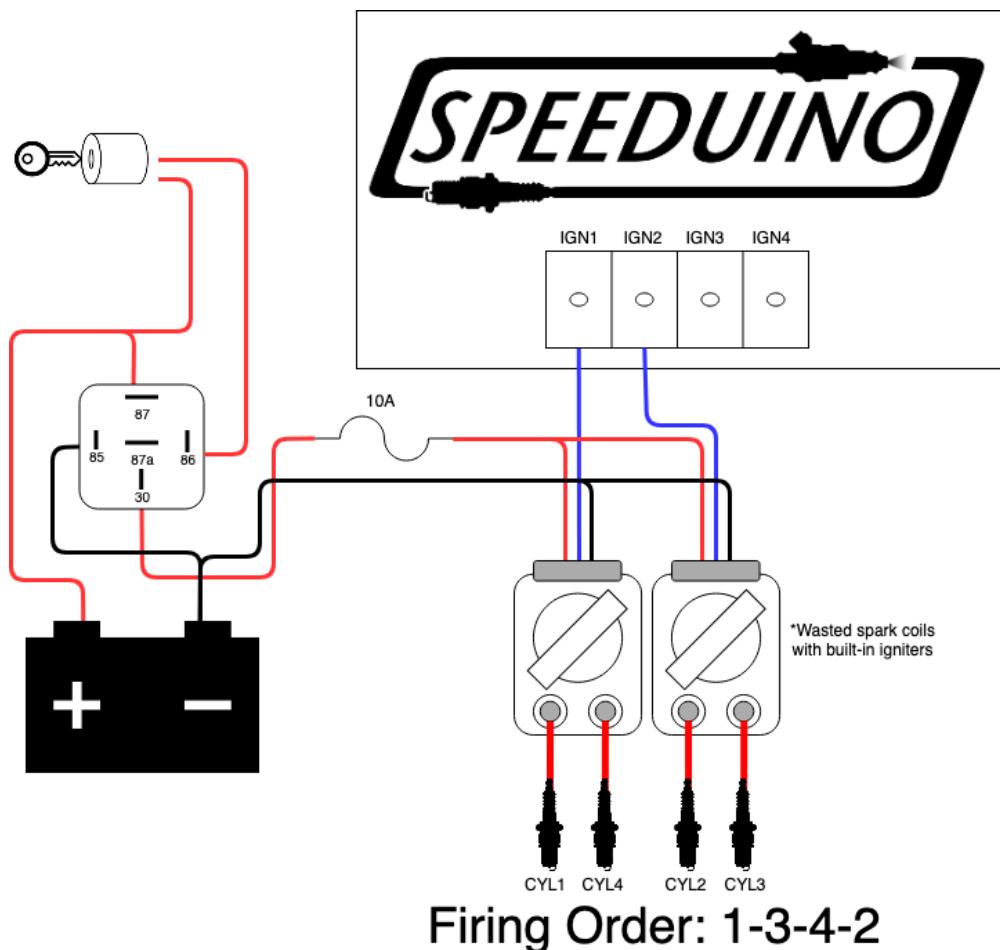


Figura 17: ign_4Cyl_COP_wasted-spark.png

Nota: O exemplo acima usa bobinas 'inteligentes' com ignições integradas. NÃO conecte bobinas de alta corrente (burras) sem adicionar um dispositivo de ignição {.is-warning}

Bobina no plugue

Como uma alternativa para uma bobina de ignição desperdiçada de pólo duplo, a bobina individual em unidades de plugue pode ser usada em uma bobina desperdiçada configuração de faísca.

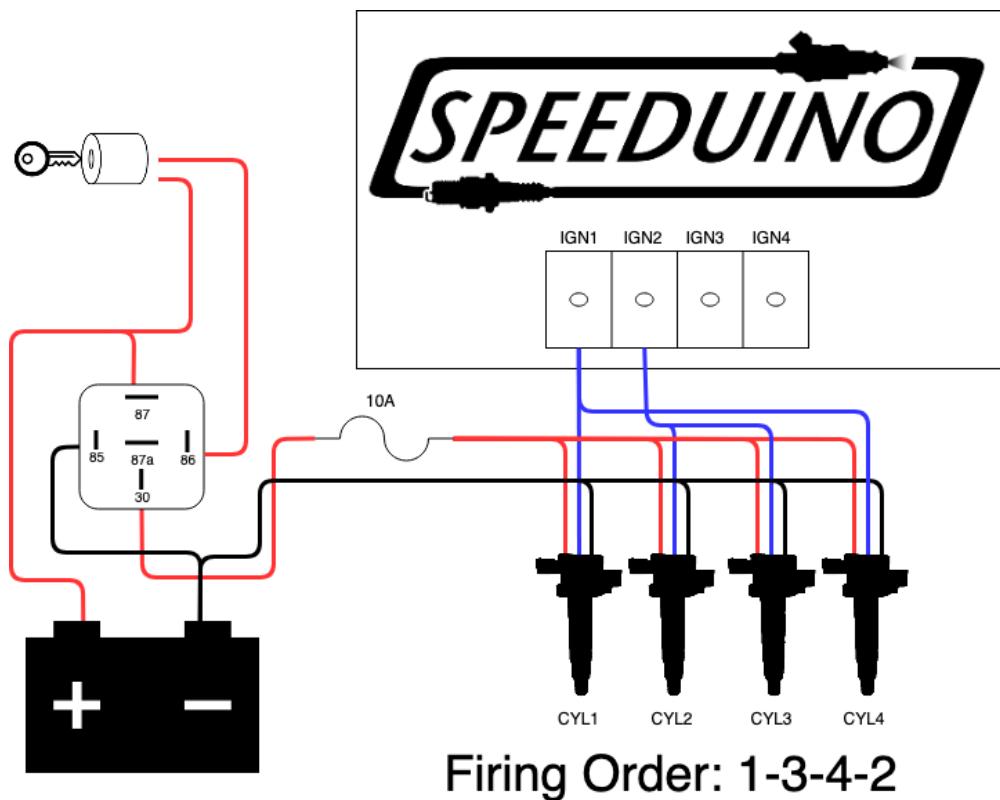


Figura 18: ign_4Cyl_COP_wasted-spark.png

Nota: Os exemplos acima usam bobinas 'inteligentes' com ignições integradas. NÃO anexe COPs idiotas (2alfinetes) sem adicionar um dispositivo de ignição {.is-warning}

Sequencial (COPs)

O controle de ignição sequencial usando Coil-on = Plugs coils simplifica dramaticamente a fiação de ignição. Com esta configuração, cada bobina (e, subsequentemente, cada cilindro) se conecta a uma única saída de ignição, conectado na ordem de disparo.

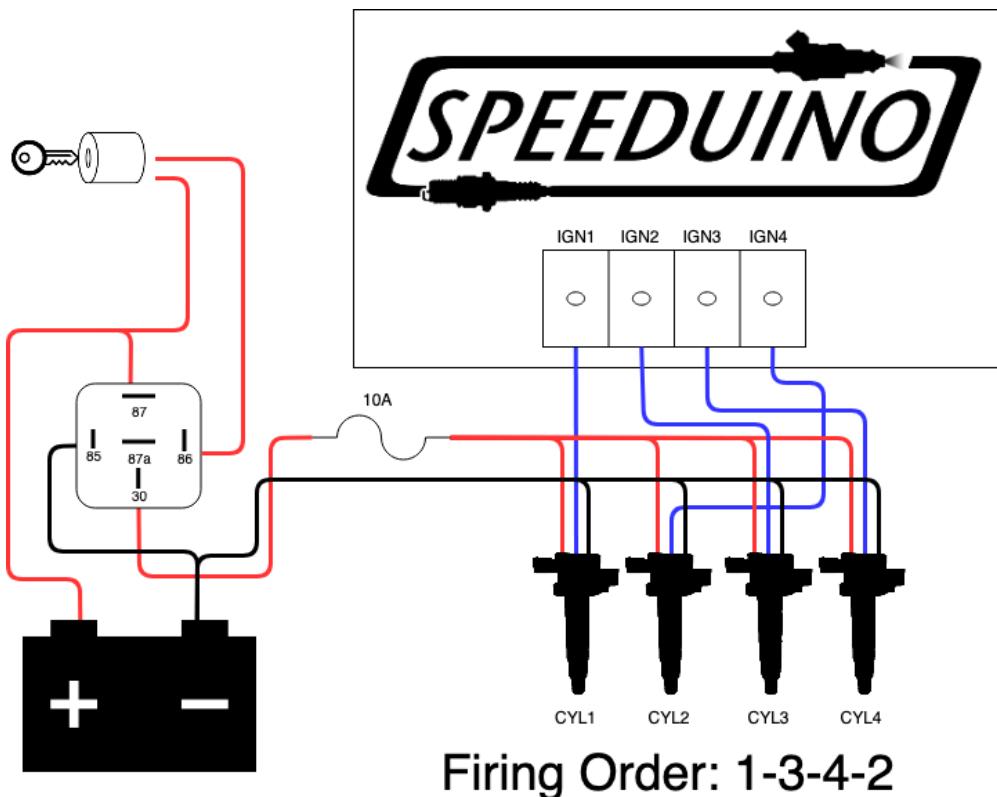
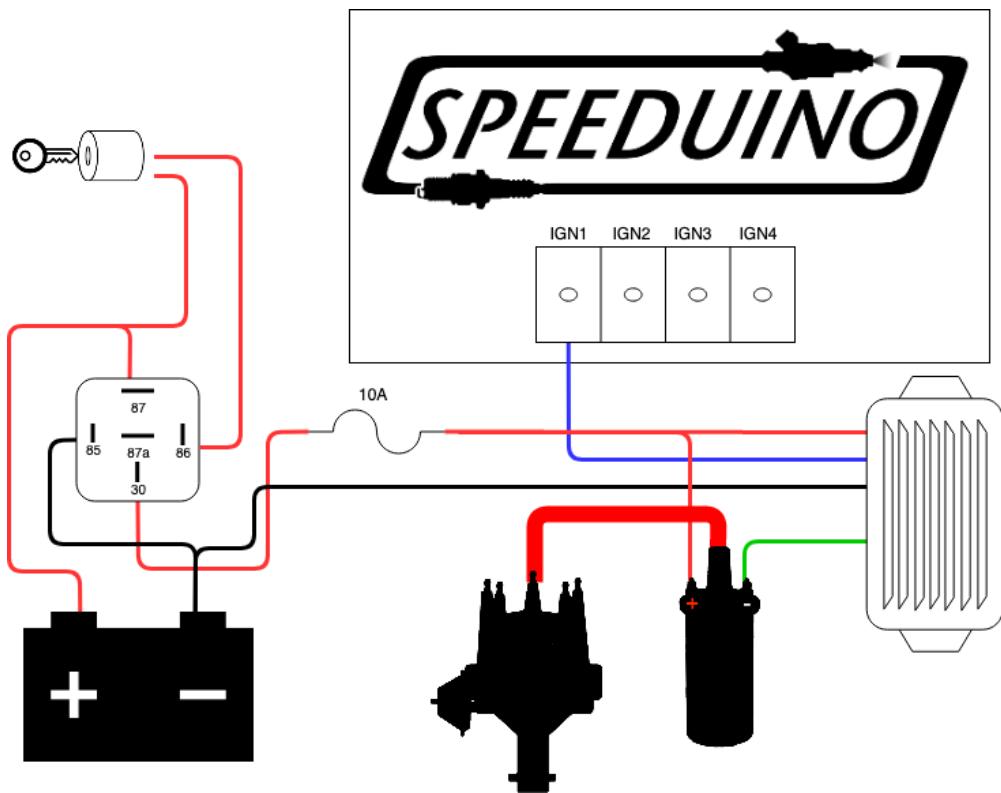


Figura 19: ign_4Cyl_COP_seq.png

Nota: O exemplo acima usa bobinas 'inteligentes' com ignições integradas. NÃO anexe COPs idiotas (2alfinetes) sem adicionar um dispositivo de ignição {.is-warning}

Distribuidor

Se um distribuidor permanecer em uso, apenas uma única saída é necessária da ECU. Isso deve ser alimentado em um módulo de ignição de canal único (como o Bosch 124 comum) que pode então acionar a bobina.



Fiação do sensor analógico

Sensores analógicos fornecem dados como temperaturas, posição do acelerador e leituras de O2 para a ECU. O diagrama abaixo mostra a fiação típica para esses sensores.

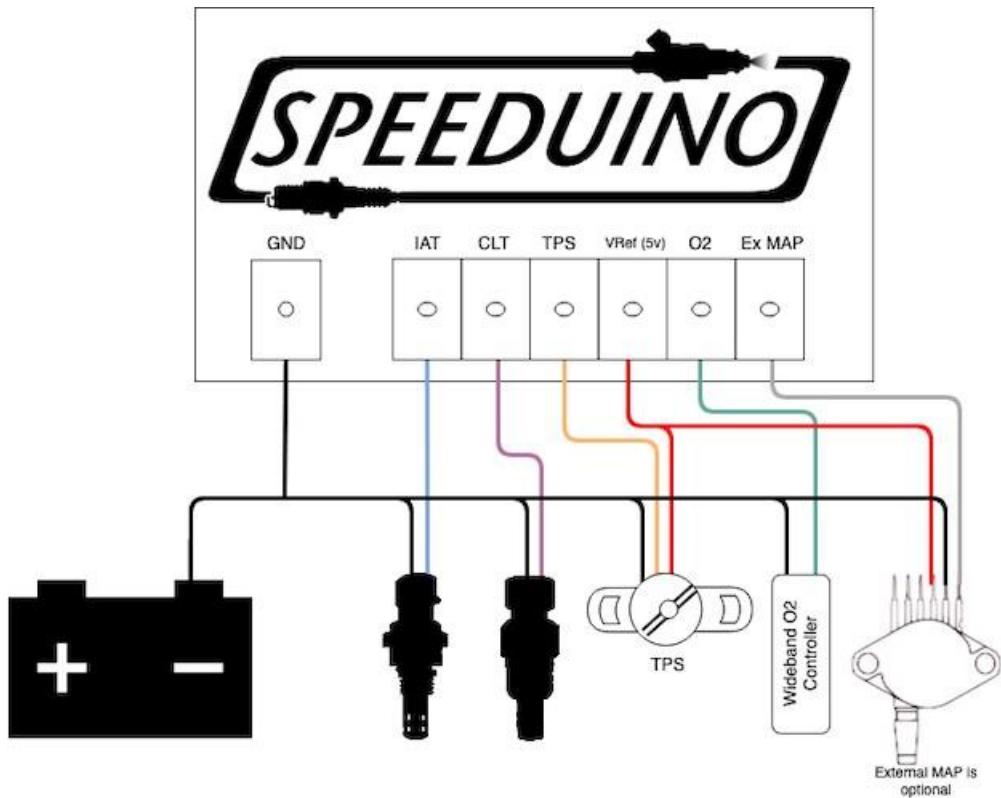


Figura 20: analog_sensors.png

Notas

- O uso de sensores de temperatura de 2 fios é altamente recomendado. Embora os sensores de 1 fio funcionem, eles são quase sempre consideravelmente menos precisos. Executando um fio terra dedicado de volta à ECUDo sensor também é recomendado.
- O sensor de MAP externo no diagrama acima é opcional e pode ser omitido se o MAP a bordo for usado. Alternativamente, um sensor Baro externo pode ser adicionado ao mesmo que um sensor externoMAPA
- É necessário um TPS variável de 3 fios. Os interruptores do acelerador tipo liga / desliga não são adequados

Constantes do motor

Visão geral

No menu Configurações, selecione Constantes



Figura 21: TS_8.png

Aqui você precisa configurar as constantes do motor. Preencha os campos na seção inferior antes de calcular o Combustível Necessário.

Configuração

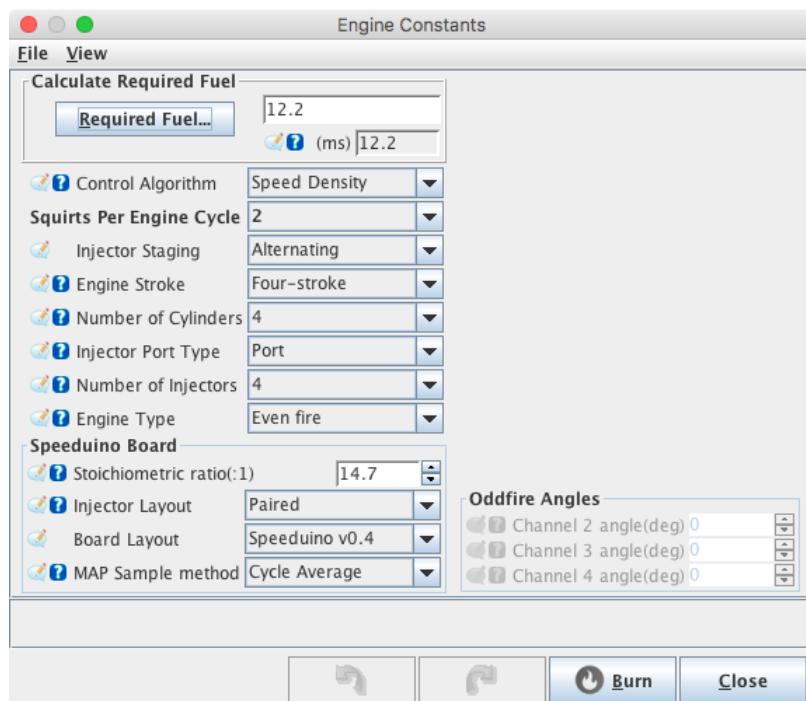


Figura 22: engine.png

Calculadora de combustível necessário

A calculadora de combustível necessário determina o tempo teórico de injeção de combustível que seria necessário em 100% VE. Isso é determinado pelo conhecimento da cilindrada do motor, do tamanho e número dos injetores de combustível e do número de esguichos que serão realizados em cada ciclo. O aumento deste número levará a um aumento geral na quantidade de combustível que é injetada em todos os pontos do mapa de VE (e vice-versa).

Você deve definir todos os valores no [Definições](#) seção abaixo antes de realizar o [Obrigatório Combustível](#) cálculo {.is-info}

Definições

- **Algoritmo de controle:** A fonte de carga que será usada para a mesa de combustível
- **Esguichos por ciclo do motor:** Quantos esguichos serão realizados ao longo da duração do ciclo do motor (por exemplo, 720 graus para um curso de 4 tempos). a maioria dos motores não exigirá valores maiores que
4. Para instalações sequenciais, isso deve ser definido como 2 com o estágio do injetor definido como 'Alternando' (internamente o Speeduino ajustará os esguichos para 1)
 - Observe que para 3 e 5 esguichos, você deve ter um sinal de came além da manivela.
- **Estágio do injetor:** Isso configura a estratégia de tempo usada para os injetores
 - **Alternando** (Recomendado para a maioria das instalações) - Os injetores são sincronizados em torno de cada cilindro TDC. O ângulo de fechamento exato pode ser específico na caixa de diálogo Características do injetor.
 - **Simultâneo** - Todos os injetores são disparados juntos, com base no PMS do cilindro 1.
- **Curso do motor:** Seja o motor é 2 tempos ou 4 tempos
- **Número de cilindros:** Número de cilindros do motor. Para motores rotativos, selecione 4.
- **Tipo de porta do injetor:** A opção não é usada pelo firmware. A seleção atualmente não importa
- **Número de injetores:** Normalmente o mesmo que o número de cilindros (para injeção de porta)
- **Tipo de motor:** Se o ângulo da manivela entre os disparos é o mesmo para todos os cilindros. Se estiver usando um carro de bombeiros ímpar (por exemplo, alguns V-Twins e Buick V6s), o ângulo de cada canal de saída deve ser específico.
- **Layout do injetor:** Especifica como os injetores são conectados
 - **Emparelhado:** 2 injetores são conectados a cada canal. O número de canais usados é, portanto, igual a metade do número de cilindros.
 - **Semi-Sequencial:** Semi-sequencial: O mesmo que emparelhado, exceto que os canais do injetor são espelhados (1 e 4, 2 e 3), o que significa que o número de saídas usadas é igual ao número de cilindros. Válido apenas para 4 cilindros ou menos.
 - **Sequencial:** 1 injetor por saída e saídas usadas é igual ao número de cilindros. A injeção é cronometrada ao longo do ciclo completo. Disponível apenas para motores com 4 ou menos cilindros.
- **Layout da placa:** Especifica o layout do pino de entrada / saída com base na placa Speeduino que você está usando. Para obter detalhes específicos sobre esses mapeamentos de pinos, consulte o arquivo utils.ino
- **Método de amostra MAP:** Como as leituras do sensor MAP serão processadas:

-
- **Instantâneo:** Cada leitura é usada à medida que é feita. É um sinal altamente flutuante, mas pode ser útil para testar

- **Média do ciclo:** A leitura média do sensor em 720 graus de manivela são usados. Esta é a média do evento são as opções recomendadas para 4 ou mais cilindros
- **Ciclo mínimo:** O mais baixo valor detectado em 720 graus é usado. Este é o método recomendado para menos de 4 cilindros ou ITBs
- **Média do evento:** Semelhante à média do ciclo, no entanto realiza a média uma vez por evento de ignição em vez de uma vez por ciclo. Geralmente oferece uma resposta mais rápida com um nível semelhante de precisão.

Os ângulos Oddfire só devem ser usados em motores Oddfire (principalmente alguns V6s específicos)

Características do injetor

Visão geral

Os injetores de combustível têm propriedades de hardware exclusivas que devem ser levadas em consideração em sua configuração. O ideal é que eles sejam fornecidos como parte das especificações de seus injetores; no entanto, em alguns casos, os dados podem não estar disponíveis ou ser difíceis de encontrar. Os valores típicos são fornecidos abaixo como pontos de partida para estes casos.

Definições

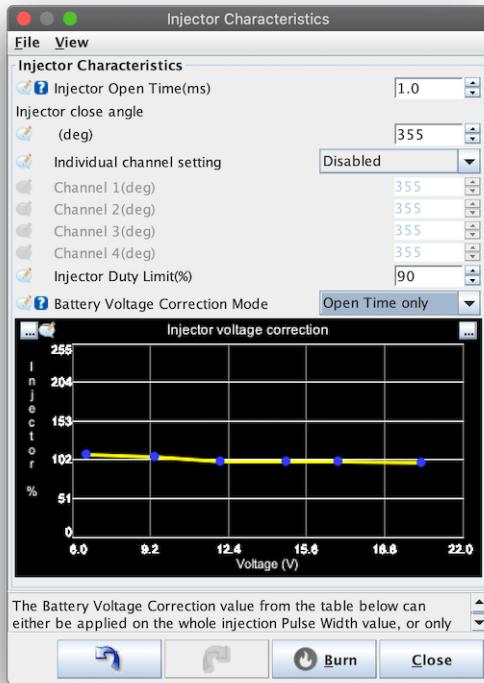


Figura 23: Características do injetor

Típica
Variável valueComment

Injecto0r. O tempo que o injetor leva para abrir completamente depois de disparado, mais o 9 Abrir - tempo necessário para fechar. Isso é específico para cada tipo e versão de injetor.

Tempo 1,5

Injecto3r55 Esta representa o ângulo, relativo ao cilindro TDC, na qual o injetor esguicho vontade. O Ângulo fim. Isso pode ser variado por canal (incluindo para fiação semi-sequencial), mas o valor padrão de 355 é adequado para a maioria das aplicações.

próxi mos

Variável value

Comment típico

Injecto8r5% The injetor abre e fecha uma vez por manivela revolução tão, taking para dentro conta a tempo aberto do injetor, o ciclo de trabalho é limitado para evitar que este ultrapasse o tempo de revolução. Um valor de 85% é recomendado, mas um valor mais alto pode ser usado para injetores de abertura mais rápida. Observe que, uma vez que o limite do ciclo de trabalho seja atingido, ele não será excedido, pois o injetor de combustível não pode fechar e reabrir rápido o suficiente para fornecer mais combustível. Isso pode causar condições enxutas em altas RPM. Se atingir este limite, considere fortemente se injetores maiores são necessários.

Injecto1r00% A porcentagem da largura do pulso do injetor varia com as mudanças na tensão de alimentação.

Volt- Um valor de 100% significa nenhuma alteração na largura de pulso.

era

Correção

VoltageOpenSe a correção de voltagem se aplica apenas ao tempo de abertura ou ao todo pulso

Mod largura de

o de tempo. só

Corr

eção

Configuração do gatilho

Visão geral

Um dos componentes mais críticos de uma configuração EFI é o Sensor de ângulo de manivela (CAS) e como ele é usado pela ECU. A caixa de diálogo de configurações do acionador é onde a configuração do acionador é definida e é de vital importância que isso esteja correto antes de tentar ligar o motor.

Com incorreto configurações, você pode ter problemas para obter a sincronização ou ver leituras de RPM erráticas.

Observe que muitas das configurações nesta caixa de diálogo dependem da sua configuração e, portanto, é normal que algumas opções podem ficar esmaecidas.

Configurações de gatilho

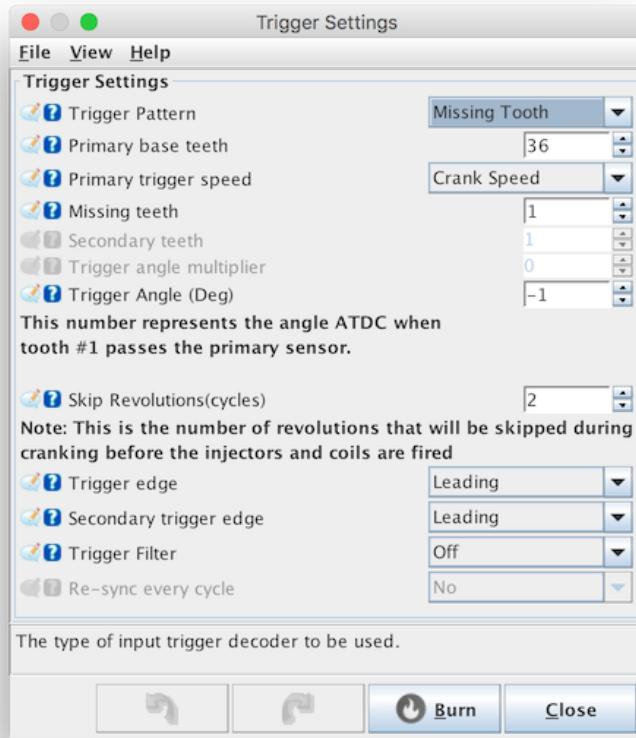


Figura 24: Caixa de diálogo de configurações do acionador

- **Padrão de gatilho** - O padrão usado pela configuração do sensor de manivela / came em seu motor. Para uma lista completados padrões suportados, consulte a página de decodificadores
- **Dentes da base primária** - Para padrões onde o número de dentes é variável (dente faltando, roda dupla, etc.), este número representa o número de dentes na roda primária. Para rodas do tipo dente faltando, este número deve ser a contagem como se não houvesse nenhum dente faltando.
- **Velocidade de disparo primária** - A velocidade em que a entrada principal gira. Está intimamente relacionado com a configuração dos dentes da Base Primária e indica se esse número de dentes passa pelo sensor uma vezcada revolução da manivela ou cada revolução do came.
- **Dentes faltando** - Se estiver usando o padrão de dente ausente, este é o tamanho da lacuna, dado em 'dentes ausentes'. Por exemplo, 36-1 tem 1 dente faltando. 60-2 tem 2 dentes ausentes, etc. Os dentes ausentes DEVEM estar todos localizados em um único

bloco, não pode haver várias lacunas de dente ausentes ao redor da roda.

- **Dentes secundários** - Como acima, mas para a entrada secundária. Esta entrada é sempre assumida para ser executada

na velocidade da câmera.

- **Multiplicador de ângulo de gatilho** - Esta opção é usada apenas no padrão não 360.
- **Ângulo de gatilho** - O ângulo da manivela, após o ponto morto superior (ATDC), quando o dente nº 1 passa pelo sensor na entrada primária (manivela). Essa configuração é crítica para o Speeduino saber com precisão o ângulo atual da manivela. Consulte a seção abaixo ('Localização do dente nº 1 e ângulo de disparo') para obter mais informações sobre como determinar este valor. Você deve usar uma luz de temporização para confirmar se o ângulo está correto depois de calculado. Sem fazer isso, seu ângulo pode estar incorreto.

Acionar opções

- **Pular revoluções** - O número de rotações que o motor deve realizar antes que o sinalizador de sincronização seja definido. Isso pode ajudar a evitar eventos de sincronização falsos ao acionar. Os valores típicos são de 0 a 2
- **Borda do gatilho** - Se o sinal primário dispara na borda ascendente ou descendente. Os condicionadores VR requerem configurações específicas, dependendo do modelo usado. Veja os requisitos de hardware
- **Borda de gatilho secundária** - Se o sinal secundário dispara na borda ascendente ou descendente
- **Filtro de gatilho** - Um filtro de software baseado em tempo que irá ignorar as entradas de manivela / came se elas chegarem antes do esperado com base no RPM atual. Quanto mais agressivo for o filtro, mais próximo do tempo esperado ele operará. Níveis mais altos de filtragem podem fazer com que os pulsos verdadeiros sejam filtrados, portanto, é recomendado usar a configuração mais baixa possível
- **Sincronizar novamente a cada ciclo** - Se definido como sim, o sistema procurará as condições de sincronização a cada ciclo, em vez de apenas contar o número esperado de dentes. Recomenda-se que esta opção seja ativada, no entanto, se você tiver um sinal de arranque / came barulhento, pode ser necessário desligá-lo, pois isso pode fazer com que a sincronização caia ocasionalmente. Depois que o Speeduino estiver totalmente sincronizado, ele continuará a funcionar no modo sequencial total, a menos que ocorra uma perda de sincronização no gatilho de crack.

Encontrando o dente nº 1 e o ângulo de gatilho

Por favor, consulte os padrões de gatilho e decodificadores para o gatilho que você está usando

Densidade IAT

Visão geral

A curva de densidade IAT representa a mudança na densidade de oxigênio da carga de entrada como temperatura sobe.

Curva de Exemplo

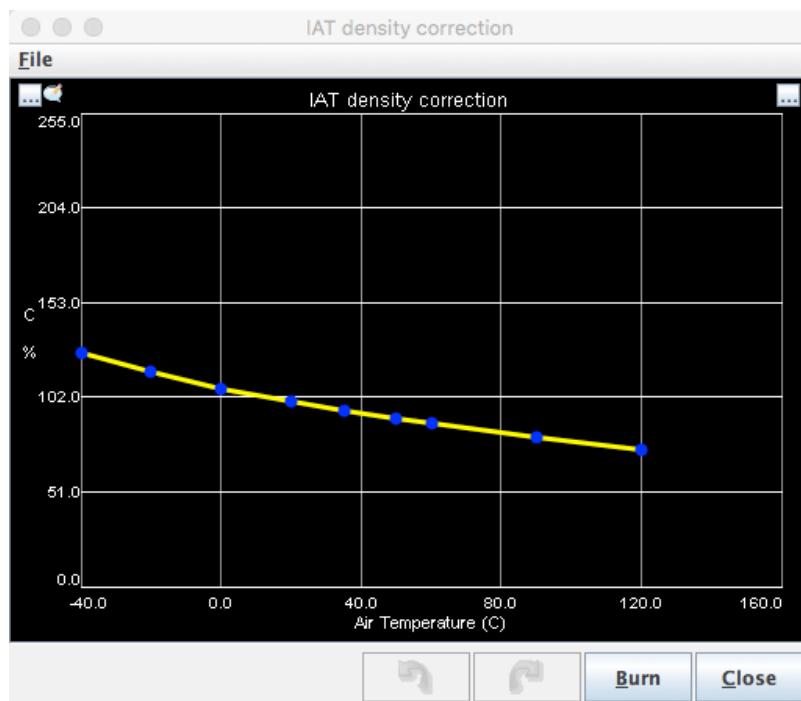


Figura 25: iatDensity.png

Esta curva padrão segue aproximadamente a lei do gás ideal e é adequada para a maioria das instalações, no entanto, se você estiver vendo temperaturas de entrada muito altas (devido ao calor no compartimento do motor ou da turboalimentação), pode ser necessário ajustar a extremidade quente desta curva.

Tabela de combustível (VE)

A tabela de combustível ou VE é o principal método de controle da quantidade de combustível que será injetado no cada ponto de velocidade / carga.

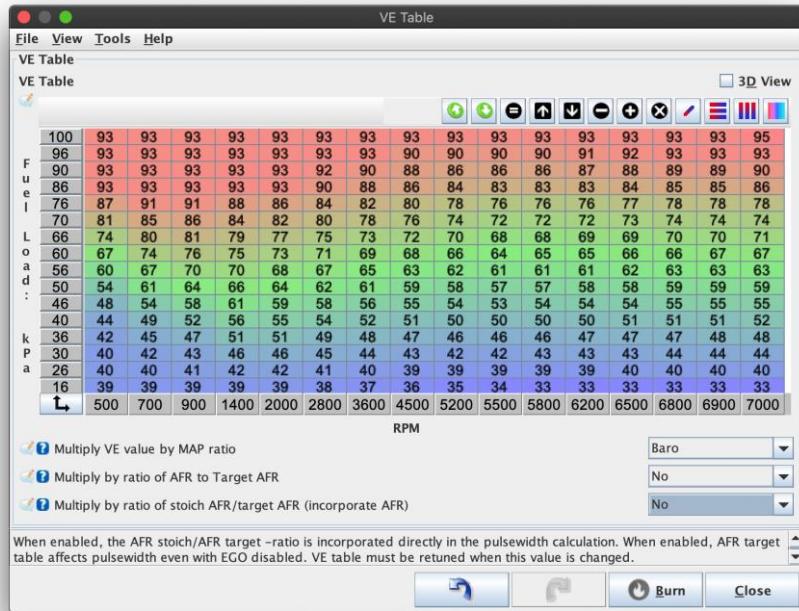


Figura 26: Tabela VE 1

Configuração

O mapa de combustível é uma tabela interpolada 3D que usa RPM e carga de combustível para pesquisar o valor VE desejado. O eixo de carga de combustível é determinado se você está usando Speed Density (MAP kPa) ou Alpha-N (TPS) para sua carga de combustível (Veja Engine_Constants)

Os valores nesta tabela representam uma porcentagem do [Combustível Necessário](#) quantidade que será injetada quando o motor está em um determinado ponto de velocidade / carga.

Opções

- **Multiplique o valor VE pela relação MAP:** Habilitar esta opção 'nivela' a tabela de combustível multiplicando o valor no ponto de velocidade / carga atual pelo valor MAP dividido pelo valor Baro (em kPa) ou 100% fixo (para compatibilidade com sintonizações de outros sistemas).
 - Você pode sintonizar com ou sem essa opção ativada, mas geralmente é recomendável ativá-la, pois permitirá resultados de sintonia mais simples e previsíveis.
 - Para novas músicas, é recomendado usar o [Baro](#) opção

Aviso: Mudando este valor exigirá o reajuste do mapa de combustível! {. is-warning}

- **Multiplique pela proporção de AFR para AFR alvo:** Esta opção é normalmente definida para [Não](#) para a maioria das configurações. Ele permite um feedback básico de circuito fechado, ajustando a quantidade de combustível base de acordo com a distância do alvo AFR que o motor está funcionando no momento.
- **Multiplique pela proporção de Stoich AFR para AFR alvo ('Incorporar AFR'):** Esta opção é semelhante à anterior, mas em vez de usar o valor AFR atual, o AFR de stoich fixo é usado.

Secundário Mesa de combustível

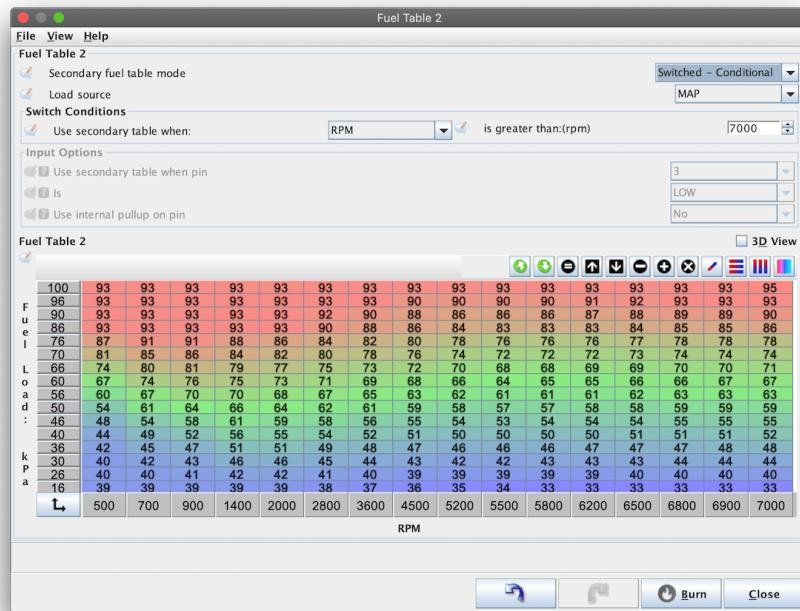


Figura 27: Combustível Secundário tabela

O Speeduino também tem a capacidade de usar uma tabela de combustível secundária que permite o abastecimento de modo combinado e alternado. Existem 2 modos combinados e 2 modos comutados disponíveis.

Os modos de combustível misturado funcionam em conjunto com a tabela de combustível primário para chegar a um único VE combinado. Os modos de combustível alternados são onde a tabela de combustível primária ou secundária é usada, mas não ambas ao mesmo tempo. A tabela que está sendo usada em um determinado momento pode ser configurada com base em uma entrada externa (por exemplo, interruptor de painel) ou definida por meio de certas condições.

% Multiplicado

Este é um modo de combustível misturado (ou seja, ele usa as tabelas de combustível primário e secundário juntas) que permite que diferentes cargas e eixos de RPM sejam combinados. Normalmente, isso é usado para ter tabelas de combustível primário e secundário com diferentes fontes de carga (por exemplo: mapa primário usando TPS e mapa secundário usando pressão múltipla).

Este modo é frequentemente usado em motores com corpos de aceleração individuais (ITBs) para permitir TPS e MAP com base tabelas a serem combinadas.

O valor final do combustível é derivado do tratamento de ambos os valores (primário e secundário) como porcentagens e multiplicando-os juntos.

Exemplo 1

- Valor da tabela de combustível primário: 75
- Valor da tabela de combustível secundário: 100
- Valor final: 75

Exemplo 2

- Valor da tabela de combustível primário: 80
- Valor da tabela de combustível secundário: 150
- Valor final: 120

Exemplo 3

- Valor da tabela de combustível primário: 90
- Tabela de combustível secundária valor: 80
- Valor final: 72

Adicionado

Este é um modo de mistura de combustível muito semelhante ao anterior [Multiplicado %](#) modo. A única diferença entre os dois é que, em vez de multiplicar os valores das tabelas primária e secundária, o2 são somados.

Este é um modo menos comumente usado, mas é uma alternativa nas mesmas configurações que você usaria

[Multiplicado %](#)

Comutado - Condisional

O modo comutado condicional permitirá o uso da 2^a tabela de combustível quando um certo valor ultrapassar um nível definido. Os valores de comutação disponíveis são:

- RPM
- Teor de etanol
- MAPA
- TPS

Dependendo do resultado desejado, isso pode ser usado para expandir a resolução da tabela principal de combustível, lidar automaticamente com combustíveis alternativos ou como um modo ITB alternativo (especialmente se o funcionamento for acelerado ITBs).

Comutado - baseado em entrada

O modo de comutação baseado em entrada permite que você altere a tabela de combustível que está em uso por meio de uma entrada externa para a ECU. As opções exigidas são:

- O pino (Arduino) ao qual a entrada está conectada
- A polaridade desta entrada (IE é a tabela de combustível secundária usada com o sinal é alto ou baixo). Para uma entrada de comutação de aterramento padrão, isso deve ser [BAIXO](#)
- Se deve usar o pullup interno nesta entrada. Para uma entrada de comutação de aterramento padrão, este deve estar [sim](#)

Enriquecimento de aceleração (AE)

O Enriquecimento de Aceleração (AE) é usado para adicionar combustível extra durante o curto período transiente após um rápido aumento no acelerador. Ele executa quase a mesma função de uma bomba aceleradora em um motor com carburador, aumentando a quantidade de combustível entregue até que a leitura da pressão do coletor se ajuste com base em uma nova carga.

Para operar o AE baseado em TPS corretamente, você deve ter um TPS variável instalado e calibrado. {é-info}

Teoria

O ajuste do enriquecimento da aceleração é baseado na taxa de mudança da posição do acelerador, uma variável conhecida como TPSdot (TPS delta ao longo do tempo). Isso é medido em% / segundo, com valores mais altos representam

pressões mais rápidas do acelerador e valores na faixa de 50% / sa 1000% / s são normais. Por exemplo:

- 100% / s = pressionando o acelerador de 0% a 100% em 1 segundo
- 1000% / s = pressionando o acelerador de 0% a 100% em 0,1s

TPSdot forma o eixo X da curva de aceleração, com o valor do eixo Y representando o aumento% em combustível.

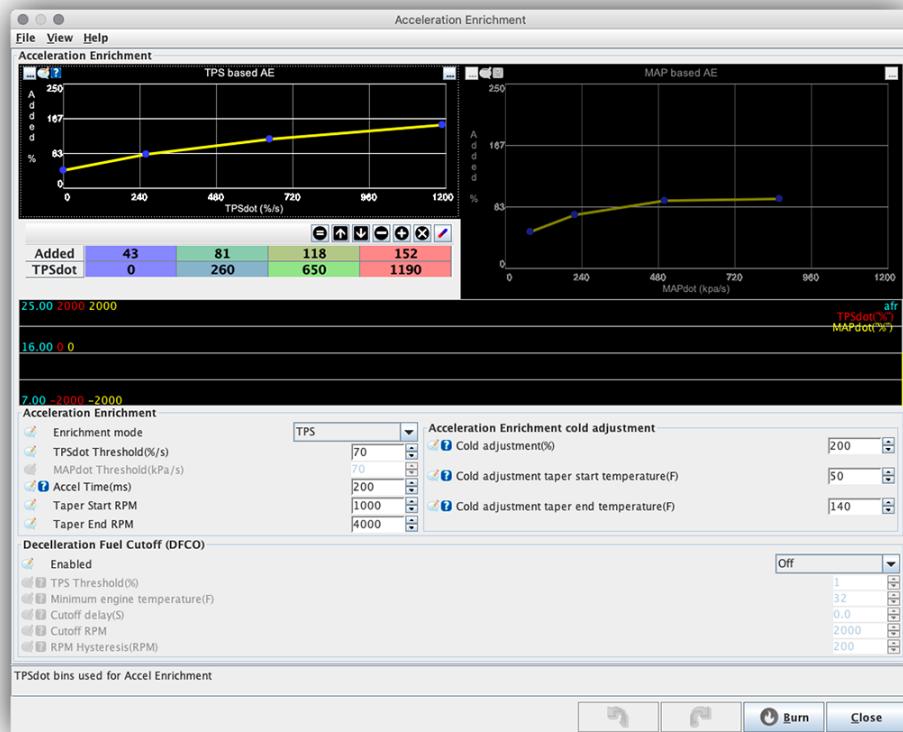


Figura 28: Curvas de enriquecimento de aceleração

Tuning

A curva de enriquecimento incluída com o ajuste básico do Speeduino é um bom ponto de partida para a maioria dos motores, mas alguns ajustes são normais dependendo do tamanho do injetor, diâmetro do acelerador, etc.

Na maioria dos casos, o ajuste da curva AE pode ser executado em um ambiente estacionário, embora o dinamômetro ou o ajuste de estrada também sejam possíveis. Blips rápidos e lentos do acelerador devem ser executados e o efeito nos AFRs monitorados usando o gráfico de linha ao vivo na caixa de diálogo AE. Este gráfico mostra os valores TPSdot e AFR em sincronia entre si, tornando os ajustes para a parte correta da curva AE mais simples para identificar.

Se você achar que o AFR é inicialmente bom, mas depois diminui brevemente, você deve aumentar o 'Tempo de aceleração' configuração, com incrementos de 10-20ms recomendados.

Falsa ativação

Nos casos em que o sinal TPS é ruidoso, picos em sua leitura podem desencadear incorretamente o enriquecimento da aceleração. Isso pode ser visto em um arquivo de log ou em um painel ao vivo no TunerStudio pela ativação do indicador 'TPS Accel' quando não há nenhum (ou pouco) movimento do acelerador ocorrendo.

Caso isso ocorra (e supondo que a fiação TPS não possa ser corrigida para reduzir o ruído), os disparos falsos podem ser impedidos de disparar AE aumentando o valor "Limite TPSdot". Isso deve ser aumentado em incrementos de ~ 5% / s, pausando entre cada aumento para observar se AE ainda está sendo ativado incorretamente.

Campos

- **Modo de enriquecimento** Escolha se deseja usar o sensor de posição do acelerador ou pressão absoluta do coletor para enriquecimento de aceleração.
- **Limite TPSdot** Porcentagem de mudança de posição do acelerador por segundo necessária para acionar o enriquecimento da aceleração. Por exemplo, se definido como 70, a posição do acelerador deve mudar a uma taxa de 70% por segundo para que o enriquecimento de aceleração se torne ativo.
- **Limiar MAPdot** Igual ao TPSdot Threshold, mas se aplica ao usar o modo de enriquecimento de MAP.
- **Tempo de aceleração** Duração do enriquecimento de aceleração. Assim que o enriquecimento for acionado, ele durará estes muitos milissegundos.
- **Taper Start RPM, Taper End RPM** Dimensiona a redução do enriquecimento em diferentes RPMs. Se o RPM for menor ou igual ao RPM inicial, o enriquecimento será 100% do valor de enriquecimento calculado, com base no valor TPSdot (ou MAPdot) visto. Se o RPM for maior ou igual ao RPM final, o enriquecimento será de 0%. À medida que o RPM aumenta, a quantidade total de enriquecimento necessário diminui. Enriquecimento é escalado linearmente entre esses valores.
- **Ajuste Frio** Dimensiona a porcentagem de enriquecimento de aceleração linearmente com base na temperatura do refrigerante. Na temperatura inicial, o ajuste será igual ao campo Ajuste de frio (%). Na temperatura final, o ajuste será 0%.
- **Combustível de desaceleração Cortar** Para a injeção de combustível quando: RPM está acima de Cutoff RPM TPS está abaixo de TPS Threshold Temperatura do motor está acima Temperatura mínima do motor As condições acima são atendidas para segundos de atraso de corte ** RPM histerese pode ser ajustada para levar em conta a flutuação Condições de RPM para evitar DFCO acidental.

AFR / O2 (combustível de circuito fechado)

AFR / O2 (para relação ar: combustível), diálogo controla o controle de combustível de circuito fechado, usado para ajustar a carga do injetor com base na entrada de um sensor de oxigênio de exaustão (sensor de O2). Em conjunto com a Tabela AFR, o sistema AFR de circuito fechado irá comparar a leitura real de O2 com a taxa de combustível alvo atual e faça os ajustes necessários.

O uso de um sensor e controlador de banda larga é altamente recomendado, no entanto, a funcionalidade básica é possível com um sensor de banda estreita se não estiver disponível.

Observe que o controle de combustível de circuito fechado não substitui um ajuste ruim. Muitas configurações boas não usam o controle de malha fechada ou permitem apenas uma autoridade de ajuste muito pequena.

Definições

O Speeduino oferece suporte a 2 algoritmos de loop fechado, cada um destinado a diferentes configurações:

1. **Simples** - Um algoritmo de 'perseguimento do alvo' baseado no tempo, em que a quantidade de ajuste de combustível depende de quanto tempo a leitura foi pobre ou rica em comparação com o alvo atual. Este algoritmo é mais adequado para sensores de banda estreita onde apenas informações básicas ricas / enxutas estão disponíveis. Em particular, este algoritmo funciona mal se você tiver um mapa de combustível que não está quase completo. Se você tiver ativado e estiver vendo oscilações na largura de pulso e / ou AFRs, mesmo quando em cruzeiro, você deve desativar o controle de circuito fechado até que o MAP de combustível de base seja melhor sintonizado.
2. **PID** - Este é o algoritmo de loop fechado preferido e fornecerá melhores resultados quando combinado com um sensor de banda larga e ajustado corretamente.

Variáveis comuns

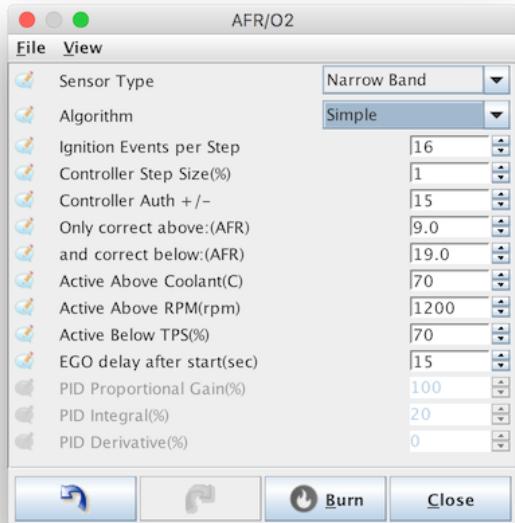


Figura 29: o2_simple.png

- **Tipo de sensor** - Banda estreita ou banda larga, dependendo da configuração do hardware. O sensor de banda estreita deve ser do tipo 0-1v, os sensores de banda larga devem ter um sinal de 0-5v. Os sensores de banda larga precisam ser calibrados na caixa de diálogo Ferramentas-> Calibrar Tabela AFR
- **Algoritmo** - Veja acima a descrição de cada algoritmo disponível
- **Eventos de ignição por etapa** - O cálculo de ajuste de AFR será executado a cada muitos ciclos de ignição. As alterações no ajuste de malha fechada normalmente têm algum retardo antes de seu impacto ser registrado pelo sensor de O2 e o aumento desse valor pode levar esse retardo em consideração. Típicos valores são 2-5.
- **Tamanho da etapa do controlador** -
- **Autenticação do controlador** - A% máxima que a largura de pulso pode ser alterada por meio desse ajuste de malha fechada. O valor recomendado não é superior a 20%.
- **Corrigir acima / abaixo de AFR** - A faixa AFR em que os ajustes de loop fechado serão aplicados. Esta faixa é normalmente limitada pelo sensor e controlador em uso.
- **Ativo acima do refrigerante** - O circuito fechado deve operar apenas quando o motor estiver na temperatura de operação. Este valor deve ser definido para corresponder à temperatura operacional padrão dos motores.
- **Ativo acima de RPM** - Os ajustes de loop fechado geralmente não devem ser feitos em modo inativo. Use este valor para especificar quando o ajuste deve começar a ser feito.
- **Ativo abaixo do TPS** - Acima deste valor TPS, os ajustes de loops fechados serão desabilitados

- **Atraso EGO após o início** - Todos os sensores de O2 requerem um período de aquecimento antes que suas leituras sejam válidas. Isso varia de acordo com o sensor em uso, mas 15s é um valor seguro na maioria dos casos.

Variáveis somente PID



Figura 30: o2_pid.png

- **P / I / D** - Ganho Proporcional PID, Integral e Derivativo percentagens.

Estas opções são adicionais às condições Simples e especificam os parâmetros do circuito fechado Operação

Limiters

Speeduino inclui uma rotação baseada em faísca limitada com cortes duros e suaves.

O limitador de corte suave irá travar o tempo em um valor absoluto para desacelerar ainda mais a aceleração. Se os RPMs continuarem a subir e atingir o limite de corte brusco, os eventos de ignição cessarão até que o RPM caia abaixo destelimiar.

Observação Como esta é uma limitação baseada na faísca, as instalações de apenas combustível não podem usar a funcionalidade do limitador de rotação {is- info}

Definições



Figura 31: Configurações do limitador de rotação

- **Limite de rotação suave:** O RPM no qual o tempo de ignição com corte suave será aplicado.
- **Tempo absoluto de limite suave:** Enquanto o motor estiver acima do limite suave de RPM, o avanço da ignição será mantido neste valor. Valores mais baixos aqui terão um efeito de corte suave maior.
- **Limite suave de tempo máximo:** O número máximo de segundos durante os quais o limitador suave irá operar. Se o motor permanecer na região de RPM de corte suave por mais tempo, o corte duro será aplicado.
- **Limitador de rotação forte:** Acima deste RPM, todos os eventos de ignição cessarão.

Combustível flex

Visão geral

O Speeduino tem a capacidade de modificar as configurações de combustível e ignição com base no teor de etanol do combustível usado, uma prática normalmente conhecida como flex fuel. Um sensor de combustível flexível é instalado nas linhas de alimentação ou retorno de combustível e um fio de sinal é usado como uma entrada na placa Speeduino.

Como o etanol é menos denso em energia, mas também possui uma octanagem equivalente mais elevada, ajustes no combustível carga e tempo de ignição são necessários.

Hardware

O Speeduino usa qualquer um dos sensores de combustível GM / Continental Flex amplamente disponíveis e usados em uma ampla variedade de veículos. Eles estavam disponíveis em 3 unidades diferentes, todas funcionalmente idênticas, com a principal diferença sendo apenas o tamanho físico e o conector. A parteos números para estes são:

- Pequeno - # 13577429
- Tamanho médio - # 13577379
- Largo - # 13577394 (igual ao de tamanho médio, mas com tubos mais longos)

Todos os 3 usam uma variante do conector da série Delphi GT150. Você pode usar um conector GT150 genérico, masvocê terá que cortar 2 guias da lateral do sensor.

Números de peça :

- Alojamento (# 13519047)
- Alfinetes (# 15326427)
- Selo (# 15366021)

Como alternativa, há uma peça GM para um conector de chicote, número da peça 13352241: <http://www.gmpartsdirect.com/oe-gm/13352241>

Fiação

Todas as unidades são conectadas de forma idêntica e têm marcações na caixa indicando para que serve cada pino (12v, terra e sinal). As placas Speeduino v0.3.5 + e v0.4.3 + têm um local de entrada em suas áreas proto ao qual o fio de sinal pode ser conectado diretamente.

Em placas anteriores a essas, você precisará adicionar um resistor pullup entre 2k e 3,5k Ohm. O valor recomendado é 3,3k, no entanto, qualquer resistor nesta faixa funcionará. Observe que esta é uma faixa relativamente restrita, valores mais genéricos como 1k ou 10k NÃO FUNCIONAM com esses sensores.

Tuning

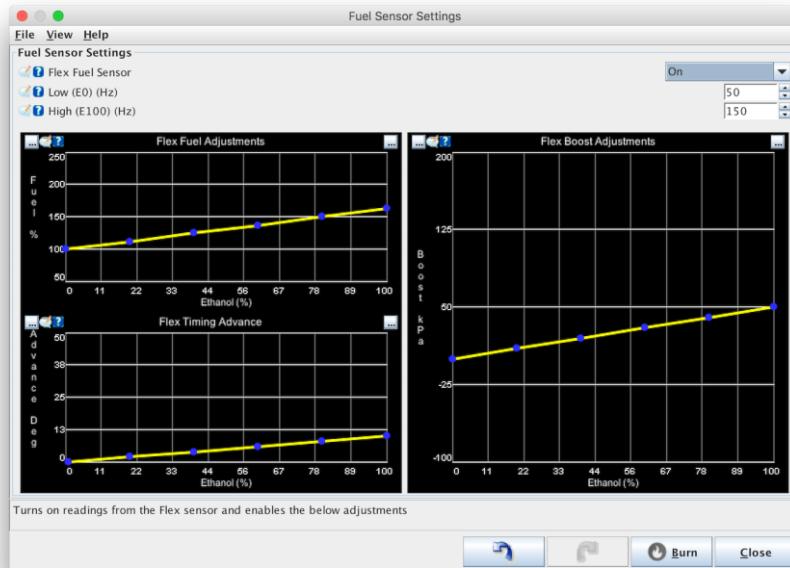


Figura 32: flex_settings.png

- **Frequência do sensor** - A frequência mínima e máxima do sensor que representam 0% e 100% de etanol, respectivamente. Para sensores flex GM / Continental padrão, esses valores são 50e 150
- **Multiplicador de combustível%** - Este é o combustível adicional que deve ser adicionado conforme o teor de etanol aumenta. O valor Baixo à esquerda representa o ajuste do mapa de combustível a 0% de etanol e normalmente será de 100% se o ajuste base foi executado com combustível E0. Se a melodia base foi feita com E10 ou E15, este valor pode ser ajustado abaixo de 100%. O valor alto representa o multiplicador de combustível em etanol 100% (E100) e o valor padrão de 163% é baseado na diferença teórica na densidade de energia entre E0 e E100. O ajuste deste valor pode ser necessário
- **Avanço adicional** - Os graus adicionais de avanço que serão aplicados conforme o teor de etanol aumenta. Este valor aumenta linearmente entre os valores baixo e alto e é adicionado apóstodos os outros modificadores de ignição foram aplicados.

Injeção faseada

Visão geral

O Speeduino tem a capacidade de controlar um estágio de combustível secundário para motores que possuem 2 conjuntos de injetores, normalmente de capacidades diferentes. Embora existam poucos motores de estoque que vêm com injetores secundários (a notável exceção sendo muitos rotativos Mazda), a injeção secundária em estágios é uma modificação comum, em particular usada sempre que grandes injetores são necessários, mas onde é desejável mantê-los menores injetores para desempenho mais suave de baixo RPM.

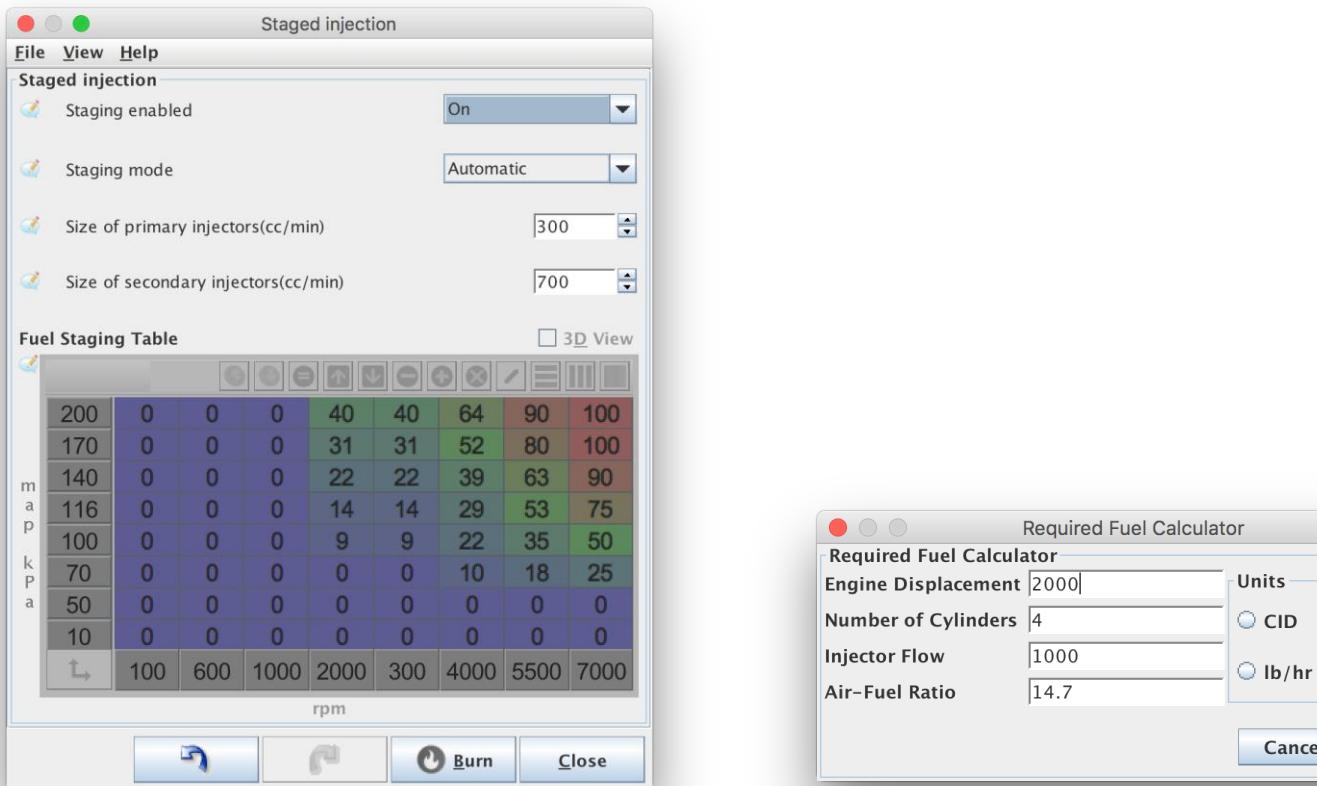
Configuração

Independentemente da estratégia de controle escolhida, você deve inserir o tamanho dos injetores primários e secundários para permitir que o Speeduino saiba a divisão no abastecimento geral.

CRÍTICO - O valor req-Fuel nas Constantes do Motor DEVE ser atualizado quando a injeção em estágios é ativada. Quando o teste está em uso, o valor inserido na calculadora req_fuel DEVE ser igual à soma dos tamanhos dos injetores primário e secundário. A falha em definir esses valores corretamente resultará em condições excessivamente ricas ou pobres. {is-perigo}

Por exemplo:

- Injetores primários: 300cc
- Injetores secundários: 700cc
- Valor inserido na calculadora req_fuel: 1000cc



Métodos de controle

O Speeduino oferece 2 modos de controle de preparação, cada um com seus próprios pontos fortes e fracos. Na maioria dos casos, é recomendado começar com o modo Automático, que requer apenas o ajuste da tabela VE padrão e uma revisão para ver se você obtém o resultado desejado. Somente se isso não puder ser ajustado para fornecer uma divisão de combustível satisfatória, a mudança para o ajuste manual da mesa seria recomendada.

Encenação automática Ao usar o método de escalonamento automático, o Speeduino leva em consideração a capacidade total dos injetores (ou seja, a soma dos 2 estágios do injetor) e realizará uma divisão deles ele mesmo. Com este método, o usuário pode simplesmente ajustar a mesa VE da mesma maneira como se apenas um único conjunto de injetores foram usados e o sistema cuidou do resto.

Neste modo, o Speeduino tentará usar os injetores primários até seu 'Limite de trabalho do injetor' (conforme configurado na caixa de diálogo Características do injetor. Quando o teste estiver sendo usado, é recomendável que este limite não seja superior a 85%. os injetores

começará a realizar qualquer abastecimento adicional dos injetores secundários. Desta forma, a tabela VE é tudo o que é necessário para o ajuste, pois o sistema cuidará de alocar a carga de combustível atual para o melhor injetor.

Controle de mesa O controle de tabela permite o uso de um mapa manual 8x8 que indica qual porcentagem da carga de combustível será realizada pelos injetores secundários - 0% = Injetores secundários desabilitados - 100%

= Injetores primários desativados

É importante observar que os valores nesta tabela NÃO correspondem diretamente à divisão do ciclo de serviço ou largura de pulso. Eles representam a porcentagem da carga total de combustível que os secundários serão solicitados a realizar. O efeito que este valor tem na largura do pulso depende da proporção do primário e capacidades do injetor secundário.

Uma desvantagem do método de ajuste de mesa é que ele não permite que a carga total de combustível dos injetores primários e secundários seja usada simultaneamente. Como a tabela é uma divisão da carga total de combustível, à medida que um conjunto de injetores tem mais desempenho, o outro tem menos desempenho.

Fiação Assumindo um motor de bombeiros uniforme de 4 cilindros, os injetores devem ser ligados em pares. Injetores primários nas saídas 1 e 2. O secundário nas saídas 3 e 4.

Para outras configurações, acesse o fórum para esclarecimentos.

Observação

O tempo morto dos 2 conjuntos de injetores é atualmente considerado o mesmo. Isso pode ser alterado em firmwares futuros, se necessário (poste uma solicitação de recurso, se necessário).

Configurações do Spark

Visão geral

A caixa de diálogo de configurações do Spark contém as opções de como as saídas de ignição funcionarão, incluindo quais das 4 saídas IGN são usadas e como. Eles são críticos e os valores incorretos farão com que o motor não dê partida e, em alguns casos, podem ocorrer danos ao hardware. Esta caixa de diálogo também contém várias opções para fixar o tempo de ignição para teste e diagnóstico.

Certifique-se de ter revisado essas configurações antes de tentar ligar o motor.

Para gerar um mapa de tempo base que fornecerá números melhores do que o mapa padrão do speedy loader, existem várias ferramentas online como: <http://www.useeasydocs.com/theory/spktable.htm> use-os por sua própria conta e risco e sempre escute se há pré-detonação / batida. É melhor ajustar as tabelas de faísca em uma estrada rolante ou dinamômetro.

Definições

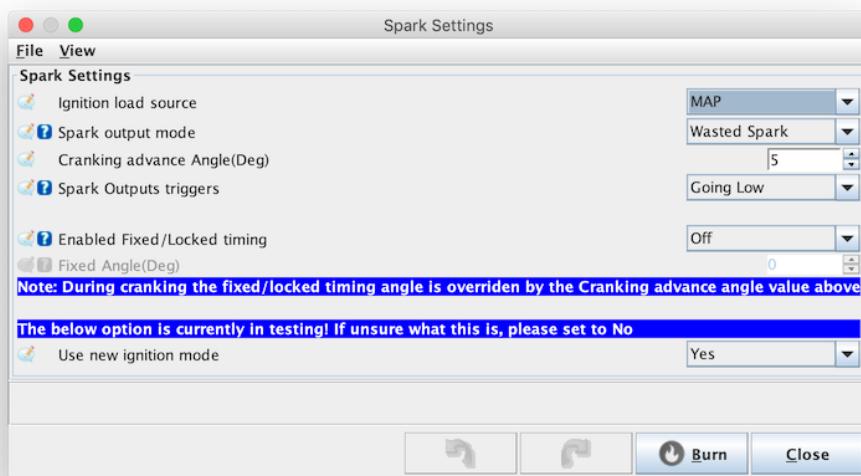


Figura 33: spark_settings.png

- **Modo de saída de faísca** - Determina como os pulsos de ignição serão emitidos e é muito específico para a fiação de ignição. Observe que, independentemente da opção selecionada aqui, os sinais de ignição SEMPRE disparam em ordem numérica (ou seja, 1-> 2-> 3-> 4) até o número máximo de saídas. A ordem de ignição do motor é contabilizada na ordem da fiação.
 - **Faísca Desperdiçada** - O número de saídas de ignição é igual à metade do número de cilindros e cada saída disparará uma vez a cada volta da manivela. Portanto, uma faísca ocorrerá durante o curso de compressão e a outra no curso do escapamento (também conhecida como faísca 'desperdiçada'). Este método é comum em muitos veículos dos anos 80 e 90 que vieram com bobinas de ignição específicas, mas também pode ser usado com bobinas individuais conectadas aos pares. A faísca desperdiçada funcionará apenas com uma referência de ângulo da manivela (por exemplo, uma roda dentada faltando sem sinal de câmera)
 - **Canal único** - Este modo envia todos os pulsos de ignição para a saída IGN1 e é usado

quando o motor contém um distribuidor (normalmente com uma única bobina). O número de pulsos de saída

por rotação (manivela) é igual à metade do número de cilindros.

- **COP desperdiçado** - Este é um modo de conveniência que usa o mesmo tempo que a opção 'Ignição de centelha', no entanto, cada pulso é enviado para 2 saídas de ignição em vez de uma. Estes são emparelhados IGN1 / IGN3 e IGN2 / IGN4 (ou seja, quando IGN1 é alto, IGN3 também será alto). Como este ainda é um modo de temporização de ignição desperdiçado, apenas a posição da manivela é necessária e haverá 1 pulso por par, por rotação da manivela. Este modo pode ser útil nos casos em que existem 4 bobinas individuais, mas a execução sequencial total não é desejada ou não é possível (por exemplo, quando não há camea referência está disponível).
 - **Sequencial** - Este modo só funciona em motores com 4 ou menos cilindros.
 - **Rotativo** - Veja abaixo todos os detalhes
- **Arranque de avanço** - O número de graus absolutos (BTDC) para os quais o tempo será definido ao dar a partida. Isso substitui todos os outros modificadores de avanço de tempo durante a partida.
 - **Gatilhos de saída de faísca** - ESTA É UMA CONFIGURAÇÃO CRÍTICA! Selecionar a opção incorreta aqui pode causar danos às ignições ou bobinas. Especifica se a bobina disparará quando a saída de ignição do Speeduino ficar ALTA ou BAIXA. A maioria das configurações de ignição VAST exigirá que isso seja definido como GOING LOW (ou seja, a bobina carrega / demora quando o sinal é alto e dispara quando o sinal é baixo). Embora IR BAIXO seja necessário para a maioria das configurações de ignição, existem algumas configurações que realizam o tempo de espera no módulo de ignição e disparam a bobina apenas quando elas recebem um sinal HIGH da ECU.
 - **Ângulo Fixo** - Isso é usado para travar o ponto de ignição em um ângulo específico para teste. Definir isso para qualquer valor diferente de 0 resultará no ângulo exato sendo usado (ou seja, substituindo quaisquer outras configurações) em todos os pontos de RPM / carga, exceto durante a partida (a partida sempre usa a configuração de avanço de partida acima). Essa configuração deve ser definida como 0 para operação normal.

Modos rotativos

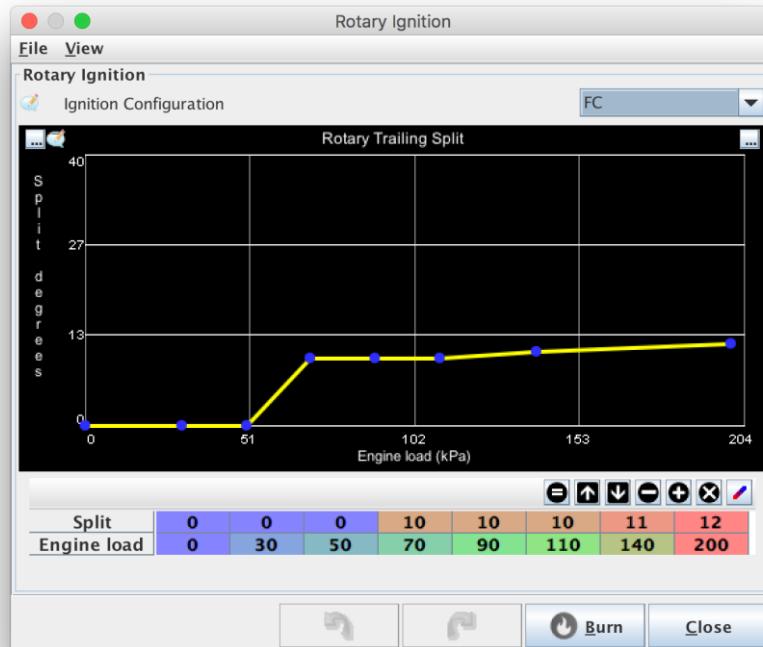


Figura 34: rotary_settings.png

O Speeduino suporta as configurações de ignição encontradas nos motores FC / FD RX7 e RX8 e esta opção torna-se disponível quando o modo de ignição rotativa é selecionado acima. O ângulo de divisão anterior / posterior pode ser definido como uma função da carga atual do motor.

- **FC** - As saídas são configuradas para Leading / Trailing configuração que foi usada em FC RX7s. A fiação é:
 - IGN1 - Fagulhas principais (desperdiçadas)
 - IGN2 - Faísca final
 - IGN3 - Seleção final
 - IGN4 - Não usado
- **FD** - Usa o mesmo sinal de faísca desperdiçado para ambas as faíscas principais como FC, mas sinais individuais para as faíscas à direita. A fiação é:
 - IGN1 - Fagulhas principais (desperdiçadas)
 - IGN2 - Rotor dianteiro arrastando
 - IGN3 - Tração do rotor traseiro

- IGN4 - Não usado
- RX8 - Saídas individuais são usadas para cada sinal de faísca. A fiação é:
 - IGN1 - Rotor dianteiro principal
 - IGN2 - Guia do rotor traseiro
 - IGN3 - Rotor dianteiro arrastando
 - IGN4 - Tração do rotor traseiro

Dwell Control

Visão geral

A caixa de diálogo de controle de parada altera o tempo de carga da bobina (parada) para as saídas de ignição do Speeduino. Deve-se ter cuidado com essas configurações, pois os dispositivos de ignição e as bobinas podem ser permanentemente danificados se permanecerem por períodos excessivos de tempo.

A partir do firmware de abril de 2017, a pausa será reduzida automaticamente quando a duração configurada for maior do que o tempo disponível no RPM atual. Isso é comum em configurações de ignição de canal único (por exemplo, 1 bobina com um distribuidor) e, em particular, em motores com maior número de cilindros.

Definições

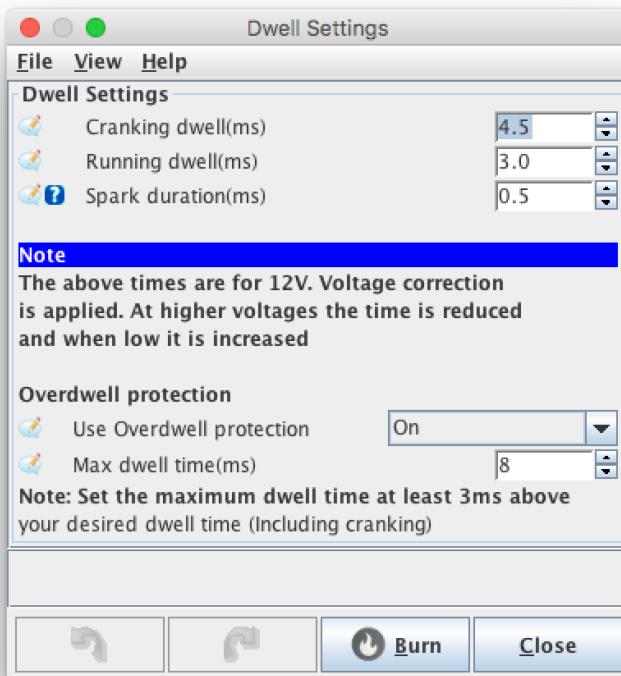


Figura 35: habitar.png

Observação: Os tempos de parada de operação e partida são valores nominais, assumidos como em uma tensão constante (normalmente 12v). O tempo de permanência real usado dependerá da tensão do sistema atual com tensões mais altas tendo tempos de permanência menores e vice-versa. Consulte a seção abaixo sobre correção de tensão

- Intervalo de arranque - O tempo de espera nominal que será usado durante o arranque. A partida é definida como sendo sempre que o RPM está acima de 0, mas abaixo dos valores de 'RPM de partida' na partidadiálogo
- Intervalo de operação - O tempo de espera nominal que será usado quando o motor estiver funcionando normalmente.
- Duração da faísca - O tempo aproximado que a bobina leva para descarregar totalmente. Este tempo é usado no cálculo de uma pausa reduzida quando em condições de tempo limitado, como mencionado acima em motores de bobina única e alta contagem de cilindros. O tempo de espera limitado é calculado tomando o tempo máximo de revolução em um determinado RPM, dividindo pelo número de saídas de faísca necessárias por rotação e subtraindo a duração da faísca. Fora dessas condições, esta configuração não é usada.

- Proteção excessiva - O sistema de proteção permanente funciona independentemente do padrão

a ignição programa e monitora o tempo que cada saída de ignição está ativa. Se o tempo ativo exceder este valor, a saída será encerrada para evitar danos às bobinas. Este valor normalmente deve ser pelo menos 3 ms mais alto do que os tempos de permanência nominais configurados acima, em ordem para permitir sobrecarga para correção de tensão.

Correção de tensão

À medida que a tensão do sistema sobe e desce, o tempo de permanência precisa reduzir e aumentar, respectivamente. Isso permite uma força de centelha consistente sem danificar a(s) bobina(s) durante condições de alta tensão do sistema. Recomenda-se que 12v seja usado como a tensão 'nominal', o que significa que o valor de % de permanência em 12v deve ser 100%.

A curva de correção no arquivo de sintonia base deve ser adequada para a maioria das bobinas / ignições, mas pode ser alterada se necessário.

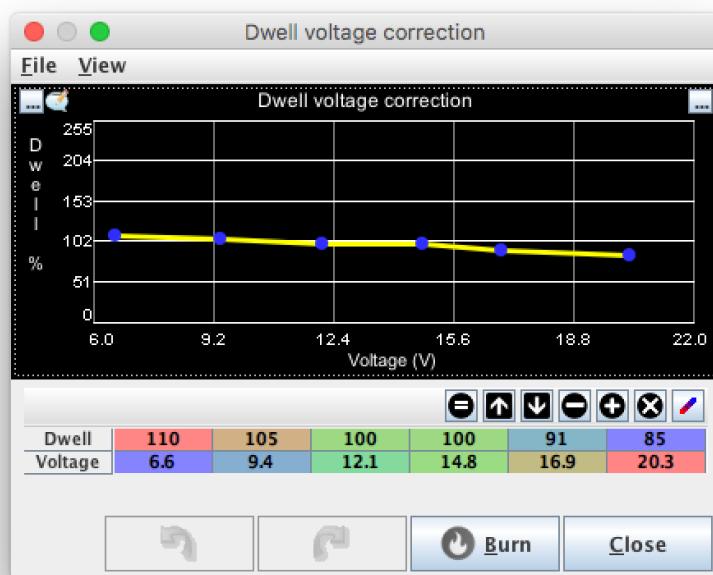


Figura 36: habitat_correction.png

Mudanças de tempo baseadas na temperatura

Mudanças na temperatura do ar de entrada (IAT), em aumentos significativos particulares durante a aceleração, podem exigir que o ponto de ignição seja puxado. As configurações de retardo IAT permitem este ajuste de tempo

Exemplo

As configurações exatas dependerão do motor, mas puxar o tempo de ignição além de 100 ° C é um cenário comum.



Visão geral

As condições de acionamento durante a partida normalmente requerem vários ajustes no controle de combustível e ignição para fornecer partidas suaves e rápidas. As configurações nesta caixa de diálogo determinam quando o Speeduino considerará o motor em uma condição de partida / partida e quais ajustes devem ser adequados dobrado durante este tempo.

Definições

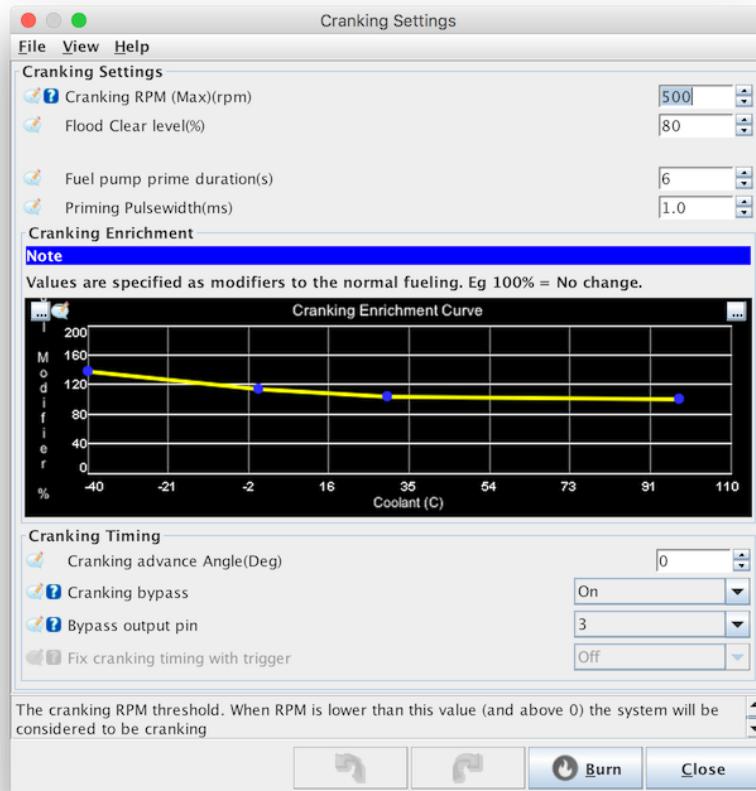


Figura 37: cranking.png

- **RPM de partida** - Isso define o limite para o Speeduino definir seu status como acionado ou em execução. Qualquer RPM acima de 0 e abaixo desse valor será considerado partida e todos os ajustes relacionados à partida serão aplicados. Geralmente é melhor definir isso para cerca de 100 rpm mais alto do que sua velocidade de partida típica para compensar os picos e fornecer uma transição mais suave para o funcionamento normal.
- **Nível de limpeza de inundação** - A limpeza de inundação é usada para ajudar a remover o excesso de combustível que entrou no(s) cilindro(s). Enquanto a limpeza de inundação estiver ativa, todos os eventos de combustível e ignição serão interrompidos e o motor pode ser ligado por alguns segundos sem risco de partida ou mais inundação. Para acionar a limpeza de inundação, o RPM deve estar abaixo da configuração de RPM de acionamento acima e o TPS deve estar acima do limite desta configuração.
- **Duração do prime da bomba de combustível** - Quando o Speeduino é ligado pela primeira vez, a saída da bomba de combustível será acionada por muitos segundos para pressurizar o sistema de combustível. Se o motor for ligado

neste tempo, a bomba simplesmente continuará funcionando, caso contrário ela será desligada após este período de tempo. Observe que a escorva da bomba de combustível ocorre apenas na hora de ligar o sistema. Se você tiver um USB conectado, o Speeduino permanecerá ligado mesmo sem um sinal de 12v.

- **Largura de pulso de preparação** - Ao ligar, o Speeduino irá disparar todos os injetores por este período de tempo. Este pulso NÃO se destina a ser uma carga de combustível inicial, mas sim para limpar o ar que pode ter entrado nas linhas de combustível. Deve ser mantido curto para evitar afogamento do motor.
- **Enriquecimento de arranque** - Enquanto a manivela está ativa (consulte RPM de partida acima), a carga de combustível será aumentada por este valor. Observe que, como um valor de correção padrão, este enriquecimento de arranque é um acréscimo a quaisquer outros ajustes que estão atualmente ativos. Isso inclui o aquecimento/enriquecimento etc.
- **Bypass de arranque** - Esta opção é especificamente para sistemas de ignição que possuem uma opção de ignição de partida por hardware. Esses sistemas foram usados ao longo dos anos 80 e início dos anos 90 e permitiram que o tempo de ignição fosse fixado e controlado pelo próprio sistema de ignição quando ativo (através de um fio de entrada). Com esta opção, você pode especificar um pino de saída que será definido como ALTO quando o sistema estiver dando partida. O número do pino especificado é o número do pino ARDUINO.
- **Corrigir o tempo de acionamento com gatilho** - Alguns padrões de disparo (geralmente de baixa resolução) são projetados para alinhar um de seus pulsos com o avanço de acionamento desejado. Isso é normalmente 5 ou 10 graus BTDC. Quando habilitado, o Speeduino aguardará esse pulso de entrada temporizado antes de disparar a saída de ignição relevante (um fator de segurança de parada ainda é aplicado caso esse pulso não seja detectado). Esta opção só está disponível quando um padrão de gatilho que suporta esta função é selecionado (Ver GatilhoConfigurar)

Enriquecimento de aquecimento

Visão geral

A caixa de diálogo Enriquecimento para aquecimento (WUE) contém configurações relacionadas ao período após a partida (ou seja, sem dar partida), mas antes de o motor atingir a temperatura normal de operação. Permite modificações para abastecendo durante este tempo para

Definições

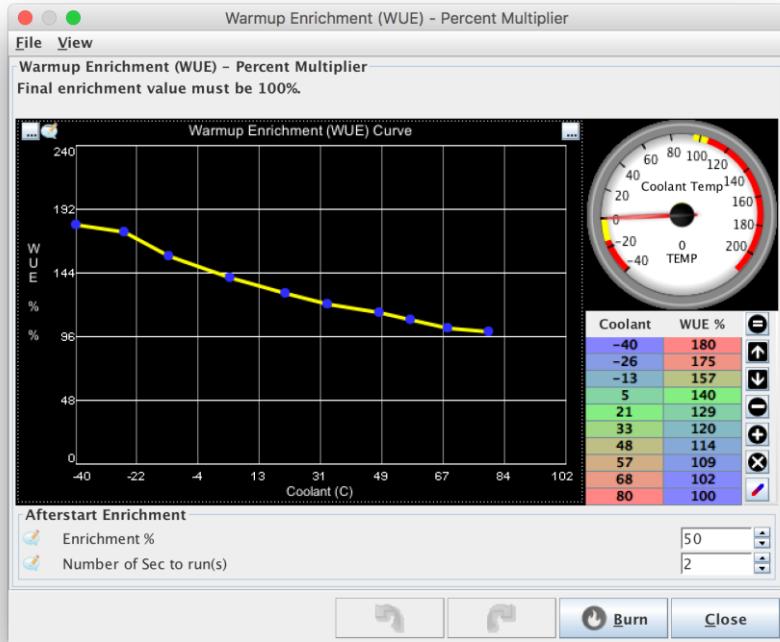


Figura 38: warmup.PNG

Curva de aquecimento

Esta curva representa a quantidade adicional de combustível a ser adicionada enquanto o motor está chegando à temperatura (com base no sensor do líquido de arrefecimento). O valor final nesta curva deve representar a temperatura normal de funcionamento do motor e ter um valor de 100% (representando nenhuma modificação do combustível a partir desse ponto).

Enriquecimento pós-início

O enriquecimento pós-partida (ASE) é um modificador de combustível separado que opera acima e acima do WUE por um período fixo de tempo após a primeira partida do motor. Normalmente, este é um período de 3 a 10 segundos, em que um pequeno enriquecimento pode ajudar a transição do motor suavemente da partida para a marcha lenta.

Controle de marcha lenta

Visão geral

As saídas de controle de marcha lenta são usadas para alterar o estado de uma válvula de controle de marcha lenta para aumentar a quantidade de ar que entra no motor em marcha lenta. Eles vêm em vários tipos (descrito abaixo) e cada um está configurado e ajustado de forma diferente.

O controle de marcha lenta de malha aberta e fechada está disponível para válvulas de marcha lenta com base em PWM e Stepper.

Compatível Tipos de válvula ociosa

Existem atualmente 3 modos de controle de inatividade disponíveis, usando liga / desliga, ciclo de trabalho PWM ou contagem de passo de passo, habilitado abaixo de uma temperatura definida do líquido de arrefecimento. Esses modos cobrem os tipos mais comuns de inatividade de mecanismos em uso.

Ligado / Desligado (também conhecido como rápido Ocio)

Esta é uma saída digital simples liga / desliga do Speeduino que dispara em uma temperatura selecionada. Destina-se a controlar uma válvula liga / desliga de marcha lenta rápida como encontrada em muitas configurações de OEM mais antigas, ou uma válvula do tipo solenóide aberta / fechada que é escolhida para esse propósito. Além das válvulas de marcha lenta OEM, exemplos de válvulas populares para reutilizar como válvulas de marcha lenta liga / desliga são válvulas maiores de vácuo, de respiro ou de purga e até mesmo válvulas de combustível. O ajuste da velocidade de marcha lenta geralmente é definido apenas uma vez, com uma restrição ajustável ou fixa em linhador, pinça de aperto ou outro método simples de controle de fluxo.

Observação: As válvulas liga / desliga podem ser usadas de várias maneiras para aumentar ou diminuir o fluxo de ar para vários fins de inatividade, além do aquecimento. Os exemplos são o uso como válvulas de painel para reduzir a perda de velocidade, recuperação da velocidade de marcha lenta para manter a velocidade do motor com cargas de acessórios, como ar condicionado, ou adição de ar para fins específicos, como controle de turbo anti-atraso. Consulte Saídas genéricas para obter informações de controle.

PWM

Embora semelhante em construção a muitas válvulas solenóides on / off; PWM As válvulas ociosas são projetadas para variar a abertura e, portanto, o fluxo através da válvula, pelo

posicionamento da válvula PWM.

Essas válvulas são abertas e fechadas variando o ciclo de trabalho do sinal enviado a elas.

Observação: Como proteção contra falhas, algumas válvulas PWM ociosas assumem o estado parcialmente aberto quando são desconectadas ou recebem 0% do ciclo de trabalho. Eles serão fechados e, em seguida, reabertos com PWM DC% crescente,

então certifique-se de pesquisar ou testar seu tipo de válvula para Operação. {.is-info}

PWM Definições As configurações no TunerStudio incluem a seleção de controle de marcha lenta PWM, configurações de temperatura e DC para aquecimento e PWM DC durante a partida sob as seguintes seleções:

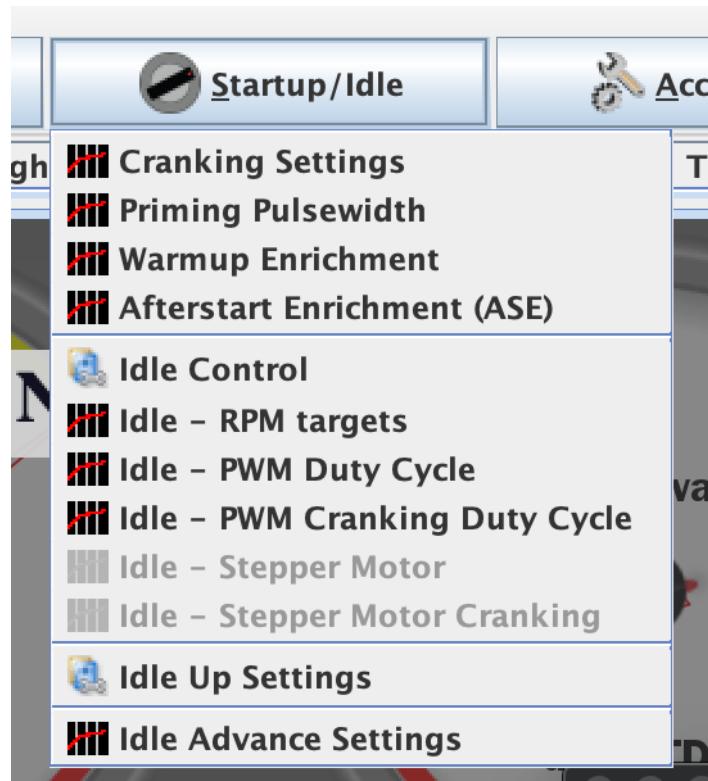


Figura 39: Exemplo de configurações de ociosidade do PWM

As opções 'Idle - PWM Duty Cycle' e 'Idle - PWM Duty Cycle' só estarão disponíveis quando 'PWM Open Loop' é selecionado nas opções de controle de inatividade

Sob Tipo de controle ocioso, PWM é selecionado:

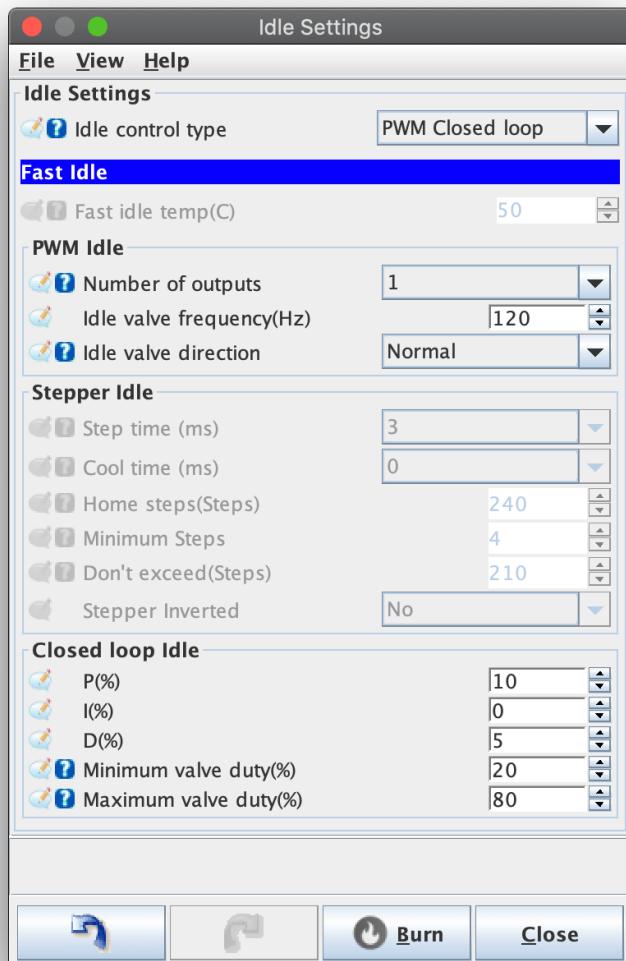


Figura 40: Exemplo Configurações ociosas do PWM

A temperatura versus DC é selecionado na seleção Idle - PWM Duty Cycle. Observe que a relação entre temperatura e PWM DC pode ser alterada simplesmente movendo os pontos azuis na curva, ou selecionando a tabela para entrada manual, conforme mostrado aqui:

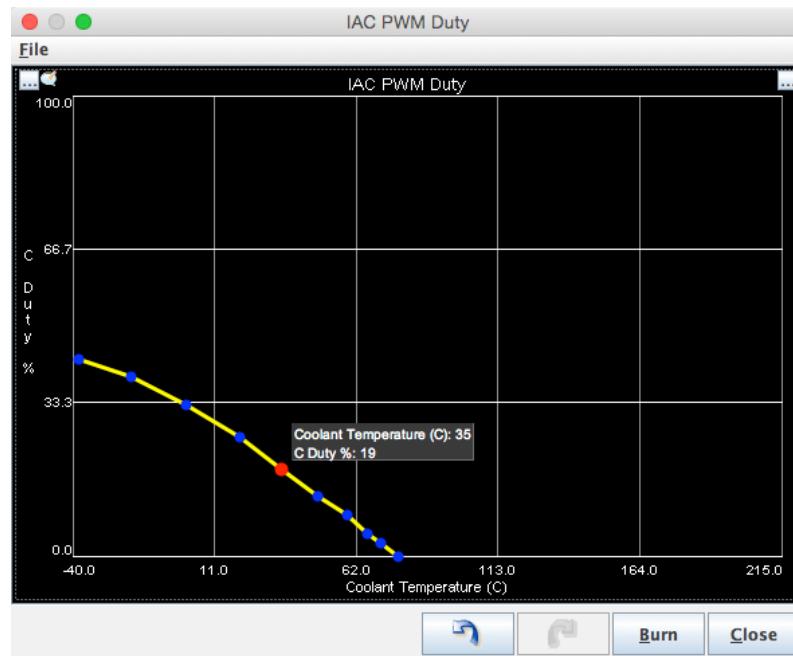


Figura 41: Curva de ociosidade do PWM de exemplo

Alguns motores preferem fluxo de ar adicional durante a partida para uma partida confiável. Este ar pode ser adicionado automaticamente apenas durante a partida usando as configurações de ciclo de trabalho de partida PWM Idle. Uma vez que o motor dá a partida e a rotação aumenta acima da rotação de partida máxima definida, o controle da marcha lenta muda para as configurações de aquecimento anteriores. Observe que a relação entre a temperatura do líquido refrigerante durante a partida e o PWM DC pode ser alterada simplesmente movendo os pontos azuis na curva ou selecionando a tabela para manual entrada conforme mostrado aqui:

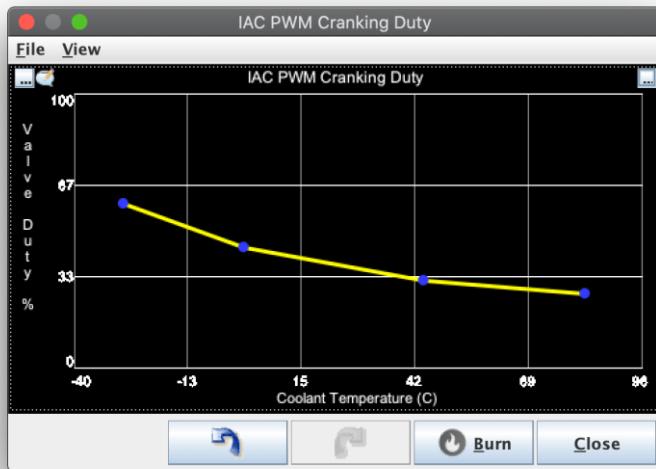


Figura 42: Exemplo de curva de acionamento PWM

NOTA: Cada motor, tipo de válvula e ajuste são diferentes. As configurações adequadas devem ser determinadas pelo sintonizador. Não deduza nenhuma configuração de ajuste das imagens neste wiki. Eles são apenas exemplos.{.is- aviso}

2 fios vs válvulas de 3 fios São suportados controladores ociosos PWM de 2 e 3 fios. Em geral, os modelos de 3 fios fornecem uma resposta mais suave do que os de 2 fios, mas a diferença nem sempre é significativa. Para válvulas de 3 fios, 2 das saídas auxiliares serão necessárias.

Motores de passo

Os controles de marcha lenta do motor de passo são muito comuns na GM e em outras configurações de OEM. Esses motores normalmente possuem 4 fios (bipolares). Eles devem ser conduzidos por transistores de potência ou um módulo de driver, como oDriver do motor de passo DRV8825 opcional para a placa v0.4. Esses módulos de driver podem ser adquiridos de forma econômica de uma variedade de fornecedores em sites como eBay, Amazon, etc.

A maioria das válvulas deslizantes de marcha lenta funciona girando uma haste rosada para dentro e para fora do corpo da válvula em uma série de etapas de giro parcial, aumentando ou diminuindo o fluxo de ar ao redor do êmbolo (na extremidade da válvula abixo) e para dentro do motor. O fluxo de ar ocioso ignora o corpo do acelerador principal:

Exemplo de um genérico Módulo de driver DRV8825 em uma placa v0.4:

Observe que a placa é montada em um ponto morto para circulação de ar e resfriamento:

As saídas do motor DRV8825 são rotuladas como A2-A1-B1-B2 e os exemplos de conexão de fiação são para esta rotulagem. Verifique seus esquemas para as conexões de saída que direcionam para estes DRV8825 out-coloca:

Exemplos de fiação para o DRV8825 motorista:

O estilo “aparafusado” da GM usado em 1982 a 2003 em muitos modelos:

Stepper Ajuste de corrente do driver O módulo driver de passo DRV8825 inclui um potenciômetro (resistor ajustável) indicado pela seta amarela na imagem abaixo. O potenciômetro é usado para definir o limite máximo de saída de corrente do driver. Como o Speeduino usa a operação de passo completo, o limite de corrente não é crítico para proteger o módulo, mas deve ser ajustado para o valor máximo do módulo para melhor operação da maioria dos IACs de passo automotivo.

Você precisará de um multímetro ou voltímetro para fazer o ajuste conforme descrito aqui. Para definir o potenciômetro para a corrente máxima antes do primeiro uso, certifique-se de que a alimentação do módulo esteja DESLIGADA e, em seguida, gire suavemente o botão do potenciômetro no sentido horário até o limite interno. Não force o ajuste além da parada interna. Ligue o Speeduino com 12 V e use o medidor para testar a tensão entre o centro do potenciômetro e qualquer ponto de aterramento de 12 V. Observe a leitura da tensão. Desligue e repita o teste, desta vez girando o potenciômetro no sentido anti-horário suavemente até o limite interno. A direção do teste que resultou em uma tensão mais alta é a configuração correta para o módulo.

Nota: Os módulos originais Pololu são normalmente ajustados no sentido horário para a tensão máxima. No entanto, os módulos clones podem ser no sentido horário ou anti-horário, o que torna esse teste necessário.

A corrente contínua nominal do módulo é de até 1,5A. Enquanto o módulo pode fornecer um pico de 2,2A de corrente; no modo full-step e com o potenciômetro ajustado para esta posição, o driver é limitado a aproximadamente 70% da corrente total, ou aproximadamente 1,5A.

Configurações de passo As configurações no TunerStudio incluem a seleção de controle de marcha lenta, configurações de temperatura e etapa para aquecimento e etapas abertas durante a partida nas seguintes seleções:

Em tipo de controle Inativo, passo a passo é selecionado. As configurações operacionais básicas de passo também estão localizadas em esta janela:

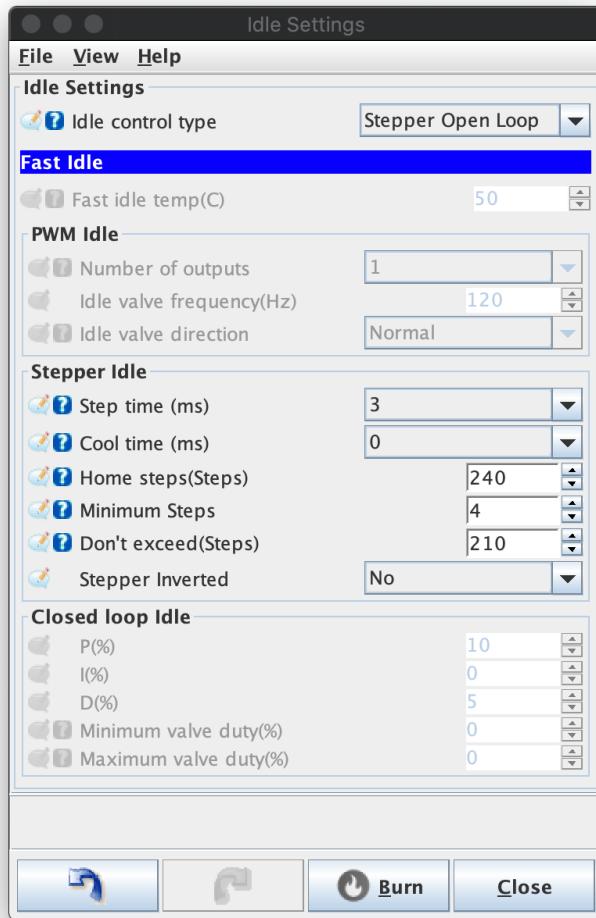


Figura 43: Stepper configurações inativas

- **Tempo da etapa:** Este é o tempo (em ms) que o motor requer para concluir cada etapa. Se for definido muito baixo, o motor ocioso não terá concluído a etapa antes que a ECU tente fazer a próxima, o que faz com que o motor 'estremeça' e não funcione corretamente. Se for definido mais tempo do que o necessário, o sistema levará mais tempo para fazer cada ajuste e a resposta geral à inatividade será mais lenta. Os valores típicos são geralmente 2ms - 4ms. O motor de passo GM comum requer 3ms.
- **Tempo frio:** Alguns motores requerem uma pequena pausa entre as etapas para funcionar corretamente. Isso é conhecido como tempo de 'resfriamento'. Normalmente, este valor será inferior a 4 ms no máximo, com muitos motores operando normalmente sem período de resfriamento (0 ms)
- **Etapas iniciais:** Os motores de passo devem ser 'homed' antes de poderem ser usados para que a ECU saiba sua posição atual. Você deve definir isso para o número máximo de etapas que o motor podem mover.
- **Passos mínimos:** A fim de permitir uma marcha lenta suave que não está flutuando continuamente, a ECU

apenas move o motor se pelo menos essas várias etapas forem necessárias. Os valores típicos estão na faixa de 2 a 6, no entanto, se você tiver uma linha de sinal de refrigerante com ruído, pode ser necessário aumentar este valor.

- **Não exceda:** Para evitar que o motor de passo tente se mover além de sua faixa máxima, este é um limite colocado no número total de etapas que serão executadas. O valor neste campo deve ser sempre menor do que o número de etapas de homing

A temperatura versus etapas é selecionada na seleção Idle - Stepper Motor. Observe que a relação entre a temperatura e as etapas do motor pode ser alterada simplesmente movendo os pontos azuis na curva, ou selecionando a tabela para entrada manual, conforme mostrado aqui:

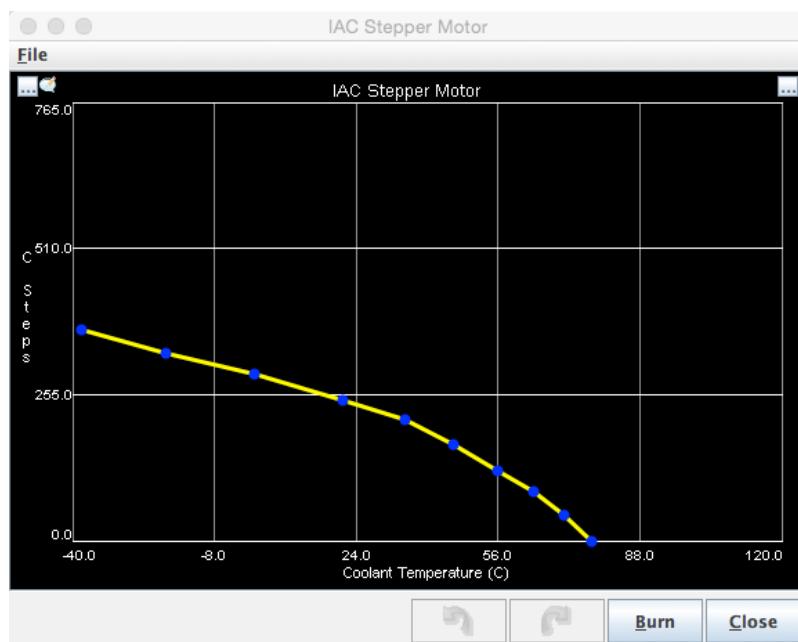


Figura 44: Curva de passo ocioso de exemplo

Alguns motores preferem fluxo de ar adicional durante a partida para uma partida confiável. Esse ar pode ser adicionado automaticamente apenas durante a partida usando as configurações de marcha lenta do motor de passo. Uma vez que o motor dá a partida e a rotação aumenta acima da rotação máxima de marcha lenta definida, o controle de marcha lenta muda para as configurações de aquecimento anteriores. Observe que a relação entre a temperatura do líquido refrigerante durante a partida e as etapas do motor pode ser alterada simplesmente movendo os pontos azuis na curva ou selecionando a tabela para entrada manual como mostrado aqui:

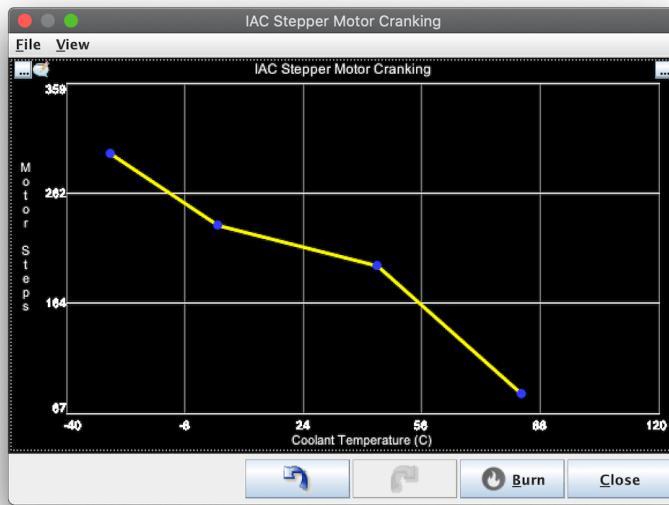


Figura 45: Exemplo de acionamento curva de passo

NOTA: Cada motor, tipo de válvula e ajuste são diferentes. As configurações adequadas devem ser determinadas pelo sintonizador. Não deduza nenhuma configuração de ajuste das imagens neste wiki. Eles são apenas exemplos aleatórios.

{.is-info}

NOTA: Consulte o Vídeo Pololu para obter instruções para definir o nível de corrente do driver DRV8825 para o máximo para a maioria dos motores de passo de passo completos automotivos.

Exemplos

Motor	Tempo da etapa	Etapas de casa
GM 4 fios	3	250
DSM 4 fios	4	270-320

NOTA: Embora a função de passo DSM normal seja vista em temperatura ambiente de 3 ms, o salto de etapa ocorre logo abaixo dessa velocidade. Temperaturas muito baixas podem causar saltos, daí a recomendação de 4ms. Teste as velocidades mais adequadas para sua configuração.

Autônomo (não eletrônico)

Embora não seja um modo de controle inativo, o Speeduino é compatível com válvulas inativas autônomas que são autocontroladas. Exemplos disso são cera térmica ou mola bimetálica ociosa ou válvulas auxiliares de ar como a mostrada abaixo. O material de expansão e contração interna abre e fecha as válvulas de ar, proporcionando maior fluxo de ar e rotação do motor quando frio para aquecimento. O Speeduino funciona para enriquecer o motor frio e ajustar para o ar adicional, da mesma forma que faria se você abrisse ligeiramente o acelerador.

Outros exemplos de válvulas autônomas são válvulas liga / desliga simples, conforme mostrado na próxima seção, controladas por interruptores térmicos baratos como estes:

Controle de malha fechada

O controle de ociosidade de loop fechado opera definindo metas de RPM em vez de configurar o ciclo de trabalho ou etapas diretamente. Um algoritmo PID é usado e pode ser ajustado para corresponder à válvula / motor que você está usando.

Configurações de

loop fechado Alvos

de loop fechado

Controle de avanço ocioso

A velocidade de marcha lenta pode ser controlada sem o uso de uma válvula de marcha lenta (IACV) ajustando o tempo. Este recurso faz referência à mesma curva alvo de RPM de marcha lenta que é usada pelo controle de marcha lenta de malha fechada e, então, ajustará o avanço com base no erro entre a RPM atual e a de destino.

Definições

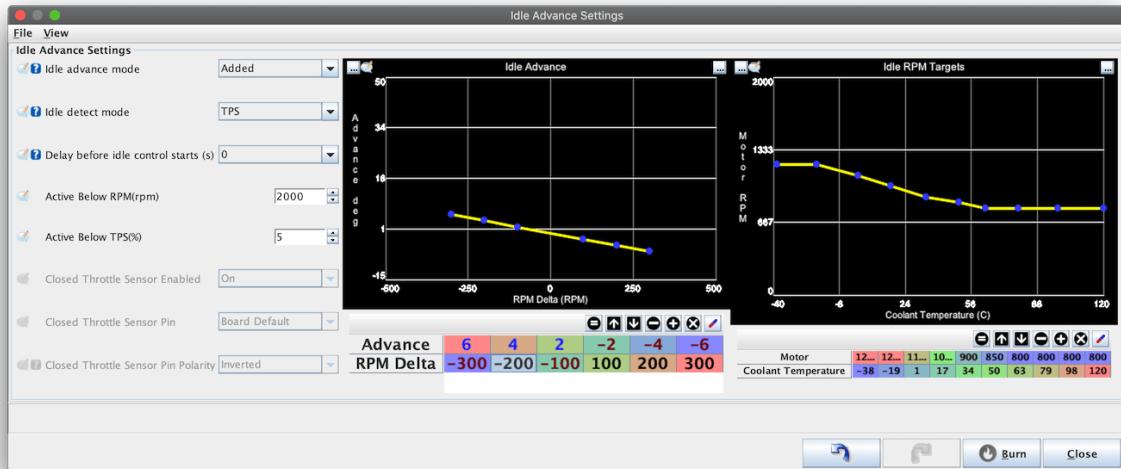


Figura 46: idle_advance.png

- **Modo de avanço ocioso**
 - **Adicionado** - Este é o modo mais comum e irá alterar a quantidade de avanço regular adicionando (ou subtraindo) um certo número de graus com base na quantidade de delta de RPM (Entre o alvo e os RPMs reais)
 - **Comutado** - O avanço da ignição mudará para os valores na curva de avanço de marcha lenta, em vez do que ajustar os valores normais de avanço
- **Modo de detecção inativo** - Esta configuração especifica como a ECU determina se está ociosa ou não. Mais comumente, isso é baseado em um TPS variável e um TPS% específico, mas se um acelerador fechadointerruptor (CTPS) está disponível, pode ser usado em seu lugar
- **Atraso antes do controle inativo** - Isso permite que o RPM ocioso se estabilize durante a desaceleração antes do o avanço da ignição é alterado.
- **Ativo abaixo** - RPM máximo sob o qual o controle de avanço de marcha lenta estará ativo
- **Ativo Abaixo** - Se o modo de detecção de marcha lenta for definido como TPS, esta é a posição do acelerador que o controle estará ativo abaixo
- As 3 configurações a seguir são usadas apenas se a detecção de inatividade usar uma entrada CTPS
 - **CTPS habilitado** - Se deve usar uma entrada CTPS
 - **Pin CTPS** - O pino Arduino ao qual o CTPS está conectado
 - **CTPS Polatiry** - Se ocioso é indicado pela entrada sendo puxada para o solo (Normal) ou indicado pela entrada sendo puxada para 5v (Invertida). No modo normal, o pullup interno

será ativado.

Curva de avanço ocioso

Esta curva especifica a quantidade de ajuste de tempo (modo adicionado) ou a quantidade de avanço absoluto (modo comutado) que será usado com base no delta (erro) do RPM alvo.

Geralmente o tempo será adicionado (valores positivos) a fim de tentar aumentar o RPM e o tempo será removido (valores negativos) para reduzir o RPM.

Curva alvo de RPM de marcha lenta

Esta curva especifica qual RPM de marcha lenta desejada é baseada na temperatura atual do líquido refrigerante. Esta tabela é compartilhada com o controle de ar de marcha lenta se estiver sendo usado em conjunto com o controle de avanço de marcha lenta.

Termoventilador

O controle de um ventilador de resfriamento (térmico) está disponível na caixa de diálogo Ventilador térmico.

Observe que apenas o controle do ventilador liga / desliga é possível no momento (não o controle de velocidade do ventilador usando PWM). {é-info}

Definições

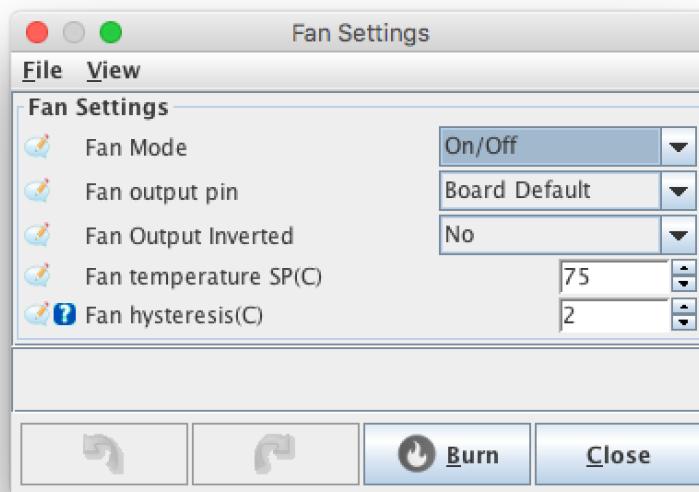


Figura 47: fan.png

- **Modo fã** - Atualmente apenas [Sobre/Fora](#) modo é suportado
- **Pino de saída** - O pino arduino que o controle do ventilador usará. Na maioria dos casos, isso deve ser deixado como [Padrão da placa](#)
- **Saída invertida** - A maioria das configurações usará [Não](#) para esta configuração, mas se você tiver um circuito de ventilador que inverte a saída, a polaridade On / Off pode ser revertida com esta configuração
- **Temperatura do ventilador** - A temperatura acima da qual o ventilador será ligado.
- **Histerese de ventilador** - O número de graus abaixo do ponto de ajuste do ventilador em que o ventilador será desligado. Isso é usado para evitar oscilações em torno do ponto de ajuste, resultando no ventilador ligando e desligando rapidamente.

Controle de lançamento e deslocamento plano

O Speeduino possui um controle de lançamento de 2 etapas combinado com um recurso de deslocamento plano. Cada um deles depende de um interruptor de embreagem (geralmente um tipo de comutação de aterrramento) sendo conectado.

Configurar

Os modos 2-step e flatshift têm estados de corte rígido e suave. Quando em corte suave, o tempo de ignição será alterado para reduzir a aceleração de RPM, embora isso geralmente não seja suficiente para parar ou limitar o aumento de RPM. Sob corte rígido, o sinal de ignição é interrompido completamente até que os RPMs caiam

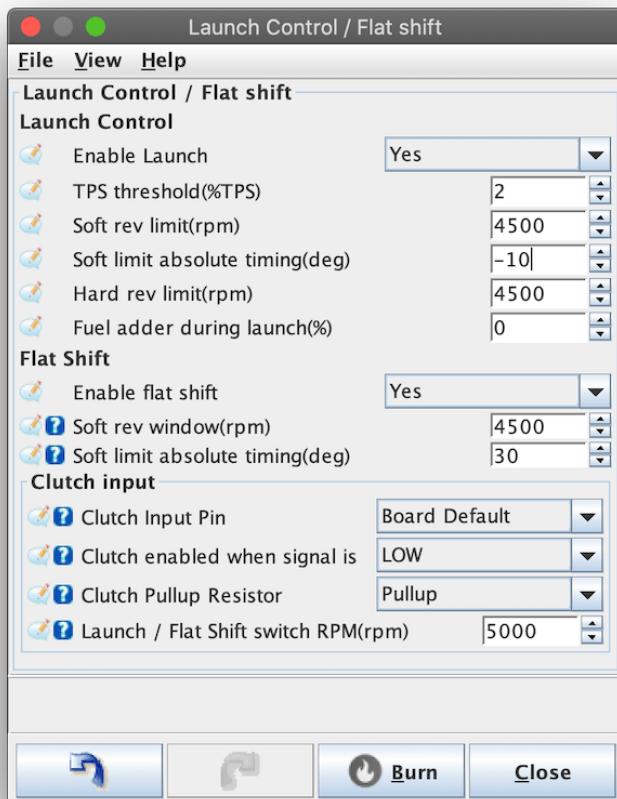


Figura 48: Configurações de lançamento e mudança plana

Lançar

- **Limiar TPS** - Um valor mínimo para o compromisso de lançamento. O limitador só será acionado acima deste RPM. Os valores típicos são 1% -3% TPS, dependendo de quanto ruído está em seu sinal
- **Limite de rotação suave** - O RPM no qual o tempo será ajustado para diminuir o aumento de RPM
- **Limite suave de tempo absoluto** - O tempo absoluto que será usado quando o limite de RPM suave for atingido. Isso substitui todos os outros ajustes de tempo neste momento

- **Limite de rotação forte** - O RPM no qual o sinal de ignição será cortado totalmente.
- **Adicionador de combustível durante o lançamento** - Um modificador de porcentagem para a largura de pulso atual para adicionar combustível extra quando o lançamento (suave ou forte) está ativo. Isso pode ajudar a aumentar o impulso em configurações de turbo no lançamentoTempo

Mudança plana

- **Janela de rotação suave** - Esta é uma janela RPM abaixo do Lançar / RPM do interruptor de mudança plana ponto durante o qual um tempo alternativo será aplicado. Os valores típicos são de 100 a 1000 rpm.
- **Limite suave de tempo absoluto** - O tempo absolute que será usado quando na mudança plana suave Janela RPM

Configurações de embreagem

Tanto o lançamento quanto a mudança plana exigem uma entrada de embreagem para serem ativados. Este é geralmente um solo ativointerruptor de tipo conectado atrás do pedal da embreagem.

- **Pino de entrada da embreagem** - O pino do Arduino ao qual o switch está conectado. A maioria das configurações deve deixar isso como padrão da placa
- **Embreagem habilitada quando o interruptor está** - A polaridade da entrada da embreagem. Normalmente, isso deve ser definido como BAIxo para um interruptor que se conecta ao aterramento quando ativado
- **Resistor de pullup da embreagem** - Se o pullup interno será habilitado nesta entrada. Normalmente, tudo isso deve ser definido como Puxar para cima se você selecionou BAIxo para a configuração acima
- **RPM do interruptor de lançamento / flat shift** - A ECU usará o ponto de RPM em que a embreagem está engatada para determinar se está no modo de lançamento ou em câmbio plano. Se a embreagem for pressionada acima deste valor de RPM, será considerado um deslocamento plano, abaixo será considerado um lançamento

O ponto de engate do interruptor da embreagem pode fazer uma diferença significativa na aplicação do controle de lançamento. O interruptor deve ser acionado o mais próximo possível do ponto de tomada das embreagens para a resposta mais rápida. { .is-sucesso }

Bomba de combustível

O controle da bomba de combustível é uma função simples, mas importante, executada pela ECU.

Atualmente, o Speeduino não executa o controle variável (PWM) da bomba. Só pode ser conectado a um relé. NÃO CONECTE DIRETAMENTE À BOMBA DE COMBUSTÍVEL.

Definições

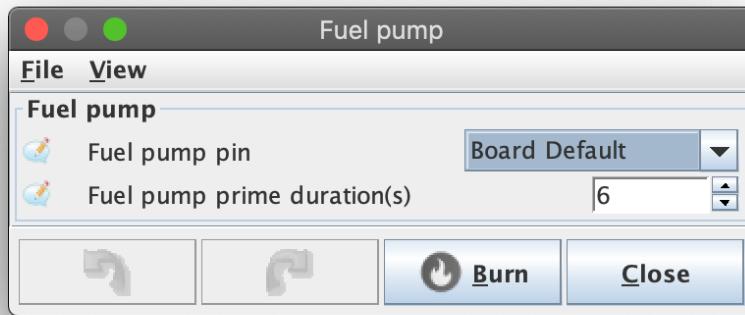


Figura 49: fuel_pump.png

- **Pino da bomba de combustível** - O pino do Arduino em que a saída da bomba de combustível está ligada. Na maioria dos casos, isso deve ser deixado para [Padrão da placa](#) a menos que você tenha um motivo específico para alterar isso.
- **Duração principal** - Por quanto tempo (em segundos) a bomba de combustível deve funcionar quando o sistema é ligado pela primeira vez. Observe que isso é acionado quando a ECU é ligada, o que nem sempre será o mesmo de quando a ignição é desligada. Se você tiver um cabo USB conectado, a ECU está já ligado.

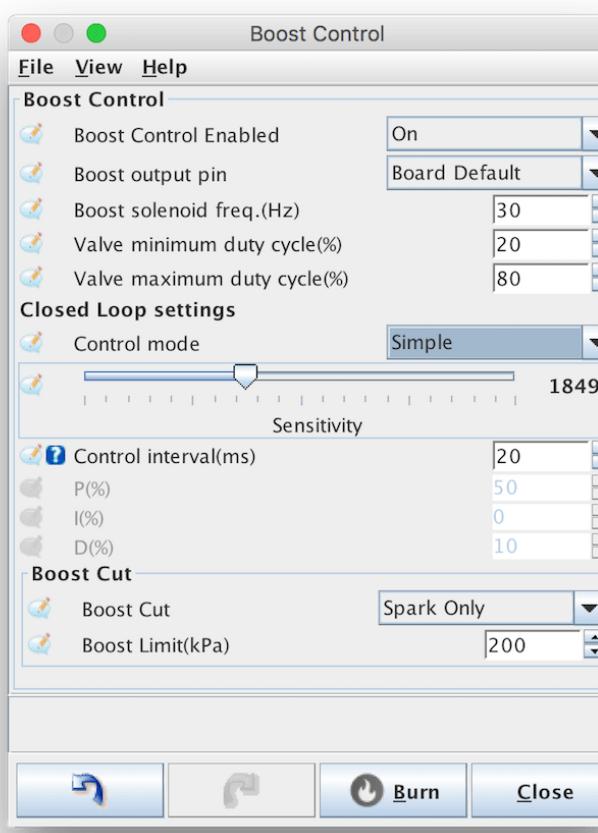
Boost Control

O Speeduino tem um controlador de reforço de circuito fechado integrado que pode ser usado para regular o padrão único configurações turbo.

A maioria dos solenóides de reforço de 3 ou 4 portas pode ser usada, com frequências entre 15 Hz e 500 Hz suportadas. Qualquer uma das saídas de alta corrente na placa pode ser conectada diretamente ao solenóide e é controlada por meio de uma tabela de alvo de reforço e ajuste de PID. A limitação do aumento excessivo também está disponível.

Definições

O controle de impulso do Speeduino usa um algoritmo PID com 2 modos de operação, Simples e Completo. Cada um tem suas próprias vantagens e desvantagens, conforme descrito abaixo.



No modo Simples, os próprios valores PID são controlados pela própria ECU e um controle deslizante de sensibilidade é usado para ajustar o quanto agressivo o ciclo de trabalho de saída será definido. O modo simples pode ser fácil e rápido de configurar, porém tem a desvantagem de evitar overboost, a sensibilidade pode precisar ser configurada baixo, o que pode aumentar o atraso.

Tabela de destino

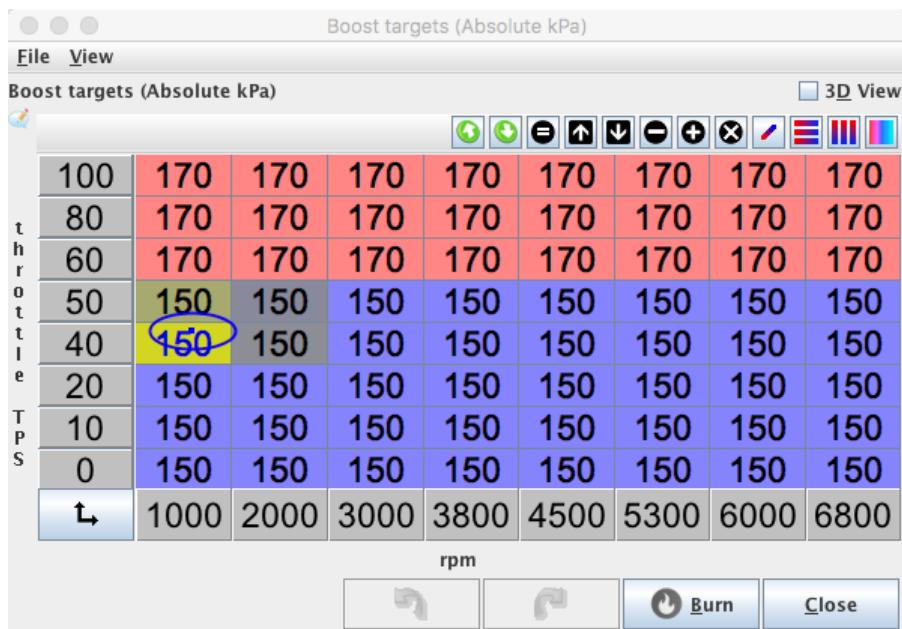


Figura 50: Exemplo de mapa de aumento

A função de mapa de aumento varia dependendo se o controle de aumento de loop aberto ou fechado foi selecionado.

- No modo de loop fechado, este mapa serve como uma tabela de destino. Os valores no mapa são as pressões de impulso desejadas (em kPa). No modo de malha fechada, esses valores alvo podem ser modificados opcionalmente por um valor de combustível flexível, se disponível.
- No modo de malha aberta, os valores do mapa são as porcentagens do ciclo de trabalho que serão usados

Controle Nitroso

O Speeduino contém um sistema de controle nitroso de 2 etapas para controlar válvulas e fazer ajustes de abastecimento para configurações a seco. Os 2 estágios operam de forma independente e podem se sobrepor (ou seja, ambos funcionam ao mesmotempo), se necessário.

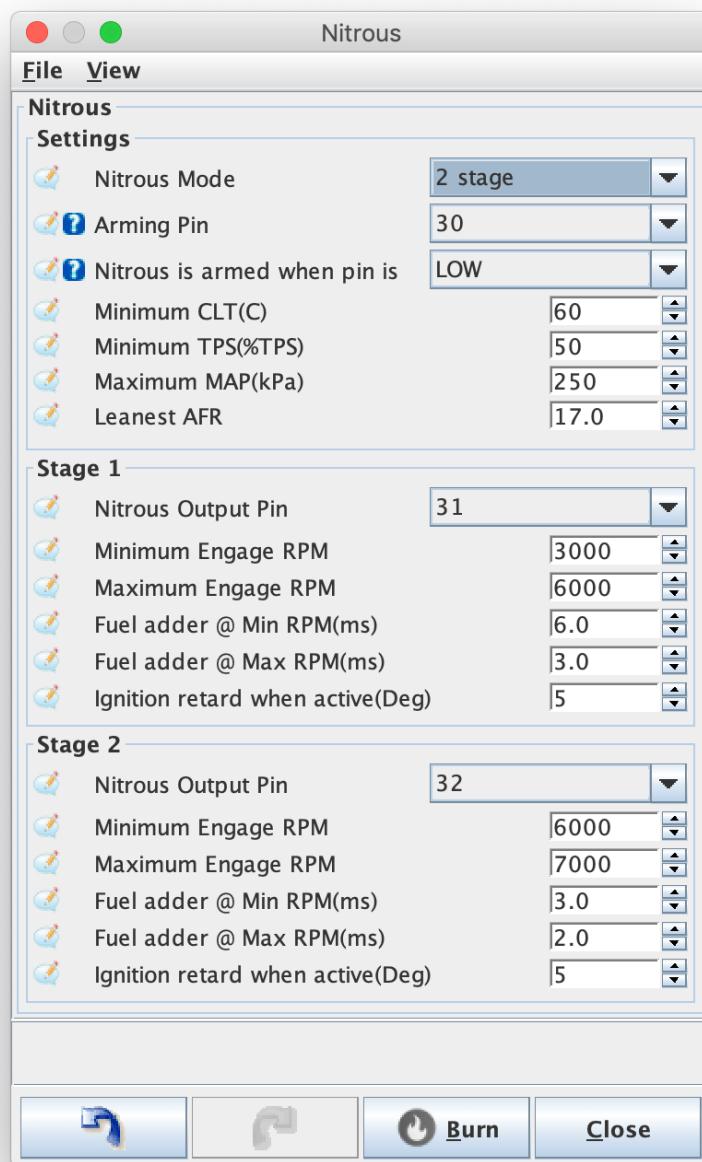


Figura 51: nitrous_settings.png

Configurações de ativação

- **Modo nitroso:** Se 1 ou 2 estágios irão ser usado

- **Pino de armamento:** O pino Arduino a ser usado para armar o controle nitroso.
- **Polaridade do pino de armação:** O estado do pino é considerado armado. Geralmente, isso será BAIXO para uma entrada de comutação de terra
- **CLT mínimo:** A temperatura mínima do refrigerante em que os estágios serão ativados
- **TPS mínimo:** O TPS mínimo em que o estágio ativará
- **MAP máximo:** Uma proteção para garantir que o nitro não seja ativado acima de um certo nível de impulso
- **AFR mais enxuto:** Nitroso só será ativado se o AFR estiver (e permanecer) abaixo deste valor

Configurações do palco

As configurações para cada estágio são idênticas e permitem que os 2 estágios sejam executados individualmente ou sobrepostos em conjunto. ping durante uma determinada janela RPM.

- **Pino de saída nitroso:** O pino (Arduino) que será elevado quando o palco estiver ativo.
- **RPM de Engate Mínimo:** O RPM em que o estágio começará
- **RPM de Engate Máximo:** O RPM em que o estágio termina
- **Adicionador de combustível @ RPM mín. / Máx .:** A quantidade de combustível a ser adicionada no mínimo e no máximo Pontos RPM.
 - A quantidade do adicionador de combustível será escalonada entre esses 2 valores à medida que os RPMs aumentam
 - Uma calculadora para esses valores de somador de combustível pode ser encontrada em:
<https://bit.ly/3a0e9WU>
- **Retardo de ignição quando ativo:** Um modificador de ignição para ser usado para retardar o tempo quando o palco é ativo
 - Observe que os valores de retardo são cumulativos. Se ambos os estágios estiverem ativos, o retardo total montante será a soma das duas etapas.

Calibração do Sensor

Antes que seu Speeduino possa interpretar corretamente os sinais dos sensores, ele deve saber quais sensores você está usando. Inserir essas informações no TunerStudio (TS) grava a calibração correta no Speeduino. É necessário realizar esta etapa antes de verificar com eficácia a compilação do Speeduino. Observe que isso não está ajustando seu sistema, mas apenas informando como entender os sinais dos sensores.

Isso deve ser concluído após concluir as configurações do seu motor. O computador deve estar conectado ao Speeduino por meio do TS para realizar as calibrações.

Sensor MAP

Abra o menu Ferramentas: tools_menu

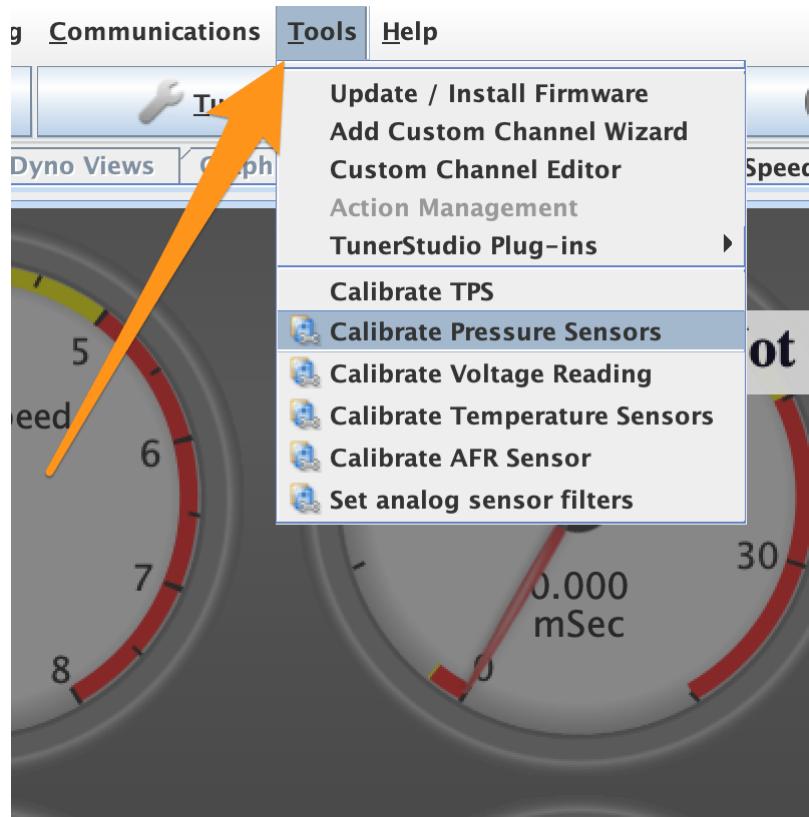


Figura 52: tools_menu.png

Selecione Calibrar Sensores de Pressão, a janela abaixo será aberta:

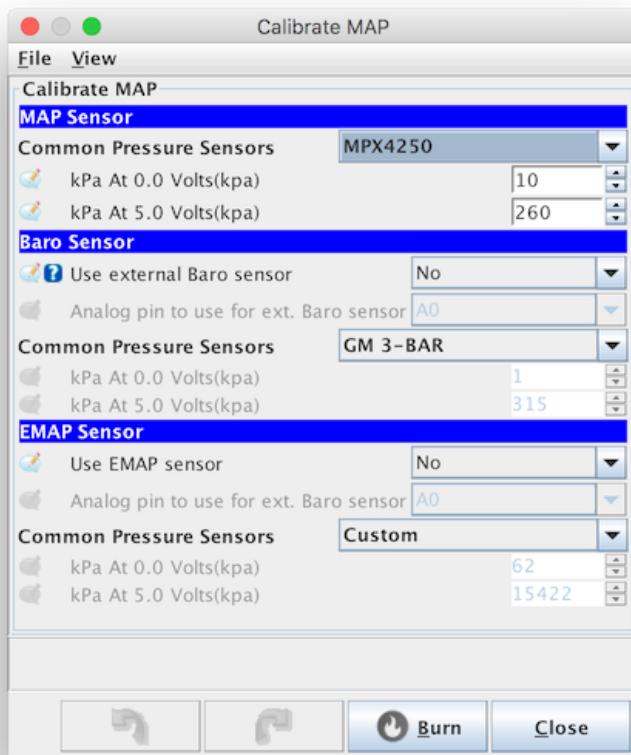


Figura 53: MAP_calibration.png

Selecione o seu sensor de mapa na lista suspensa. Se você usou o sensor MAP na lista de materiais, este será o MPX4250A. Se você estiver usando outro MAP ou do fabricante do motor, selecione-o na lista. Clique em Gravar para enviar as informações ao seu Speeduino.

Se usados, os sensores externos Baro e EMAP (pressão de exaustão) podem ser calibrados no mesmo menu.

Sensores de temperatura de admissão e refrigerante

Abra o menu Ferramentas e selecione Calibrar Termistor Mesas:

O sensor selecionado será o sensor de temperatura do líquido refrigerante. Selecione seu sensor na lista suspensa Common Sensor Values. Isso colocará os valores corretos nos gráficos de temperatura e resistência e no valor do resistor de polarização. Se o seu sensor não estiver listado, consulte Inserindo valores personalizados abaixo.

Observe que a construção padrão do Speeduino deve ter um resistor de polarização de 2490 ohms, que é o padrão para sensores usados pela maioria dos fabricantes. Se o seu sensor exigir outro valor, pode ser necessário alterar o resistor R3 para o valor correto para o seu sensor. Você pode tentar substituir o valor do resistor de polarização com 2490 ohms, mas verifique se o seu sensor está lendo corretamente no TS.

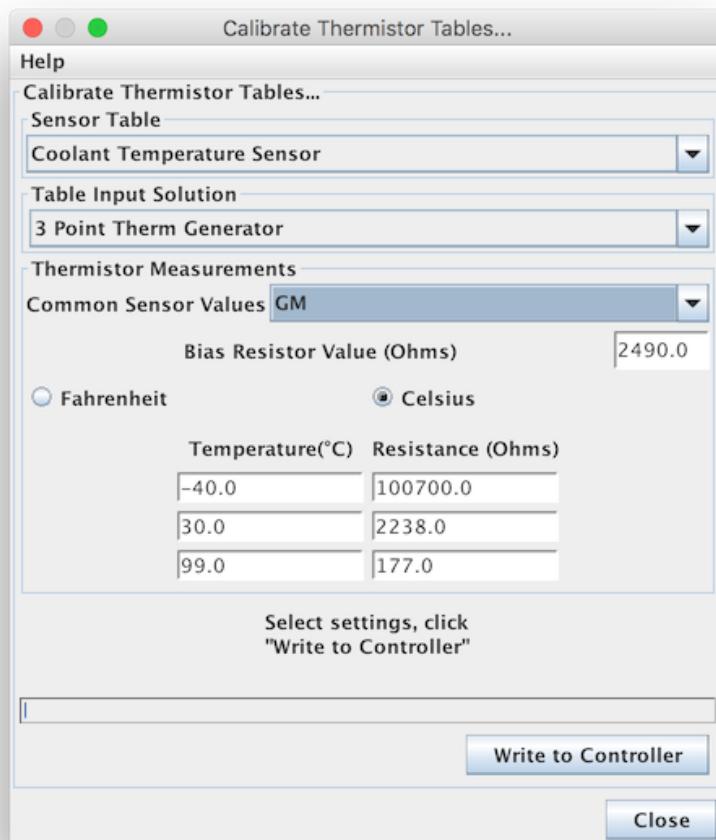


Figura 54: CLT_calibration.png

A mesma calibração pode então ser realizada para o sensor de temperatura do ar de entrada (IAT) alterando a tabela do sensor para o sensor de temperatura do ar:

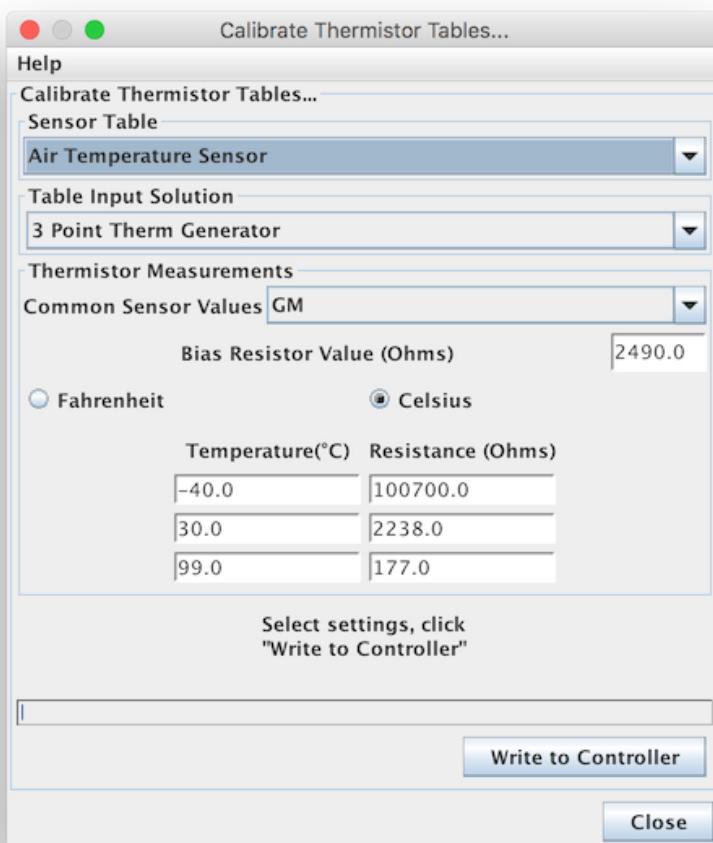


Figura 55: IAT_calibration.png

Selecione seu sensor na lista suspensa Common Sensor Values. Isso colocará os valores corretos nos gráficos de temperatura e resistência e no valor do resistor de polarização. Clique em Gravar no controlador para enviar essas informações ao Speeduino. Se o seu sensor não estiver listado, consulte Como inserir valores personalizados abaixo de.

Observe que a construção padrão do Speeduino deve ter um resistor de polarização de 2490 ohms, que é o padrão para sensores usados pela maioria dos fabricantes. Se o seu sensor exigir outro valor, pode ser necessário alterar o resistor R3 para o valor correto para o seu sensor. Você pode tentar substituir o valor do resistor de polarização com 2490 ohms, mas verifique se o seu sensor está lendo corretamente no TS.

Entrando Valores Personalizados

Alguns sensores não estão listados nas tabelas dos sensores comuns. Se o seu não estiver listado, você precisará inserir os valores nos campos você mesmo. Você precisará de dois bits de informação: 1. O valor do seu resistor de polarização (2490 se você usou os valores padrão na lista de materiais, ou se tiver um Speeduino pré-fabricado) e 2. A resistência do seu sensor em três temperaturas diferentes.

A resistência do sensor pode ser gerada medindo-se a resistência do sensor no ar ambiente, colocando-o no freezer e depois em água fervente. Você precisará de um bom multímetro e um termômetro preciso que mede -10C a 100C (14 ° F a 212 ° F). É melhor usar fios de jumper para permitir que a resistência do sensor seja lida sem segurá-lo em sua mão (alguns sensores reagem rapidamente às mudanças de temperatura). Alguns sensores reagem lentamente às mudanças de temperatura, portanto, permita que o sensor atinja uma temperatura estável por pelo menos 10 minutos e registre a temperatura e a resistência observadas.

Na tela Calibrate Thermistor Tables, primeiro certifique-se de que a unidade de temperatura correta é selecionada (F ou C). Em seguida, registre o valor do resistor de polarização e os valores de temperatura / resistência nos campos. Clique em Gravar no controlador para enviar essas informações ao Speeduino.

Observe que seu procedimento também pode ser usado para inserir os valores de resistência em simuladores para teste e solução de problemas. Dois pontos devem ser lembrados se você usar valores de simulador - primeiro, nunca insira zero para resistência. Embora seu simulador possa chegar a zero, insira um valor pequeno acima de zero, digamos 10 ohms. Inserir zero leva a valores falsos no firmware. Segundo - lembre-se de entrar no corrigir os valores do sensor antes de instalar o Speeduino!

Sensor de oxigênio

Abra o menu Ferramentas novamente e selecione Calibrar Tabela AFR:

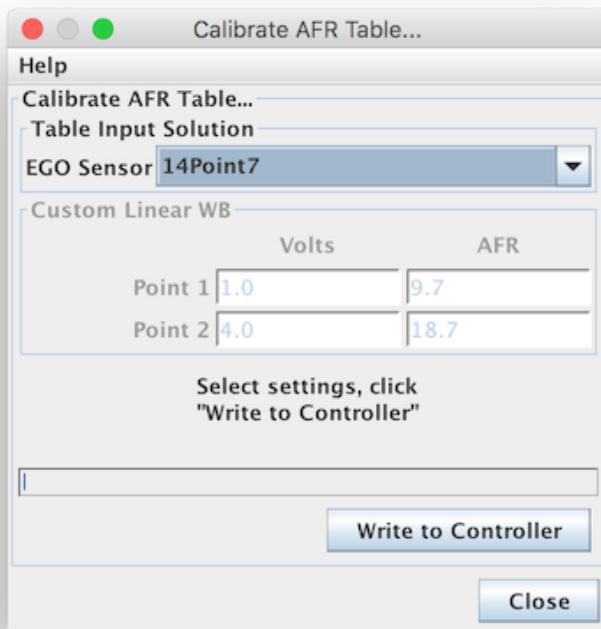


Figura 56: Calibração de O2

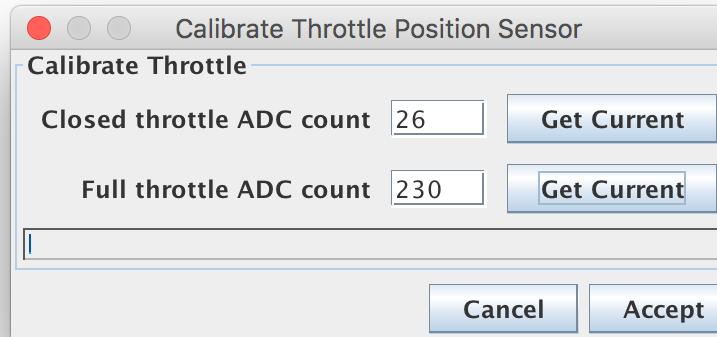
Selecione o sensor do seu sensor de oxigênio na lista suspensa Valores comuns do sensor. Se você estiver usando um controlador de sensor de oxigênio personalizado, selecione Custom Linear WB e, em seguida, você pode inserir os valores de Volts e AFR em dois pontos (devem ser publicados no manual do controlador).

Clique em Gravar no controlador para enviar essas informações ao Speeduino.

Isso irá configurar seu Speeduino para que você também possa executar simulações para verificar sua compilação antes da instalação ção.

Sensor de posição do acelerador

Antes que o Speeduino possa funcionar corretamente com seu motor, você também precisará calibrar o sensor de posição do acelerador. Isso deve ser feito usando o corpo do acelerador e TPS usado no motor. É melhorfaça isso enquanto o corpo do acelerador está instalado no motor.



Abra o menu Ferramentas e selecione Calibrar TPS:

Com o acelerador fechado, clique no botão Obter atual ao lado do campo de contagem ADC do acelerador fechado. Em seguida, mova o acelerador para a posição aberta e mantenha-o assim. Em seguida, clique no botão Obter Atual ao lado de Completo Campo de contagem de ADC do acelerador.

Clique em Aceitar para salvar as informações no Speeduino.

Configuração IO Auxiliar

O Speeduino também suporta a leitura de até 16 canais de entrada adicionais. Essas entradas podem ser Pinos Analógicos ou Digitais no Mega2560 (ou outro MCU em uso) ou de um dispositivo de aquisição de dados remoto (como o DxControl GPIO, por exemplo) através da porta serial secundária ou da interface Canbus (se disponível/capaz).

Um canal de dados configurado aqui terá os dados brutos disponíveis no TunerStudio como um medidor e também ser logável também.

Como usar

A configuração é dividida principalmente em duas categorias,

- **Pin MCU local** - Como configurar para usar um pino MCU local
- **Fonte de Dados Externa** - Como configurar para usar um externo Fonte de dados

Como configurar para usar um pino MCU local

A página de configuração é acessada a partir do menu suspenso Acessórios dentro do TunerStudio, selecione o Opção "configuração de canal de entrada auxiliar local"

Esta janela será aberta.

Na imagem acima, os primeiros dois canais foram configurados como um analógico local e um digital local entrada respectivamente.

- **Alias de entrada** - Este é um nome alternativo definido pelo usuário (até 20 caracteres) para o canal de entrada
- **Habilitar canal de entrada auxiliar** - Isso ativa / desativa o canal de entrada
- **ALFINETE** - O pino está selecionado (apenas para opções locais)

Entrada Pseudônimo O alias de entrada pode ser qualquer nome de caractere ascii que o usuário desejar, com até 20 caracteres. Isso também pode ser deixado como padrão.

Canal de entrada Aux Habilitar As opções aqui são

- **FORA** - O canal está desativado
- **EXT / CAN** - O canal está atribuído para uma fonte de dados externa

(esta opção só é visível se CAN_COMMANDS estiver habilitado nas propriedades do projeto. Veja aqui para mais em formação

- **Local_analog** - Selecione um pino MCU analógico local
- **Local_Digital** - Selecione um pino mcu digital local

ALFINETE Esta configuração está disponível apenas para seleções locais de pinos MCU. É o nome real do pino MCU.

Como configurar para usar uma fonte de dados externa

Para usar os canais de entrada auxiliares para dados externos, o IO secundário deve estar habilitado. Veja aqui para mais informações sobre como fazer isso.

A página de configuração é acessada a partir do menu suspenso Acessórios dentro do TunerStudio, selecione o Opção "Configuração de canal de entrada auxiliar externa"

Esta janela será aberta.

Para que as entradas de dados externas estejam ativas, a opção "Habilitar entrada de dados externos" deve ser habilitada.

Na imagem acima, os três primeiros canais foram configurados como analógico local e digital local entrada e uma entrada externa respectivamente.

- **Alias de entrada** - Este é um nome alternativo definido pelo usuário (até 20 caracteres) para o canal de entrada
- **Habilitar canal de entrada auxiliar externo** - Isso ativa / desativa o canal de entrada
- **Endereço CAN da fonte** - Is Real Can address do dispositivo de origem
- **Dados de Origem Byte inicial** - É o primeiro byte (dos 8 bytes em um comando canbus) onde estão os dados pode ser encontrado
- **Parâmetro de entrada Número de Bytes** - É o número de bytes em que os dados são armazenados (lsb primeiro)

Entrada Pseudônimo O alias de entrada pode ser qualquer nome de caractere ascii que o usuário desejar, com até 20 caracteres. Isso também pode ser deixado como padrão.

Canal de entrada Aux externo Habilitar As opções aqui são

- **FORA** - O canal está desativado
- **EXT / CAN** - O canal está atribuído para uma fonte de dados externa
- **Local_analog** - Selecione um pino MCU analógico local
- **Local_Digital** - Selecione um pino mcu digital local

Fonte CAN Endereço Este é o endereço hexadecimal do dispositivo remoto

Início dos dados de origem Byte Um comando de lata de dados tem até 8 bytes. Este valor define o primeiro byte de dados o valor dos dados começa em.

Número do parâmetro de entrada de Bytes O byte de dados pode ser feito de um único byte ou dois (palavra ou Valor de 16 bits)

Padrões de gatilho suportados

Falta de padrão de dente

Visão geral

Um gatilho de pedivela de dente ausente é usado como equipamento padrão por uma série de OEMs, principalmente a Ford, mas também é muito popular como acessório de reposição.

É composto de uma roda de manivela com um determinado número de dentes uniformemente espaçados e um ou mais dentes "ausentes". Os valores comuns são normalmente 60-2, 36-1, 24-1, 12-1 e 4-1, onde o primeiro número representa o número total de dentes que a roda teria se não faltasse nenhum. O segundo número depois do traço “-” indica o número de dentes que faltam.

Observação: Se houver um terceiro número (por exemplo, 36-1-1), os dentes ausentes não são sequenciais e este decodificador não se aplica. Não confunda contagens com barras “/”, já que os números após as barras representam dentes de came - não faltam dentes. As rodas com “+” indicam dentes adicionados em vez de faltantes e, novamente, este decodificador não se aplica.
.is-warning}

Formulários

As rodas dentadas perdidas podem ser usadas em praticamente qualquer motor e é uma das opções de reposição mais populares. Ele fornece uma resolução muito boa nas versões com maior contagem de dentes (por exemplo, 36-1 e 60-2) sem exigir muito da CPU para decodificar.

Requisitos de tempo

A manivela de dente ausente e os decodificadores de came exigem que a roda gire mais ou menos na mesma velocidade em toda a rotação. Para decodificadores de dente único ausente: Se o próximo dente não estiver dentro de $1,5 * \text{O Tempo Delta dos últimos 2 dentes}$, presume-se que acabamos de observar o dente ausente. Para mais de um decodificador de dente ausente, há um pouco mais de margem de manobra se o próximo dente não estiver dentro de $2 * \text{O Tempo Delta dos últimos 2 dentes}$, presume-se que acabamos de observar os dentes ausentes.

Normalmente, isso pode ser consertado garantindo que o motor de partida tenha corrente suficiente disponível para alimentar qualquer ponto mais difícil por meio dos cames de fechamento de rotação / abertura / acessórios do motor.

Se o motor de partida estiver bom e recebendo a tensão correta, certifique-se de que os componentes mecânicos do motor estão corretos.

Sintonizador Configuração de estúdio

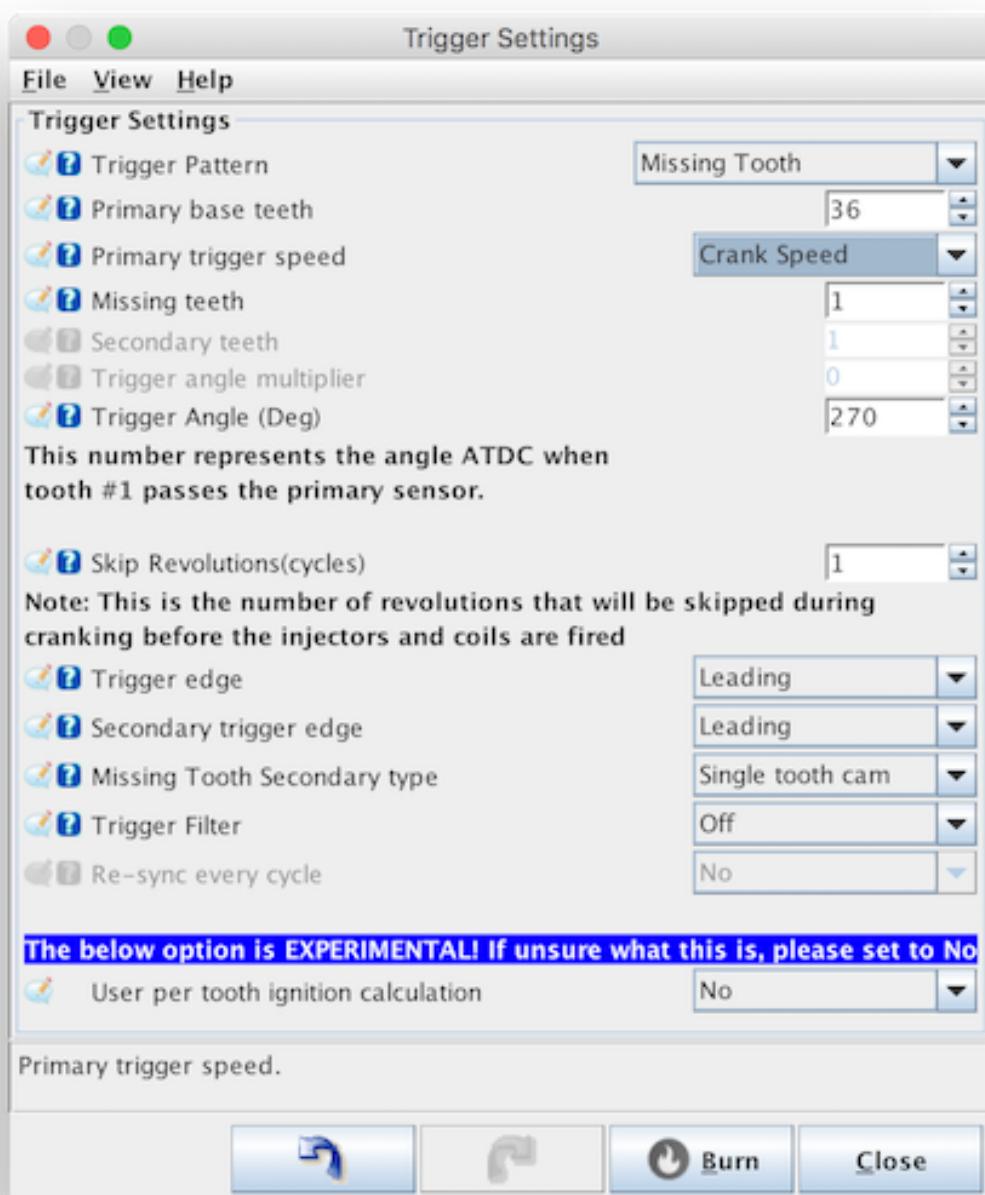


Figura 57: missingtooth_triggerconfig.png

Campos:

- Dentes da base primária: este é o número de dentes que a roda teria se não faltasse nenhum. Por exemplo, uma roda 36-1 tem apenas 35 dentes reais, mas você inseriria 36 neste campo.
- Dentes ausentes: o tamanho da 'lacuna' no número de dentes. Esses dentes ausentes devem estar situados em um único bloco (ou seja, há apenas uma única lacuna nos dentes)
- Ângulo de gatilho: Este é o ângulo em graus de manivela APÓS TDC (ATDC) do primeiro dente seguinte a lacuna

Configuração de tempo

O ângulo de disparo pode ser encontrado colocando o motor em TDC e, em seguida, calculando o quanto ele deve ser girado até o primeiro dente após a lacuna atingir o sensor.

Operação sequencial

O decodificador de dente ausente oferece suporte à operação sequencial se uma entrada de came adicional estiver presente. Se o modo Sequencial for selecionado para o sincronismo do combustível ou o sincronismo da faísca, o sistema esperará ver um sinal de came e não sincronizará corretamente sem ele. Observe que este é SOMENTE o caso se sequencial for selecionado para um ou ambos de combustível e tempo de ignição.

Este sinal de came deve assumir a forma de um único pulso a cada ciclo completo. Este pode ser um dente curto ou um arranjo do tipo meia-lua, desde que eletricamente haja apenas uma subida (ou descida) pulso por ciclo.

Diagrama de gatilho

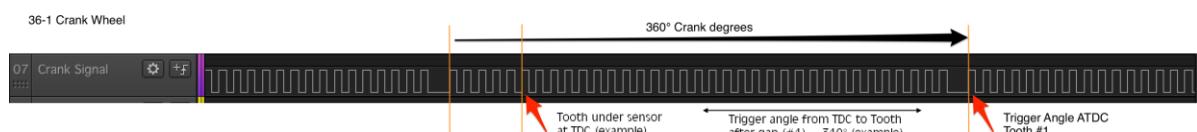


Figura 58: missingtooth_trace1.png

Dente faltando (velocidade do Cam)

O gatilho de velocidade do came com dente ausente é uma inovação do Speeduino, que permite uma função semelhante a uma configuração de roda dupla, permitindo assim a operação sequencial ou desperdiçada da centelha das rodas montadas no came ou distribuidor. A operação

seções primeiro para familiarização, pois esta seção irá apenas destacar as diferenças fundamentais para aqueles decodificadores comuns.

Este decodificador é composto de uma única roda de velocidade de came na mesma configuração de uma roda de dentes faltantes montada na manivela. O número de dentes deve ser uniformemente divisível em 720° . À medida que gira à meia velocidade da manivela, o sensor lê metade dos dentes da roda em cada revolução de 360° da manivela e os dentes restantes na próxima rotação da manivela. Um único dente ausente aparecerá em apenas uma das duas rotações da manivela e será usado como um indicador de fase, da mesma forma que o sistema de roda dupla usa o sinal de came.

Formulários

O came de dente ausente ou as rodas distribuidoras podem ser usadas com modificação ou fabricação do came ou da roda distribuidora, pois nenhum sistema OEM os usa originalmente. A roda deve ter pelo menos tantos dentes quanto cilindros, sem incluir o dente faltante. Isso geralmente requer o dobro do número de dentes como cilindros ou mais. Recomenda-se o maior número possível de dentes, ranhuras ou outros recursos legíveis (alvos de sensores) no espaço limitado para atender a esse requisito e maximizar a resolução. O sensor deve ser capaz de ler com segurança dentes menores ou próximos.

Devido aos dentes normalmente limitados, apenas metade dos dentes sendo lidos em cada revolução e o potencial de precisão reduzida devido ao desgaste do acionamento de sincronismo; a precisão do tempo pode ser reduzida em comparação aos sistemas de roda de manivela. Uma figura de erro não pode ser prevista aqui, pois o desgaste ou 'slop' de um determinado motor será único. No entanto, deve ser razoável supor que o erro de temporização não excederá a precisão de um sistema acionado por came equivalente ao OEM, como sistemas de distribuição típicos, ou possivelmente melhor devido a mais alvos de sensor.

Requisitos de tempo

A manivela de dente ausente e os decodificadores de came exigem que a roda gire mais ou menos na mesma velocidade em toda a rotação. Para decodificadores de dente único ausente: Se o próximo dente não estiver dentro de $1,5 * \text{O Tempo Delta}$ dos últimos 2 dentes, presume-se que acabamos de observar o dente ausente. Para mais de um decodificador de dente ausente, há um pouco mais de margem de manobra se o próximo dente não estiver dentro de $2 * \text{O Tempo Delta}$ dos últimos 2 dentes, presume-se que acabamos de observar os dentes ausentes.

Normalmente, isso pode ser consertado garantindo que o motor de partida tenha corrente suficiente disponível para alimentar qualquer ponto mais difícil por meio dos cames de fechamento de rotação / abertura / acessórios do motor.

Se o motor de partida estiver bom e recebendo a tensão correta, certifique-se de que os componentes mecânicos do motor estão corretos.

Sintonizador Configuração de estúdio

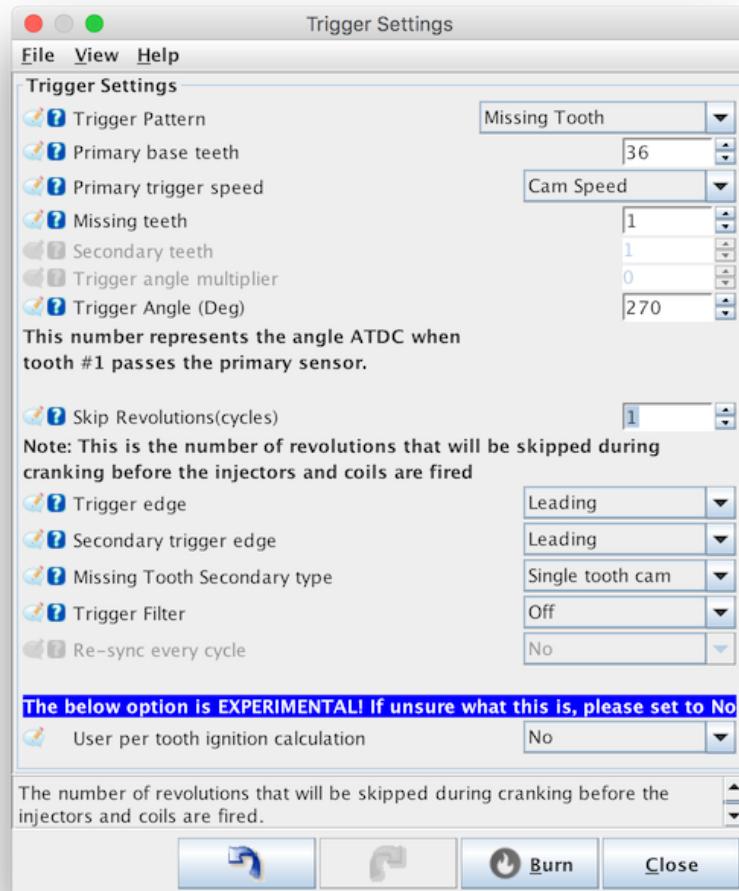


Figura 59: missingtooth_cam_triggerconfig.png

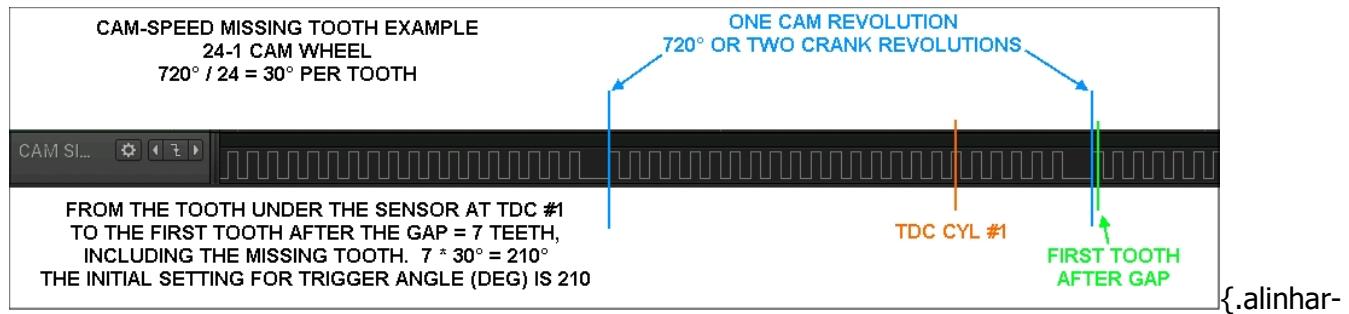
Campos:

- Dentes da base primária: este é o número de dentes que a roda teria se não faltasse nenhum, por exemplo, uma roda 36-1 tem apenas 35 dentes reais, mas você inseriria 36 neste campo.
- Dentes ausentes: o tamanho da 'lacuna' no número de dentes. Esses dentes ausentes devem estar situados em um único bloco (ou seja, há apenas uma única lacuna nos dentes). Recomenda-se a falta de um dente.
- Ângulo de disparo: este é o ângulo em graus de manivela APÓS o PMS (ATDC) do primeiro dente após a lacuna. Este número varia de -360 ° a + 360 °.
- Velocidade da câmera: certifique-se de que esta caixa esteja marcada para este sistema de velocidade da câmera.

Configuração de tempo

O ângulo de disparo é definido em Graus CRANK, não came.

Padrão de gatilho



Roda Dupla

Um gatilho de roda dupla é aquele em que existe uma roda primária com vários dentes combinada com um único pulso secundário para fornecer informações de localização. A entrada principal não deve conter dentes perdidos. Ambos os pulsos podem ser executados na velocidade do came ou da manivela, mas as operações sequenciais requerem que o pulso secundário esteja localizado no came. O projeto do gatilho secundário pode variar (por exemplo, um único dente curto, roda em meia-lua, etc.), desde que forneça apenas um único pulso por revolução.

Tal como acontece com outras rodas de contagem de dentes arbitrárias, o número de dentes deve ser dividido uniformemente em 360 (ou 720 se rodando na velocidade do came).

O dente # 1 é definido como o primeiro dente na roda primária APÓS o pulso na roda secundária roda.

Formulários

Os gatilhos de roda dupla são equipamento padrão em vários carros da marca Euro, especialmente os da VW e Audi. Eles também são um acessório de reposição popular devido à sua simplicidade e facilidade de instalação.

Sintonizador Configuração de estúdio

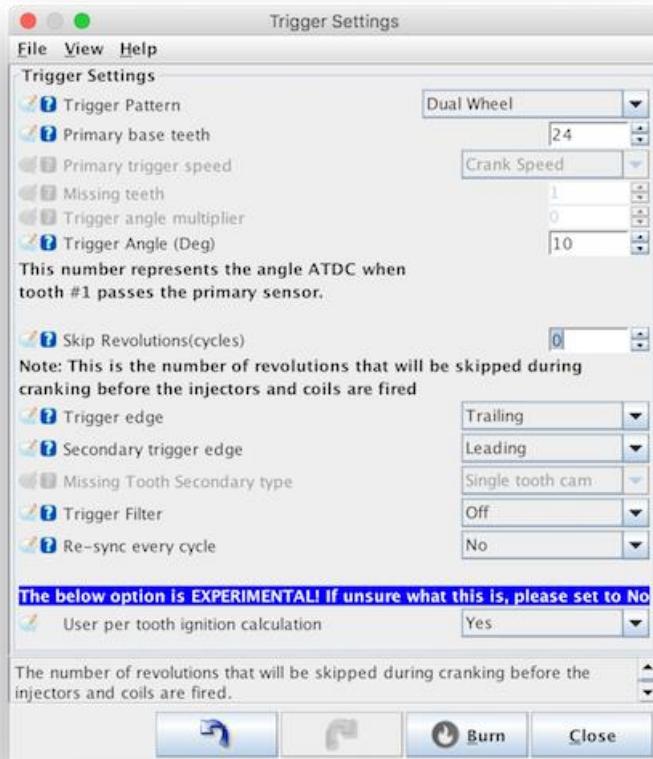


Figura 60: dualwheel_triggerconfig.png

Campos:

- **Dentes da base primária:** Este é o número de dentes na roda de entrada primária. Se a roda primária estiver localizada no came (ou de outra forma funcionando na velocidade do came), divida seus dentes por dois para esta configuração
- **Ângulo de gatilho:** Este é o ângulo em graus de pedivela APÓS o PMS (ATDC) do primeiro dente na entrada primária, seguindo o pulso único na entrada secundária.
- **Borda do gatilho:** Se o gatilho será retirado da borda anterior (ascendente) ou posterior (descendente) da entrada primária
- **Borda de gatilho secundária:** Como acima, mas para a entrada secundária
- **Sincronizar novamente a cada ciclo:** Se o sistema irá redefinir o nível de sincronização toda vez que a entrada secundária for vista. Isso pode ser útil para acionadores de manivela barulhentos que, do contrário, poderiam perder a sincronização permanentemente

e não recupere até uma reinicialização.

Configuração de tempo

O ângulo de disparo pode ser encontrado colocando o motor em TDC e, em seguida, calculando o quanto ele deve ser girado até o primeiro dente decíduo após o pulso secundário.

Operação sequencial O decodificador de dente ausente oferece suporte à operação sequencial se a entrada secundária estiver funcionando na velocidade do came. Se o modo Sequencial for selecionado para o tempo de combustível ou tempo de faísca, o sistema irá esperar que a entrada secundária esteja funcionando na velocidade do came e fornecerá apenas metade de pulsos de saída se este não for o caso.

Este sinal de came deve assumir a forma de um único pulso a cada ciclo completo. Este pode ser um dente curto ou um arranjo do tipo meia-lua, desde que eletricamente haja apenas uma subida (ou descida) pulso por ciclo.

Distribuidor Básico

O acionador do Distribuidor Básico é um decodificador muito simplista que espera uma entrada como um dis- saídas do tributor. Ou seja, 1 pulso por cilindro por ciclo.

Por esse motivo, o sinal não tem qualquer sinal de posição do cilindro e, portanto, sem um dente ausente / adicionado ou referência de sinal do eixo de comando, o Speeduino não pode calcular o ângulo do virabrequim, a fase do ciclo ou a atribuição do cilindro. Um distribuidor deve ser usado para encaminhar as faíscas resultantes para os cilindros corretos (com exceção dos motores monocilíndricos).

O sinal pode ser tão simples quanto os pontos do disjuntor de um antigo distribuidor pré-eletônico para um virabrequim sem ranhuras anormais, extras ou ausentes, desde que seja adequadamente condicionado para 0v-5v. A maioria dos que instalaram tacômetros de reposição está familiarizada com a simplicidade do sinal, sendo a única variação o número de pulsos em cada rotação do virabrequim.

Uma nota sobre a resolução

Por sua natureza, a resolução de uma roda distribuidora simples é bastante baixa. A resolução exata depende do número de cilindros (quanto mais cilindros, maior a resolução), mas mesmo com um motor de 8 cilindros existem apenas 4 pulsos por revolução. Isso pode afetar a precisão do tempo se executar o controle de ignição do Speeduino e, na maioria dos casos, atualizar para uma roda do gatilho de resolução mais alta for fortemente recomendado.

Sinal de gatilho

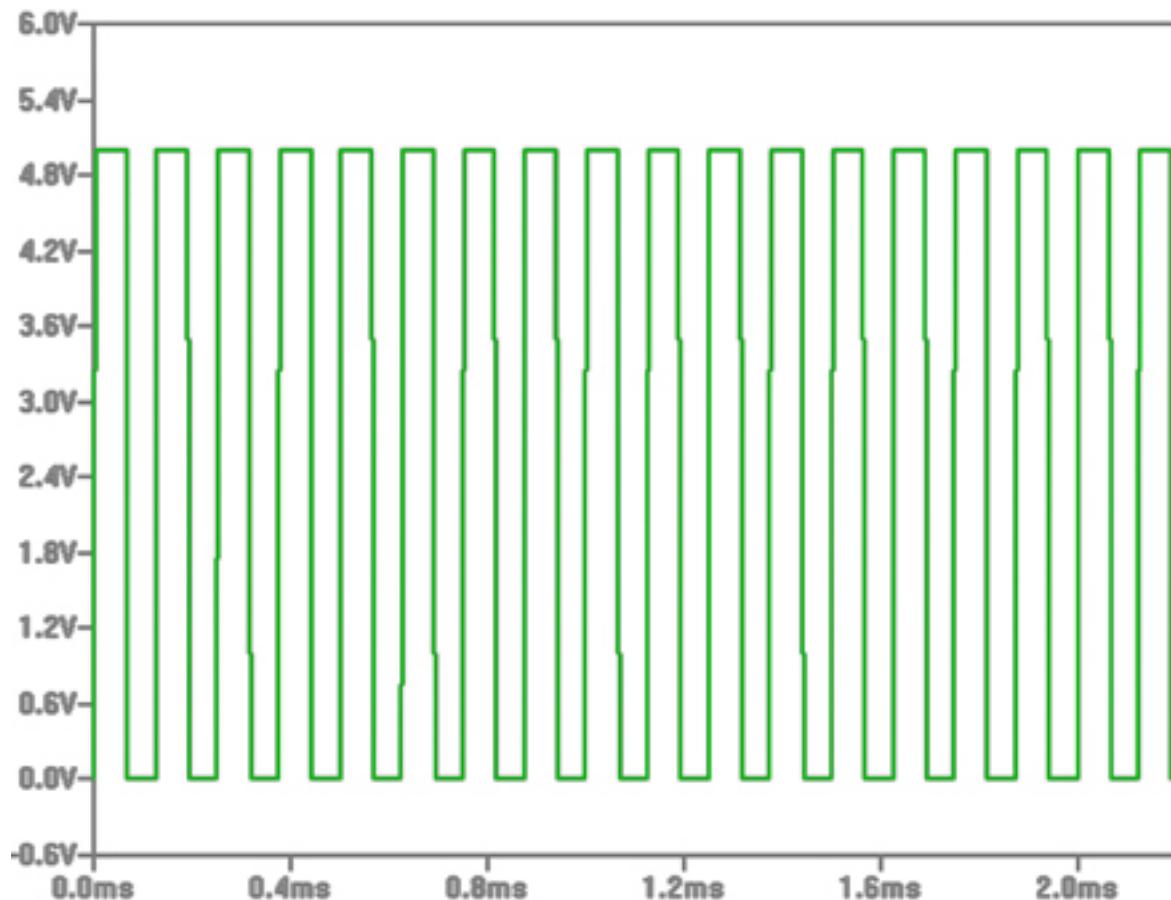


Figura 61: Basic_sistributor case.png

GM 7X

Este decodificador usa uma roda de gatilho GM com seis entalhes espaçados uniformemente e um entalhe desigual. O entalhe irregular é contado como # 3 com um total de sete entalhes.

Padrão 4G63

O gatilho 4g63 é usado em um grande número de motores Mitsubishi e Mazda de 4 cilindros. Ver abaixo para aplicativos.

É composto de sinais de manivela e came fornecidos por um sensor Hall ou um sensor óptico. O sinal é eletricamente o mesmo em ambos os casos.

Formulários

- Mitsubishi Lancer
- NA Miata / MX-5 (até 1997)

Sintonizador Configuração de estúdio

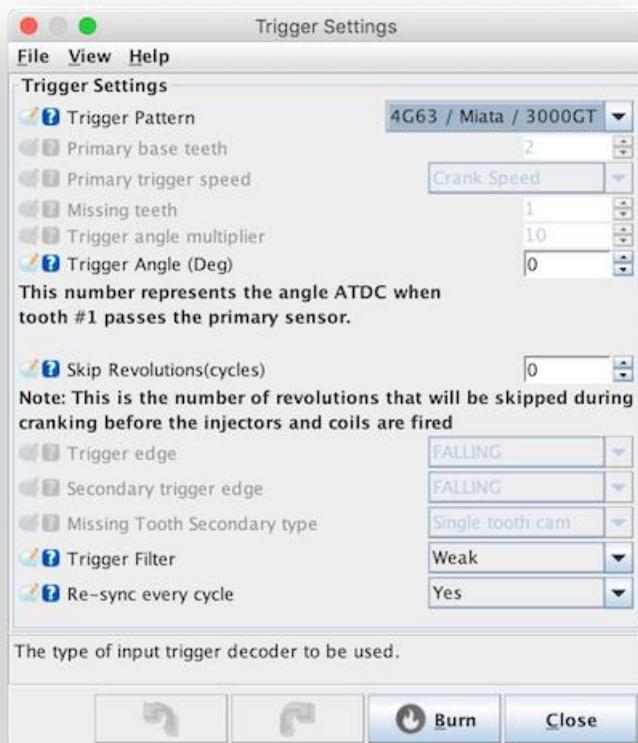


Figura 62: 4g63_triggerconfig.png

NOTA Dentro do `Opções de arranque` diálogo, certifique-se de que o `Corrija o tempo de acionamento com acionar` opção está ativada {`.is-warning`}

Ajuste de tempo

Na maioria dos casos, a alteração do ângulo do gatilho não deve ser necessária, no entanto, há uma pequena variação entre as versões OEM deste gatilho, portanto, alguns ajustes menores podem ser necessários. Uma vez que você tenha

o motor ligou, defina um ângulo de ignição fixo e verifique o tempo com uma luz de tempo. Se estiver alguns graus fora ($<20^\circ$), ajuste o ângulo de disparo aqui. Se estiver mais de 20° para fora, pode haver um maior problema.

Padrão de gatilho

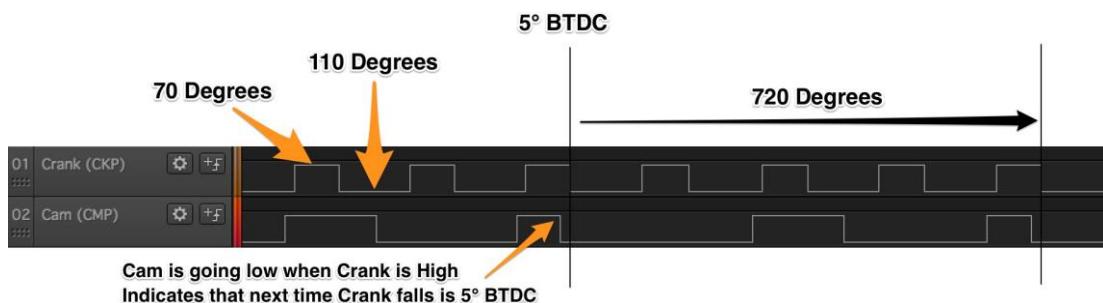


Figura 63: 4g63_trace.png

Visão geral

Esta é uma roda de 24 dentes com 12 de largura dentes e 12 dentes estreitos. O estreito fornece 3 graus de pulso, enquanto o largo fornece 12. Todas as bordas de queda estão a 15 graus de distância. Este decodificador usa as bordas descendentes, exigindo o sinal do came para determinar o ângulo da manivela.

O decodificador de "roda dupla" definido para 24 dentes e [CAINDO](#) borda deve ser usada em vez do decodificador 24X dedicado. Uma versão atualizada do decodificador 24X dedicado permanece um WIP. {é-perigo}

Sinal de gatilho

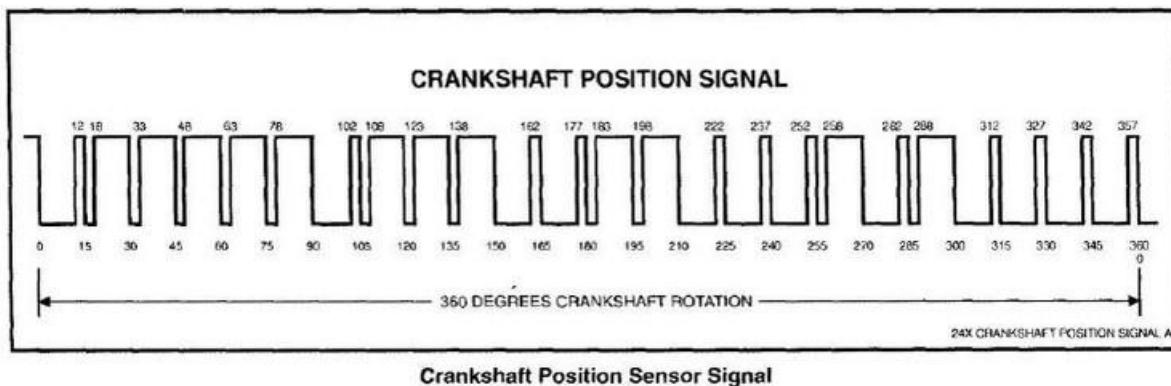


Figura 64: Traço de virabrequim GM 24X

Visão geral

Existem dois sinais, um da roda da manivela e outro do came. A roda da manivela emite uma série de quatro pulsos a cada 120 graus. Cada um dos quatro pulsos tem 20 graus de diferença e dura apenas 2 graus. As vantagens da roda do came uma vez a cada 360 graus ou 720 graus de manivela. O pulso dura 180 graus ou 360 graus de manivela.

Sinal de gatilho

Arquivo: Syncsignal.jpg

Harley Evo

A harley O padrão EVO é usado em motores V-Twin de '86 a '99. Este padrão funcionará em todos os motores EVO injetados.

Visão geral

O decodificador Honda D17 se aplica à família de motores Honda usando uma roda de virabrequim 12 + 1. O sinal da árvore de cames 4 + 1 não é usado atualmente com o Speeduino. Sem o sinal de came, todos os modos padrão de combustível e ignição até semi-sequencial e faísca perdida são suportados.

Formulários

- TBA

Configuração do Tuner Studio

Ajuste de tempo

Na maioria dos casos, a alteração do ângulo do gatilho não deve ser necessária, no entanto, há uma pequena variação entre as versões OEM deste gatilho, portanto, alguns ajustes menores podem ser necessários. Depois de ligar o motor, defina um ângulo de ignição fixo e verifique o tempo com uma luz de tempo. Se estiver alguns graus fora ($<20^\circ$), ajuste o ângulo de disparo aqui. Se estiver mais de 20° para fora, pode haver um maior problema.

Padrão de gatilho

A roda do gatilho da manivela consiste em 12 dentes uniformemente espaçados mais 1 dente 13 adicional que fornece informações de posição. O primeiro dente após este 13 é considerado Dente # 1



Figura 65: honda_D17.png

Miata 99-05

De MY99 em diante, Miatas mudou para um novo padrão de gatilho que, embora semelhante ao usado no 4g63, é mais tolerante a ruído e não depende de ambas as bordas de um dente sendo rastreado. Crucialmente, também permite o movimento do sinal de came em relação ao sinal de manivela que é necessário devido à adição de sincronização de came variável nesses motores. A sincronização pode ser determinada da mesma maneira, independentemente se o came variável está em seu movimento máximo ou mínimo.

O gatilho consiste em uma roda de 4 dentes localizada no virabrequim e uma roda de 3 dentes no came. Os dentes em ambas as rodas estão espaçados de forma desigual.

Formulários

NB Miatas de 1999 até 2005.

Sintonizador Configuração de estúdio

- O ângulo de disparo não deve precisar ser alterado uma vez que este padrão tenha sido selecionado (ou seja, certifique-se de está definido como 0)
- Ambas as bordas do gatilho devem ser definidas para [AUMENTAR](#)
- Para a maioria das instalações, A filtragem do acionador definida como Desligada ou Fraca é suficiente.
- No [Iniciando/Ocioso](#) -> [Configurações de arranque](#) diálogo certifique-se de que as seguintes opções são ligadas:
 - 'Corrigir o tempo de acionamento com gatilho'
 - 'Use o novo modo de ignição'

Padrão de gatilho

A roda de manivela contém 4 dentes, separados por 70 e 110 graus alternados.

A sincronização é determinada pela contagem do número de pulsos secundários (came) que ocorrem entre os pulsos primários (manivela) e podem ser confirmados em 2 pontos do ciclo. O primeiro pulso de manivela após 2 pulsos de came é o dente nº 6 e o primeiro pulso de manivela após um único pulso de came é o dente nº 2. O dente # 1 está localizado a 10 graus BTDC e não pode ser identificado diretamente, apenas em relação aos dentes # 2 e # 6. À medida que a sincronização da árvore de cames é movida como parte do VVT, os pulsos secundários permanecem dentro da mesma 'janela' em relação aos pulsos primários. A sincronização pode, portanto, antes confirmada em todas as cargas e velocidades, não importa o que VVTvalor está sendo usado atualmente.



Figura 66: miata9905.png

Não 360 Decodificador

Esta é uma variação do decodificador de roda dupla que pode ser usado com contagens de dentes que não se dividem uniformemente em 360 °. Este sistema de decodificador é geralmente exclusivo para uma determinada marca ou série de motor e, portanto, já recebeu um nome para identificar o tipo, como o decodificador Audi 135. Embora este decodificador de "divisor desigual" possa ser usado com uma variedade de contagens de dentes, nem todas as contagens de

Visão geral

Os gatilhos Daihatsu +1 são usados em vários motores de 3 e 4 cilindros da Daihatsu. Ver abaixo para aplicativos.

É composto de um único sinal de came fornecido por um sensor Hall. Isso deve ser alimentado no RPM1entrada no Speeduino

Formulários

- TBA (3 cilindros)
- TBA (4 cilindros)

Configuração do Tuner Studio

Basta selecionar a opção de gatilho Daihatsu +1.

Ajuste de tempo

Na maioria dos casos, a alteração do ângulo do gatilho não deve ser necessária, no entanto, há uma pequena variação entre as versões OEM deste gatilho, portanto, alguns ajustes menores podem ser necessários. Depois de ligar o motor, defina um ângulo de ignição fixo e verifique o tempo com uma luz de tempo. Se estiver alguns graus fora (<20 °), ajuste o ângulo de disparo aqui. Se estiver mais de 20 ° para fora, pode haver um maior problema.

Padrão de gatilho

Em motores de 3 cilindros, existem 3 dentes uniformemente espaçados a 0, 240 e 480 graus de manivela. Há um dente adicional (+1) localizado a 30 graus da manivela para fornecer informações de posição

O 4 cilindro é o mesmo, exceto com 4 dentes uniformemente espaçados. Os 5 dentes estão, portanto, localizados em 0,30, 180, 360 e 540 (graus de manivela, ATDC)

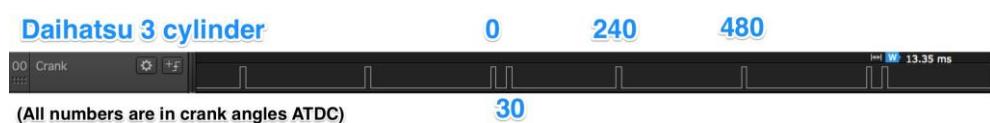


Figura 67: daihatsu_3_1.png

Subaru 36-2-2-2

A roda 36-2-2-2 é comum em muitos motores Subaru de 4 e 6 cilindros de aprox. 2000 em diante. Ele utiliza uma roda de gatilho de manivela contendo 36 dentes nominais, espaçados de 10 graus de manivela e 3 grupos de 2 dentes ausentes. Esses agrupamentos de dentes ausentes permitem que a sincronização seja determinada em mais 1/2 volta de manivela.

As primeiras rodas foram acionadas por VR, no entanto, após a mudança para o tempo de válvula variável, o Subaru mudou para os sensores Hall. A maioria das configurações são emparelhadas com um ou dois sensores de came 4-1, no entanto, estes não são necessário para sincronização no Speeduino.

Observe que existem 2 variações do padrão 36-2-2-2, o H4 e o H6. Embora visualmente muito semelhantes, os padrões têm diferentes agrupamentos de dentes e não são compatíveis. O suporte para a variante H6 deste gatilho foi adicionado no firmware 202103 e não funcionará anteriormente versões { .is-info }

Configuração

Ângulo de gatilho: 0 Trigger Edge: FALLING Trigger edge: N / A Pular Revoluções: 1 Trigger **Filtro:** Fraco (dependendo da instalação)

Padrão de gatilho

Os 3 conjuntos de 2 dentes ausentes estão localizados de forma que um grupo fique sozinho e os outros dois estejam localizados adjacentes um ao outro, com um único dente entre eles. A sincronização pode ser determinada detectando os 2 dentes ausentes e, em seguida, ver se há outro conjunto de dentes ausentes imediatamente após.

A compressão do cilindro 1 TDC ocorre no quarto dente após a lacuna única. O Speeduino observa qualquer período de dente ausente e, em seguida, espera para confirmar se ele é seguido por outro. A sincronização pode, portanto, ser determinada desta maneira em 2 pontos em uma única rotação da manivela.

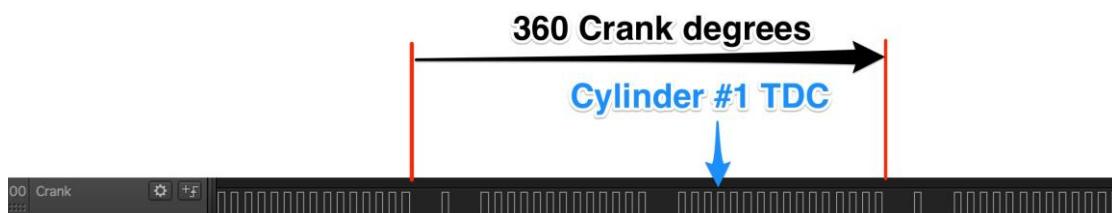


Figura 68: 36-2-2-2.png

H4 Padrão Nota: Muitos diagramas e imagens da roda do gatilho disponíveis online mostram a roda na parte traseira, fazendo com que ela gire no sentido anti-horário. Para a orientação correta, ao olharmos para frente do motor, a roda gira no sentido horário.

Placa V0.4

Visão geral

A placa v0.4 é uma placa de teste que foi desenvolvida com o objetivo de reproduzir os recursos das placas v0.3 existentes, mas com as seguintes melhorias:

- Custo mais baixo (principalmente devido ao tamanho reduzido, mas também algumas alterações de componentes)
- Mais compatível com caixas / gabinetes prontos para uso
- Stepper-style Opção de driver IAC
- Possui um único conector de 40 pinos para todos IO (excluindo alimentação de 12 V)

Observação: O v0.4 NÃO se destina a substituir a linha de placas v0.3! Os 2 são projetados com objetivos diferentes em mente. O v0.4 destina-se a ser integrado mais estreitamente à fiação existente, com o objetivo de que as placas de interface possam ser usadas para conectar facilmente através do conector IDC40. A menos que você entenda a interface da placa v0.4 e acredite que é a melhor opção para sua instalação, a v0.3 pode muito bem ser a melhor opção para você. {.is-warning}

Recursos da placa

As placas v0.4 incluem os seguintes recursos:

- 4 canais injetores
- 4 saídas de ignição
- Canais de entrada totalmente protegidos para CLT, IAT, TPS e O2
- Montagem de condicionador VR opcional nas entradas de manivela e came
- Local de montagem do sensor MAP
- Localização de montagem do módulo de passo DRV8825
- 4 saídas sobressalentes de média corrente (por exemplo, bomba de combustível, termoelétrica ventilador, controle de aumento, VVT, etc)
- 5 não preenchidos / configurados saídas sobressalentes opcionais de baixa corrente na seção "proto", incluindo saída do tacômetro
- Um único conector IDC de 40 pinos inclui todos os pinos necessários para a placa, com exceção do Entrada 12v

Formato físico

Observe que existem algumas diferenças entre as várias versões da placa, no entanto, a pinagem no conector IDC40 principal permanecem os mesmos.

Observação: os pinos do injetor têm marcações 1/2 e 2/2 para direcionar os injetores de maneira mais fácil e clara para os modos semi-sequencial e em lote. Se a aplicação exigir menos de 4 injetores, basta usar o pino 1/2 ou 2/2. Se a aplicação requer 5 ou mais injetores, é recomendado usar 1/2 e 2/2 quando disponíveis para distribuir mais uniformemente a corrente das bobinas do injetor que disparam. Consulte a Fiação do Injetor para detalhes mais específicos.

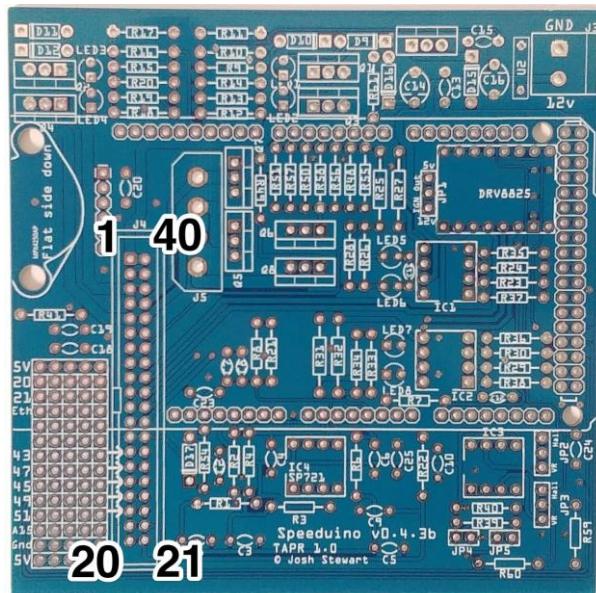


Figura 69: v0_4_board_annotation_1.jpg

Alfinete # Função

- | | |
|---|----------------------|
| 1 | Injetor 1 - Pino 1/2 |
| 2 | Injetor 2 - Pino 1/2 |
| 3 | Injetor 3 - Pino 1/2 |
| 4 | Injetor 3 - Pino 2/2 |
| 5 | Injetor 4 - Pino 1/2 |
| 6 | Injetor 4 - Pino 2/2 |
| 7 | Ignição 1 |

Alfinete # Função

- | | |
|----|--|
| 8 | Ignição 4 |
| 9 | Chão |
| 10 | Chão |
| 11 | Sensor MAP (0v-5v) |
| 12 | Chão |
| 13 | 5v |
| 14 | Proto Área 1 (0.4.4b + Sensor Flex) |
| 15 | Proto Área 2 (0.4.4b + Ventilador) |
| 16 | Proto Área 3 (0.4.4b + Bomba de Combustível) |
| 17 | Proto Área 4 (0.4.4b + Tacômetro) |
| 18 | Proto Área 5 (0.4.4b + Embreagem) |
| 19 | Refrigerante (CLT) |
| 20 | Entrada Temperatura do Ar (IAT) |
| 21 | Sensor O2 |
| 22 | Entrada TPS |
| 23 | Chão |
| 24 | Entrada Cam / VR2 + |
| 25 | Entrada de manivela / VR1 + |
| 26 | VR2- (não usado para sensor Hall) |
| 27 | VR1- (não usado para sensor Hall) |
| 28 | 5v |
| 29 | Idle Stepper 2B |
| 30 | Idle Stepper 2A |
| 31 | Idle Stepper 1A |
| 32 | Idle Stepper 1B |
| 33 | Ignição 3 |
| 34 | Ignição 2 |
-

Alfinete # Função

35	Impulso
36	Idle 2 (para uso com válvulas inativas de 3 fios)
37	PWM inativo
38	VVT
39	Injetor 2 - Pino 2/2
40	Injetor 1 - Pino 2/2

Montagem do Conselho

A montagem de uma placa completa é virtualmente idêntica à v0.3 e permanece relativamente simples, com todos os componentes passando por orifícios e etiquetados na placa. Embora não importe tecnicamente quais componentes do pedido são instalados, o seguinte é recomendado para simplificar:

1. Todos os resistores
2. Todos os diodos (incluindo LEDS)
3. Todos os capacitores> Observe que C14 e C16 são polarizados capacitores, o que significa que eles devem ser colocados da maneira correta ao redor. Os capacitores devem ser marcados com um sinal + em um lado. No PCB, o lado positivo é indicado por uma linha no símbolo do capacitor. {.is-warning}

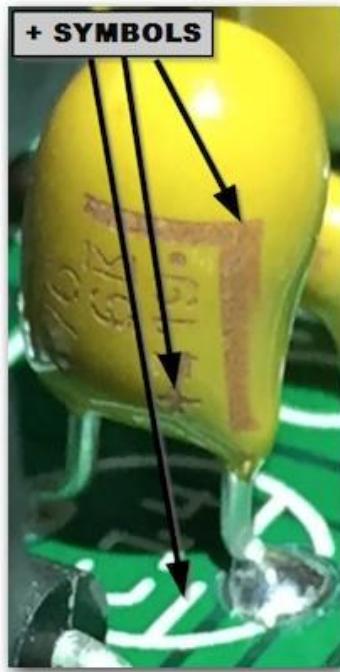


Figura 70: capacitor_orientation.png

4. Todos os cabeçalhos de jumpers (JP *)
5. Pinos do Arduino:
 - **Método sugerido:** Quebre os pinos do cabeçalho nos comprimentos necessários e insira em um Arduino Mega. Coloque a placa sobre os pinos e solde no lugar
 - Observe que nem todos os pinos na fila dupla final precisam ser preenchidos (embora não haja nenhum mal em fazer isso). Os pinos com números ímpares (por exemplo, D23, D25 .. DD53) não precisam de pinoseis.
6. Conector IDC 40
7. Soquetes IC
8. Todos os terminais de parafuso
9. Todos os MOSFETs
10. Regulador de energia
11. Sensor MAP (se usado)> NOTA: TODAS as placas de automontagem têm o sensor MAP com o orifício na parte superior. Todas as placas montadas normalmente terão o orifício na parte inferior {.is-warning}

Vídeo de instruções de montagem

Este vídeo é para a placa v0.3, mas se aplica amplamente aos designs da v0.4 também.

Configuração da placa

A placa pode ser configurada de várias maneiras, dependendo do hardware que você usa e da forma como sua configuração está configurado.

Saídas padrão da placa

Várias funções no Speeduino têm saídas ajustáveis ou podem ser definidas como Padrão da Placa. A seguir estão os pinos de saída padrão para o v0.4, no entanto, todas essas funções podem ser reatribuídas a outros pinos, se necessário (por exemplo, para usar as saídas de alta corrente onboard)

Função	Saída da placa	Pin Arduino
Controle de impulso	IDC Pin 35	7
VVT	IDC Pin 38	4
Idle 1	IDC Pin 37	5
Idle 2 (válvulas inativas de 3 fios)	IDC Pin 36	6
Bomba de combustível	Área proto (45) (0.4.4b + IDC 16)	45
Ventilador	Área proto (47) (0.4.4b + IDC 15)	47
Tacho	Área proto (49) (0.4.4b + IDC 17)	49
Lançamento / Embreagem	Área proto (51) (0.4.4b + IDC 18)	51

Componentes Opcionais

Se estiver usando um sensor de manivela VR, a placa exigirá a adição de um condicionador VR. A placa foi projetada para funcionar com o condicionador VR dual da JBPerf(http://www.jbperf.com/dual_VR/index.html)que se conectarão diretamente. Eles estão fora de estoque para compra direta há algum tempo, mas podem ser compilados a partir da lista de peças, pois as instruções ainda estão disponíveis.

Há também um quadro VR oficial que pode ser comprado na loja speeduino, que também se conecta diretamente.

A maioria dos revendedores parceiros tem seus próprios condicionadores com outros recursos, como LEDs, para quando o sinal está disparando alto / baixo. Outros condicionadores de RV de terceiros também funcionarão, mas obviamente nem todas as configurações podem ser

Proteção contra sobretensão SP721 Para usuários com dificuldade de obter o SP721 usado em algumas versões sões, consulte as informações na página SP721 Diode Alternate

Jumper Configs

Dependendo do tipo de manivela e sensores de came que você tem, há uma série de jumpers que irão precisa ser definido.

Alguns sensores VR podem enviar alta tensão AC para a placa Arduino. Se você não tiver certeza do tipo de sensor, identifique-o antes de conectá-lo à placa. Usar um sensor VR com os pinos 'diretos' fechados (JP2) e / ou (JP3) pode causar danos ao microprocessador. Perigo!

Os saltadores que configuração necessária são:

- JP1 - Define se as saídas de ignição são 12v ou 5v. Observe que mesmo se você definir isso para 12v, ** NÃO ** deve conectar diretamente a uma bobina de alta corrente. Essas saídas devem sempre ir para uma bobina de nível lógico ou um dispositivo de ignição
- JP2 - Se a entrada RPM1 (Crank) deve ou não ser roteada por meio do condicionador VR (opcional). Isso deve ser definido como VR ao usar um sensor de VR ou um sensor Hall que alterna entre 0v-12v
- JP3 - O mesmo que JP2, mas para a entrada RPM2 (Cam)
- JP4 - resistor pullup de 1k para entrada RPM1. Deve ser jumpeado ('On') quando um sensor é usado que alterna entre solo e flutuante (que é a maioria dos sensores de efeito Hall)
- JP5 - O mesmo que JP4, mas para a entrada RPM2 (Cam)

Para simplificar, os tipos de sensores mais comuns e suas configurações necessárias estão abaixo:

Sensor de manivela	Sensor Cam	JP2	JP3	JP4	JP5
Sensor Hall	-	corr edor	Fora e	Sobr a	For
Sensor VR	-	VR	Fora	Fora	For a
Sensor Hall	Sensor Hall	corr edor	corr edor	Sobr e	Sob re
Sensor VR	Sensor Hall	VR	corr edor	Fora Sob	Sob re

Conexão de 40 pinos

Você pode soldar os fios diretamente na placa ou usar conectores IDC (contato de deslocamento de isolamento). O IDC de 40 pinos é o conector que foi usado nos cabos de fita da unidade de

cabos de computador podem ser usados. Um cabo mais pesado, chamado cabo DuPont, é recomendado para uso de longo prazo. Mais tarde, na vida das interfaces IDE / ATA, a velocidade foi aumentada e isso exigiu um novo cabo fino de 80 fios. Esses cabos NÃO são compatíveis. Alguns dos pinos estão conectados, causando o azul mágico fumaça para ser liberada.

Revisões do conselho

Versão	Descrição	Ação
V0.4.4	Abnew ground up, todo SMD, design de placa que inclui drivers adicionais integrados e circuitos de proteção. É eletricamente e fisicamente compatível com todos os outros v0.4 versões.	BOM Não requerido
V0.4.4	Modificado para uma montagem automatizada mais fácil, incluindo alguns componentes SMD e montagem do sensor de pressão com o lado plano para cima. Adicionada opção de execução / programa. Apenas vendido oficialmente como placas completas	Não requerido Download
V0.4.3	Capacitores de filtro adicionados às entradas RPM primária e secundária. Pinça de tensão adicionada à entrada RPM secundária. Entrada de combustível flexível adicionada à área proto	Download
V0.4.2	Número considerável de melhorias de roteamento. Layout de área proto mais limpo. Pinça de tensão adicionada ao RPM primário entrada	Igual à v0.4.2
V0.4.1	Adicionada área de proto. Matriz de diodos substituída por SP721. Adicionado soquete de saída auxiliar de alta corrente opcional (J5). Diodo realocado no circuito de alimentação para evitar que o USB volte a alimentar 5v no trilho de 12v quando a ignição for desligada	Download
V0.4	Inicial liberação	Download

Placa V0.3

Visão geral

A placa v0.3 foi a primeira blindagem Speeduino amplamente disponível e é adequada para

Manual Speeduino 23/04/21
Muitas aplicações típicas de injeção e ignição de 1-4 cilindros (exceto motores de injeção direta).
Ele usa terminais de parafuso para todas as conexões, a fim de tornar a fiação de teste simples e
rápida para a prototipagem.

Recursos da placa

As placas v0.3 incluem os seguintes recursos:

- 4 canais injetores
- 4 saídas de ignição
- Canais de entrada totalmente protegidos para CLT, IAT, TPS e O2
- Montagem de condicionador VR opcional nas entradas de manivela e came
- Local de montagem do sensor MAP
- 4 saídas sobressalentes de corrente média (por exemplo, bomba de combustível, ventilador térmico, etc)
- Todas as E / S por meio de terminais de parafuso na placa
- Área proto com IO, SPI e interrupções de energia.

Formato físico

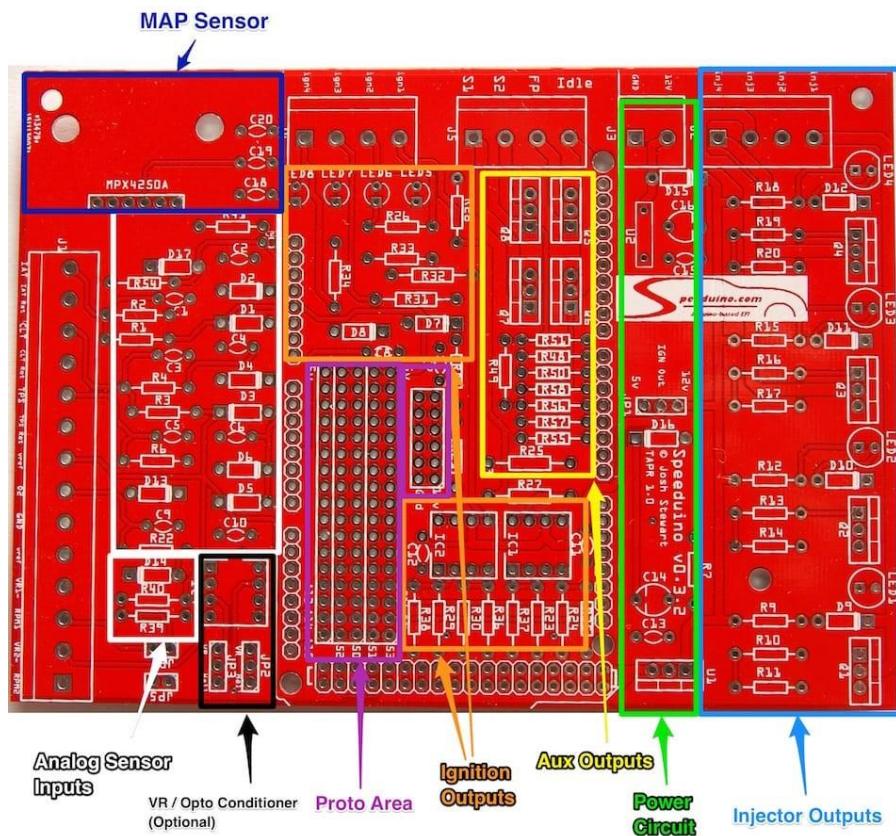


Figura 71: v0_3_2_board_annotation.jpg

Área proto

A protoárea pode ser usada para adicionar seus próprios circuitos ao Speeduino, se necessário, ou simplesmente como um ponto de acesso conveniente para várias conexões. As conexões quebradas para a placa protoestão:

- 5v e 12v
- Motivos
- Pinos SPI (MOSI, MISO, SCK e SS). Alternativamente, eles podem ser usados como IO digital genérico (Arduinopinos 50-53)
- 3 entradas analógicas genéricas (13-15)



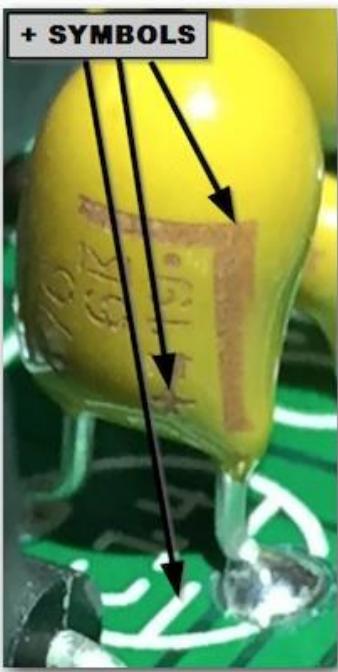
Figura 72: v0_3_2_proto_annotation.jpg

Montagem do Conselho

Consulte as revisões do conselho para obter um link para a lista de materiais (BOM) do seu conselho específico.

A montagem de uma placa completa é relativamente simples, com todos os componentes passando por orifícios e etiquetados na placa (consulte o BoM mencionado acima para obter a lista de peças). Embora não importe quais componentes do pedido são instalados, o seguinte é recomendado para simplificar:

1. Todos os resistores
2. Todos os diodos (incluindo LEDS)
3. Todos os capacitores
 - Observe que C14 e C16 são capacitores polarizados, o que significa que devem ser colocados da maneira correta. Os capacitores devem ser marcados com um sinal + em um lado. No PCB, o lado positivo é indicado por uma linha no símbolo do capacitor.



Orientação correta do capacitor 4. Todos os cabeçalhos de jumper (JP *) 5. Pinos do Arduino: * Método sugerido: quebre os pinos do cabeçalho nos comprimentos necessários e insira em um Arduino Mega. Coloque a placa sobre os pinos e solde no lugar * Observe que nem todos os pinos na extremidade da fila dupla precisam ser preenchidos (embora não haja nenhum dano em fazer isso). Os pinos com números ímpares (por exemplo, D23, D25 .. DD53) não precisam de pinos neles. 6. Soquetes IC 7. Sensor MAP (se usado) * NOTA: TODAS as placas de automontagem têm o sensor MAP com o orifício na parte superior. 8. Todos os terminais de parafuso 9. Todos os MOSFETs 10. Regulador de energia

Vídeo de instruções de montagem

Configuração da placa

A placa pode ser configurada de várias maneiras, dependendo do hardware que você usa e da forma como sua configuração está configurado.

Saídas padrão da placa

Várias funções no Speeduino têm saídas ajustáveis ou podem ser definidas como Padrão da Placa. O seguinte segue os pinos padrão para o v0.3

Observação: Esses padrões são aplicáveis ao firmware de janeiro de 2017 e mais recentes

Função	Saída da placa	Pin Arduino
Controle de impulso	Terminal de parafuso S2	7
VVT	Terminal de parafuso S1	6
Idle 1	Terminal de parafuso ocioso	5
Idle 2 (válvulas inativas de 3 fios)	Área proto (rotulada 53)	53
Bomba de combustível	Terminal de parafuso FP	4
Lançamento / Embreagem	Área proto (etiquetada como 51)	51

Componentes Opcionais

Se estiver usando um sensor de manivela VR, a placa exigirá a adição de um condicionador VR. A placa foi projetada para funcionar com o condicionador VR dual da JBPerf(http://www.jbperf.com/dual_VR/index.html)que se conectará diretamente. Outros condicionadores de RV provavelmente também funcionarão, mas não foram testados.

Agora também existe uma placa oficial de RV que pode ser usada, consulte o link à esquerda.

SP721 Sobretensão Proteção Para usuários com dificuldade de obter o SP721 usado em algumas versões sessões, veja informações sobre o SP721 Diode Alternate página.

Configuração de jumpers

Dependendo do tipo de manivela e sensores de came que você tem, há uma série de jumpers que irão precisar ser definido. Os jumpers que precisam de configuração são:

- JP1 - Define se as saídas de ignição são 12v ou 5v. Observe que mesmo se você definir isso para 12v, ** NÃO ** deve conectá-los diretamente a uma bobina de alta corrente. Essas saídas devem sempre ir para uma bobina de nível lógico ou um dispositivo de ignição
- JP2 - Se a entrada RPM1 (Crank) deve ou não ser roteada por meio do condicionador VR (opcional). Isso deve ser definido como VR ao usar um sensor de VR ou um sensor Hall que alterna entre 0v-12v
- JP3 - O mesmo que JP2, mas para a entrada RPM2 (Cam)
- JP4 - resistor pullup de 10k para entrada RPM1. Deve ser jumpeado ('On') quando um sensor é usado que alterna entre solo e flutuante (que é a maioria dos sensores de efeito Hall)
- JP5 - O mesmo que JP4, mas para a entrada RPM2 (Cam)

Para simplificar, os tipos de sensores mais comuns e suas configurações necessárias estão abaixo:

Sensor de manivela	Sensor Cam	JP2	JP3	JP4	JP5
Sensor Hall	-	corre do r	Fora e	Sobr e	For a
Sensor VR	-	VR	Fora	Fora	For a
Sensor Hall 0v-12v (requer condicionador VR)	-	VR	Fora	Fora	For a
Sensor Hall	Sensor Hall Flutuante	corre do r	corr edor	Sobr e	Sobre
Sensor VR	Sensor Hall Flutuante	VR	corr edor	Fora	Sobre

Revisões do conselho

VersionChanges	BOM
V0.3.7 Bluetooth adicionado cabeçalho	Mesmo que v0.3.6
V0.3.6 Adicionadas tampas de filtro para manivela e came entradas	Download
V0.3.5 Flex adicionado entrada de combustível para a área proto. Muitas melhorias de roteamento.	Download
V0.3.4 Limpeza de roteamento e mais útil serigrafia	Igual a v0.3.3
V0.3.3 Matriz de diodos substituída com SP721	Download
V0.3.2 Adicionada área de proto. Soquete IC sobressalente removido (não tinha sido usado)	Download
V0.3.1 Sensor MAP movido para mais perto da borda da placa. Roteamento reforçado nas saídas de alta corrente (incluindo injetores)	Download
V0.3 Inicial liberação	Download

Este trabalho está licenciado sob um Creative Commons Atribuição-NãoComercial-Compartilhamento pela mesma Licença 4.0 Internacional Licença internacional.