Pinpoint

- Especificações técnicas
- Instalação
- Inicio Rápido
- Troubleshooting

Especificações técnicas

O Pinpoint Odometry Computer é um coprocessador destinado a aplicações de robótica competitiva. Ele lê dois pods de odometria de “dead wheel” (encoders que leem a posição de uma roda não motorizada) e uma IMU interna para rastrear a posição e a orientação de um robô em tempo quase real.

Ele fornece informações processadas por meio de uma interface de comunicação I²C.

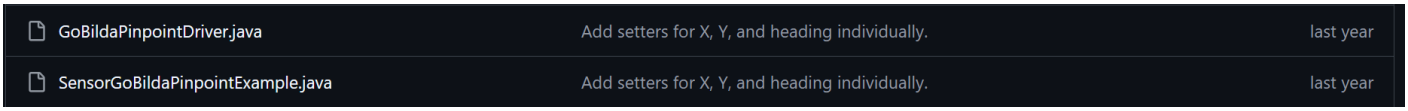
Especificações do produto

Especificação	Valor	Especificação	Valor
Tensão de entrada	3.3V - 5V	Velocidade máxima de rotação da IMU	2000RPM
Corrente de entrada	80mA	Frequência de atualização do algoritmo	1500hz
Velocidade máxima do encoder	256,000 eventos/segundo	Tipo de conector	4-Pos JST PH [FH-MC]
Endereço I²C	0x31	Frequência máxima de I²C (testada)	400khz

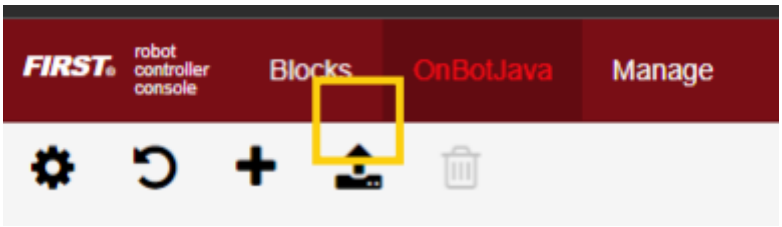
Instalação

OnBot Java

1. Acesse a página: Exemplo Pinpoint
2. Faça download dos seguintes arquivos:



3. Faça upload no Onbot

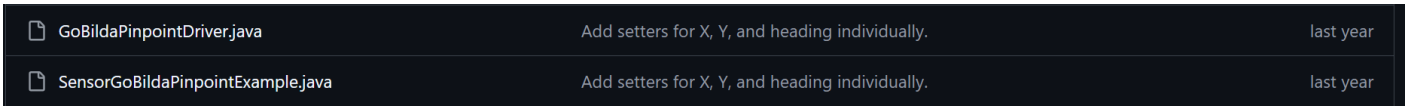


4. Construa o código
5. Após o Control Hub reiniciar você deve ver o código de exemplo na Driver Station.

> Parabéns, você concluiu a instalação!

Android Studio

1. Acesse a página: Exemplo Pinpoint
2. Faça download dos seguintes arquivos:



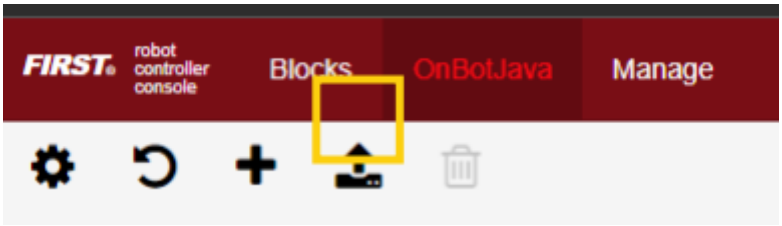
3. Copie os arquivos e cole-os na pasta TeamCode
4. Compile o código para o Control Hub
5. Após o Control Hub reiniciar você deve ver o código de exemplo na Driver Station.

>

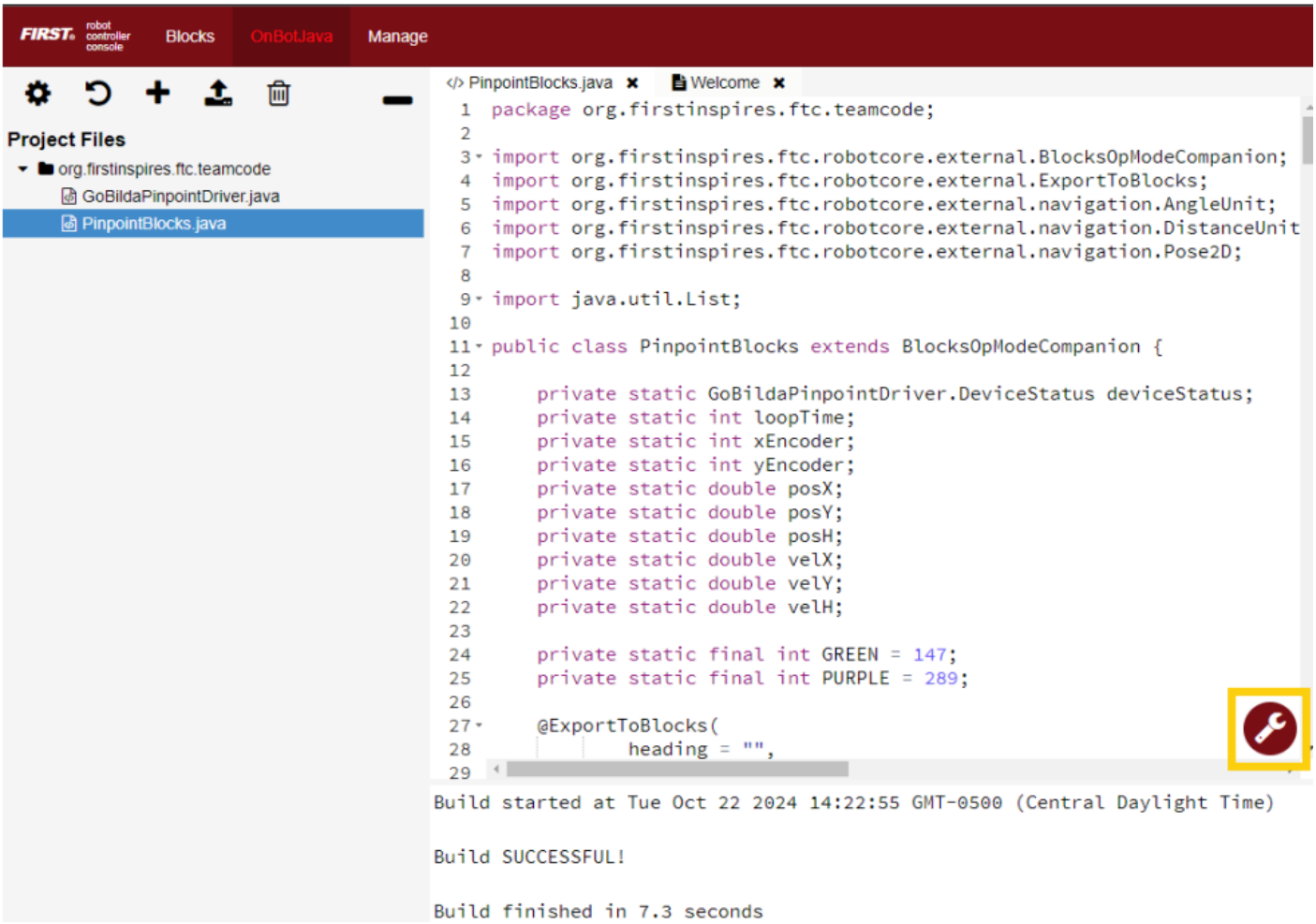
> Parabéns, você concluiu a instalação!

Java Blocks

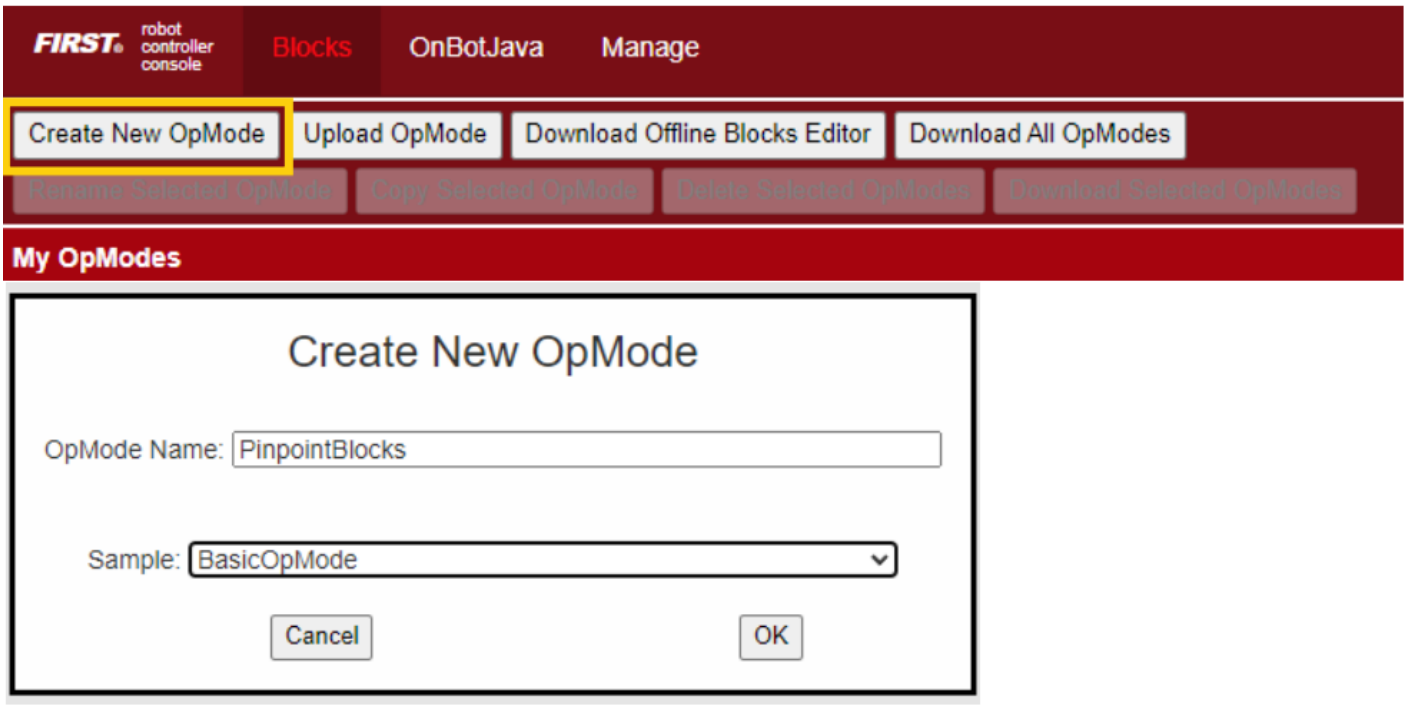
1. Instale o seguinte arquivo: Driver Blocks
2. Descompacte o arquivo
3. Faça upload dos arquivos recém extraídos no Onbot Java



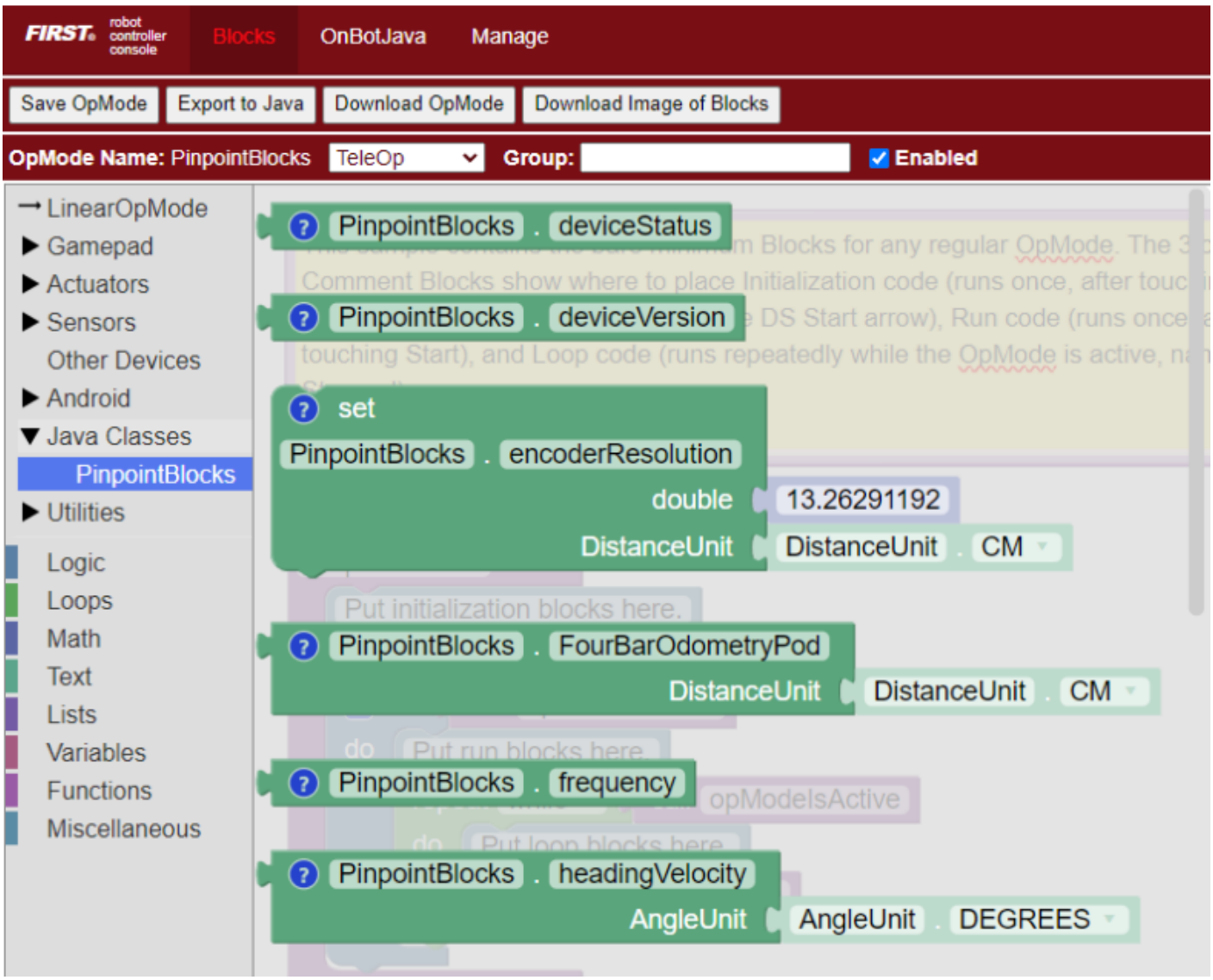
4. Construa o código



5. Crie um novo OpMode em blocos



6. Abra na aba esquerda *PinpointBlocks* e verifique se os blocos do Pinpoint aparecem



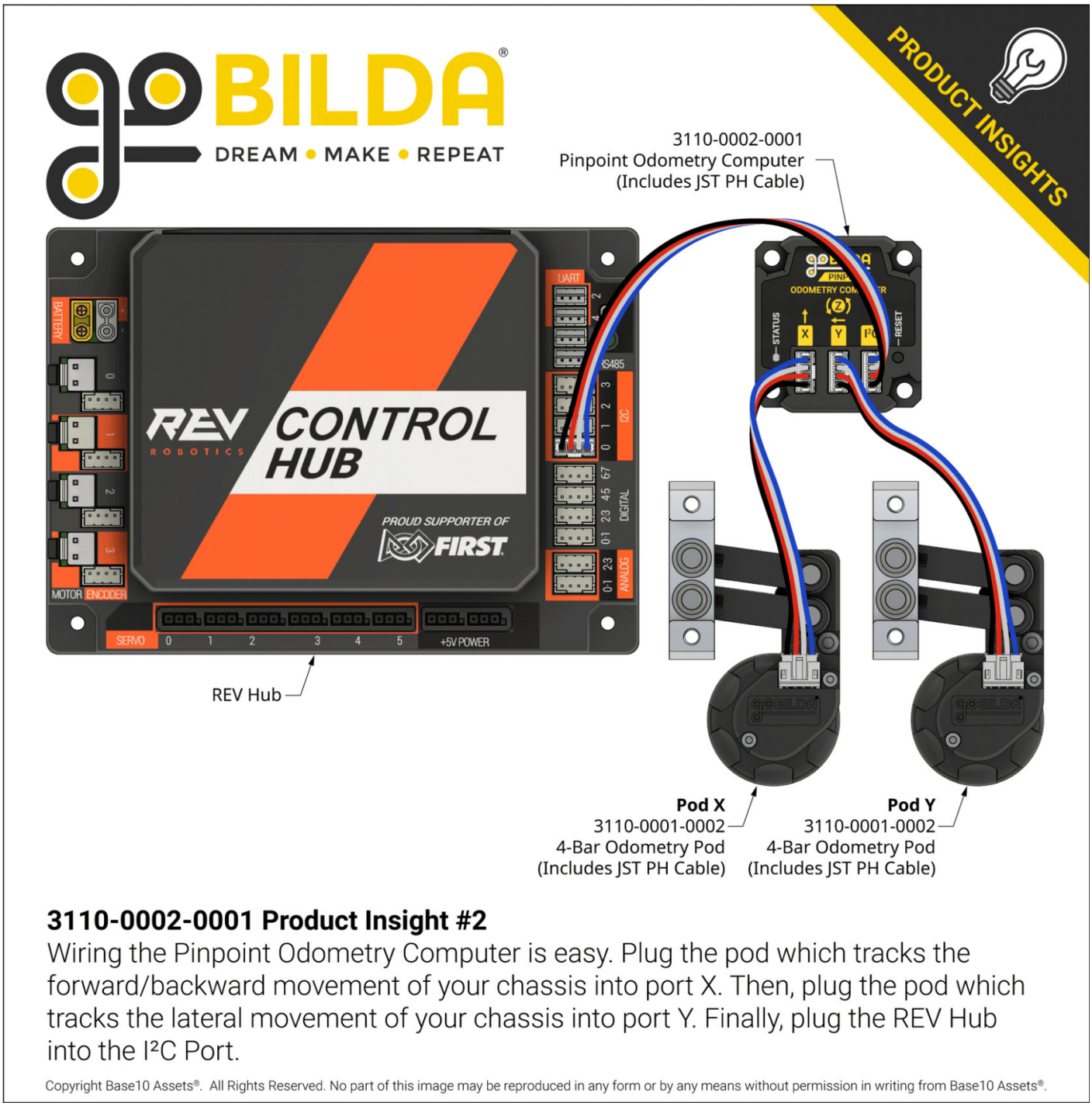
Parabéns, você concluiu a instalação!

Inicio Rápido

Veja a seção anterior para colocar os códigos de exemplos no seu projeto de programação.

> Isto vai orientar você na configuração de um Odometry Computer em um robô com dois Odometry Pods.

1. Monte o seu Pinpoint com o adesivo voltado para cima no chassi.
2. Monte dois pods de odometria no robô, com um rastreando para frente (o pod X) e um rastreando lateralmente (o pod Y).
3. Conecte eletricamente o seu sistema.



4. Certifique-se de que você está lendo dados do dispositivo com o código de exemplo.
   - Estaremos fazendo referência à posição estimada do robô nestes passos.
5. Defina a calibração da resolução do Encoder para os pods de odometria que você está usando.

```
odo.setEncoderResolution(GoBildaPinpointDriver.GoBildaOdometryPods.goBILDA_4_BAR_POD);
//odo.setEncoderResolution(13.26291192, DistanceUnit.MM);
```

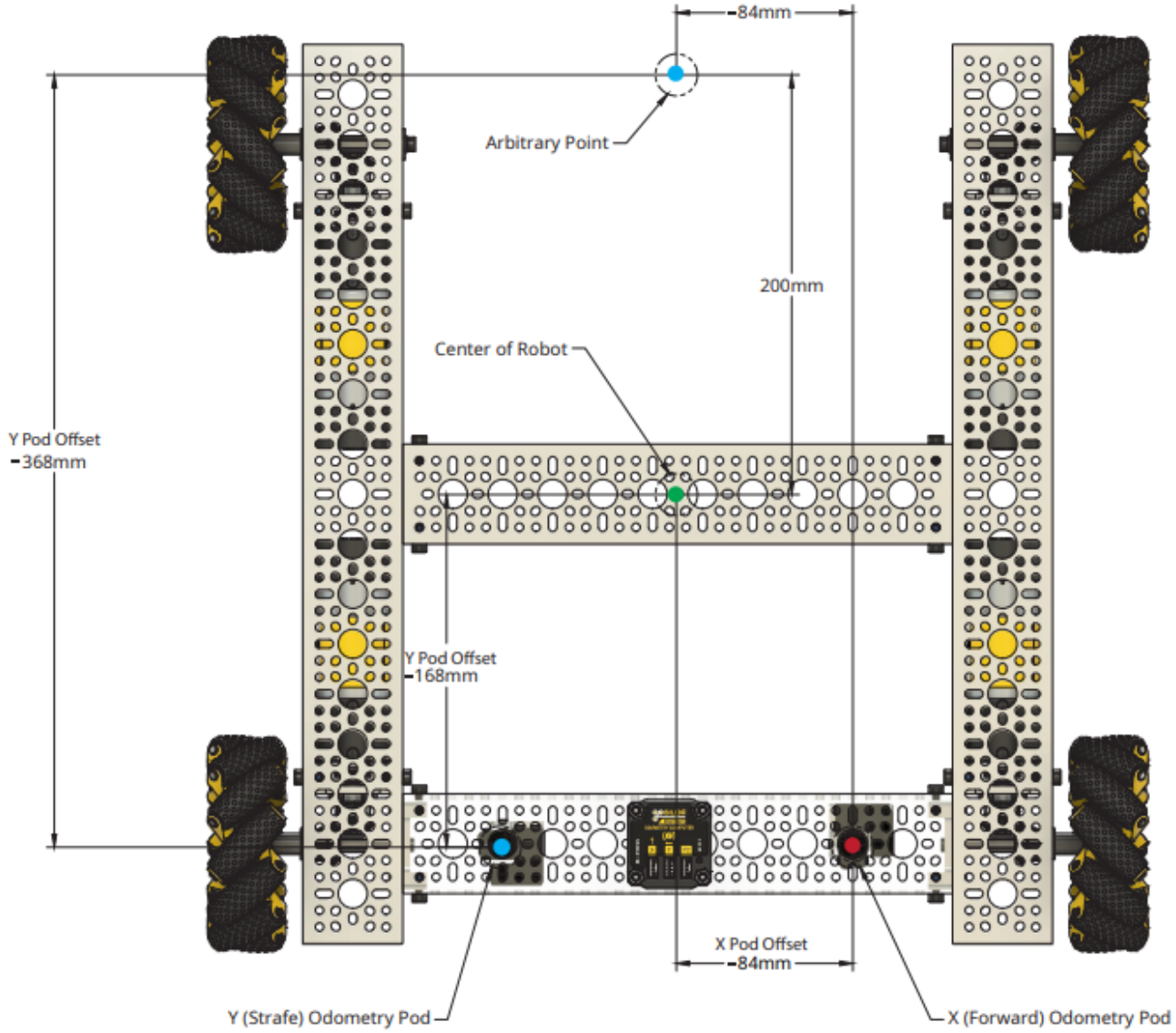
6. Certifique-se de que as leituras de heading e posição do robô estão estáveis. Elas não devem mudar enquanto o robô não estiver se movendo.

- Se mudarem, envie um comando ***resetPosAndIMU*** (com o código de exemplo isso pode ser feito clicando o botão A do controle).
- Ou clique no botão ***reset*** no Pinpoint.

7. Verifique se o **Status do Dispositivo** está como ***Ready*** ou se o LED no Pinpoint está verde.
   - Se estiver roxo, azul, ou laranja, verifique seus pods e certifique-se de que eles estão conectados corretamente.
8. Mova o robô para frente sem girá-lo. Certifique-se de que a posição X estimada aumenta.
   - Se X diminuir, inverta a direção do Encoder X definindo a X Encoder Direction como reversed.
   - Se Y se mover mais do que X, certifique-se de que o pod conectado à porta X está rastreando para frente.
   - Se nenhum dos valores se mudar, certifique-se de que seu pod X está funcionando.
9. Mova o robô para a esquerda sem girá-lo. Certifique-se de que a posição Y aumenta.
   - Se Y diminuir, inverta a Y Encoder Direction.
   - Se nenhum dos valores mudar, certifique-se de que seu pod Y está funcionando.

```
odo.setEncoderDirections(GoBildaPinpointDriver.EncoderDirection.FORWARD,
GoBildaPinpointDriver.EncoderDirection.FORWARD);
```

10. Meça os seus Pod Offsets.



11. Grave os Offsets no código.

```
odo.setOffsets(-84.0, -168.0, DistanceUnit.MM);
```

12. Gire o seu robô ao redor do ponto de rastreamento sem deslizar o robô para frente ou para os lados. As posições X e Y devem permanecer relativamente baixas.
    - Se elas variarem em mais de ~100mm ou 4”, verifique novamente os seus Pod Offsets. O problema mais comum aqui é que um dos offsets está positivo quando precisa estar negativo.
13. Gire o robô uma volta completa no sentido anti-horário.
    - O heading deve indicar muito próximo de exatamente uma rotação positiva.
14. Usando uma fita métrica, mova o seu robô ao longo do campo e meça o quanto ele se moveu, e compare isso com o quanto o Pinpoint informa que ele se moveu.
    - Se esses valores forem significativamente diferentes, verifique novamente a sua configuração de ticks-per-mm.

15. Redefina a posição estimada (com o botão ou com o seu código) e dirija o robô um pouco antes de retornar ao ponto de partida. Os valores devem estar próximos de zero.
    - Se tudo estiver funcionando, espere ver o seu robô retornar para dentro de cerca de 10mm (~0.5”) da posição inicial. Um valor maior é um bom indicativo de que algo está errado.
16. Em circunstâncias normais, não é necessário ajustar o Yaw Offset.

Troubleshooting

> Se você estiver tendo algum problema com seu Pinpoint Odometry Computer, consulte as sugestões de solução de problemas abaixo para tentar resolver o seu problema. Se o problema que você estiver enfrentando não estiver documentado abaixo, ou se você não conseguir resolvê-lo, não hesite em entrar em contato com o nosso suporte técnico!

Luzes de status nunca ficam verdes:

A luz de status do Pinpoint deve ficar verde em operação normal.
Se ela estiver sempre Roxa, Azul, ou Laranja, então o Pinpoint não está detectando um, ou ambos os seus pods de odometria. Esse problema provavelmente é um fio com defeito, mas também pode ser um Pinpoint com defeito, ou um pod de odometria com defeito.
Se apenas um pod for detectado, troque os pods X e Y. Se o erro/cor mudar (de azul para laranja, ou de laranja para azul) então o problema é um pod ou um fio. Tente trocar o fio entre os dois pods. Se a cor mudar novamente, então substitua o fio. Se o problema acompanhar o pod, então talvez seja necessário substituir o pod.
Se a cor estiver sempre Roxa, Azul, ou Laranja independentemente de quais pods estejam conectados, entre em contato com o suporte técnico.
Se a luz de status estiver sempre Vermelha, certifique-se de que o botão não está preso pressionado. Se ela ficar vermelha quando você executa seu programa no controlador, então certifique-se de que você não está chamando “resetPosAndIMU” ou “recalibrateIMU” constantemente. Se nenhuma dessas ações resolver o seu problema, entre em contato com o suporte técnico.

Dados de I²C sempre em 0

Certifique-se de que o seu código chama update(); toda vez que você precisar de novos dados!

IMU com deriva

Uma deriva muito, muito pequena no heading pode ser normal, mas o Pinpoint não deve derivar constantemente. Ele consegue detectar até mudanças muito, muito pequenas no ambiente, e é normal ver o heading estimado flutuar mesmo se alguém estiver apenas caminhando perto do robô.
Se você observar esse problema, clique no botão de reset enquanto tiver certeza de que o robô

não está se movendo. Se você ainda observar uma deriva de heading maior que 1° de deriva em 1min com o robô parado, entre em contato com o suporte técnico.

Imprecisões no rastreamento

Pequenas imprecisões no rastreamento são esperadas em qualquer sistema de odometria. Se você consistentemente observar a posição do seu robô corretamente reportada dentro de 6mm (~0.25”), e essa imprecisão for consistente, então, embora ajustes adicionais possam ser possíveis, você provavelmente está operando dentro da faixa normal de precisão para sistemas com dead wheel.

Se você observar grandes imprecisões no rastreamento, então é provável que exista um problema de configuração no seu setup. Siga os passos 6-13 do guia de configuração na página anterior. Se você ainda observar grandes imprecisões no rastreamento, primeiro veja se elas são consistentes. Execute o mesmo trajeto pré-programado repetidamente. Se você observar grandes mudanças de uma execução para outra, então certifique-se de que sua IMU não está derivando, e confira novamente se seus pods estão firmemente conectados, e montados no robô de forma que não fiquem balançando. Uma conexão solta ou um pod mal montado é a causa mais provável por trás de imprecisões inconsistentes.

Se a configuração estiver ajustada corretamente, e você observar grandes imprecisões consistentes, entre em contato com o suporte técnico para uma solução de problemas mais aprofundada.