

# Implementação de uma CNC pick-and-place

Vinicius Santiago Oliveira  
Instituto Federal de Educação, Ciência  
e Tecnologia do Rio Grande do Sul - IFRS  
Avenida São Vicente, 785,  
Farroupilha, RS, Brasil  
Email: [vinicius.santoliv@gmail.com](mailto:vinicius.santoliv@gmail.com)

Matias Rossato Muraro - Orientador  
Instituto Federal de Educação, Ciência  
e Tecnologia do Rio Grande do Sul - IFRS  
Avenida São Vicente, 785,  
Farroupilha, RS, Brasil  
Email: [matias.muraro@farroupilha.ifrs.edu.br](mailto:matias.muraro@farroupilha.ifrs.edu.br)

**Resumo**—As placas de circuito impresso são uma parte essencial do mundo moderno e estão presentes em quase todo tipo de aparelho eletrônico. A prototipagem de placas pode ser uma ferramenta útil para o aprendizado prático destas tecnologias, sendo portanto interessante produzi-las em um ambiente de ensino. Realizar a colocação dos componentes na placa manualmente, no entanto, demanda muito tempo e concentração do soldador devido ao tamanho ínfimo dos componentes SMDs, os mais usados na indústria atualmente e, por isso, convém encontrar um modo de automatizar essa etapa do processo de produção. Uma forma de conseguir isso é através de uma *pick-and-place*, uma máquina capaz de realizar a colocação dos componentes sobre uma placa de modo rápido e preciso. Este Trabalho de Conclusão de Curso propõe o desenvolvimento de um protótipo de uma máquina *pick-and-place* a partir da adaptação da estrutura de uma máquina CNC utilizando o controlador Arduino Mega e o *software* OpenPnP. Os resultados obtidos mostram que a CNC *pick-and-place* consegue realizar a colocação dos componentes na placa de maneira correta a partir de um projeto de uma placa de circuito impresso.

**Palavras-chave**—CNC, *pick-and-place*, OpenPnP, PCB

## I. INTRODUÇÃO

O surgimento da placa de circuito impresso, ou PCB, em meados da década de 1950 mudou a indústria da eletrônica. O método mais usado até então na montagem de circuitos eletrônicos era a montagem ponto-a-ponto, em que componentes eram soldados uns aos outros através de fios. As PCBs tornaram este método obsoleto, por facilitarem a criação de circuitos mais complexos e o método mais usado passou a ser a tecnologia *through-hole* (THT), em que os terminais dos dispositivos eletrônicos são inseridos em buracos na placa e soldados do outro lado, sendo que a placa serve ao mesmo tempo como suporte físico e como condutor elétrico [1].

Já na década de 1960, contudo, foi desenvolvida a tecnologia de montagem em superfície (*surface-mount technology*, ou SMT), onde componentes são montados diretamente sobre a placa sem a necessidade de furos. Isso reduzia o custo de produção das PCBs e permitia que os dispositivos SMD (*surface-mount devices*), como são chamados os componentes eletrônicos construídos usando essa tecnologia, pudessem assim ser montados nos dois lados da lâmina. Sua maior vantagem, porém, era a possibilidade de miniaturização dos circuitos. Sem precisar atravessar a placa, muitos componentes podiam ter suas dimensões reduzidas, o que permitia uma maior densidade dos componentes

nas placas. Essas vantagens acabaram fazendo com que os componentes SMD crescessem em popularidade, e atualmente a indústria tem preferido produzir a maioria dos circuitos integrados nesse formato, deixando os componentes THT em segundo plano. [2]

As dimensões dos SMDs são padronizadas e vão desde o padrão 01005, que mede 0,4mm x 0,2mm e é usado para componentes passivos, até padrões maiores como o SOIC, que é usado para circuitos integrados [3]. Seu tamanho reduzido, contudo, apresenta um problema, que é a dificuldade da montagem manual de um circuito eletrônico, exigindo um soldador experiente e muita atenção. Na produção de muitas placas esta atividade pode se tornar física e mentalmente cansativa, além de demandar muito tempo de um soldador.

Atualmente, a maior parte da produção em larga escala consiste em linhas de montagem onde são feitos, em ordem: a aplicação da pasta de solda na placa, geralmente usando-se estênceis ou impressão a jato; o posicionamento dos componentes, usando-se as chamadas máquinas *pick-and-place*; e, por fim, a soldagem de refluxo, que é onde a pasta de solda se funde e fixa os componentes na placa, normalmente feita em fornos de refluxo especializados, embora também seja possível usar chapas aquecidas ou mesmo ar quente para este fim. [4]

As máquinas *pick-and-place*, ou máquinas P&P, em específico são o foco deste trabalho. Elas consistem em máquinas robóticas que pegam componentes eletrônicos e os posicionam de forma autônoma em placas de circuito impresso nos seus locais corretos, sendo também mais rápidas e precisas que um soldador humano. Dessa forma, esse trabalho propõe o desenvolvimento de uma P&P a partir da conversão de uma máquina CNC utilizando para tanto o controlador Arduino Mega, responsável pela integração das diferentes partes da máquina; e o *software* OpenPnP, utilizado para realizar a operação e configuração do sistema. Esse protótipo poderá ser usado em ambiente estudantil onde poderá ser usado por estudantes e professores para tornar o processo de prototipagem de placas de circuito eletrônico mais rápido e simples.

Este trabalho será dividido da seguinte maneira. Inicialmente é feita uma introdução ao assunto, descrevendo brevemente o problema e a proposta do trabalho. A seção II apresenta a revisão bibliográfica, explicando em maiores

detalhes alguns conceitos-chave sobre o funcionamento do sistema e as soluções encontradas atualmente na literatura. A seção III apresenta em detalhes o protótipo construído e as alterações realizadas ao longo do trabalho. A seção IV mostra as configurações necessárias feitas no *firmware* e no OpenPnP e a seção V explica o processo de operação do programa. Na seção VI são mostrados os resultados encontrados no trabalho e a seção VII apresenta a conclusão do projeto. Por fim, a seção VIII inclui em anexo tabelas com as configurações vistas no trabalho e vídeos demonstrando o protótipo em operação.

## II. FUNDAMENTAÇÃO TEÓRICA

### A. Placa de Circuito Impresso

A placa de circuito impresso serve ao mesmo tempo como suporte mecânico e conexão elétrica entre componentes eletrônicos, através de trilhas condutivas gravadas em folhas de cobre laminadas sobre um substrato não-condutor. O substrato pode ser feito de vários materiais, mas é normalmente composto por fenolite, fibra de vidro ou ligas poliméricas. Sobre o substrato vão as trilhas condutoras de cobre que podem ser gravadas na placa de inúmeras formas, seja quimicamente através de ácidos ou fotolitografia, seja fisicamente através do desenho das trilhas por fresagem, embora já se pesquisem técnicas de impressão 3D para o mesmo fim [1].

### B. Tecnologia SMT

Os componentes usados em máquinas P&P são normalmente do tipo SMD, que ocupam um lado da placa de circuito e são geralmente menores que seus equivalentes THT, muito embora também existam máquinas capazes de fazer a montagem automatizada de placas que usam componentes deste tipo.

Como o melhor aproveitamento do espaço sobre a placa é de vital importância para a indústria e as dimensões reduzidas dos SMDs são sua principal vantagem, com o passar dos anos foram sendo desenvolvidos formatos cada vez mais compactos. Como consequência disso, foram desenvolvidos componentes com o espaçamento entre os condutores chegando a 0,5mm, bem como tecnologias distintas como o *ball grid array* (BGA), em que os condutores do componente se situam diretamente sob o mesmo em forma de matriz de pontos [5].

### C. Soldagem

Para que o circuito funcione como se espera, é preciso que os componentes do circuito tenham seus terminais ligados de forma permanente às trilhas de cobre da placa, e a forma como isso é feito é por meio da soldagem. A soldagem de um componente SMD é realizada usando pasta de solda. A pasta de solda é um composto homogêneo cuja composição inclui pó de liga de solda e é capaz de formar ligações metalúrgicas sob certas condições, tornando-a facilmente adaptável para a produção automatizada de juntas de solda [6].

A pasta é colocada sobre os contatos da PCB, sobre os quais serão depois colocados os SMDs, bastando então aquecê-la para realizar a solda e fixar os componentes sobre a placa.

### D. Máquinas CNC

O controle numérico computadorizado (CNC) é um sistema onde um microcontrolador é usado para operar uma máquina e é parte integrante do equipamento. Surgindo na década de 1970 nos Estados Unidos como uma evolução do controle numérico, este método substituiu o controle por *hardware* pelo controle por *software*, por possibilitar a programação dos movimentos das máquinas usando sistemas integrados de projeto CAD/CAM. Isso aumentou a versatilidade e precisão das máquinas e passou a permitir a verificação de eventuais erros no processo de produção através da simulação do programa CNC [7].

A tecnologia de controle computadorizado é usada atualmente em inúmeros tipos de máquinas e processos de manufatura, incluindo mas não se limitando a cortadoras, tornos, puncionadoras, retificadoras, furadoras e fresadoras. Muitas dessas máquinas consistem em robôs cartesianos com movimento independente em três eixos ortogonais, o que torna sua estrutura, movimentação e controle bastante similares ao de uma máquina P&P [8].

### E. Máquinas Pick-and-Place

Máquinas P&P são usadas na indústria desde meados da década de 1980 e são parcialmente responsáveis pela popularização do formato SMD. Inicialmente a colocação dos componentes em PCBs consistia numa linha de montagem com duas máquinas em sequência: primeiro uma máquina de baixa precisão e alta velocidade de operação conhecida como *chip shooter* selecionava e posicionava os componentes usando uma única torre rotativa. Neste método a placa é movida e o bocal se mantém fixo, o que limita a usabilidade da máquina a componentes pequenos que não saiam do lugar com o movimento da PCB. A placa seguia então para um outro robô, mais próximo das P&Ps atuais, usado para manipular peças maiores ou de formatos incomuns. Este robô era mais lento que um *chip shooter*, mas apresentava maior versatilidade e podia manipular componentes maiores. Com o passar dos anos, o alto custo de se adquirir e manter duas máquinas funcionando para posicionar componentes acabou levando muitos fabricantes da indústria a procurar uma nova técnica de produção. A solução encontrada foi incrementar as funcionalidades das P&Ps, para que todo o trabalho pudesse ser executado com apenas uma máquina [9].

Assim, as P&Ps se tornaram mais complexas: novos algoritmos foram desenvolvidos para otimizar o caminho que a máquina percorre para posicionar os componentes, e reduzir o número de trocas de bocal necessárias para manipular componentes de tamanhos distintos. Múltiplas ferramentas independentes operando em paralelo no cabeçote passaram a permitir a manipulação mais rápida das peças, acelerando o processo. Com essas melhorias as máquinas P&P ficaram mais rápidas e hoje são capazes de atingir velocidades de operação de mais de 150000 componentes colocados por hora [10].

Muito embora braços robóticos possam ser usados como máquinas P&P, sua forma mais comum é a de um

robô cartesiano operando no plano XY, semelhante a uma impressora 3D ou uma máquina CNC.

O funcionamento da máquina se dá da seguinte forma: uma placa é posicionada na sua área de trabalho. No *software* são definidos pelo operador os parâmetros da placa e outras preferências que dirão à máquina onde e como posicionar os componentes na placa. O robô então fará a calibração das posições usando sensores e iniciará o posicionamento dos componentes, pegando-os nos alimentadores e os colocando sobre a placa em suas posições e orientações corretas.

A construção desta máquina exige pelo menos 4 motores independentes, que correspondem ao número de movimentos que a máquina pode fazer simultaneamente: um para movimentar o cabeçote no eixo X, um para realizar o movimento do eixo Y, outro para realizar o movimento do eixo Z, esse com a finalidade de erguer e baixar a ferramenta de manipulação dos componentes no cabeçote e, por fim, um motor para rotacionar os componentes na ponta do manipulador.

Devido ao tamanho diminuto dos componentes SMD, a sua manipulação é realizada através de sucção por um tubo estreito de pressão negativa. O bocal de sucção é a ferramenta que de fato manipula a peça, e consequentemente a parte mais importante da máquina. Através de uma bomba de ar comprimido na qual o bocal afunila o ar gerando uma diferença de pressão, os componentes são levantados e segurados. Pode-se desse modo levar os componentes da área de coleta até seu local designado na placa. Em máquinas mais complexas utiliza-se vários bocais de diâmetros distintos para melhor manipular componentes de tamanhos diferentes e ainda, algumas P&P também são capazes de realizar a troca de bocal automaticamente. [11]

Para que a máquina P&P opere corretamente ela precisa saber, com precisão, onde a placa e os componentes estão e como estão posicionados. Isso pode ser realizado automaticamente através do uso de câmeras, mas neste trabalho optou-se por definir as coordenadas manualmente através do OpenPnP.

#### F. Alimentador SMD

Para simplificar e agilizar o processo produtivo, máquinas P&P costumam ter um local em sua área de trabalho dedicado a componentes ainda não posicionados de modo que, quando seja preciso posicionar um componente em específico, a máquina já saiba onde encontrá-lo e não perca tempo procurando por ele. Por essa razão é comum o uso de alimentadores de componentes, ou *SMD feeders*, em conjunto com máquinas P&P [5].

Como muitos componentes SMD são vendidos na forma de fitas em bobinas, com um tipo de componente único por rolo, um alimentador pode tomar a forma de um dispensador com inúmeras bobinas de componentes, uma para cada tipo usado na montagem da(s) placa(s). A posição das bobinas é informada à P&P de modo que cada componente seja selecionado de forma inequívoca. Uma vez que a ponteira esteja posicionada, a máquina retira um componente da

fita para posicionar na placa e a bobina correspondente se desenrola um pouco, de modo a garantir que a próxima peça esteja no mesmo lugar que a anterior. Este tipo de alimentador é adequado para grandes volumes de produção fazendo uso de amplas quantidade e variedade de circuitos integrados.

Para a produção em menor escala pode-se utilizar alimentadores mais simples, como bandejas nas quais os componentes situam-se em posições fixas com espaçamento padronizado. Neste formato, diferente do que acontece nos alimentadores de fita, os componentes não são levados a um ponto fixo onde são buscados pela ponteira, ficando estáticos em suas posições e o programa se encarrega de levar a ponteira até a coordenada de cada componente individualmente. Essa abordagem é menos eficiente que o uso de bobinas motorizadas mas é igualmente funcional.

#### G. OpenPnP

O controle das máquinas P&P é semelhante ao de máquinas CNC ou impressoras 3D, e pode ser feito por *softwares* dedicados pelos quais é feita sua calibração e operação. Existem programas gratuitos destinados a fazer a integração e controle de um sistema P&P, como o LitePlacer e o OpenPnP. Este último foi escolhido para ser usado neste projeto por ter uma documentação extensa e minuciosa e por ser um programa de código-aberto, permitindo que alterações sejam realizadas em seu código, caso isso se mostre necessário durante o projeto.

O OpenPnP se encarrega do controle do sistema, apresentando uma interface visual para o operador, recebendo e interpretando os arquivos CAD com as dimensões da placa e das posições dos componentes utilizados, controlando o funcionamento dos motores, e permitindo a integração do sistema com câmeras para fazer a calibração dos movimentos da máquina e a verificação visual dos componentes que estão sendo manipulados. A análise de visão computadorizada é feita usando o programa OpenCV, que também é gratuito e de código-aberto e vem incluso no OpenPnP [11].

#### H. Microcontrolador

Para realizar o controle desta máquina é necessário um microcontrolador para fazer a integração entre a máquina P&P e o computador onde se situa o *software* de controle do sistema. O microcontrolador faz uso de *softwares* auxiliares, ou *firmwares*, para ler as instruções enviadas pelo computador e traduzí-las em sinais para realizar a movimentação dos motores da máquina. Um dos *firmwares* recomendados pelo projeto OpenPnP chama-se Marlin e apesar de ser desenvolvido para impressoras 3D ele é também capaz de operar máquinas P&P, contanto que algumas alterações sejam realizadas.

### III. PROTÓTIPO

#### A. Estrutura Básica

A estrutura básica da máquina P&P tem a forma de um robô cartesiano com três eixos de movimento linear e foi originalmente desenvolvida por [12]. Para se executar a

movimentação do cabeçote nos eixos X, Y e Z são necessários motores de passo. Eles são motores elétricos que dividem a rotação em termos de passos, energizando cada fase em sequência, um estágio por vez [13], o que os torna mais precisos que motores CC. A CNC usada como base para este projeto faz uso de 3 motores de passo Nema 17hs4401, escolhidos por seu tamanho compacto e torque de 4,2 Kgf/cm. São usados na máquina *drivers* de potência DVR8825, que são necessários para o acionamento dos motores de passo. A máquina também faz uso de chaves micro switch KW11-3Z-2, as quais são empregadas como sensores de fim de curso para definir os limites máximo e mínimo de deslocamento dos 3 eixos.

A área de trabalho da CNC mede 220mm x 177mm, dimensões essas que correspondem ao deslocamento máximo do cabeçote nas direções X e Y da máquina. Somado a isso, o eixo Z ainda oferece originalmente 6mm de deslocamento vertical. O controle dos motores é realizado através da plataforma Arduino UNO e sua integração com os *drivers* é realizada através de uma *shield* CNC V3. A alimentação do sistema é feita por uma fonte chaveada de 12V com capacidade de 10A.

### B. Alterações e Componentes Adicionados

Nessa seção são apresentadas as alterações realizadas na CNC original para fazer a mesma operar como uma P&P. A máquina P&P desenvolvida é apresentada na Figura 1, na qual os seus principais componentes são apresentados.

O motor escolhido para permitir a rotação das peças no eixo Z se trata de um motor de passo bifásico Nema 8 com eixo central oco, desenvolvido para o uso em máquinas P&P. Ele contém um suporte fixo para ponteiras de sucção SMD e é bastante compacto, o que permite que ele seja facilmente fixado ao cabeçote da máquina.

O bocal ou ponteira da máquina P&P é um ponto crucial do projeto porque, sendo a única parte da máquina que entra em contato com os componentes eletrônicos, é o principal responsável por definir quais componentes poderão ser manipulados pelo sistema. Os 3 bocais escolhidos seguem um padrão feito para ser compatível com máquinas P&P JUKI e são denominados 505, 504 e 502, com diâmetros externos de 3,5mm, 1,5mm e 0,7mm e diâmetros internos de 1,7mm, 1,0mm e 0,4mm respectivamente. Estes bocais não são facilmente removíveis, logo, no protótipo desenvolvido a troca desses bocais deve ser realizada manualmente e a devida configuração deve ser realizada no OpenPnP, indicando quais componentes podem ser pegos com o bocal em uso. Cabe salientar que o OpenPnP tem suporte para realizar trocas automáticas de bocais, porém essa funcionalidade não é usada na P&P desenvolvida.

A máquina utiliza-se de um sistema de sucção para levantar os componentes e para isso emprega-se uma bomba de ar, instalada com a entrada conectada à mangueira que se conecta à ponteira. A bomba utilizada opera com tensão de 12V e tem potência de 10W. Adicionalmente, foram instaladas

mangueiras de ar comprimido de 1/4 de polegada para conectá-la à ponteira da máquina.

Originalmente, a CNC utilizava o *firmware* grbl com suporte para 3 eixos mas este se mostrou inadequado e precisou ser substituído. Muito embora existam diversas versões adaptadas do grbl para uso em máquinas com 4 ou mais eixos, não se teve êxito nos seus usos neste projeto, assim optando-se pelo uso do *firmware* Marlin.

O Marlin consiste em um *firmware* de controle especializado para impressoras 3D e é citado como uma alternativa ao grbl no site do OpenPnP. Essa mudança resultou na substituição do Arduino UNO por um Arduino Mega, um microcontrolador com mais capacidade e compatível com o novo *firmware*. A adoção do novo microcontrolador, por sua vez, exigiu a troca da *shield* CNC V3 pela *shield* RAMPS 1.4, que é construída de modo a funcionar em conjunto com o Arduino Mega. Uma versão adaptada do Marlin, chamada MarlinOnRamps4OpenPnP [14], elimina funções e bibliotecas desnecessárias para uma máquina P&P e foca especificamente na sua integração com o OpenPnP e com o *hardware* utilizado, sendo por isso a versão escolhida para este trabalho.

Como originalmente a CNC continha 3 eixos, foi preciso adicionar um quarto *driver* para se poder controlar o motor de rotação do bocal. Para regular a corrente no *driver* A4988 regulou-se a tensão no parafuso de ajuste conforme a equação 1:

$$V_{ref} = imax * 8 * R_s \quad (1)$$

onde  $V_{ref}$  é a tensão de referência,  $imax$  é a corrente máxima do motor de passo e  $R_s$  assume o valor de 0,1 no caso do *driver* A4988. A corrente nominal do motor de passo da ponteira é de 0,6A, por segurança optou-se operar com corrente de 0,4A, assim obtendo-se um valor de  $V_{ref} = 0,32V$ . Os 4 *drivers* dos motores de passo foram ajustados para operarem com uma resolução de 1/4 de passo por pulso de entrada, instalando para tanto um jumper no par central dos pinos de ajuste localizados sob o *driver* no *shield* RAMPS 1.4. O ajuste foi realizado dessa forma para evitar perdas de passo durante a execução do deslocamento do cabeçote ou da rotação da ponteira.

A máquina originalmente contava com três pares de sensores de fim de curso, sendo dois sensores para cada eixo, um para cada extremidade. Os sensores eram ligados em série, tornando os pontos de máximo e mínimo deslocamento iguais para o *firmware* instalado no Arduino Mega. Para que o sistema funcione adequadamente os sensores foram separados de modo a se obter sinais distintos de máxima e mínima distâncias de operação. A máquina CNC utilizada como base para o projeto inclui um filtro passa-baixa para redução do ruído no sinal dos sensores e, a fim de garantir o bom funcionamento da máquina, foi elaborado um novo circuito contendo 6 filtros independentes. Também foi necessária a instalação de novos condutores elétricos. O diagrama esquemático do filtro passa-baixa utilizado nesse protótipo é mostrado na Figura 2.

Por fim, foi elaborado um molde para organizar a área de trabalho da máquina em seções, visando facilitar



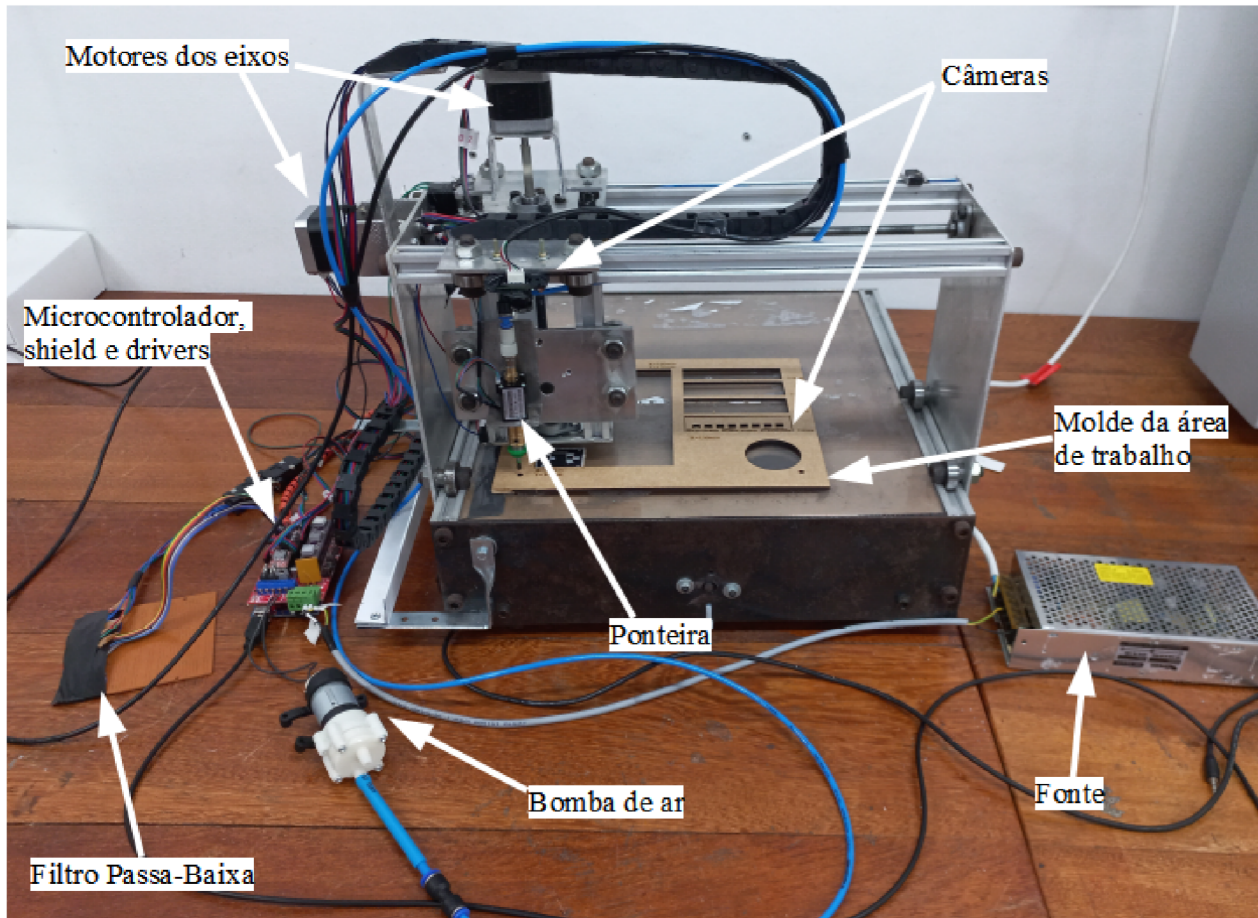


Figura 1: Visão geral do protótipo.

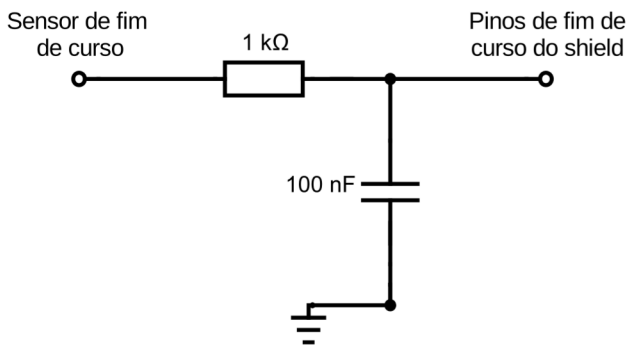


Figura 2: Diagrama do filtro passa-baixa utilizado em cada um dos sensores de fim de curso no protótipo.

o posicionamento da placa que será trabalhada e dos componentes que serão colocados sobre ela. O molde, cujo projeto pode ser visto na Figura 3, mede 210mm x 170mm. Ele contém recortes para posicionamento de uma ou mais PCBs e de até 4 alimentadores do tipo *tray*, além de um recorte circular para não bloquear a visão da câmera instalada sob a área de trabalho da máquina. Com esse molde

pode-se assegurar o preciso posicionamento da placa e dos alimentadores, simplificando o processo de configuração da máquina. O molde foi confeccionado em MDF de 3mm de espessura através de uma máquina de corte a laser.

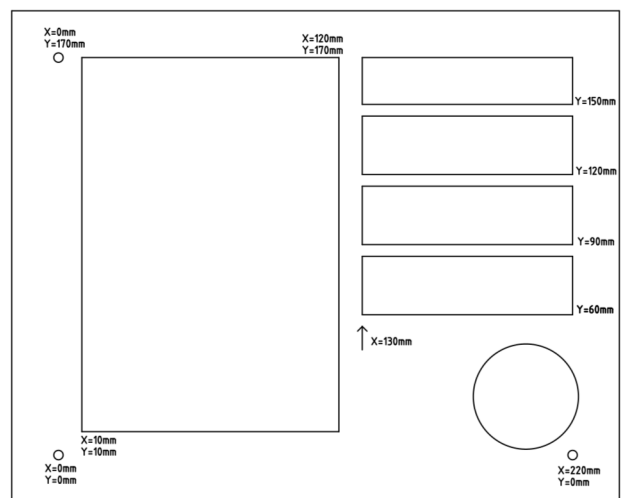


Figura 3: Projeto do molde.

#### IV. CONFIGURAÇÕES

##### A. Firmware

1) *Passo dos eixos*: Para a correta operação da P&P é essencial que o valor de deslocamento que é solicitado pelo OpenPnP seja corretamente realizado pela CNC. Para isso, o passo dos eixos no *firmware* deve ser adequadamente configurado. O STEP dos eixos representa quantos pulsos o Arduino deve enviar a um drive para que seja realizado um deslocamento de 1mm em um eixo linear que está conectado ao motor controlado por esse drive. Representa também quantos pulsos devem ser enviados a um drive para produzir uma rotação de 1° em um eixo rotacional.

A Figura 4 representa o percurso realizado pela informação e pelos comandos para o controle de um motor que aciona um eixo linear. A Figura 5, por sua vez, representa o percurso realizado pelas informações e pelos comandos para o controle de um motor que aciona um eixo rotativo.

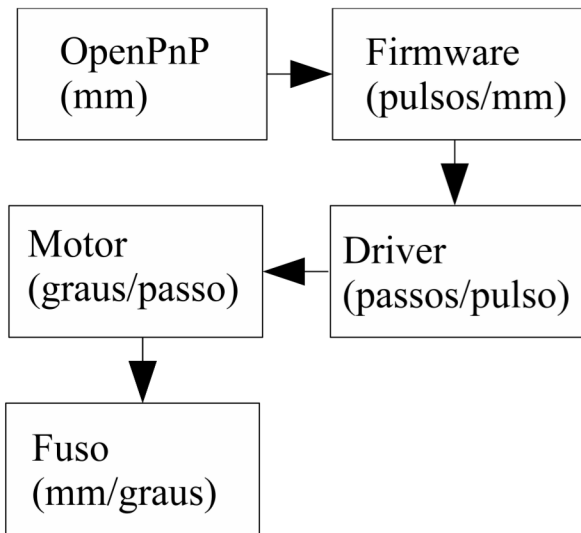


Figura 4: Percurso realizado pelas informações e comandos dos motores dos eixos lineares.

A equação 2 representa as conversões das informações e comandos para o acionamento de um eixo linear. Onde  $D$  é o deslocamento realizado,  $Kl$  é o número de pulsos que o Arduino deve enviar ao drive para produzir o deslocamento de 1mm,  $Kv$  é o número de pulsos por passo definido no *driver*,  $Km$  é o valor do ângulo de cada passo do motor,  $Kf$  é o deslocamento linear do fuso em milímetros por volta e  $Df$  é o deslocamento solicitado pelo OpenPnP.

$$D = Kl * Kv * Km * Kf * Df \quad (2)$$

$$(mm) = \left(\frac{pulso}{mm}\right) * \left(\frac{passo}{pulso}\right) * \left(\frac{grau}{passo}\right) * \left(\frac{mm}{grau}\right) * (mm)$$

A CNC apresenta os seguintes valores das constantes:  $Kf=1,25mm/360^\circ$ ,  $Km=1,8^\circ$  e  $Kv=0,25$  passos/pulso, dessa forma, para que  $D$  seja igual a  $Df$ , o valor para o  $Kr$  é de 640 pulsos/mm.

A Equação 3 representa as conversões das informações e comandos para o acionamento do eixo rotacional. Onde  $R$  é a rotação realizada,  $Kr$  é o número de pulsos que o Arduino deve enviar ao drive para produzir a rotação de 1°,  $Rf$  é a rotação solicitada pelo *firmware*.

$$R = Kr * Kv * Km * Rf \quad (3)$$

$$(grau) = \left(\frac{pulso}{grau}\right) * \left(\frac{passo}{pulso}\right) * \left(\frac{grau}{passo}\right) * (grau)$$

Para que  $R$  seja igual a  $Rf$ , com os mesmos valores de  $Kv$  e  $Km$  utilizados para o eixo linear, se obtém um valor de 2,22 pulsos/grau para  $Kr$ , que é o STEP do eixo rotacional.

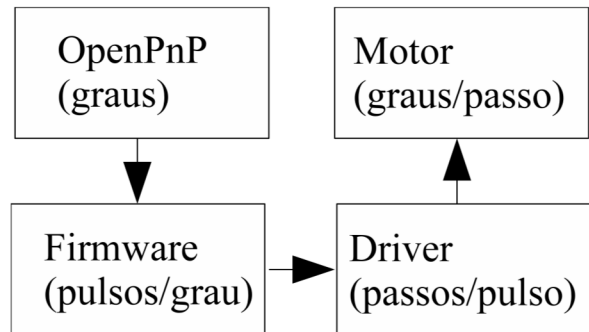


Figura 5: Percurso realizado pelas informações e comandos do eixo rotacional.

2) *Velocidade Máxima e Aceleração*: Foram realizados testes para definir os valores máximos de velocidade e aceleração dos motores da máquina. Utilizando a comunicação serial do Arduino foram enviados ao controlador os comandos em Gcode para movimentar cada eixo individualmente. Após a realização dos testes foi verificado que para não ocorrer perda de passo, os valores apropriados para a velocidade e aceleração dos motores são 7mm/s e 1mm/s<sup>2</sup>, respectivamente.

3) *Fins de Curso*: Para evitar colisões que podem causar o trancamento da máquina ou mesmo a queima dos motores, a opção de leitura contínua dos fins de curso foi ativada.

As configurações no *firmware* que devem ser alteradas para a operação da P&P são apresentadas na Tabela I da seção de Anexos.

##### B. OpenPnP

No OpenPnP, a CNC deve ser virtualmente construída e configurada. As configurações realizadas para operação da máquina desenvolvida são:

1) *Eixos*: Os eixos são criados na opção Axes e devem ser criados conforme sua existência na CNC. Dessa forma foram criados 3 eixos lineares e 1 eixo rotacional. Para a correta operação da máquina os eixos devem ser nomeados conforme a configuração realizada no *driver* na aba Gcode, conforme mostrado na Figura 6. Deve-se salientar que no eixo Z, responsável por realizar o movimento vertical da ponteira, é preciso habilitar a zona de segurança e definir a altura máxima e mínima de trabalho, de forma que a máquina evite colisões

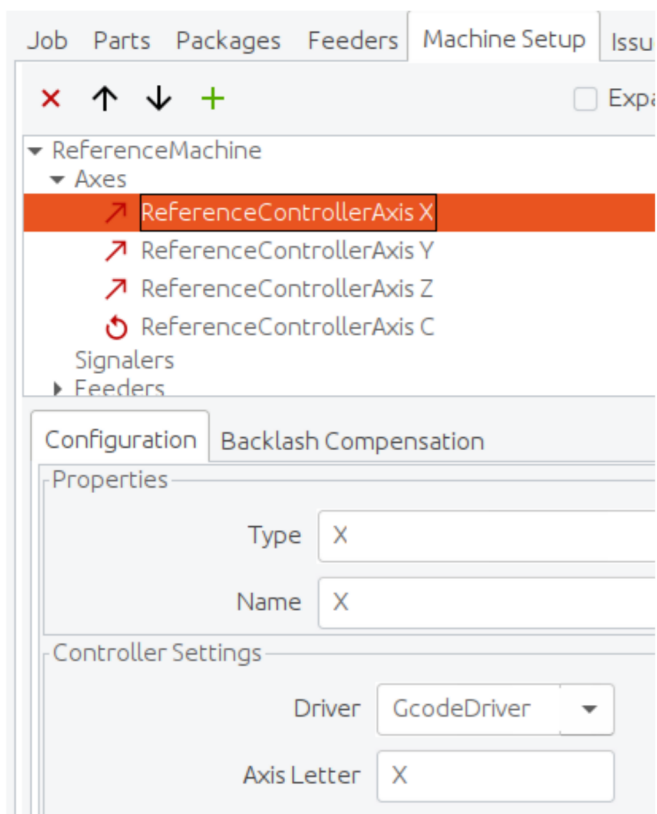


Figura 6: Criando e configurando os eixos.

com objetos na área de trabalho durante sua operação. Isso é mostrado na Figura 7. Ainda, para o eixo Z é configurado que a posição mais baixa está na coordenada -25mm, que é o mesmo valor utilizado por padrão no Marlin. Isso permite uma correta correspondência entre a posição do eixo Z no OpenPnP, no *firmware* e fisicamente na CNC.

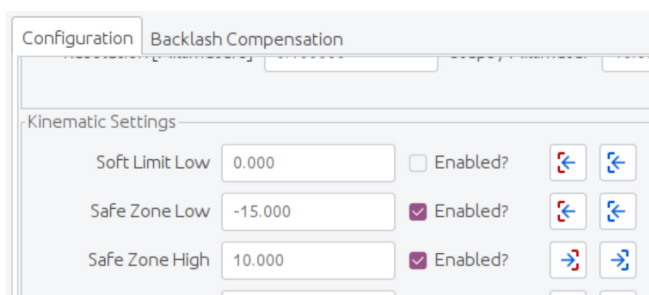


Figura 7: Definindo a zona de segurança do eixo Z.

2) *Cabeçote*: No cabeçote devem ser criados e configurados a ponteira, a câmera superior caso seja utilizada e o atuador que aciona a bomba de vácuo. Em máquinas P&P com múltiplas ponteiras é possível configurar um atuador para a bomba de vácuo no cabeçote e um atuador para a válvula de cada ponteira.

3) *Driver*: É preciso definir o *driver* utilizado para realizar a comunicação do OpenPnP com a CNC, sendo o GcodeDriver

uma opção recomendada no site do OpenPnP e a opção escolhida neste trabalho. Para utilizar este *driver* deve-se, na aba Machine Setup, adicioná-lo sob o menu Drivers, conforme a Figura 8.

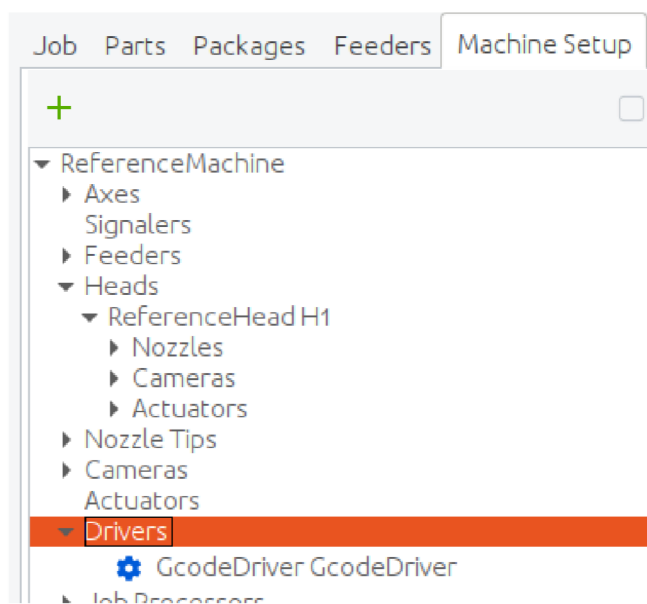


Figura 8: Criando o Driver.

Uma vez adicionado, deve-se entrar na aba de configurações do *driver* para definir qual porta USB do computador será utilizada e qual o *baud rate* da comunicação, no caso da utilização do Marlin esse valor é de 250000kbps.

Como o *driver* já foi criado, pode-se atribuir um *driver* aos eixos neste ponto, que será o GcodeDriver. Também pode-se alterar a resolução do movimento dos eixos, nesse projeto ela foi mantida em 0,1.

As configurações que efetivamente permitem a comunicação entre o OpenPnP e a CNC são realizadas na aba Gcode das configurações do *driver*, com os parâmetros das configurações apresentadas em anexo na tabela II. Inicialmente deve-se adicionar o texto **G28**, que é a função home, na função HOME\_COMMAND; e o texto **G01 {X:X%f} {Y:Y%f} {Z:Z%f} {C:C%f} F{feedrate:%0f}** na função MOVE\_TO\_COMMAND, que é a linha que permite ao programa interagir com os motores físicos. Essas configurações permitem movimentar os eixos da máquina através do OpenPnP.

4) *Bomba de vácuo*: Para que a máquina opere como uma P&P ainda são necessárias algumas configurações: Na função COMMAND\_CONFIRM\_REGEX do *driver* deve-se adicionar o texto **ok**, isso é utilizado para que o microcontrolador reconheça o retorno de comandos realizado pelo Marlin. Na função MOVE\_TO\_COMPLETE\_COMMAND adiciona-se o texto **M400**, que se trata de uma das funções próprias do Marlin em Gcode. Este comando faz com que o sistema espere o término da realização de uma ação antes de enviar o



próximo comando, permitindo assim que a bomba de vácuo seja acionada no momento correto da operação. Em testes foi observado que esta função atua adequadamente se o *Command Timeout* do GcodeDriver for configurado com um valor elevado, como 30000ms, caso contrário uma mensagem de erro de timeout pode ocorrer durante o funcionamento da máquina e ocasionar erro na operação. Esse valor deve ser elevado devido a CNC ter baixa velocidade de deslocamento, necessitando de vários segundos para completar grandes movimentos.

Deve-se configurar o atuador que ativará a bomba e para isso é preciso criar um atuador no cabeçote. Isso é feito na aba Machine Setup sob os menus Heads>>ReferenceHeadH1>>Actuators. No caso desse projeto foi usado o ReferenceActuator, conforme a Figura 9.

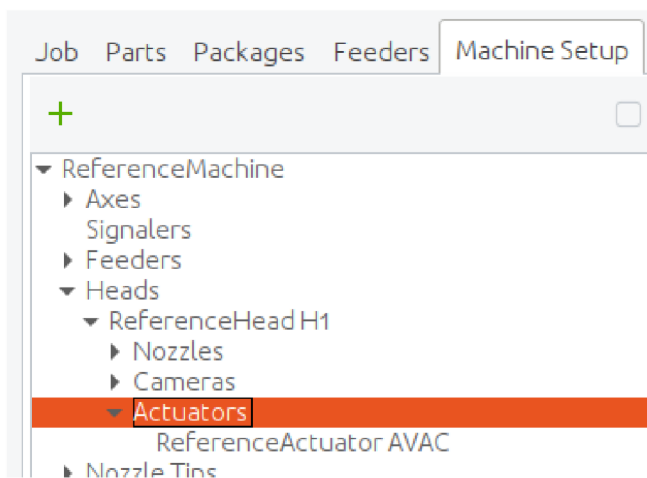


Figura 9: Criando um atuador.

Na seção *Properties* do atuador atribui-se a ele o nome que se desejar. Na seção *Machine Coordination*, deve-se marcar as opções *Before Actuation?* e *Before Read?*. Na seção *General* define-se o tipo do valor da variável, como o atuador utilizado deverá ligar e desligar a bomba quando necessário, será utilizado o tipo de variável *Boolean*. Ainda na mesma seção, deve-se definir o comportamento do atuador para cada estado da máquina. No menu *Enabled* define-se o atuador como *ActuateOn* para que a bomba possa ser ativada quando a máquina estiver operante; e nos menus *Homed* e *Disabled* define-se o atuador como *ActuateOff*, conforme mostrado na Figura 10.

É necessário fazer com que o sistema reconheça este atuador como o controlador da bomba. Para isso, no submenu Heads>>ReferenceHeadH1>>Nozzles como é mostrado na Figura 11, deve-se selecionar a ponteira que será usada em conjunto com a bomba, no caso desse projeto, há apenas uma chamada *ReferenceNozzle N*. Nas configurações da ponteira existe uma aba chamada *Vacuum* e dentro desta existe um menu chamado *Vacuum Actuator*, na qual deve ser selecionado o nome previamente definido para o atuador.

Por fim, com o atuador criado e sua função definida, deve-se configurar sua ação. No *GcodeDriver*, na aba *Gcode*

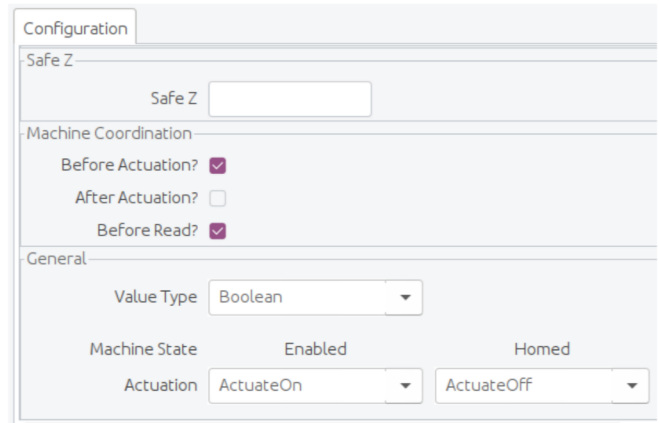


Figura 10: Configurando o atuador.

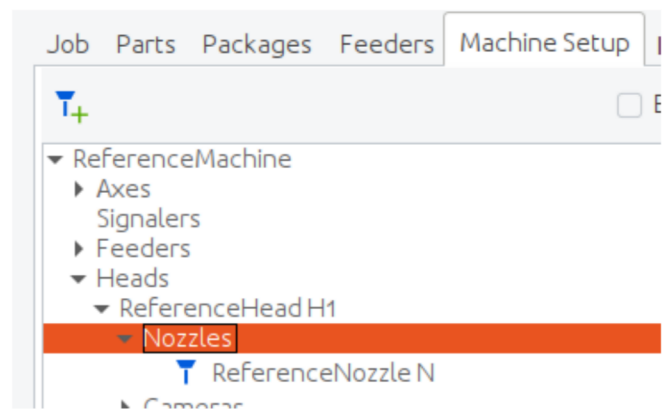


Figura 11: Configurando a ponteira.

e sob o menu *Head Mountable*, deve-se selecionar o atuador que será configurado. No menu *Setting*, a função *ACTUATE\_BOOLEAN\_COMMAND* deve receber o texto `{True:M42 P9 S255}{False:M42 P9 S0}`. O comando M42 é uma função Gcode genérica do Marlin usada para controlar o estado de pinos. O P9 significa que o pino controlado será o pino D9 da *shield* Ramps 1.4, que por se tratar de uma placa desenvolvida para operar impressoras 3D, contém saídas feitas para operar mesas aquecidas e extrusoras de filamento plástico. Essa saída fornece 12V e suporta mais de 1A de corrente, permitindo assim o controle direto da bomba de ar e dispensando o uso de um relé. As variáveis S255 e S0 definem a potência da saída em PWM sendo 255 a potência máxima da bomba, utilizada quando se deseja ligar o atuador, e 0 representando a potência mínima, utilizada quando se deseja desligar o atuador.

5) *Alimentadores*: Para que a máquina P&P consiga buscar os componentes na sua área de trabalho é necessário configurar os alimentadores, ou *feeders* em inglês. Os alimentadores podem assumir várias formas, como bobinas ou peças soltas. Nesse projeto, a fim de simplificar o processo, foi escolhido o formato de bandeja, ou *Tray*, em que os componentes ficam em posições fixas e espaçadas regularmente, como

uma matriz. Para adicionar um alimentador é preciso, na aba *Feeders*, adicionar um *ReferenceTrayFeeder* ao projeto, conforme mostrado na Figura 12. Nas suas configurações, se deve designar a posição central da primeira peça em relação à origem da área de trabalho, a quantidade de peças por fileira e coluna, a quantidade de fileiras e colunas e a distância entre elas nos eixos X e Y, como visto na Figura 13.

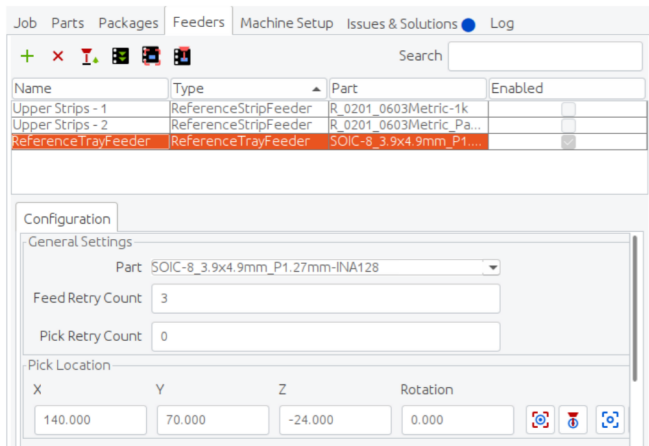


Figura 12: Criando o alimentador.

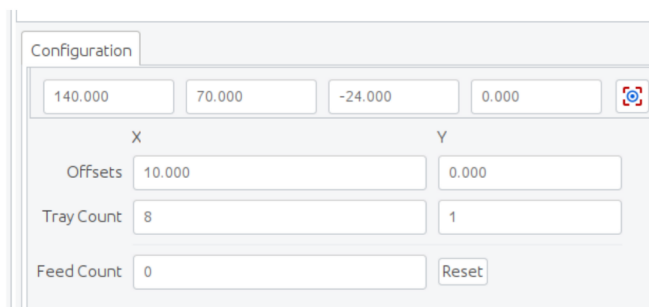


Figura 13: Configurando o alimentador.

## V. OPERAÇÃO DO OPENPNP

### A. Geração da Placa no KiCAD

Será tratado nessa subseção como gerar placas para a P&P através do programa KiCAD. A placa pode ser projetada com componentes que serão posicionados pela P&P e por componentes que não serão posicionados.

O projeto desenvolvido deve ser compatível com as dimensões máximas suportadas pela máquina e as posições dos componentes devem, a fim de garantir uma colocação precisa, se situar em coordenadas fracionadas de no mínimo 0,5mm. Deste desenho são gerados um arquivo com a extensão .pos e uma lista de componentes (*bill of materials*). Essa lista de materiais é gerada no programa Schema do KiCAD e deve ser gerada pelo script "bom\_csv\_grouped\_by\_value".

É de suma importância adicionar no desenho da placa uma origem, pois é em relação a ela que o sistema calcula a posição dos demais componentes e sem isso o sistema

não funciona adequadamente. Essa origem no KiCAD é denominada "Deslocamento de Furo e Posicionamento", que se encontra no menu Inserir no programa Pcbnew. As marcas fiduciais nas bordas da placa também são necessárias caso se deseje realizar a calibração usando a câmera.

### B. Utilização do OpenPnP

A descrição da utilização do OpenPnP é realizado para utilização com a máquina P&P elaborada nesse projeto e para o estado atual de desenvolvimento.

No OpenPnP, o projeto de colocação de componentes se chama *Job*, e cada *job* pode conter várias PCBs, que podem ser iguais ou diferentes. Os componentes que são possíveis de serem posicionados pela P&P são chamados de *Parts*, e cada componente deve ter um alimentador associado, denominado *Feeder*. O posicionamento dos componentes é realizado para aqueles que estão selecionados como *Enabled* na aba *Job*, e a P&P os posiciona levando em consideração as suas posições em relação a placa e a posição e rotação desta.

Assim, com o arquivo da placa gerado e a máquina configurada, é preciso então criar ou abrir um *job*. Isso é realizado na tela inicial do programa, na aba *Job*, conforme a Figura 14:

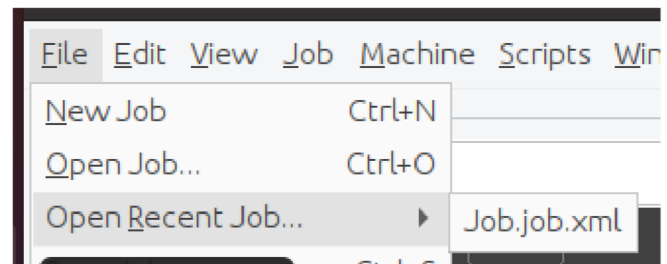


Figura 14: Criação do Job.

Com o *job* criado, deve-se inserir o arquivo da placa em que será feita a manipulação. Isso é feito na seção *Boards* da aba *Job*, adicionando uma nova placa no botão +. Após a criação da *Board*, se pode importar a placa projetada no Kicad, como mostrado na Figura 15. Neste ponto define-se a camada da placa que se está trabalhando e marca-se a opção que importa componentes ausentes no programa, conforme a Figura 16.

Na aba *Job*, na opção *Boards*, deve-se informar também a posição da placa e sua rotação, valores que são determinados em função da posição do deslocamento de furo e posicionamento inserido no KiCAD e da posição física em que a placa foi posicionada dentro do molde na área de trabalho da P&P. O próximo passo é configurar a altura dos componentes que serão posicionados. Isso é feito na aba *Parts*, escrevendo-se a altura em milímetros do componente na coluna *Height*, conforme a Figura 17.

Na aba *Packages*, é preciso atribuir um bocal para cada componente que será colocado em uma placa neste *job*. Isso é feito selecionando um componente e marcando na aba *Nozzle Tips* o bocal desejado como compatível, assim como mostrado na Figura 18.

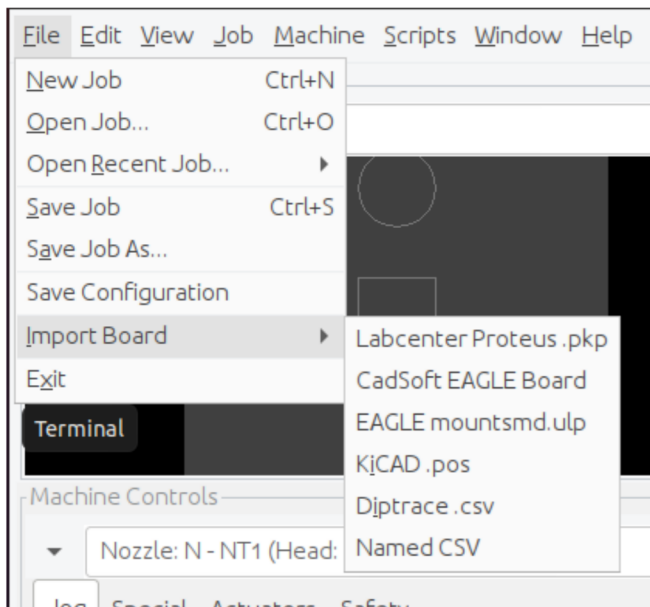


Figura 15: Importação da placa do KiCAD.

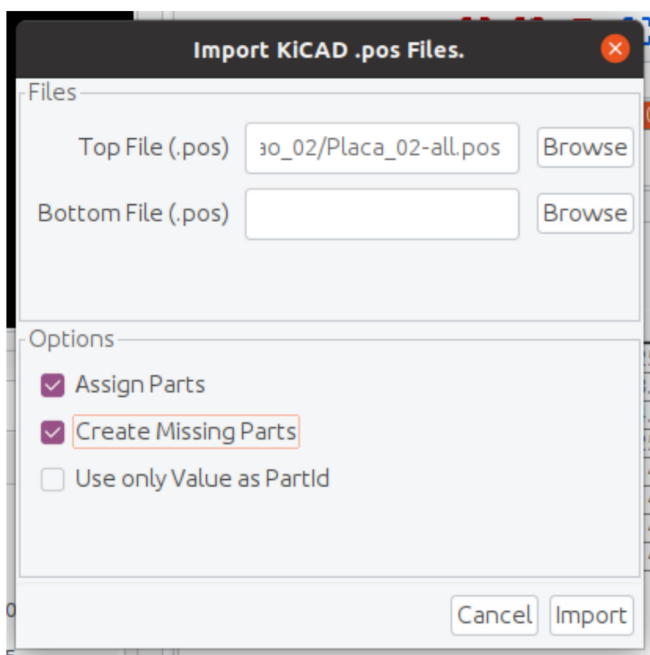


Figura 16: Inserção dos componentes ausentes.

Por fim, é preciso habilitar os componentes que serão posicionados marcando a opção *Enable* na aba *Job*. Caso sejam usadas marcas fiduciais é preciso defini-las como *Fiducials* na coluna *Type*. Com isso feito, a máquina deve estar pronta para uso, bastando conectar a máquina no botão verde da Figura 20, realizar a função *Home* no botão acima deste. Assim pode-se dar início ao processo de posicionamento dos componentes.

ID	Description	Height	
FIDUCIAL-1X2-FID...		0.000	FIDU
FIDUCIAL-HOME		0.000	FIDU
Fiducial_0.5mm_M...		0.000	Fidu
Fiducial_1mm_Ma...		0.000	Fidu
R_0201_0603Metri...		1.000	R_02
R_0201_0603Metri...		1.000	R_02
SOIC-8_3.9x4.9m...		2.000	SOIC

Figura 17: Definindo a altura dos componentes.

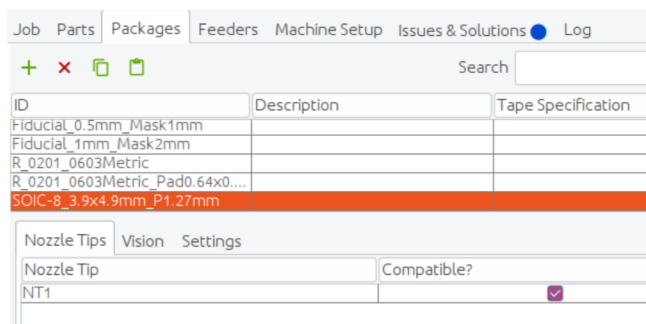


Figura 18: Atribuindo um bocal aos componentes.

## VI. RESULTADOS

Após as configurações, a integração do sistema se mostrou bem-sucedida e pôde-se realizar o controle dos 4 eixos e da bomba através do *software* OpenPnP conforme esperado.

O programa OpenPnP se mostrou funcional porém instável quando usado no sistema operacional Windows 10. Durante sua utilização, o programa deixou de reconhecer permanentemente o microcontrolador em duas ocasiões distintas em dois computadores diferentes, na primeira vez devido ao sistema entrar em modo de economia de energia e derrubar as portas USB e na segunda, sem razão aparente. Optou-se, por fim, pela execução do OpenPnP no sistema operacional Linux Ubuntu 20.10 através da utilização de uma máquina virtual, onde a operação do programa ocorreu sem maiores problemas.

O molde da área de trabalho se mostrou de grande valia, por facilitar o posicionamento adequado das placas e dos alimentadores. Ainda, com o uso do molde não é necessário realizar a medição da posição das placas e dos alimentadores toda vez que o sistema for utilizado.

Sobre o posicionamento automático de componentes, a P&P desenvolvida realizou a colocação de 2 circuitos integrados SMD SOIC-8 em um desenho de uma PCB mostrado na Figura 21. A placa mede 40mm x 30mm e os circuitos integrados utilizados medem 5,8mm x 6,2mm incluindo os pinos. As posições onde os componentes foram colocados foram apropriadas, o que permite um adequado processo de soldagem por refusão se assim for desejado.



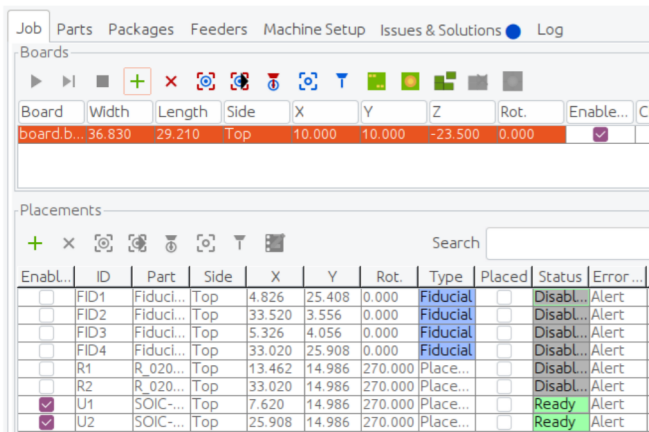


Figura 19: Habilitando os componentes desejados.

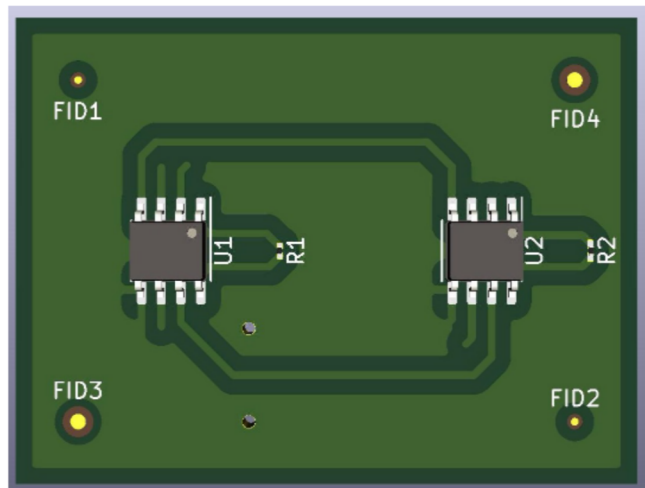


Figura 21: PCB usada nos testes do protótipo.

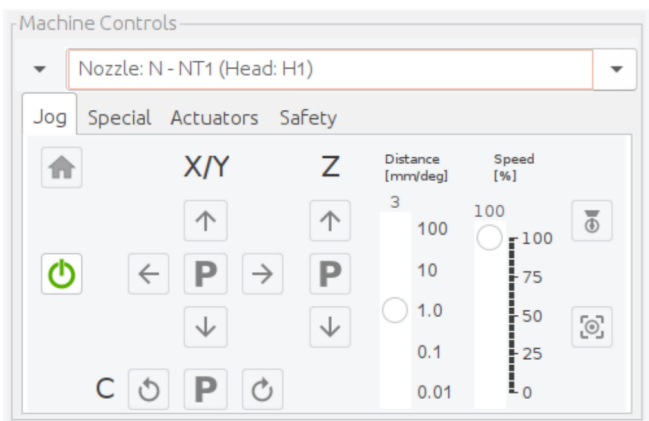


Figura 20: Conectando a máquina.

## VII. CONCLUSÃO

A integração da máquina com o OpenPnP demandou muito tempo e assim não houve tempo para realização de muitas baterias de testes, sendo testado apenas um projeto de placa com um tipo de componente. Nas 4 vezes em que o processo foi executado o protótipo se mostrou capaz de realizar todas as suas operações com sucesso, portanto pode-se afirmar que a integração do OpenPnP com o protótipo desenvolvido foi bem-sucedida. A máquina se mostrou capaz de operar com precisão e estabilidade dentro dos testes realizados.

Numa eventual continuação do projeto deve-se realizar novas baterias de testes a fim de averiguar adequadamente seu desempenho. Há também diversas melhorias que podem ser implementadas no protótipo a fim de aprimorar o funcionamento do sistema. Algumas dessas melhorias incluem a fixação da bomba, da fonte e do microcontrolador em local adequado, a integração das câmeras no sistema, a aceleração do movimento do cabeçote e a implementação de alimentadores mecanizados de bobinas.

## AGRADECIMENTOS

A Júlio César Sebben e Marcelo Schardong, pelo apoio e ajuda durante o desenvolvimento da P&P.

A Matias Rossato Muraro, pela orientação e paciência sem as quais este trabalho não seria possível.

À minha família, pelo apoio incondicional que me permitiu chegar até aqui.

## REFERÊNCIAS

- [1] J. Brown, "The History of Printed Circuit Boards – Infographic," *PCB Solutions*, pp. 1–5, 2013. [Online]. Available: <https://www.pcb-solutions.com/pcb-market-monitor/the-history-of-pcb-infographic/>
- [2] T. H. Technology, "The difference What is Through Hole," pp. 1–2.
- [3] "SMT / SMD Components & packages, sizes, dimensions, details." [Online]. Available: [https://www.electronics-notes.com/articles/electronic\\_components/surface-mount-technology-smd-smt/packages.php](https://www.electronics-notes.com/articles/electronic_components/surface-mount-technology-smd-smt/packages.php)
- [4] N. Seidle, "Reflow Skillet," pp. 1–20, 2006. [Online]. Available: <http://www.sparkfun.com/tutorials/59>
- [5] R. PRASAD, *Surface Mount Technology: Principles and Practice*, 1997. [Online]. Available: <https://books.google.com.br/books?id=cVHtBwAAQBAJ&pg=PR5&ots=roCpO5PDi1&dq=surfacemounttechnology&lr&hl=pt-BR&pg=PR5#v=onepage&q&f=false>
- [6] J. S. Huang, *Solder Paste in Electronics Packaging: Technology and Applications in Surface Mount, Hybrid Circuits, and Component Assembly*, 2012.
- [7] J. P. P. Marcicano, "Introdução ao controle numérico." [Online]. Available: <http://sites.poli.usp.br/d/pmr2202/arquivos/aulas/cnc.pdf>
- [8] M. F. MAYER, "Projeto de router CNC para corte de materiais para prototipagem," Ph.D. dissertation, IFSC - Campus Florianópolis, 2020. [Online]. Available: [https://repositorio.ifsc.edu.br/bitstream/handle/123456789/1757/TCC\\_Marconi.pdf?sequence=1](https://repositorio.ifsc.edu.br/bitstream/handle/123456789/1757/TCC_Marconi.pdf?sequence=1)
- [9] M. Ford, "Circuits Assembly Online Magazine - A History of Placement Programming and Optimization," pp. 1–13, 2016.
- [10] Yamaha Motor Co Ltda, "Ultra-High-Speed Modular Z:TA-R YSM40R Overview." [Online]. Available: <https://global.yamaha-motor.com/business/smt/mounter/ysm40r/>
- [11] OpenPnP, "OpenPnP Documentation." [Online]. Available: <https://github.com/openpnp/openpnp/wiki>
- [12] J. C. Sebben, "Projeto Integrador: MESA CNC 3 EIXOS," 2021.
- [13] R. S. A. V. A. P. R. and V. R. M., "Pick and Place Robot for Surface Mounting Devices," *International Research Journal on Advanced Science Hub*, vol. 2, no. 8, pp. 75–81, 2020.
- [14] "MarlinOnRamps4OpenPnP." [Online]. Available: <https://github.com/mgrl/MarlinOnRamps4OpenPnP>

## VIII. ANEXOS

Configuração	Arquivo	Linha	Parâmetro de Entrada
Passo dos eixos	Configuration.h	529	DEFAULT_AXIS_STEPS_PER_UNIT = {640,640,640,2.22}
Velocidade máxima dos motores	Configuration.h	536	DEFAULT_MAX_FEEDRATE = {7, 7, 7, 7}
Aceleração máxima dos motores	Configuration.h	544	DEFAULT_MAX_ACCELERATION = {1, 1, 1, 1}
Leitura contínua dos fins de curso	Configuration_adv.h	254	#define ENDSTOPS_ALWAYS_ON_DEFAULT

Tabela I: Configurações do *firmware*

Head Mountable	Setting	Parâmetro de Entrada
Default	HOME_COMMAND	G28
Default	MOVE_TO_COMMAND	G01 {X:X%f} {Y:Y%f} {Z:Z%f} {C:C%f} F{feedrate:%0f}
Default	COMMAND_CONFIRM_REGEX	ok
Default	MOVE_TO_COMPLETE_COMMAND	M400
Actuator	ACTUATE_BOOLEAN_COMMAND	{True:M42 P9 S255}{False:M42 P9 S0}

Tabela II: Configurações do GcodeDriver no OpenPnP

Vídeo 1: <https://drive.google.com/file/d/1DztHNfc6cpDp5nuZNOTRpKDAYTf6l6uI/view?usp=sharing>

Vídeo 2: [https://drive.google.com/file/d/1E4-qSuGxuoZHyOhvLZ34NI3FRUaH7e\\_9/view?usp=sharing](https://drive.google.com/file/d/1E4-qSuGxuoZHyOhvLZ34NI3FRUaH7e_9/view?usp=sharing)