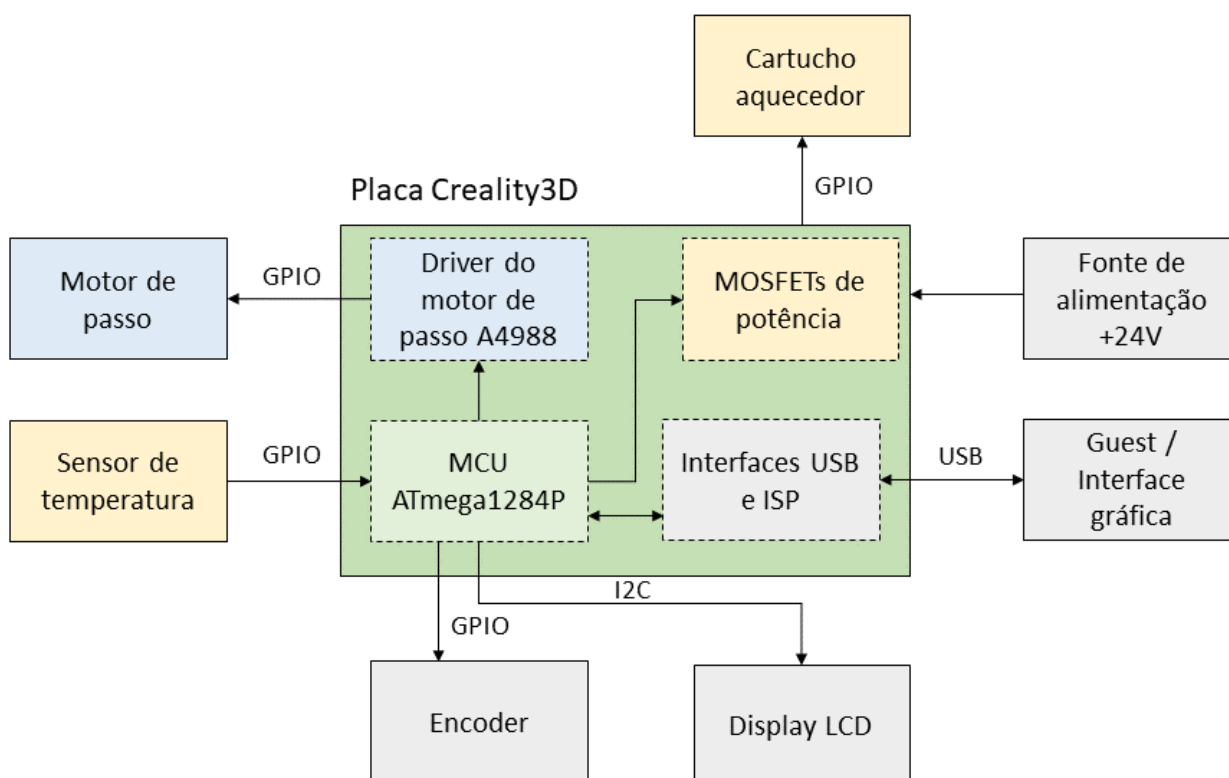


Metodologia: Introdução

* A seção de metodologia abrange a explicação detalhada das diretrizes e princípios que orientam o desenvolvimento dos sistemas do dispositivo de reciclagem de garrafas PET em filamento para impressão 3D. De modo geral, o funcionamento do dispositivo pode ser dividido em duas categorias principais: eletrônica e mecânica. A eletrônica é responsável pelo controle do sistema, enquanto a parte mecânica realiza a integração com atuadores controlados pelo primeiro grupo. O diagrama funcional da eletrônica do dispositivo pode ser visualizado na figura a seguir:



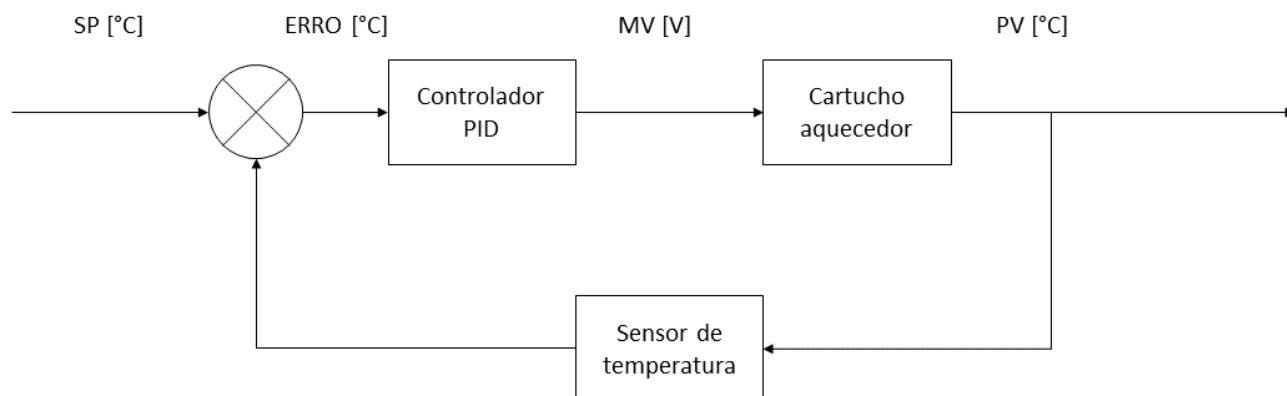
* Como esse artigo propõe reaproveitar uma placa já desenvolvida e comercializada, parte da seleção dos periféricos é limitada pela construção de hardware da placa. No que tange à parte de controle, propõe-se elaborar uma interface de comando montada no dispositivo via display LCD e encoder. Além disso, como a placa também possui interface de comunicação serial via USB, é possível desenvolver uma Interface Homem-Máquina (IHM) de comando para controle do dispositivo de maneira totalmente independente, através de um computador.

* Quanto à programação do microcontrolador, será utilizado o ambiente de desenvolvimento integrado (IDE) de Arduino. Vale ressaltar, no entanto, que o microcontrolador ATmega1284P não é um microcontrolador oficialmente suportado em placas de desenvolvimento Arduino. Para programar esse chip da maneira proposta, um pacote adicional de Arduino para microcontroladores AVR chamado MightyCore será utilizado. Com a IDE preparada, basta gravar o "bootloader" (inicializador) de Arduino no microcontrolador utilizando um gravador AVR USBasp e a própria interface de programação em sistema (ISP) da placa. Assim, é possível programar a placa diretamente através da IDE Arduino via comunicação serial USB no computador.

Metodologia: Hotend

* Como já mencionado, parte da escolha de hardware do sistema de controle de temperatura é limitada pela construção da placa, que já possui entradas preparadas com circuitos de filtro para entrada analógica de dados do sensor de temperatura e MOSFETs de potência para controle do cartucho aquecedor. Pelo fato de o cartucho aquecedor ser um elemento resistivo, o princípio de seu funcionamento é baseado na modulação da tensão aplicada em seus terminais e, consequentemente, a corrente que passa por ele. É possível obter esse resultado ao controlar o sinal de porta do MOSFET de potência via modulação por largura de pulso (PWM). Como a programação é feita pela IDE de Arduino, é válido aproveitar a biblioteca nativa da placa de desenvolvimento com algumas funções que podem ser úteis para essa aplicação. A "analogWrite()", por exemplo, modula um sinal PWM em uma determinada porta a uma frequência constante de aproximadamente 490 Hz. Ao controlar um MOSFET de potência por PWM, é necessário atentar-se à frequência do pulso gerado pelo microcontrolador. Frequências muito altas podem causar perdas de comutação, interferência eletromagnética e limitações de resposta do sistema. Frequências muito baixas, por sua vez, podem causar uma resposta lenta e controle impreciso. A frequência de 490 Hz permite uma resposta rápida do sistema de controle, garantindo que o aquecimento seja regulado com precisão, sem os riscos já mencionados de altas frequências.

* Assim como é feito no próprio controle de temperatura da impressora 3D controlada pela Creality3D, esse artigo propõe o desenvolvimento de um sistema de controle de temperatura em malha fechada com controlador PID. A figura a seguir ilustra seu princípio de funcionamento:



Onde:

SP = setpoint

MV = variável manipulada

PV = variável de processo

* Nesse sistema, um sinal de referência do tipo degrau, chamado de "set point" (SP), é inserido e comparado com o valor de temperatura atual (PV) pelo controlador. O PV é fornecido pelo sensor de temperatura. O controlador é responsável por comparar o sinal de referência com o valor de temperatura atual e calcular o erro correspondente. Com base nesse erro, o controlador calcula o sinal de saída utilizando equações e parâmetros PID pré-definidos. Esse sinal de saída é então convertido em uma forma de onda quadrada com largura de pulso modulada de acordo com o cálculo realizado pelo módulo PID. Esse sinal de saída é conhecido como variável manipulada (MV) e é expresso em tensão (V).

* Em seguida, o MOSFET de potência recebe o sinal da variável manipulada e atua no fornecimento de energia para o cartucho aquecedor. Dessa forma, por meio da modulação da tensão aplicada ao

cartucho aquecedor e, conseqüentemente, da corrente que passa por ele, o elemento resistivo varia sua temperatura. Essa mudança de temperatura é detectada pelo sensor de temperatura, que fornece um sinal elétrico de realimentação ao controlador, repetindo assim o cálculo do erro e o processo de controle sucessivamente.

* Nesse contexto, para definir a estratégia de implementação do sistema de controle PID, é importante explicar os princípios fundamentais dessa técnica. O termo P é proporcional ao erro atual (diferença entre o ponto de ajuste e o valor do processo), multiplicado por um fator de ganho K_p , resultando em uma saída de controle proporcional ao erro. No entanto, o controle proporcional sozinho pode levar a um erro estável entre o ponto de ajuste e o valor do processo. O termo I integra os erros passados ao longo do tempo, eliminando o erro residual acumulado e adicionando um efeito de controle proporcional ao valor cumulativo do erro. À medida que o erro diminui, o efeito proporcional diminui, mas é compensado pelo efeito crescente do termo integral. O termo D estima a tendência futura do erro com base na taxa de mudança atual, proporcionando um "controle antecipatório" que busca reduzir o efeito do erro por meio da influência do controle baseada na velocidade da mudança do erro. Quanto mais rápida a mudança, maior será o efeito de controle ou amortecimento.

* Após compreender os princípios básicos do controlador PID, é possível avançar para a implementação efetiva do sistema. A taxa de amostragem é um aspecto crucial nesse tipo de controle, tanto para a leitura da temperatura quanto para o cálculo do erro. Para otimizar os recursos do microcontrolador, foi decidido utilizar um único timer para determinar a taxa de amostragem e o cálculo do erro, garantindo uma taxa constante. O "Timer/Counter3", um periférico de 16 bits com recursos integrados, como controle por PWM e múltiplas fontes de interrupção, foi selecionado para lidar com o módulo de temperatura. Para controlar a taxa de amostragem e atualização dos parâmetros PID, configurou-se o temporizador para gerar interrupções a cada 0,5 segundos (2 Hz), permitindo o acesso a esses eventos por meio da rotina de serviço de interrupção (ISR) correspondente. O controle da frequência de atualização pode ser descrita como na equação a seguir:

$$f_{timer} = \frac{f_{clk_{I/O}}}{N \cdot (2^n - TCNT3)}$$

onde:

f_{timer} = frequência desejada [Hz];

$f_{clk_{I/O}}$ = frequência de clock do microcontrolador [Hz];

N = fator de pré-escala;

n = número de bits do registrador;

$TCNT3$ = pré-carga no registrador que determinará o tempo para ocorrer o evento de overflow.

Metodologia: Motor

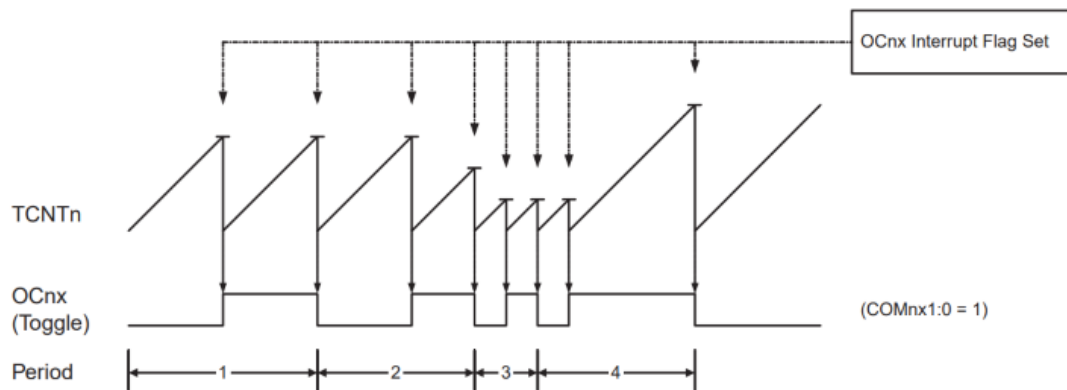
* Quanto ao controle de velocidade do motor, na placa Creality3D, há 4 drivers para controle dos motores de passo originalmente instalados na impressora 3D. Nesse artigo, apenas um motor será utilizado. O driver instalado nessa placa é o A4988, um controlador de motor de passo de baixo custo e fácil utilização, capaz de fornecer corrente e controle de direção para motores de passo bipolares. Sendo assim, a escolha do motor é limitada a um motor de passo bipolar com 4 fios. O motor escolhido foi o NEMA 17 modelo VZS1740-048-1004, fabricado pela Vurtz Motors. Este é um motor de passo bipolar

que atende as tensões de operação da placa, com um torque de 4,8 kgf.cm e corrente de operação de até 1,0 A. Além disso, possui um ângulo de passo de 1,8°/passo (±5%), ou seja, 200 passos/revolução.

* Para um simples controle de velocidade de um motor de passo através da modulação da frequência dos pulsos para o driver A4988, diversas estratégias de controle de frequência podem ser adotadas. Para esse dispositivo, foi escolhido o "Timer/Counter2" do microcontrolador, fazendo uso do recurso gerador de forma de onda (WGM) no modo de limpar o "timer" no evento de comparação de correspondência (CTC). A frequência de saída se dá através do pino OC2A do microcontrolado, associado à configuração de "timer" escolhida. Cada pulso gerado no pino da porta representa um passo dado pelo motor, comandado pelo driver controlador.

* O modo CTC utiliza o registrador OCR2A para controlar a resolução do contador. Nesse modo, o contador é reiniciado quando o valor do contador (TCNT2) é igual ao OCR2A. O valor definido no registrador OCR2A determina o limite máximo do contador, influenciando assim a sua resolução. Esse modo possibilita um maior controle sobre a frequência de saída de correspondência de comparação (evento de compare match). A figura a seguir ilustra o diagrama de tempos para o Timer no modo CTC.

Figure 15-5. CTC Mode, Timing Diagram



* Para produzir uma forma de onda no modo CTC, é possível configurar a saída OC2A para alternar seu nível lógico a cada evento de compare match, selecionando os bits do modo de saída de comparação (COM) como modo de alternância (COM2A1:0 = 1). O valor de OC2A não será visível no pino de saída do microcontrolador até que os dados sejam direcionados para a saída. A frequência máxima da forma de onda gerada será $f_{OC2A_{m\acute{a}x}} = \frac{f_{clk_{I/O}}}{2}$ quando OCR2A estiver configurado como zero (0x00) e o fator de pré-escala for igual a 0x01. A frequência da forma de onda é determinada pela seguinte equação:

$$f_{OC2A} = \frac{f_{clk_{I/O}}}{2 \cdot N \cdot (1 + OCR2A)}$$

onde:

$f_{clk_{I/O}}$ = frequência de clock do microcontrolador [Hz];

N = fator de pré-escala (1, 8, 32, 64, 128, 256 ou 1024).

* Essa frequência, no entanto, não é a frequência do sinal que controla os passos (steps) do driver do motor de passo, mas sim a frequência com que o sinal no pino de saída OC2A é alternado. Ou seja, um ciclo completo para um pulso do driver do motor de passo deve contemplar 2 ciclos de alternância de sinal do registrador. Assim, como o período e frequência são inversamente proporcionais, é possível deduzir a fórmula:

$$f_{step} = \frac{f_{OC2A}}{2} = \frac{f_{clk_{I/O}}}{4 \cdot N \cdot (1 + OCR2A)}$$

Como a placa Crealty3D utilizada possui a finalidade primária de impressão 3D, o driver A4988 é montado na placa com uma configuração de microstepping de 1:16. Microstepping é uma técnica usada para obter um movimento mais suave em motores de passo, dividindo cada etapa completa em etapas intermediárias menores. O que significa dividir cada passo completo em 16 micropassos. Em um motor de passo, um passo completo representa uma variação angular α calculada por $\alpha = \frac{360^\circ}{n}$, onde n é o número de passos necessários para uma volta completa ou um movimento de 360 graus. Portanto, para o motor utilizado neste projeto, que possui 200 passos por revolução, uma revolução com o uso de microstepping resultará em $200 \cdot 16 = 3200$ passos. A resolução original do motor era de $\alpha = \frac{360^\circ}{200_{passos}} = 1,8^\circ/\text{passo}$, mas com o microstepping, a resolução passa a ser $\alpha = \frac{360^\circ}{3200_{passos}} = 0,1125^\circ/\text{passo}$. É importante destacar que à medida que o fator de microstepping aumenta, o torque do motor diminui. No entanto, para resolver o problema de redução de torque, foram projetadas e fabricadas engrenagens impressas em 3D, que são conectadas ao motor através de um sistema de redução. Mais detalhes sobre esse sistema de redução por associação de engrenagens são fornecidos em outra seção da metodologia. Tendo definido as equações de frequência de pulsos (steps) e quantidade n de passos do motor (3200), pode-se definir a velocidade efetiva do motor. Comumente, a velocidade de motores é dada em RPM (rotações por minuto), uma unidade de frequência:

$$v_{motor} = f_{motor_{Hz}} \cdot 60 = \frac{1}{T \cdot n} \cdot 60 = \frac{1}{\frac{1}{f_{step}} \cdot n} \cdot 60 = \frac{f_{step}}{n} \cdot 60 = \frac{f_{clk_{I/O}} \cdot 60}{4 \cdot N \cdot (1 + OCR2A) \cdot n}$$

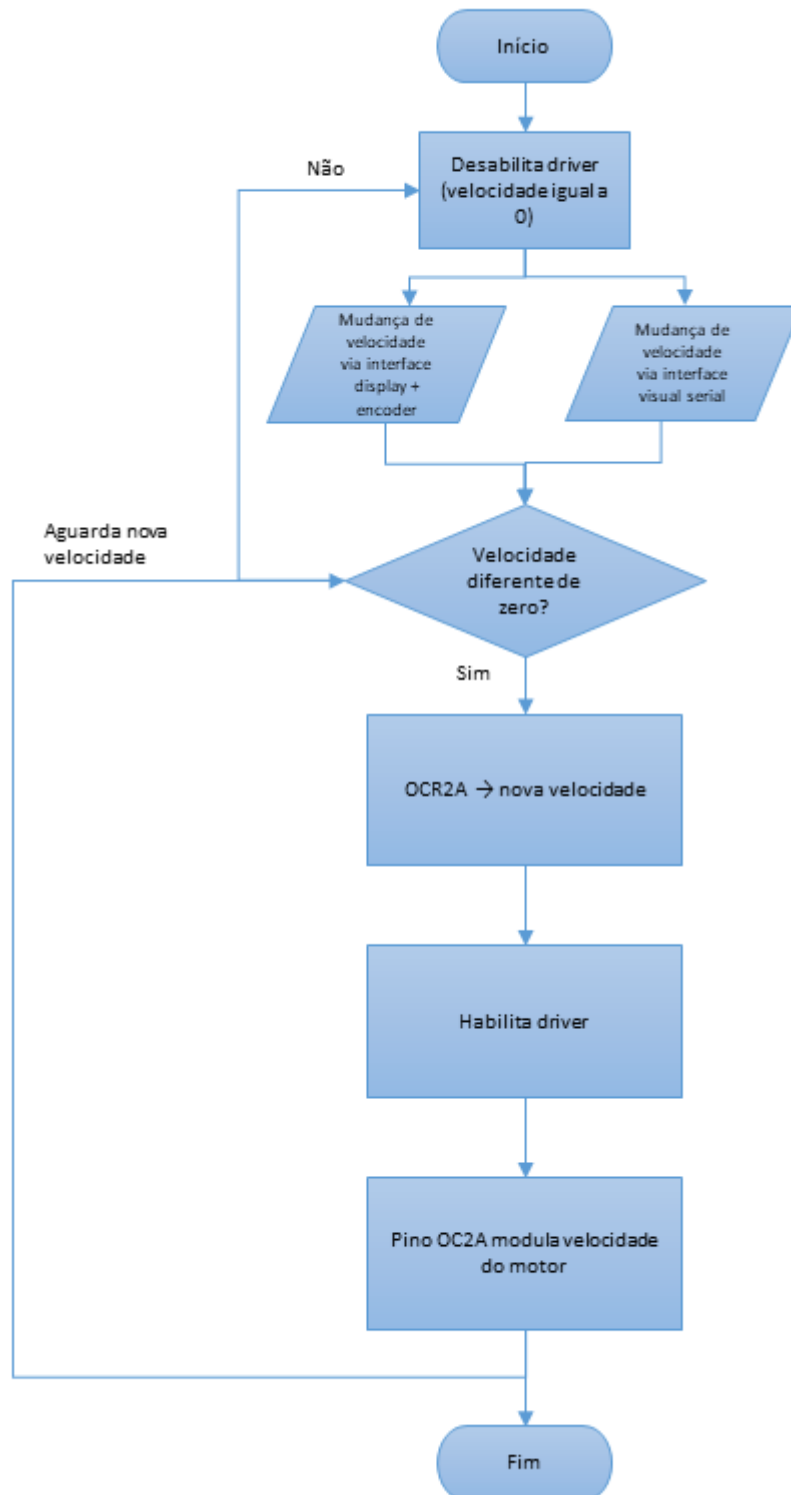
onde:

$f_{motor_{Hz}}$ = frequência do motor para uma volta com n pulsos [Hz];

T = período do pulso de um passo [s];

60 = fator de conversão para $f_{motor_{Hz}}$ de Hz para RPM.

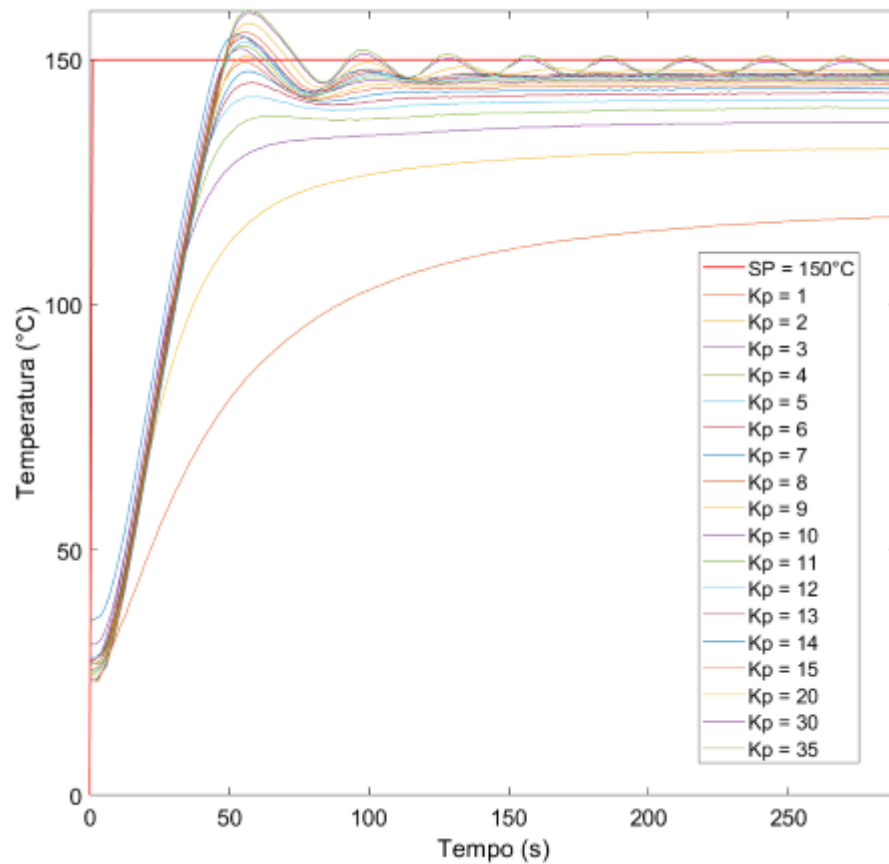
* Com base nos princípios de controle de velocidade do motor de passo para esse dispositivo, é possível implementar esse controle por meio da configuração do timer selecionado e da manipulação do registrador OCR2A, responsável por modular a saída pelo pino OC2A, de acordo com a configuração escolhida. Para esse projeto específico, foi constatado que um fator de pré-escala de 64 abrange uma faixa de velocidades adequada para a operação do dispositivo. Além disso, é essencial desenvolver uma interface que permita selecionar a velocidade desejada, definir o sentido de rotação, bem como o momento de iniciar e interromper o processo de tracionamento. O controle do motor de passo pode ser resumido com o diagrama a seguir:



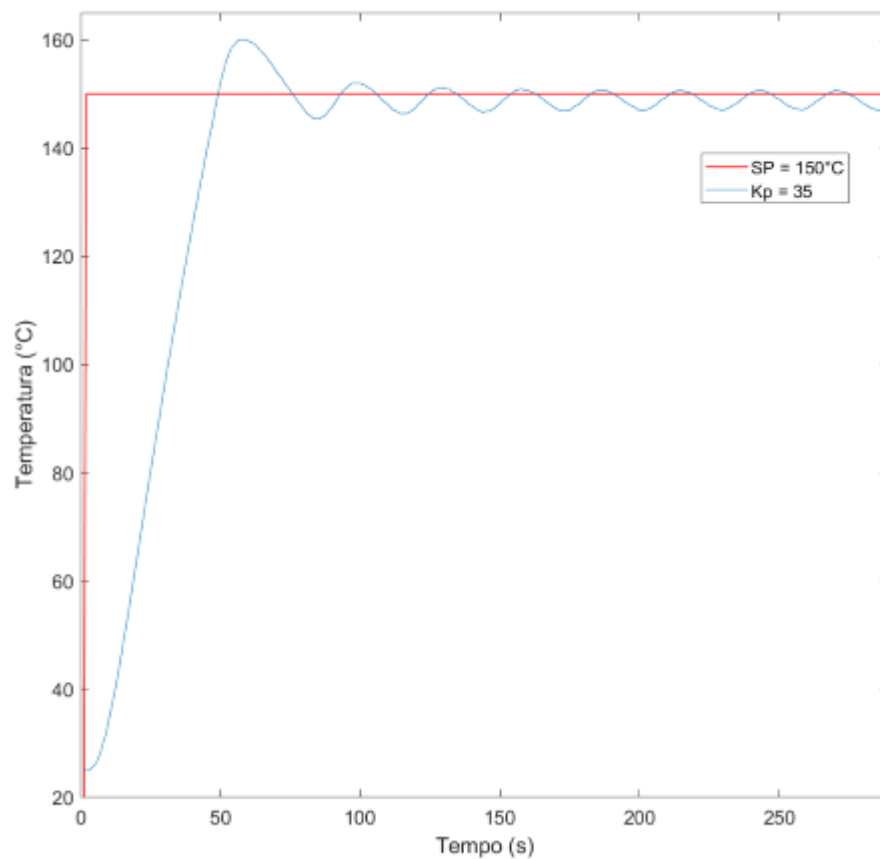
Resultados: Hotend

* Tendo definido e implementado o controlador PID em programação de texto estruturado C/C++ via IDE Arduino e gravado o programa no microcontrolador, foi possível dar início aos testes de controle de temperatura. Alguns firmwares de impressora 3D de código aberto utilizam uma temperatura de referência média para calibrar os parâmetros proporcional (K_p), integral (K_i) e derivativo (K_d), como é o caso do Marlin, um driver para impressoras RepRap de código aberto. Como o elemento de aquecimento escolhido para o controle de temperatura é um produto já comercializado em impressoras 3D, optou-se por sintonizar os parâmetros PID com a mesma temperatura de referência utilizada no

firmware Marlin: 150°C. Na figura a seguir, é possível observar os resultados obtidos de resposta ao degrau de entrada com diferentes valores de ganho proporcionais, ainda sem termos integral e derivativo:



No teste do gráfico anterior, foi possível observar a resposta ao degrau com ganho proporcional variando de 1 até 35. Nota-se que, para o setpoint (SP) escolhido, ganhos proporcionais $< K_p = 9$ o sistema sequer atinge a temperatura desejada. Somente a partir de $K_p = 9$ a temperatura desejada é atingida, ainda que somente em seu valor de pico, estabilizando abaixo do SP. Além disso, nota-se também a presença de uma curva característica para controle de sistemas: o ganho proporcional crítico, que inicia uma oscilação constante em um sistema de controle. Esse comportamento é observado quando $K_p = 35$, conforme a figura a seguir:



Esse ganho, de acordo com o método do período crítico de Ziegler-Nichols, pode ser usado para sintonizar os parâmetros proporcional, integral e derivativo em controladores PID. O método se resume em duas etapas principais:

- Com termos integral (K_i) e derivativo (K_d) do controlador zerados, aumentar o ganho proporcional K_p até que a resposta do sistema oscile periodicamente. Esse ganho é denominado ganho proporcional crítico (K_u);
- Medir o período da oscilação ao encontrar K_u . O período da oscilação é comumente representado por P_u .

Tendo determinado os termos K_u e P_u , pode-se calcular os parâmetros PID com as seguintes equações:

$$K_p = 0,6 \cdot K_u$$

$$K_i = \frac{2}{P_u}$$

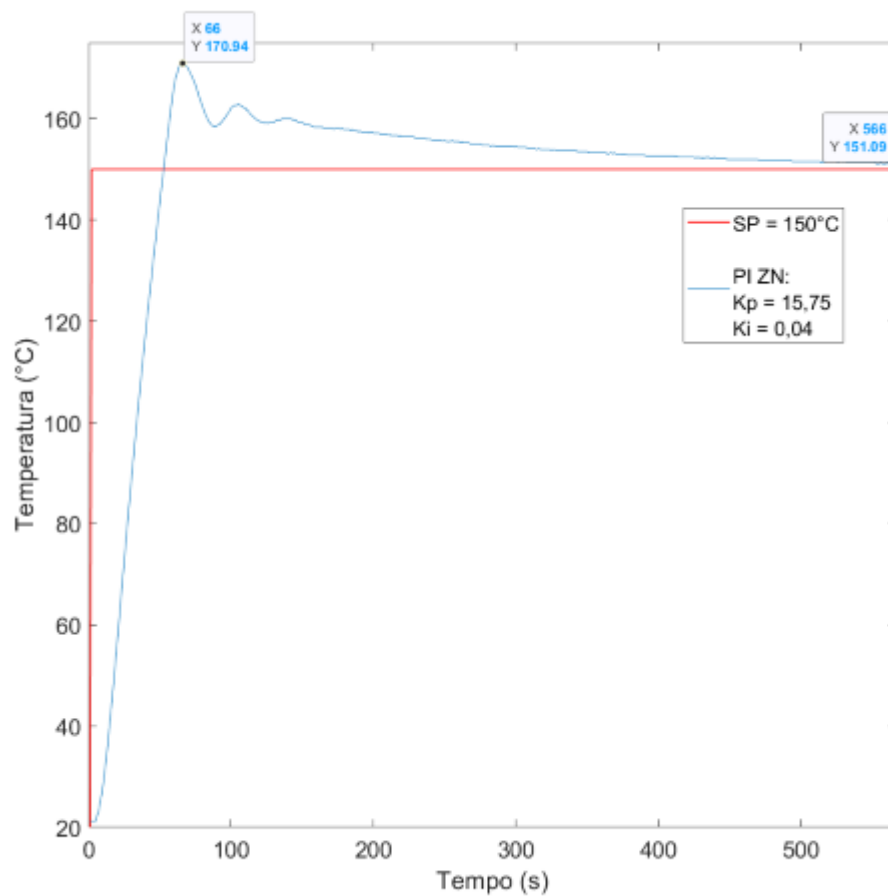
$$K_d = \frac{P_u}{8}$$

Além da sintonia de um controlador PID, o método de Ziegler-Nichols também contempla outros controladores, como PI e PD. As seguintes equações valem para um controlador PI:

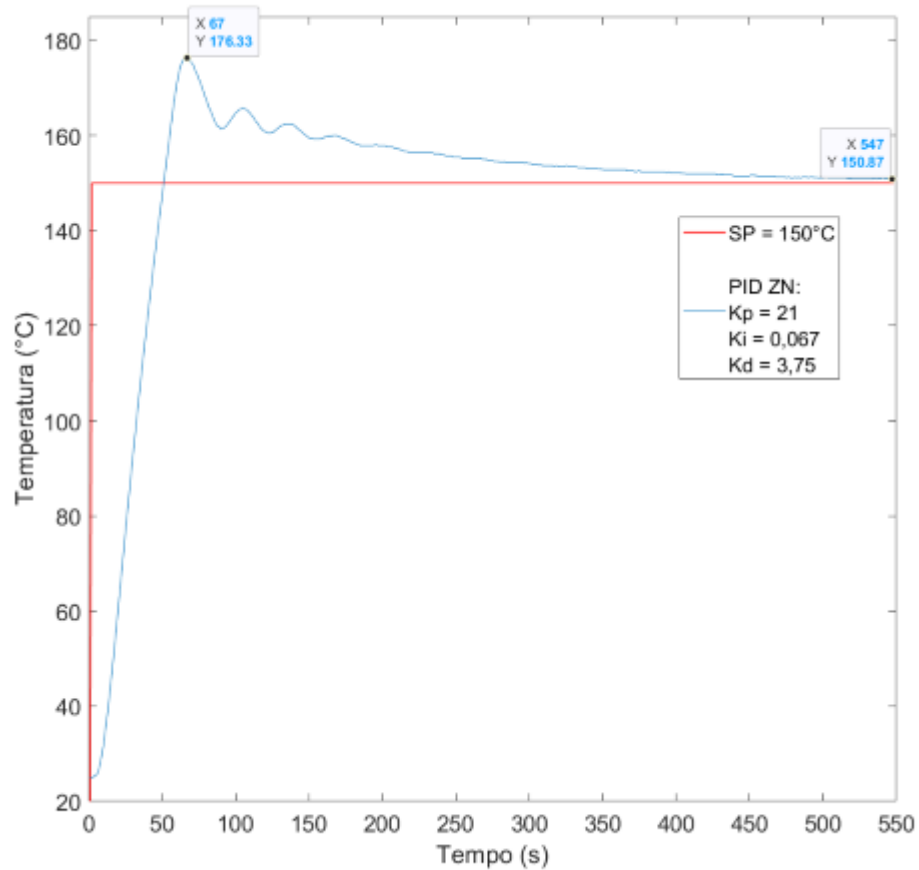
$$K_p = 0,45 \cdot K_u$$

$$K_i = \frac{1,2}{P_u}$$

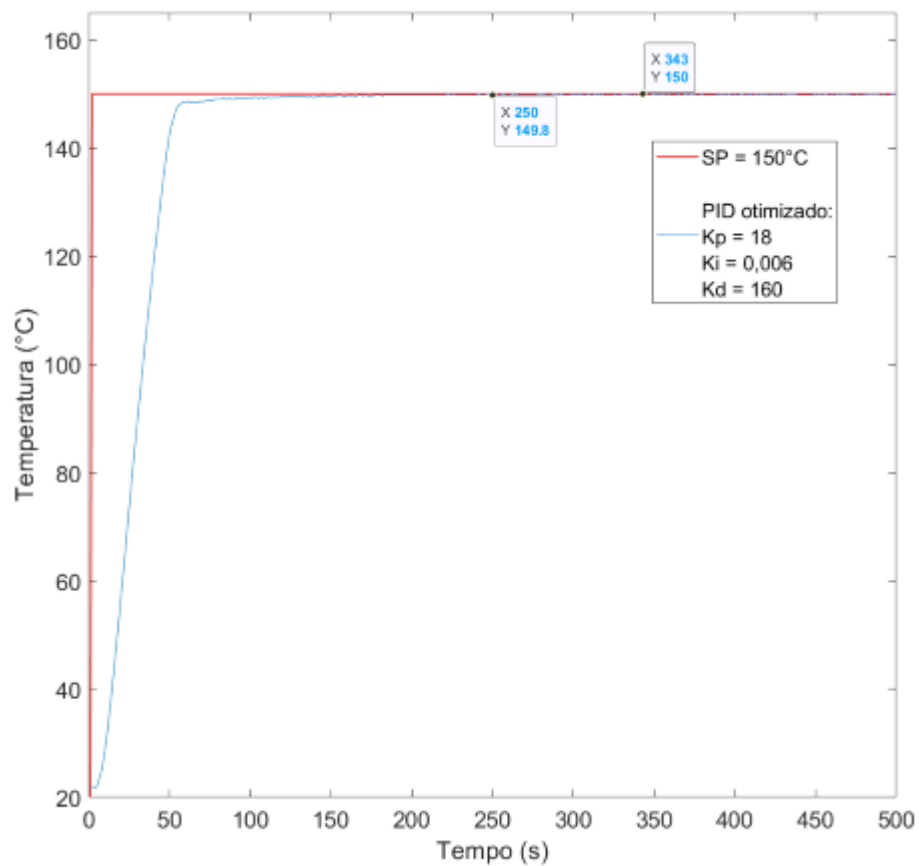
Sabendo que o ganho proporcional crítico gerou um período de oscilação de 30s, foram calculados os parâmetros para um controlador PI, conforme o método de Ziegler-Nichols. A figura a seguir ilustra a resposta do sistema:



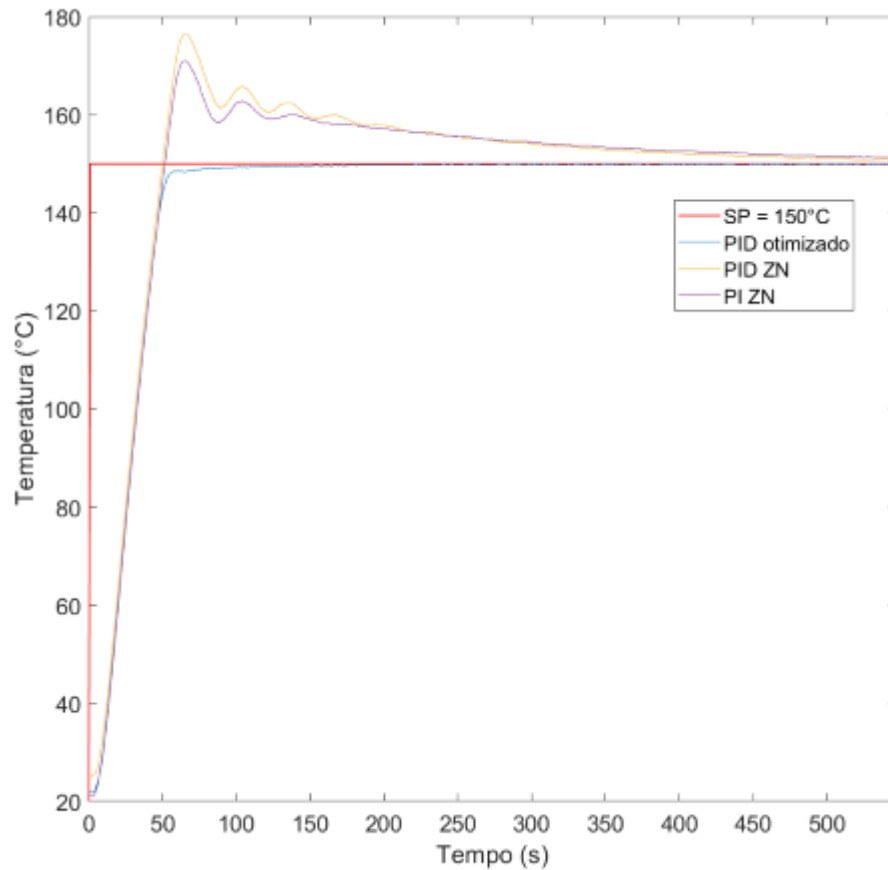
Observa-se que o sistema tende a estabilizar próximo de 151,09°C, o que resulta em um erro de estado estacionário (e_{ss}) = 1,09°C, cerca de 0,73% do SP. Esse valor, no entanto, está longe de ser ótimo, pois ele só é estabiliza cerca de 9 minutos depois do início do processo. Sabe-se que impressoras 3D atingem temperaturas até mais altas para fusão do material e estabilizam a temperatura em cerca de 2 minutos. Ademais, nota-se um sobressinal considerável no início do aquecimento: o sistema atinge um valor de pico de 170,94°C, 13,96% do SP. Esse sobressinal, apesar de não trazer grandes riscos para a situação de sintonia onde o SP é de 150°C, pode apresentar risco de danificar o sistema ou até de incêndio em situações normais de trabalho. Espera-se realizar o trabalho de reciclagem do filete de garrafa PET em filamento numa faixa de trabalho de 220 a 230°C. Se aplicarmos um sobressinal de 13,96% a um SP de 230°C, pode-se atingir um pico de 262,11°C, teoricamente. Essa temperatura, no entanto, já está fora da temperatura de trabalho permitida em impressoras como a Creality Ender 3, por exemplo. Uma alternativa para tratar do problema de sobressinal seria diminuir o ganho K_i . Outra alternativa, no entanto, é a inserção do termo derivativo K_d no sistema, que possui a capacidade de reduzir o sobressinal, além de reduzir o tempo de resposta do sistema. A figura a seguir ilustra a resposta da implementação de um controlador PID, de acordo com o método de Ziegler-Nichols:



Ao recalculer os parâmetros PID de acordo com as equações de Ziegler-Nichols, foi obtido um K_i maior. Assim, apesar da presença do termo derivativo K_d no sistema, ainda temos a presença de um sobressinal significativo no sistema de 17,55%. Outro ponto a se observar é que o sistema estabilizou ligeiramente mais rápido por volta de 150,87°C, representando um e_{ss} de 0,87°C, 0,58% do SP. Ainda assim, os resultados obtidos não são ideais para a aplicação proposta. O grupo propôs-se, então, a realizar a sintonia pelo método de tentativa e erro, aumentando gradativamente somente o ganho proporcional K_p e posteriormente os ganhos integral e derivativo K_i e K_d , respectivamente, de modo a otimizar o sobressinal, o erro estacionário e o o tempo de acomodação do sistema. A figura a seguir ilustra os parâmetros encontrados e a resposta do sistema aos novos parâmetros PID:

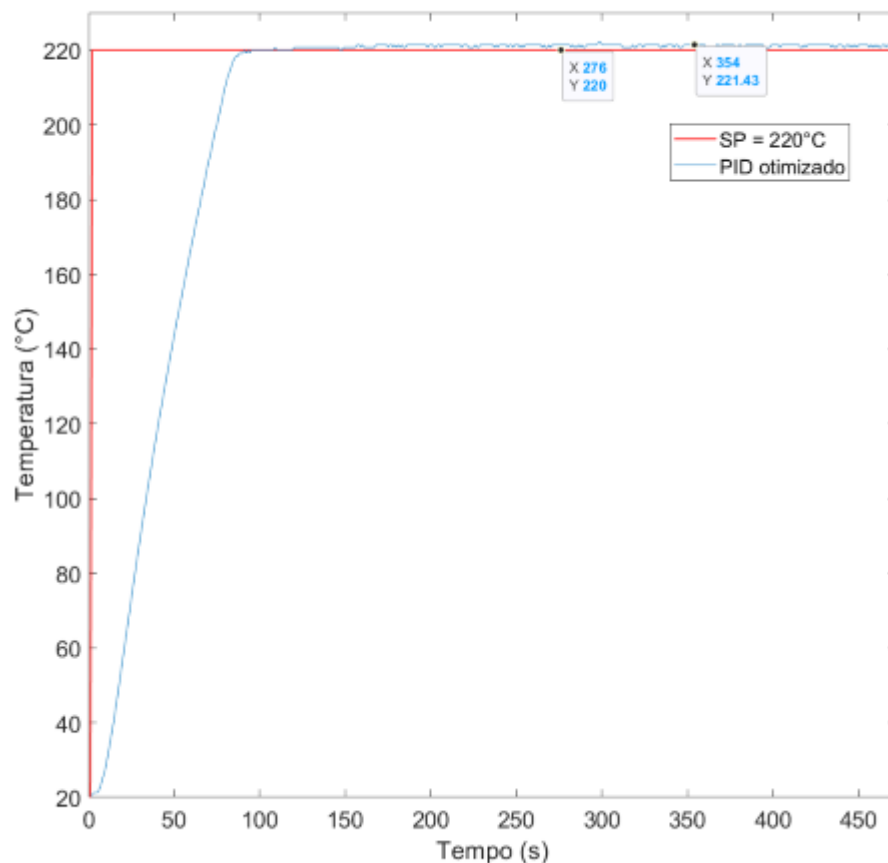


Com os parâmetros encontrados pelo método de tentativa erro, foi obtido um e_{ss} de 0,2°C, 0,13% do SP. Além disso, observa-se também a ausência de sobressinal e um rápido tempo de acomodação, em cerca de 83 segundos. O sistema possui um perfil bem conservador com um ganho derivativo consideravelmente alto, de modo a evitar o sobressinal. Além disso, o ganho derivativo mais alto atribui maior sensibilidade às variações de temperatura do sistema. Apesar dessa característica não ser muito desejável em sistemas de controle de temperatura em função do tempo de resposta mais lento em comparação com outros sistemas que integram controladores PID, pode-se verificar que o efeito produzido é satisfatório. Abaixo temos uma comparação da resposta do sistema com controladores PI (Ziegler-Nichols), PID (Ziegler-Nichols) e PID otimizado:



Apesar do método de Ziegler-Nichols ter uma taxa de eficiência relativamente alta para sintonia de controladores PID, em alguns casos - como nesse sistema - pode não ser o método mais adequado. Uma explicação provável para a baixa eficiência do método de Ziegler-Nichols nesse caso é a presença de um tempo morto de cerca de 5 segundos no sistema. Tempo morto é o atraso ou a falta de resposta do sistema a uma mudança no estímulo de entrada (degrau de 150°C). O método do período crítico de Ziegler-Nichols não contempla a presença do tempo morto e isso pode influenciar na resposta do sistema. Outros métodos para tratar desse problema: Cohen-Coon: Kappa-Tau, continua [nesse artigo](#).

Como o método de tentativa e erro apresentou um comportamento satisfatório, optou-se por permanecer com ele. A fim de validar a eficiência do controle de temperatura para situações de trabalho, foi adotado um SP de 220°C, conforme demonstrado por TYLMAN e DZIERZEK (2020). A figura a seguir demonstra a resposta do sistema com PID otimizado por tentativa e erro e degrau de entrada de 220°C:



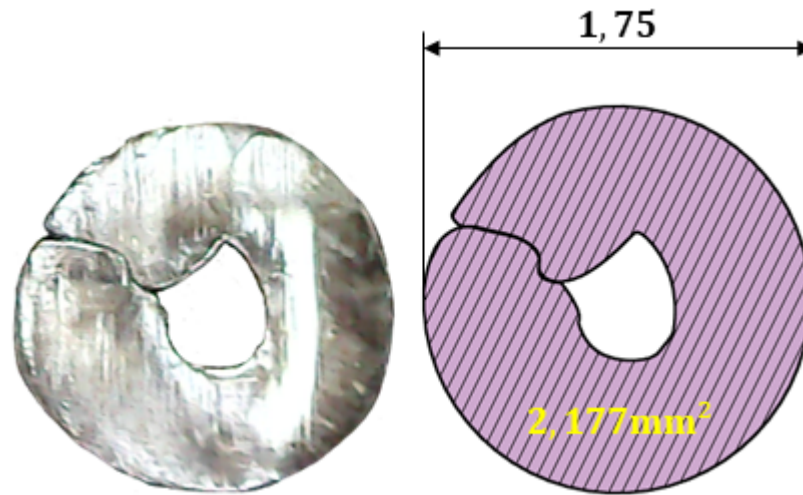
Em temperaturas de trabalho ideais, observou-se um sistema relativamente estável com mais ruído na resposta - possivelmente em função da alta sensibilidade gerada pelo ganho derivativo mais alto. Esse ruído, no entanto, não chega a ser significativo, haja visto que o erro não chega a ser maior do que 1,43°C, 0,65% do SP. Ademais, a fim de verificar a estabilidade do sistema a perturbações externas (como a trefilação/extrusão do filete de garrafa PET, por exemplo), foi feito um teste no qual foi passado um filete de garrafa PET pela matriz aquecida. Nesse teste, não foram observadas grandes variações de temperatura além do que fora registrado na figura anterior.

No que tange ao teste do filamento produzido, foi obtido um filamento com superfície uniforme e diâmetro de $(1,75 \pm 0,025)\text{mm}$. A figura a seguir apresenta uma amostra produzida.



Ainda sobre o filamento produzido, uma análise em corte de uma seção transversal evidencia que o material não preenche a área por completo. Com auxílio da ferramenta de CAD Fusion 360, da

Autodesk, é possível estimar a área coberta por material:



A secção transversal do filamento produzido ocupa uma área de aproximadamente $2,177\text{mm}^2$, cerca de 90,51% da área total para um filamento sólido com diâmetro de 1,75mm.

Resultados: Motor

* Para fins de teste e validação, foi adotado um valor arbitrário de $OCR2A = 16$. Adotando um clock de CPU de 16 MHz e fator de pré-escala de 64, podemos calcular a velocidade do motor:

$$v_{motor} = \frac{16000000 \cdot 60}{4 \cdot 64 \cdot (1 + 16) \cdot 3200} = 68,93RPM$$

* Em laboratório, a fim de testar a fidelidade da velocidade calculada e a velocidade real do motor, foi utilizado um tacômetro Minipa MDT-2245C (precisão de $\pm 0,05\%$) e a velocidade medida para essa mesma configuração foi de 69 RPM. Portanto, pode-se concluir que foi estabelecido um controle de velocidade satisfatório para o sistema, ainda que essa não seja a variável de processo principal a ser controlada.