

UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

ANDRÉ LUÍS MACIEL PINTO

AUTOMAÇÃO DO PROCESSO DE *LOAD* E *UNLOAD* DE PCB'S UTILIZANDO
BRAÇO ROBÓTICO E PROCESSAMENTO DIGITAL DE IMAGEM

Manaus
2024

ANDRÉ LUÍS MACIEL PINTO

AUTOMAÇÃO DO PROCESSO DE *LOAD* E *UNLOAD* DE PCB'S UTILIZANDO
BRAÇO ROBÓTICO E PROCESSAMENTO DIGITAL DE IMAGEM

Trabalho Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade do Estado do Amazonas como requisito para a obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Almir Kimura Jr.

Manaus

2024

ANDRÉ LUÍS MACIEL PINTO

AUTOMAÇÃO DO PROCESSO DE LOAD E UNLOAD DE PCB'S UTILIZANDO
BRAÇO ROBÓTICO E PROCESSAMENTO DIGITAL DE IMAGEM

Trabalho de Conclusão de Curso (TCC) apresentado à Escola Superior de
Tecnologia da Universidade do Estado do Amazonas como parte dos requisitos
necessários para obtenção do título de Bacharel em Engenharia de Controle e
Automação.

Manaus, 19 de julho de 2024

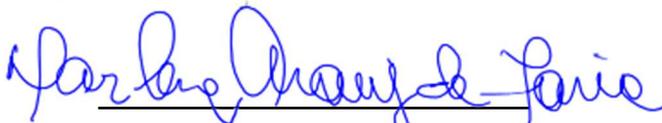
Banca Examinadora:



Prof. Almir Kimura Junior

Orientador(a)

Instituição: Universidade do Estado do Amazonas



Profa. Marlene Araújo de Faria

Avaliador

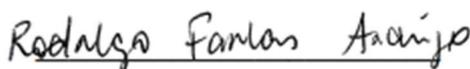
Instituição: Universidade do Estado do Amazonas

Documento assinado digitalmente
gov.br CLETO CAVALCANTE DE SOUZA LEAL
Data: 31/07/2024 17:08:15-0300
Verifique em <https://validar.it.gov.br>

Prof. Cleto Cavalcante De Souza Leal

Avaliador

Instituição: Universidade do Estado do Amazonas



Prof. Rodrigo Farias Araújo

Avaliador

Instituição: Universidade do Estado do Amazonas

RESUMO

A automação tem gerado mudanças significativas nas indústrias nos tempos atuais. Impulsionada por avanços tecnológicos e pela busca por eficiência, o desenvolvimento de automações cria impactos positivos em diversos aspectos. Otimizando processos e impulsionando produtividade é possível observar uma redução das tarefas manuais repetitivas beneficiando assim indústria e trabalhadores. Com planejamento adequado, a automação contribui para um futuro promissor.

Através de recursos oriundos da Lei de Informática n.º 8.387, esta automação teve como objetivo substituir um processo manual de uma empresa multinacional que presta serviços no setor de manufatura de produtos eletrônicos para uma empresa de telefonia. Neste processo placas de circuito impresso dos celulares são colocadas em equipamentos que fazem diversos testes de recursos de comunicação destas placas. O processo era feito manualmente com cada operador recebendo a placa montada do processo anterior e levando a placa até o equipamento que realiza o teste, este processo é repetido várias vezes. Em caso de falha dos testes, o operador repete o processo em um equipamento diferente, caso a falha seja recorrente a placa é segregada e analisada por técnicos. Tendo este processo como base foi elaborada uma automação funcional que tem como objetivos: identificar a posição da placa em uma esteira de entrada utilizando um sistema de visão computacional, levar esta placa corretamente utilizando uma garra robótica até o equipamento de teste também indexado utilizando visão computacional, retirar esta placa do equipamento de teste e levar esta placa até um palete que também indexado utilizando visão para que ela esteja pronta para o processo seguinte. Cada módulo automatizado possui 16 caixas de teste que em um processo manual é manipulado por um operador. É previsto que o processo em funcionamento pleno aumente em até 20% o tempo médio entre reparos (MTTR) das caixas de teste e reduza em até 5% a quantidade de placas danificadas. Foi também analisado que cada módulo trará um retorno de investimento (ROI) após 2,5 anos.

Palavras-Chave: Automação; Sistema de Visão Computacional; Polo Industrial; Produção; Pesquisa e Desenvolvimento; P&D.

ABSTRACT

Automation has generated significant changes in industries nowadays. Driven by technological advances and the search for efficiency, the development of automation creates positive impacts in many aspects. By optimizing processes and boosting productivity, it is possible to observe a reduction in repetitive manual tasks, thus benefiting industry and workers. With proper planning, automation contributes to a promising future.

Using resources from Information Technology Law No. 8,387, this automation aimed to replace a manual process of a multinational company that provides services in the electronics manufacturing sector for a telephone company. In this process, cell phone printed circuit boards are placed in equipment that conducts various tests of the communication capabilities of these boards. The process was done manually with each operator receiving the assembled plate from the previous process and taking the plate to the equipment that performs the test, this process is repeated several times. In case of test failure, the operator repeats the process on a different piece of equipment; if the failure is recurring, the board is segregated and analyzed by technicians. Using this process as a basis, a functional automation was developed with the following objectives: identifying the position of the plate on an input conveyor using a computer vision system, taking this plate correctly using a robotic gripper to the test equipment, also indexed using computer vision, remove this plate from the test equipment and take this plate to a pallet that is also indexed using vision so that it is ready for the next process. Each automated module has sixteen test boxes that in a manual process are manipulated by an operator. The process in full operation is expected to increase the mean time between repairs (MTTR) of test boxes by up to 20% and reduce the number of damaged boards by up to 5%. It was also analyzed that each module will bring a return on investment (ROI) after 2.5 years.

Keywords: Automation; Computer Vision System; Industrial Hub; Production; Research and Development; R&D

LISTA DE FIGURAS

Figura 1 - Placas de circuito impresso de celulares	12
Figura 2 - Caixa de teste funcional de placas de circuito	13
Figura 3 - Vários tipos de sensores.....	17
Figura 4 - Vários tipos de atuadores	18
Figura 5 - Manipulador robótico 6 eixos	18
Figura 6 - Robô sendo movimentado manualmente.....	20
Figura 7 – Camadas do protocolo TCP/IP.....	22
Figura 8 – Exemplo de arquitetura de comunicação Modbus TCP/IP	25
Figura 9 - CLP MODICON	26
Figura 10 - Hierarquia de tarefas de processamento de imagens	27
Figura 11 - Exemplo de segmentação de imagem	29
Figura 12 – Fluxograma simplificado de desenvolvimento do projeto	30
Figura 13 - Fixtures para testes de PCBs	32
Figura 14 – Layout proposto para o módulo principal da solução	34
Figura 15 – Disposição dos sensores nas esteiras	35
Figura 16 – Layout proposto para o módulo de saída da solução	35
Figura 17 - Modelos de robô TM Omron	36
Figura 18 – Layout proposto para o módulo de saída da solução	37
Figura 19 – Estação montada para testes.....	38
Figura 20 – Berço de placas utilizado para testes.....	38
Figura 21 – Ferramental do robô com 3 atuadores e ventosas	39
Figura 22 – Vista inferior o ferramental do robô	40
Figura 23 – Módulo de saída	40
Figura 24 – Esteiras do módulo principal	41
Figura 25 – Quadros elétrico de drives e Quadro elétrico de rede e segurança	42
Figura 26 - Quadro elétrico de fontes e disjuntores.....	43
Figura 27 – Racks de Teste	43
Figura 28 – Representação da estrutura de proteção	44
Figura 29 – Berço impresso da placa.....	45
Figura 30 – Placa iluminada pela iluminação do robô	46
Figura 31 – a) Imagem com a referência buscada bem destacada pela iluminação. b) Parte do contorno da referência selecionada não pode ser mais visto devido a modificação da posição da iluminação.....	46
Figura 32 – a) Pode-se observar o contorno da referência selecionada. b) Parte do contorno da referência selecionada não pode ser mais visto devido a modificação da posição o que gerou sombra na placa.	47
Figura 33 – Desenho da peça para suporte da iluminação na garra do robô	48
Figura 34 – Iluminação utilizada no teste e peça impressa para suporte da iluminação na garra do robô	49
Figura 35 – Sistema de iluminação	50
Figura 36 – visão da placa a uma distância de 150mm da câmera.	50
Figura 37 – Componente referenciado na placa utilizando iluminação difusa e processamento da câmera do robô TM-12	51
Figura 38 – Sistema não encontrando a referência.....	51

Figura 39 – Anel de Iluminação multiangular branca	52
Figura 40 – Garra do robô com o anel de iluminação.....	53
Figura 41 - Foto da placa antes de ser processada	53
Figura 42 - Foto da placa após extração do plano de cor cinza	54
Figura 43 - Foto da placa após aprimoramento do contraste	54
Figura 44 - Foto da placa após binarização	55
Figura 45 - Foto do berço antes do processamento.....	56
Figura 46 - Foto do berço após o processo extração do plano de cor verde	56
Figura 47 - Foto do berço após o processo de aprimoramento de contraste.....	57
Figura 48 - Foto do berço após o processo de binarização.....	57
Figura 49 – Placa no berço antes do processamento do sistema de visão	58
Figura 50 – Foto da placa no berço após o processamento do sistema de visão.....	58
Figura 51 – Foto do berço antes do processamento do sistema de visão	59
Figura 52 – Foto do berço após o processamento do sistema de visão	59
Figura 53 - Foto do palete antes de ser processada	60
Figura 54 - Foto do palete após aprimoramento do contraste	61
Figura 55 - Foto do palete após binarização	61
Figura 56 – Referência encontrada no palete	62
Figura 57 – Placa na cavidade do palete	62
Figura 58 – Processamento de imagem referenciando a placa.....	63
Figura 59 – Exemplo de programação no software TMFlow	63
Figura 60 – Exemplo de programação em ladder no programa FPWin Pro 7	67
Figura 61 – Início do programa do robô	70
Figura 62 - Teste com a primeira calibração	73
Figura 63 - Testes após segunda calibração	74
Figura 64 - Teste com a primeira calibração	76
Figura 65 - Testes após segunda calibração	76
Figura 66 – Teste de encaixe da placa no palete.....	77
Figura 67 – Comparação de desempenho da produção máquina x produção manual	78

LISTA DE TABELAS

Tabela 1 - Quantidade de placas produzidas e falhas durante as semanas de validação	74
Tabela 2 - 5 erros com maior recorrência	75

SUMARIO

1.	INTRODUÇÃO	11
1.1	Definição do problema	11
1.2	Hipótese	12
1.3	Justificativa	12
1.4	Objetivo Geral	14
1.5	Objetivos Específicos	14
2.	REFERENCIAL TEÓRICO	16
2.1	Automação Industrial	16
2.2	Sensores e Atuadores	16
2.3	Robótica Industrial	18
2.4	Programação de Robôs	19
2.4.1	Programação on-line	19
2.4.2	Programação <i>off-line</i>	21
2.5	Protocolos de comunicação	21
2.5.1	Protocolo TCP	22
2.5.2	<i>Modbus</i> TCP	24
2.6	CLP	26
2.7	Processamento digital de imagens	27
2.7.1	Segmentação de imagens	28
3.	METODOLOGIA	30
4.	DESENVOLVIMENTO	32
4.1	Análise de requisitos	32
4.2	Pré-Montagem da automação	37
4.3	Montagem e integração	40
4.4	Programação e Ajustes de automação	45
4.4.1	Sistema de visão	45
4.4.1.1	Testes do sistema de visão	45
4.4.1.2	Processamento para referenciamento da placa	53

4.4.1.3	Processamento para referenciamento do berço de teste	55
4.4.1.4	Processamento de calibração cruzada	57
4.4.1.5	Processamento para referenciamento do palete	60
4.4.2	Programação do robô	63
4.4.3	Programação do CLP	67
4.4.4	Comunicação com o sistema supervisorio	68
4.4.4.1	Comunicação com o robô	68
4.4.4.2	Comunicação com o CLP	71
5.	RESULTADOS E DISCUSSÕES	73
6.	CONCLUSÃO	78
6.1	Trabalhos futuros	79
7.	REFERENCIAL BIBLIOGRÁFICO	81

1. Introdução

Quando uma tarefa é relativamente simples, repetitiva ou perigosa para um humano, então uma escolha apropriada pode ser implementar uma automação desta tarefa utilizando um robô. Os robôs estão aumentando em inteligência, com a adição de sentidos como a visão é possível a realização de tarefas mais complexas e de maneira mais precisa. (RIBEIRO, 1997). Algumas razões que levam um processo manual a ser substituído por uma automação utilizando um robô são aumento da produtividade, redução dos custos do trabalho, redução ou eliminação das atividades manuais repetitivas, aumento da segurança do trabalhador e melhoria na qualidade do produto. (ROGGIA e CARDOZO FUENTES, 2016).

Para se manterem mais competitivas e eficientes, as empresas do polo industrial de Manaus tem focado em melhorar e aprimorar seus processos. Com o incentivo da Lei nº 8.387 de 30 de dezembro de 1991, regulamentada pelo Decreto nº 6.008 de 29 de dezembro de 2006, denominada de Lei de Informática essas empresas têm investido em atividades de pesquisa, desenvolvimento e inovação. (NOGUEIRA DO NASCIMENTO, 2021). As empresas investem esses recursos em institutos de ciência e tecnologia que então desenvolvem novos processos ou protótipos para atingir esses objetivos. Uma das principais abordagens para atingir esse objetivo é a automação de processos que são atualmente feitos manualmente.

1.1 Definição do problema

O processo de *load* e *unload* de placas de celulares em caixas de teste de radiofrequência é feita manualmente por operadores. O manuseio dessas placas e caixas de teste por esses operadores nesta etapa do processo pode causar um aumento nas quantidades de falhas, uma quantidade maior de falhas das caixas por erro humano e complicações ao operador por realizar por muitas horas uma tarefa repetitiva.

1.2 Hipótese

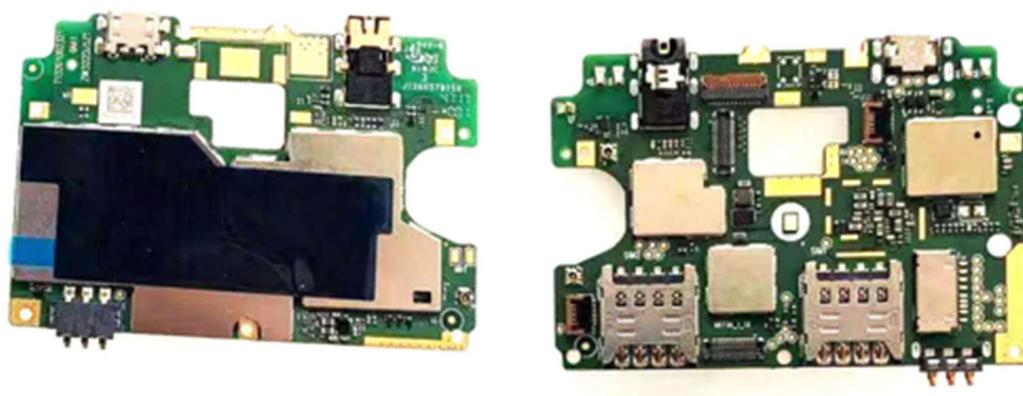
É possível automatizar esse processo de *load* e *unload* de placa utilizando um braço robótico com visão computacional para o encaixe de placas nos berços melhorando a qualidade do processo e aumentando o tempo entre falhas das caixas de teste?

1.3 Justificativa

Uma empresa de montagem de celulares localizada no polo industrial de Manaus possui linhas de montagem que cobre desde a montagem da placa de circuito impresso do celular até a montagem final dos celulares. Para este projeto a etapa selecionada foi a de testes de radiofrequência destas placas de circuito impresso (PCBA).

As placas de circuitos impresso semelhantes a placas mostrada na Figura 1 precisam passar por uma série de testes funcionais, no caso do processo selecionado para esse projeto de automação testes de radiofrequência, antes de serem levadas para a montagem final dos celulares para garantir que estão funcionando da maneira correta.

Figura 1 - Placas de circuito impresso de celulares

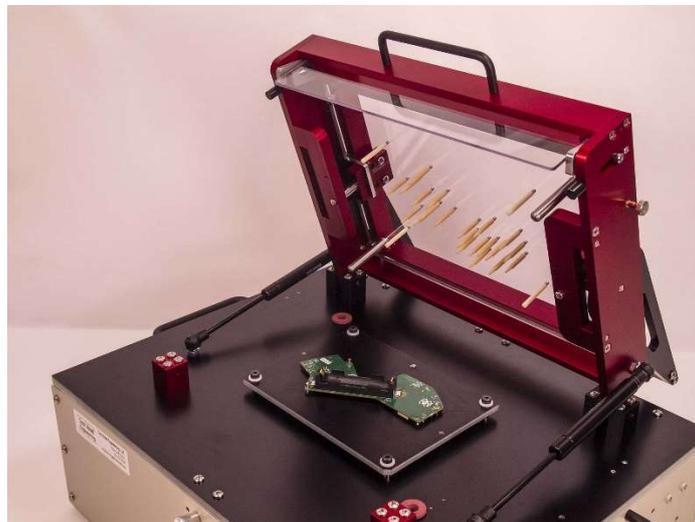


Fonte: *AliExpress*

<https://pt.aliexpress.com/i/4000421761100.html>

Para isso existe um processo manual onde essas placas são inseridas em caixas de teste semelhantes à mostrada na Figura 2.

Figura 2 - Caixa de teste funcional de placas de circuito



Fonte: *Test Head Engineering*

<https://www.testhead.com/products/functional-test-fixtures/>

Para isso o operador responsável por este posto segue os seguintes passos:

- O operador recebe placas do posto anterior para serem testada;
- O operador coloca as placas em todas as caixas de testes disponíveis primeiro abrindo as caixas, depois posicionando as placas da maneira correta no berço de teste com auxílio de guias;
- Após ocupar todas as caixas de teste o operador coloca as placas sobressalentes que ainda precisam ser testadas em buffers próximos as caixas que estão ocupadas;
- Após uma caixa concluir um teste o operador, abre a caixa, retira a placa testada e coloca no lugar a placa que está no buffer esperando para ser testada;
- Caso o teste seja concluído com sucesso o operador coloca as placas testada em bandejas que comportam até 12 placas de cada vez para serem levadas para o processo seguinte;
- Caso o teste seja concluído com alguma falha a placa é retestada e em caso uma segunda falha é levada para análise e reparo.

O tempo desse processo é terminado pelo tempo de teste que varia para cada produto. No caso de um tempo de teste muito longo é necessário que o processo seja equipado com mais caixas de teste para manter a taxa de placas produzidas por hora e para tempos de teste mais rápidos é necessário menos caixas. O tempo médio de um operador realizando esse processo de colocada e retirada das placas é calculado em aproximadamente 12 segundo.

Tomando como ponto de partida este processo manual descrito, este projeto teve como objetivo o desenvolvimento de um protótipo de uma célula robótica para a automação parcial deste processo. A automação foi identificada como parcial pois foi planejada para manipulação de apenas 16 caixas de teste de cada vez sendo necessário então para o caso de produtos que precisam de mais caixas de teste um trabalho em conjunto com o processo manual ou a aplicação de mais células para a automação completa do processo.

Está célula utiliza um braço robótico para a manipulação dessas placas, processamento digital de imagem para o referenciamento e encaixe correto dessas placas nos berços de teste.

É esperado que esta automação gere uma diminuição nas falhas, aumento no tempo entra falhas das caixas, redução do esforço repetitivo dos operadores, aumento da produtividade.

1.4 Objetivo Geral

Desenvolver, construir e testar um protótipo de uma célula robótica para automatizar o processo de *load* e *unload* de placas de circuito impresso de celulares em caixas de teste de radiofrequência para obter um ganho com a diminuição de operadores necessários para o processo, diminuição de falhas com a redução do manuseio das placas e das caixas e um aumento da produtividade.

1.5 Objetivos Específicos

- Analisar o processo manual e todos os pontos necessários para automatizá-lo
- Desenvolvimento do conceito do protótipo da célula robótica

- Estudo e seleção do melhor método para o referenciamento da placa e do berço de teste para a melhor taxa de sucesso de encaixe;
- Seleção e programação do braço robótico;
- Seleção e programação do CLP para controle de esteiras e outros componentes do sistema.

2. Referencial Teórico

2.1 Automação Industrial

A automação industrial é a aplicação de diversos tipos de equipamentos para a manipulação de processos na indústria por meios mecânicos e automáticos com o objetivo de substituir o trabalho humano. A utilização dessas aplicações se justifica pelo objetivo de melhorar estratégias de produção para aumentar a competitividade em termos de custo, disponibilidade, inovação e qualidade de um produto. PEREIRA (2003).

Segundo ROGGIA e FUENTES (2016), é possível dividir automações industriais de 2 maneiras. Quanto ao tipo de processo podem ser de:

- Processos de manufatura onde há grande movimentação de partes mecânicas como uma grande linha de montagem
- Processos contínuos onde há pouca movimentação mecânica como a automação de uma estação de tratamento de água.

A outra classificação proposta por ROGGIA e FUENTES (2016) está relacionada ao grau de flexibilidade da automação podendo ser:

- Automação rígida projetada para fabricação de um produto específico com alta produção e tendo assim inflexibilidade na variedade de produção;
- Automação programável com capacidade de acomodar diferentes configurações de produtos controlados por diferentes programas muito utilizado quando se tem uma produção baixa de cada produto diferente;
- Automação flexível que reúne características das dos tipos anteriores tendo assim um equipamento específico, mas que pode ser programado ou controlado por programas diferentes para atender especificações e características de diferentes produtos.

2.2 Sensores e Atuadores

Sensores são dispositivos sensíveis à estímulos de energia de variadas fontes. Essas fontes podem ser por exemplo luminosa, térmica, elétrica, cinética.

Esses estímulos são então relacionados a grandezas físicas que podem ser medidas, como: pressão, velocidade, corrente, aceleração, posição etc. Normalmente para se utilizar um sensor da maneira correta é necessário manipular o seu sinal de saída utilizando um sistema de controle. Geralmente se utiliza um circuito de interface que, por exemplo, amplia o sinal para que esse possa ser utilizado efetivamente por um controlador. WENDLING (2010).

A Figura 3 mostra vários tipos diferentes de sensores.

Figura 3 - Vários tipos de sensores



Fonte: Próprio autor

Atuadores são dispositivos que modificam uma variável controlada ao receber um sinal e agem sobre um sistema controlado. Os atuadores mais comuns podem ser elétricos, hidráulicos ou pneumáticos. Alguns exemplos de atuadores são: Válvulas, Relés, Cilindros, motores e solenoides.

A Figura 4 mostra tipos diferentes de atuadores.

Figura 4 - Vários tipos de atuadores



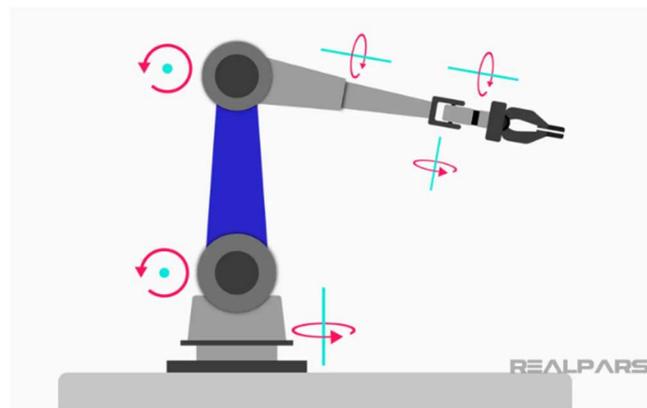
Fonte: Próprio autor

2.3 Robótica Industrial

Para FRASCÁ e BRUNO (2021), um robô industrial é composto por partes rígidas para aguentar o esforço aplicado sobre ele, se movimenta através de elos e juntas e executa tarefas através de acessórios e ferramentas anexadas. Funcionando de maneira análoga ao corpo humano as partes rígidas são como o braço, as juntas as articulações e a ferramenta como a mão.

A Figura 5 mostra uma característica importante de manipuladores robôs que são os eixos. O grau de liberdade de um robô é determinado pela quantidade de posições precisam ser especificadas para que o mecanismo seja localizado. No caso de robôs industriais típicos é comum que o grau de liberdade seja igual ao número de juntas. CRAIG (2005).

Figura 5 - Manipulador robótico 6 eixos



Fonte: realpars

<https://realpars.com/6-axis-robot/>

Algumas classificações comuns de robôs são: cartesiano, articulado, cilíndrico, esférico e SCARA.

Através da linguagem de programação é possível instruir para o robô trajetórias e tarefas. É comum que essas trajetórias sejam especificadas por uma sequência de pontos referenciados em relação ao sistema de coordenadas do robô. Juntamente com essas informações o software de programação do robô pode especificar também outras características do movimento como velocidade, aceleração e outras que possam melhorar estes movimentos. CRAIG (2005).

2.4 Programação de Robôs

A programação de robôs industriais pode ser feita utilizando dois tipos diferentes de métodos programação *on-line* ou *off-line*.

Sobre a linguagem utilizada para a programação, vários deles podem ser programados utilizando linguagem comuns, porém existem também robôs com linguagem específicas para programação de movimentos ou comandos para acessar sensores e atuadores. Segundo LOZANO-PEREZ (1983), essas linguagens são referidas como linguagens explícitas ou a nível de robô e a principal vantagem é que elas permitem que os dados de sensores externos, como visão e força, sejam usados na modificação dos movimentos do robô e a principal desvantagem seriam que elas exigem que o programador do robô seja especialista em programação de computadores e no projeto de estratégias de movimento baseadas em sensores dificultando assim o acesso ao trabalhador típico no chão de fábrica. Porém essa desvantagem vem sendo mitigada com o avanço de técnicas de programação de alguns fabricantes. É possível hoje em dia que robôs industriais sejam programados com linguagens a nível de robô utilizando ambientes de programação gráfica que simplificam a complexidade da programação deixando assim a programação mais acessível para pessoas com menos conhecimento em programação.

2.4.1 Programação on-line

De acordo com M D ANTUNES (2015), a programação *on-line* requer a utilização física do robô, pois utiliza ele para gerar as movimentações necessárias que permitem que as trajetórias sejam definidas. Essas movimentações podem ser definidas utilizando um software próprio do robô em auxílio com uma tela ou computador conectados com o robô que faz os movimentos desejados e maneira assistida e os grava gerando assim a trajetória desejada ou movimentando o manualmente Figura 6 com uma pessoa levando manualmente o robô para os pontos desejados de trajetória gerando dessa maneira a movimentação necessária para que o robô realize a tarefa desejada.

Ainda segundo M D ANTUNES, (2015), esse tipo de programação tem como principal vantagem a facilidade de criação do programa por um operador e como desvantagem principal a necessidade de disponibilidade do robô causando assim uma parada dele enquanto está sendo programado.

Figura 6 - Robô sendo movimentado manualmente



Fonte: TM Collaborative Robots Tutorial 3 – How to Program Using Flow Chart Software
https://www.youtube.com/watch?v=7jGXwGiYV3g&ab_channel=OmronIndustrialAutomationEM

2.4.2 Programação *off-line*

Outro método utilizando na programação de robôs é a programação *off-line* que para PAN e colab., 2012, apresenta várias vantagens do método *on-line* como por exemplo o fato de não ser necessário estar na presença do robô ou parar sua operação para que seja feita a sua programação, possibilidade de início da programação do robô mais cedo no cronograma de um projeto, pois não é necessário que o robô esteja na sua estrutura final para que seja iniciada a programação, flexibilidade na programação com mudanças, melhorias e ajustes podendo ser implementados de maneira mais rápida e possibilidade do desenvolvimento de simulações.

2.5 Protocolos de comunicação

As entidades em diferentes sistemas podem se comunicar entre si por meio de redes de computadores. Uma entidade é qualquer coisa capaz de transmitir ou receber dados. Mais do que isso, as duas entidades não podem simplesmente trocar um fluxo de dados e esperar que o conteúdo seja entendido. As entidades devem concordar com o protocolo a ser usado para que possa haver comunicação. Um conjunto de diretrizes para controlar a transmissão de dados é chamado de protocolo. O protocolo estabelece o que será comunicado, como será comunicado e quando. Sintaxe, semântica e sincronização (ou timing) são componentes essenciais de protocolo. (FOROUZAN, 2006)

- **Sintaxe:** Refere-se à estrutura ou formato dos dados e à ordem em que são apresentados. Por exemplo, um protocolo simples pode especificar que o primeiro byte indique o endereço de origem, o segundo byte o endereço de destino, e o restante dos dados seja a mensagem propriamente dita.
- **Semântica:** Trata do significado de cada conjunto ou seção de bits. Define como um padrão específico será interpretado e que ação será tomada com base nessa interpretação. Por exemplo, um endereço pode identificar uma rota a ser seguida no roteador ou o endereço final da mensagem.

- Timing: Relaciona-se a quando os dados devem ser enviados e a que velocidade. Por exemplo, se uma fonte de dados produz dados a 100 Mbps, mas o destino só pode receber a 1 Mbps, a transmissão sobrecarregará o receptor e os dados serão perdidos.

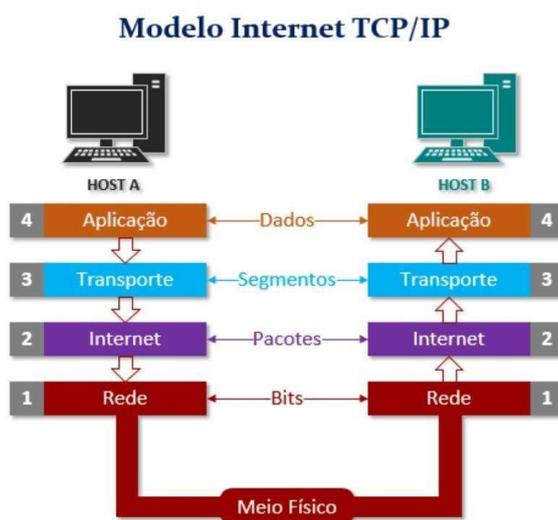
2.5.1 Protocolo TCP

O conjunto de protocolos TCP/IP permite a comunicação perfeita entre computadores de vários tamanhos, fabricados por diferentes fornecedores e operando em sistemas operacionais completamente distintos. A iniciativa de pesquisa financiada pelo governo sobre redes de comutação de pacotes, que começou no final da década de 1960, evoluiu para o método mais prevalente de redes de computadores na década de 1990. A definição do conjunto de protocolos e a maioria de suas implementações são acessíveis ao público sem nenhum ou mínimo custo. (STEVENS, 1994)

TCP/IP é um protocolo normalmente considerado como tendo 4 camadas.

Figura 7

Figura 7 – Camadas do protocolo TCP/IP



Fonte: Protocolo TcP/ip Como Funciona E Quais as Suas Camadas

<https://www.tpsearchtool.com/images/protocolo-tcpip-como-funciona-e-quais-as-suas-camadas>

Segundo (STEVENS, 1994) cada camada tem uma responsabilidade diferente.

- A camada de enlace, às vezes chamada de camada de enlace de dados ou camada de interface de rede, normalmente inclui o driver de dispositivo no sistema operacional e a placa de interface de rede correspondente no computador. Juntos, eles lidam com todos os detalhes de hardware da interface física com o cabo (ou qualquer tipo de mídia que esteja sendo usada).
- A camada de internet controla a movimentação de pacotes pela rede. O roteamento de pacotes, por exemplo, ocorre aqui. IP (*Internet Protocol*), ICMP (*Internet Control Message Protocol*) e IGMP (*Internet Group Management Protocol*) fornecem a camada de rede no conjunto de protocolos TCP/IP.
- A camada de transporte fornece um fluxo de dados entre dois hosts, para a camada de aplicação acima. No conjunto de protocolos TCP/IP existem dois protocolos de transporte muito diferentes: TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*). O TCP fornece um fluxo confiável de dados entre dois hosts. Ele se preocupa com coisas como dividir os dados transmitidos a ele pelo aplicativo em pedaços de tamanho apropriado para a camada de rede abaixo, confirmar os pacotes recebidos, definir tempos limite para garantir que a outra extremidade reconheça os pacotes enviados e assim por diante. Como esse fluxo confiável de dados é fornecido pela camada de transporte, a camada de aplicação pode ignorar todos esses detalhes. O UDP, por outro lado, fornece um serviço muito mais simples para a camada de aplicação. Ele apenas envia pacotes de dados chamados datagramas de um host para outro, mas não há garantia de que os datagramas cheguem ao outro lado. Qualquer confiabilidade desejada deve ser adicionada pela camada de aplicação. Existe uma utilização para cada tipo de protocolo de transporte, que veremos quando observarmos as diferentes aplicações que utilizam TCP e UDP.

- A camada de aplicação trata dos detalhes da aplicação específica. Existem muitos aplicativos TCP/IP comuns que quase todas as implementações fornecem:
 - Telnet para login remoto,
 - FTP, o protocolo de transferência de arquivos,
 - SMTP, o protocolo *Simple Mail Transfer*, para correio eletrônico,
 - SNMP, o Protocolo Simples de Gerenciamento de Rede

2.5.2 Modbus TCP

O protocolo *Modbus* foi introduzido em 1979 pela empresa "Modicon", líder no mercado de controladores lógicos programáveis (CLP). Foi concebido como o protocolo interno de comunicação ponto a ponto entre os CLPs Modicon e os painéis de programação usados para programar os controladores. O protocolo continua até hoje a ser utilizado por sua facilidade de entender e aplicar. Além disso, é um sistema aberto e pode ser usado gratuitamente. O *Modbus* pode ser encontrado em diversos setores de automação, incluindo automação industrial.

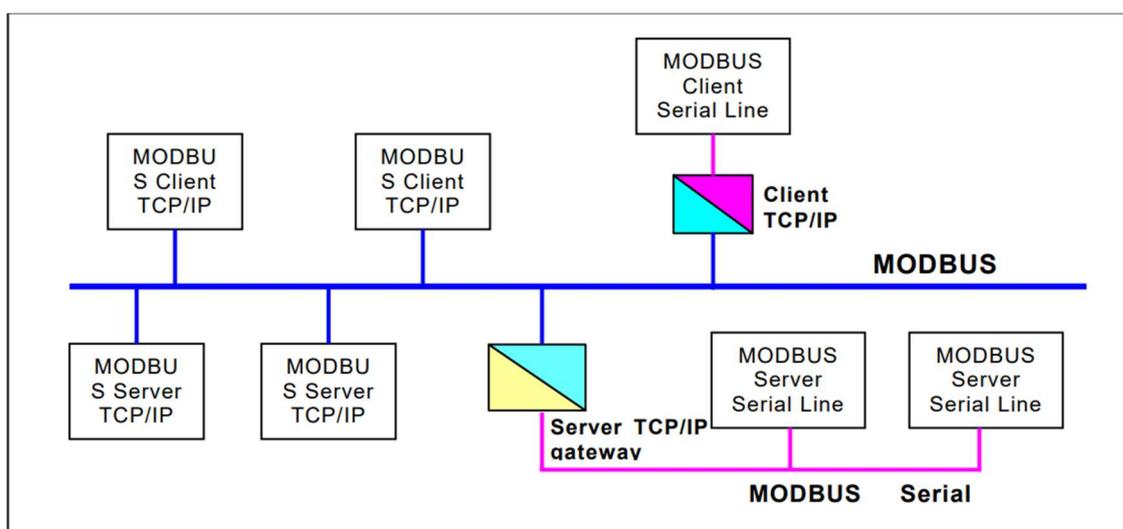
Segundo (*MODBUS ORGANIZATION*, 2005) as maiores vantagens de se utilizar o protocolo *Modbus* TCP são:

- Simplicidade: O *Modbus* TCP/IP simplesmente pega o conjunto de instruções *Modbus* e envolve o TCP/IP em torno dele. Se você já possui um driver *Modbus* e entende de Ethernet e soquetes TCP/IP, poderá ter um driver instalado e funcionando e conversando com um PC em poucas horas. Os custos de desenvolvimento são excepcionalmente baixos.
- Ethernet padrão: Não são necessários chipsets específicos e é possível usar placas Ethernet de PC padrão para se comunicar com seu dispositivo recém-implementado.
- Gratuito e de aberto para modificações: O protocolo *Modbus* foi transferido da Schneider Electric para a Organização *Modbus* em abril de 2004. A especificação está disponível gratuitamente para download e não há taxas de licenciamento subsequentes exigidas para o uso dos protocolos *Modbus* ou *Modbus* TCP/IP.

- Disponibilidade de muitos dispositivos: A interoperabilidade entre dispositivos de diferentes fornecedores e a compatibilidade com uma grande base instalada de dispositivos compatíveis com *Modbus* tornam o *Modbus* uma excelente escolha.

A comunicação de sistemas utilizando o protocolo Modbus TCP/IP funciona inclui basicamente dois dispositivos conectados através de uma rede TCP/IP utilizando normalmente ethernet como o meio físico. Um deles fazendo o papel de cliente e o outro de servidor. A Figura 8 mostra um exemplo de arquitetura *Modbus*.

Figura 8 – Exemplo de arquitetura de comunicação *Modbus* TCP/IP



Fonte: *Modbus messaging implementation guide*

O protocolo Modbus TCP permite acesso e manipulação dos seguintes dados de equipamentos:

- Registradores: permite ler e escrever valores
- Bobinas de saída: permite ler e escrever valores
- Leitura e escrita de entradas discretas: permite ler e escrever valores em entradas discretas.
- Leitura e escrita de registradores de entrada: permite ler e escrever valores em registradores de entrada.

2.6 CLP

Controladores lógicos programáveis (CLPs) são amplamente utilizados na automação industrial. Esses dispositivos, considerados a tecnologia de controle de processos mais difundida, funcionam como computadores industriais programáveis para executar funções de controle. Os CLPs substituíram a fiação complexa associada aos circuitos de controle convencionais com relés, oferecendo benefícios como facilidade de programação, instalação, alta velocidade de resposta, compatibilidade de rede, detecção de defeitos e confiabilidade. (FRANK D. PETRUZELLA, 2014)

Figura 9 - CLP MODICON



Fonte: Modicon: 50 anos de Pioneirismo e Inovações

<https://blog.se.com/br/automacao-industrial/2018/11/30/modicon-50-anos-de-pioneirismo-e-inovacoes/>

De acordo com FRANCHI e DE CAMARGO, 2008 primeiro controlador lógico totalmente programável (Figura 9) foi desenvolvido em 1969 por uma firma de engenharia, Bedford Associates, depois chamada de Modicon. Foi projetado como um sistema de controle a computador dedicado construído especialmente para atender as especificações de um edital da General Motors Hydramatic Division, com as seguintes características:

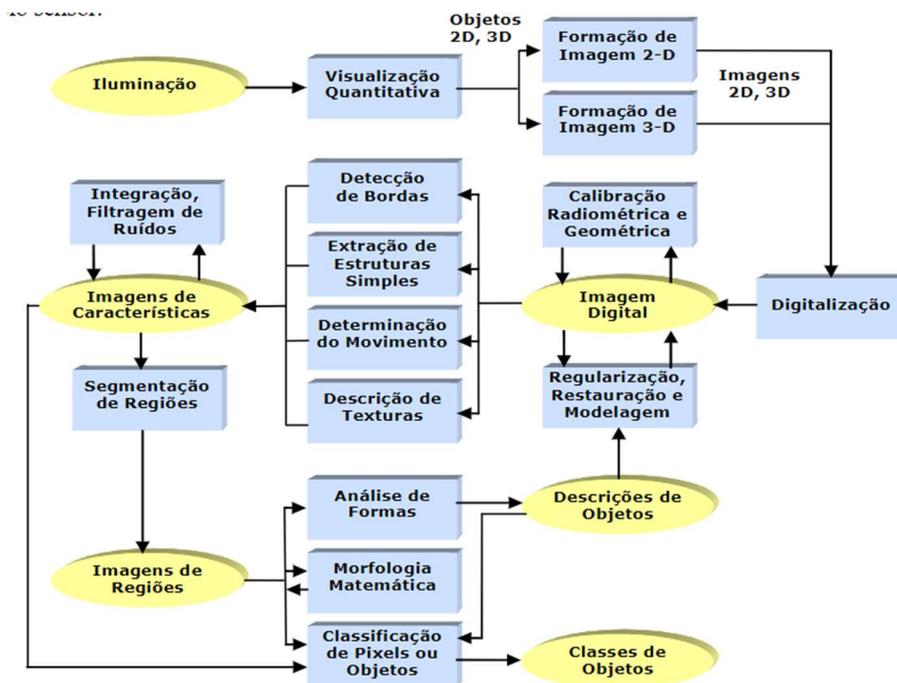
- Facilidade de programação e reprogramação;
- Facilidade de manutenção e reparo;
- Confiabilidade;

- Menor tamanho que o sistema equivalente em relés;
- Competitivo em custo com painéis de relés e eletrônicos equivalentes;
- Ter possibilidade de expansões com alterações mínimas;
- Ter uma memória programável;

2.7 Processamento digital de imagens

Segundo EUSTÁQUIO (2001), o processamento digital de imagem envolve um conjunto de tarefas interconectadas que se inicia na captura de uma imagem realizadas através de um sistema de aquisição que realiza essa captura de acordo com a iluminação refletida na superfície dos objetos, passa por um processo de digitalização para que essa imagem seja propriamente representada para um tratamento computacional.

Figura 10 - Hierarquia de tarefas de processamento de imagens



Fonte: EUSTÁQUIO (2001)

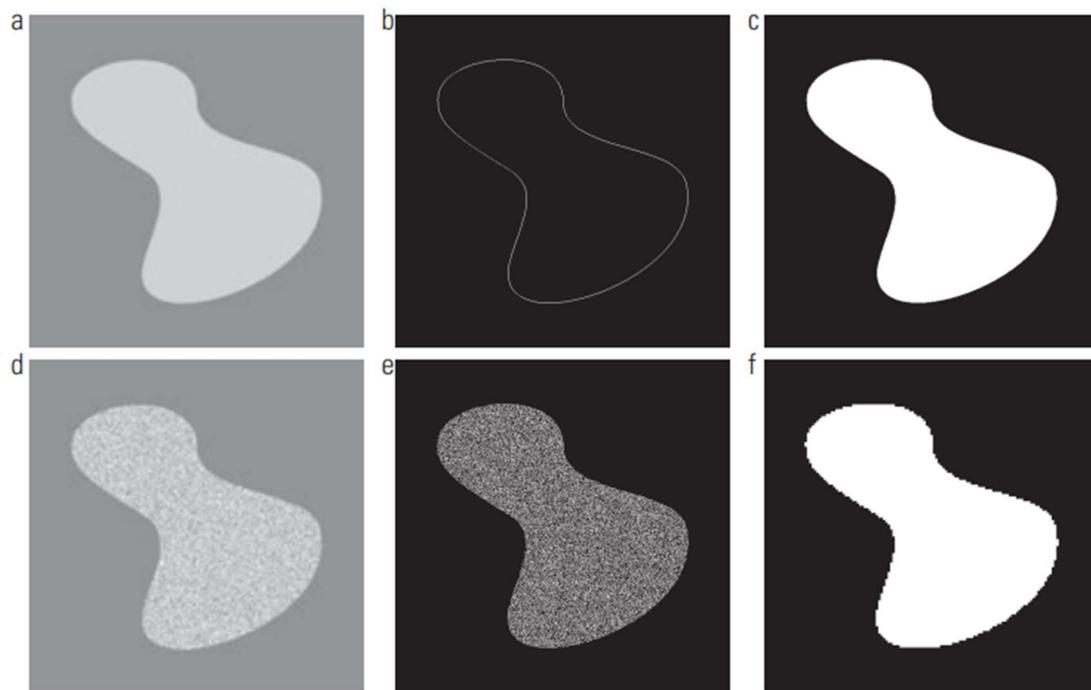
2.7.1 Segmentação de imagens

Segundo (GONZALEZ e WOODS, 2010), “A segmentação subdivide uma imagem em regiões ou objetos que a compõem. O nível de detalhe em que a subdivisão é realizada depende do problema a ser resolvido. Ou seja, a segmentação deve parar quando os objetos ou as regiões de interesse de uma aplicação forem detectados. Por exemplo, na inspeção automatizada de componentes eletrônicos, o interesse está em analisar as imagens dos produtos com o intuito de determinar a presença ou ausência de anomalias específicas, como a falta de componentes ou circuitos de conexão interrompidos. Não há sentido nenhum na execução da segmentação além do nível de detalhamento necessário para identificar esses elementos.”

Ainda segundo GONZALEZ e WOODS, 2010, as abordagens de segmentação de imagem podem ser divididas em duas categorias: descontinuidade e similaridade. A abordagem de descontinuidade se divide uma imagem com base nas mudanças bruscas de intensidade, como as bordas. Já as de similaridade estão baseadas na divisão de uma imagem em regiões que sejam semelhantes de acordo com um conjunto de critérios predefinidos. A limiarização, o crescimento de região e a divisão e fusão de regiões são exemplos dos métodos desta categoria.

Na Figura 11 é mostrado dois exemplos de segmentação. No primeiro a imagem possui uma região com intensidade constante, a borda dessa região é detectada e então segmentada na última imagem. No segundo exemplo o mesmo ocorre, porém a imagem não possui uma região de intensidade constante e sim uma textura.

Figura 11 - Exemplo de segmentação de imagem

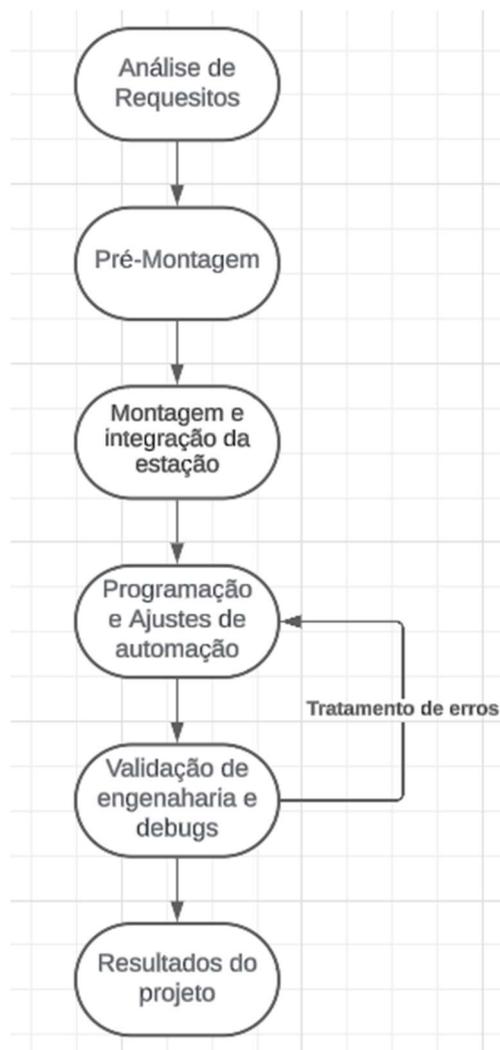


Fonte: GONZALEZ e WOODS, 2010

3. Metodologia

O desenvolvimento deste projeto nasceu da necessidade de automação de um posto de trabalho com o objetivo de reduzir o manuseio e aumentar a produtividade. A característica principal deste projeto e que será abordada neste trabalho é a combinação do processamento digital de imagem e programação de robôs. A utilização de processamento digital de imagem nesta abordagem vem para facilitar a automatização do processo tendo em vista que com essa técnica é possível diminuir a quantidade de montagens precisas e calibrações mecânicas. Isso se faz possível por causa dos avanços de técnicas de processamento digital de imagem dos últimos anos.

Figura 12 – Fluxograma simplificado de desenvolvimento do projeto



Fonte: Próprio autor

A Figura 12 mostra de maneira simplificada as etapas que foram seguidas no desenvolvimento desse projeto.

O ponto de partida foi a análise do processo feito manualmente. Analisando detalhadamente como esse processo era feito, cada atividade do operador, o nível de montagem da placa neste ponto do processo, entre outros. A partir deste ponto foram definidos os requisitos mínimos e selecionadas ferramentas que iriam auxiliar na construção e implementação desta automação na fábrica. Foi partindo desta análise que foram selecionados o robô que seria utilizado, a técnica de processamento de imagem utilizada e outros detalhes do projeto.

Após essa análise definições o projeto seguiu para uma fase de pré-montagem para que fossem feitos os testes iniciais de validação dos equipamentos selecionados na etapa de análise de requisitos.

Com os equipamentos já definidos e validados o projeto seguiu para a etapa de montagem e integração onde foi feita a montagem final do projeto.

Com todas as partes do projeto montadas e integradas o projeto segue para a parte de programação e ajustes de automação onde foram feitas as programações do robô e CLP além de teste de validação do método utilizado pelo robô para manipulação das placas utilizando sistema de visão.

Por último o projeto entra na fase de validações de engenharia onde são testados e validados todos os pontos feitos nas atividades anteriores. Nesse ponto são analisadas todas as falhas ainda presentes na máquina que são tratados através de programação e ajustes de automação. Tendo corrigido ou mitigado todas as falhas encontradas durante essa etapa o projeto então foi concluído e teve seus resultados analisados.

A seguir serão explorados com mais detalhes cada uma das etapas adotadas para a concretização desse projeto, desde a análise inicial dos requisitos até a validação prática do sistema. O objetivo é demonstrar como foi montado e validado esse projeto com destaque principal para as etapas de programação do robô, processamento de imagem e resultados de encaixe das placas em dois cenários: Encaixe das placas em um berço e encaixe das placas em paletes.

4. Desenvolvimento

4.1 Análise de requisitos

A primeira fase do desenvolvimento envolve uma análise detalhada dos requisitos do projeto. Isso inclui compreender as especificações das placas de PCB, os tipos de componentes a serem manuseados, as tolerâncias de posicionamento e as velocidades necessárias, a maneira como o processo é feito manualmente. Com base nessa análise, é elaborado um plano de programação que define os objetivos específicos, as etapas do processo, as estratégias de detecção e manipulação. A seleção da linguagem de programação adequada e do ambiente de desenvolvimento e do robô que realizará a tarefa também é realizada nessa fase.

Neste processo o operador recebe uma pilha de bandejas com placas do processo anterior, leva essas bandejas até um ponto próximo à estação de teste, coloca as placas na estação de teste similar à mostrada na Figura 13, retira a placa após o teste ser finalizado e coloca a placa em bandejas que serão empilhadas e levadas até o processo seguinte.

Figura 13 - Fixtures para testes de PCBs



Fonte: Circuit check

<https://www.circuitcheck.com/product/functional-test-fixtures/>

Após o estudo, o processo foi dividido nos seguintes passos:

- Operador retira a placa da bandeja

- Operador escaneia *track-id* da placa
- Operador abre o testador e insere a placa
- Operador fecha o testador
- Placa passa por todos os testes
- Operador abre o testador e retira a placa já testada
- Caso a placa esteja Ok ela é armazenada pelo operador em uma bandeja para ser levada para o próximo posto
- Caso a placa não esteja Ok ela é separada para ser reparada ou retestada

Foi concluído então que para que a automação pudesse atender todos os requisitos levantados pela análise feita os passos que deveriam ser seguidos deveriam ser aproximadamente os seguintes:

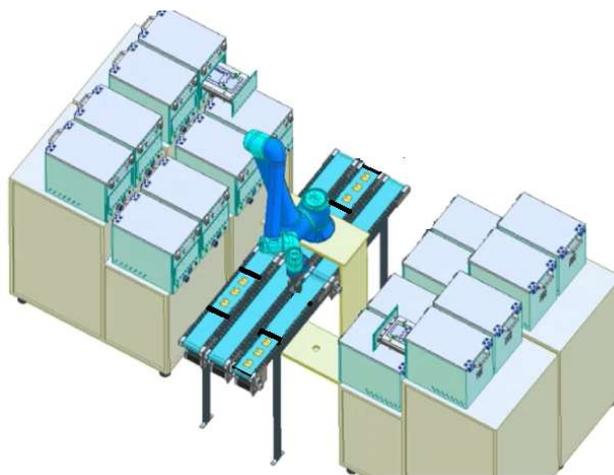
- A placa que deve ser testada deve ser colocada do processo anterior em uma esteira;
- Um braço robótico vai até o ponto de pega da placa, tirar uma foto para adquirir um referenciamento dela;
- O robô pega a placa, leva até um leitor de *QRCode* para decidir se a placa pode ser testada e indicar para o sistema de tracking da fábrica que aquela placa entrou naquele processo;
- O robô leva a placa até o berço de teste;
- Após a placa testada, o robô pega a placa e leva até a saída podendo ter dois modelos diferentes de saída que funcionam de acordo com a necessidade da placa. A saída pode ser em bandejas ou em paletes.

Com esses passos que automação deve seguir definidos se seguiu então para o processo design da solução e seleção de equipamentos que seriam utilizados na montagem, teste e validação dessa solução. Foi pesado então em uma solução dividida em dois módulos. Um módulo principal e um módulo de saída. Se chegou a essa conclusão visando dar mais flexibilidade para a automação. Com módulos fica mais fácil incrementar a solução futuramente ou

adaptar para outros tipos de finalidades deixando assim o processo inteiro mais flexível.

Na Figura 14 é mostrado o layout do módulo principal da solução que inclui a parte de pega da placa, deixa da placa no testador, teste da placa e pega da placa testada no testador.

Figura 14 – Layout proposto para o módulo principal da solução



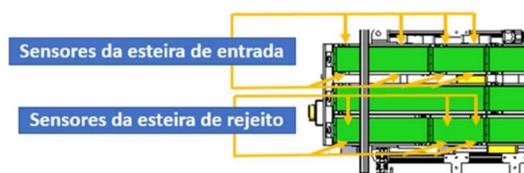
Fonte: Próprio autor

A solução mostrada na imagem mostra:

- Três esteiras de lona utilizadas para transportar as placas, cada esteira composta por um conjunto de sensores para garantir seu funcionamento ideal. A esteira principal, mais à esquerda na imagem, tem sensores no ponto de entrada de placas, no ponto de parada de pega de placa 1, no ponto de parada de pega de placa 2, no ponto de parada de buffer 1 e no ponto de parada de buffer 2. A esteira de rejeito, mais a direita na imagem, tem sensores no ponto de deixa de placa de rejeito 1, no ponto de deixa da placa de rejeito 2 e na saída da esteira;
- Um braço robótico TM-12 com câmera integrada;
- Quatro racks de testadores já vendidos pelo fabricante como elaborados pensando na automatização desse processo, pois são racks cujas gavetas podem ser abertas remotamente por sinais enviados pelo sistema de controle;

Na Figura 15 é possível observar melhor a disposição dos sensores e montagem das esteiras descritas acima.

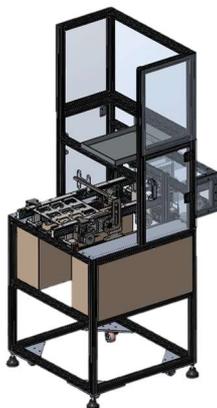
Figura 15 – Disposição dos sensores nas esteiras



Fonte: Próprio autor

A Figura 16 mostra o layout que foi proposto para um módulo de saída da placa. Neste módulo estão inclusas as funções de deixa da placa já testada no palete. Este módulo é composto por uma esteira que tem a capacidade de armazenar até dois paletes, um que está sendo preenchido de placas testadas e um segundo suspenso por um mini elevador que permite se criar esse *buffer* para agilizar a troca de um palete cheio por um vazio e permite que o palete cheio passe por baixo dele para seguir a diante para o processo seguinte.

Figura 16 – Layout proposto para o módulo de saída da solução



Fonte: Próprio autor

O braço robótico selecionado foi o TM-12 da Omron que foi selecionado por possuir um ambiente de programação fácil e prático para as aplicações previstas para a automação, pelo alcance de 1300mm que permite com que o braço possa gerenciar o a deixa de placas em até 16 berços de teste e por este modelo vir de fábrica com câmera no seu ferramental e com software para processamento digital de imagem já integrado no seu controlador.

Na Figura 17 é mostrado todos os modelos analisados de robô TM incluindo o TM12 que foi selecionado. A figura mostra os 4 modelos de mercado da empresa Omron disponíveis os modelos menores foram eliminados da seleção somente pelo seu alcance máximo que não iria satisfazer o requisito do projeto.

Figura 17 - Modelos de robô TM Omron

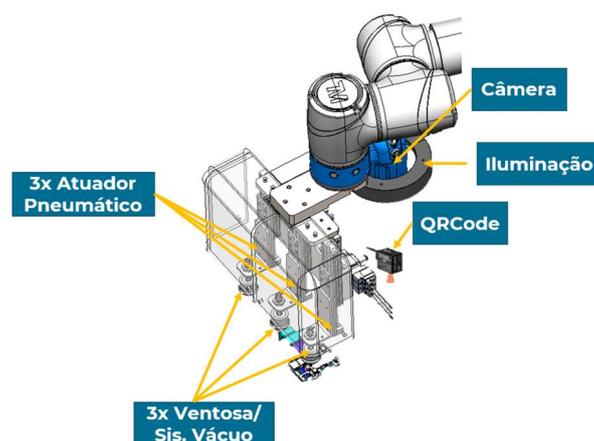


Fonte: Omron Industrial Automation

<https://www.ia.omron.com/products/family/3739/>

Foi determinado ainda nesta etapa o design do ferramental do robô. A Figura 18 mostra o ferramental elaborado para o robô que é composto por 3 atuadores pneumáticos com ventosas na ponta e sistema de vácuo, um leitor de *QRCode*, um anel luminoso para auxiliar a câmera integrada do robô. São usados 3 atuadores para transportar duas placas e ter um espaço vazio para realizar a troca da placa no testador. Logo, o espaço vazio tira a placa do testador e depois é colocada uma nova placa a ser testada que já está sendo transportada pelo ferramental. Isso foi determinado com o objetivo de se ganhar tempo de operação aumentando assim a capacidade operacional da solução.

Figura 18 – Layout proposto para o módulo de saída da solução



Fonte: Próprio autor

4.2 Pré-Montagem da automação

Com o processo analisado e os equipamentos necessários para criação da solução definidos o próximo passo seguido foi iniciar a montagem de uma estação semelhante a configuração final da automação para validar testes de movimentação do robô. Um dos principais objetivos dessa montagem é iniciar a programação do robô, testes de movimentação, programação do sistema de visão, testes de visão, testes de encaixe das placas nos berços de teste e testes de rotina do robô em paralelo ao design e construção mecânica do módulo principal.

Na Figura 19 é possível observar uma foto da estação montada para os testes iniciais. A estação é composta por um robô TM-12 e uma estrutura montada com perfis de alumínio. As dimensões, pontos de pega e deixa das placas foram todas retiradas do projeto final para que as movimentações do robô estivessem o mais próximas o possível da final afim de realizar a validação da velocidade e percurso que o robô deveria fazer para conseguir atender toda a área de trabalho prevista. A esteira de lona azul vista nesta imagem foi o ponto utilizado para a pega da placa representando assim a entrada do sistema.

Figura 19 – Estação montada para testes

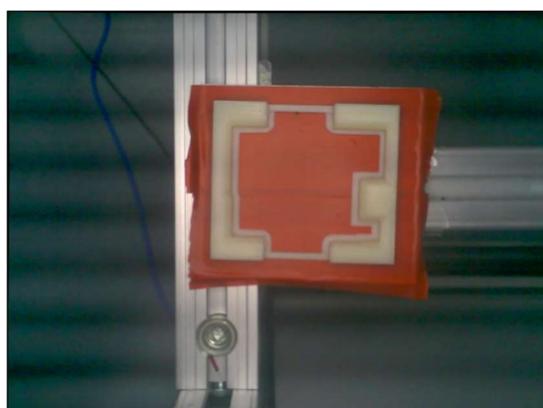


Fonte: Próprio autor

Para simular o ponto de deixa das placas na estrutura foi criado um berço com o formato da placa utilizando impressão 3D. Esses berços foram posicionados na estação simulada nas 16 posições que na estação final seriam ocupadas pelos racks de teste. Após algumas análises foi decidido adicionar no fundo desse berço uma fita vermelha com o objetivo de melhorar o contraste do berço o fundo facilitando assim o processamento digital de imagem que seria utilizado para realizar a localização dele. A

Figura 20 mostra como ficou este berço após impresso e colocado na posição com a fita vermelha em baixo.

Figura 20 – Berço de placas utilizado para testes



Fonte: Próprio autor

Foi também montado o ferramental do robô utilizando os requisitos e equipamentos que foram decididos no estudo anterior. Os acionamentos deste ferramental foram montados e conectados para que o pudessem ser controlados pela própria programação do robô. A Figura 21 mostra como ficou a montagem desse ferramental com uma estrutura metálica usinada construída para facilitar o encaixe no braço robótico. Foram feitas todas as conexões pneumáticas e elétricas necessárias para o seu funcionamento.

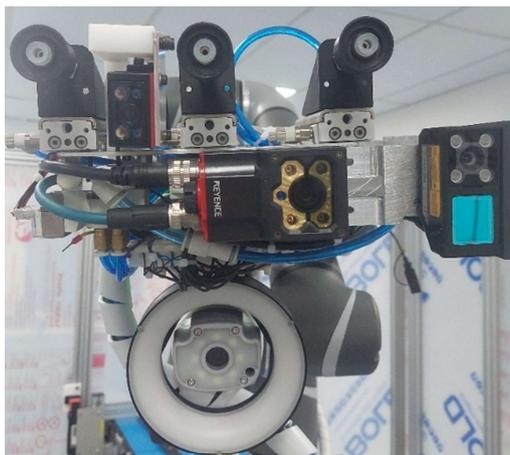
Figura 21 – Ferramental do robô com 3 atuadores e ventosas



Fonte: Próprio autor

O ferramental do robô é composto também por mais dois elementos uma iluminação externa e um leitor de *track-id*. A iluminação serve para melhorar o processamento de imagem, pois após alguns testes foi identificado que somente a iluminação do robô não seria suficiente para realizar o processamento de imagem com exatidão. O leitor de *track-id* é utilizado para fazer o rastreamento das placas no processo.

Figura 22 – Vista inferior o ferramental do robô



Fonte: Próprio autor

Também foi montado em cima da estrutura final do módulo de saída o ponto utilizado como saída do sistema afim de testar o fluxo completo bem como o encaixe das placas em paletes. Este ponto do projeto foi de extrema importância e crítico, pois os paletes possuíam uma folga muito pequena exigindo assim um processamento um pouco mais cuidadoso. A Figura 23 mostra a estrutura montada para ser utilizada como módulo de saída.

Figura 23 – Módulo de saída



Fonte: Próprio autor

4.3 Montagem e integração

Enquanto os testes eram feitos na bancada montada, se avançou na montagem do módulo principal. Nesse módulo foi feita a montagem das esteiras, quadros elétricos, quadros pneumáticos.

Foi determinado que a solução teria três esteiras com funções diferentes. Essas esteiras são:

- Esteira de entrada: Uma esteira dividida em três secções. Uma para entrada das placas, uma para parada das placas que serão manipuladas pelo robô e um para levar as placas para módulos seguintes. A divisão se deu com intenção de que fosse possível criar um *buffer* de placas, o robô pode manipular até duas placas de cada vez, mas outras 4 podem ficar no *buffer* esperando para serem manipuladas.
- Esteira de placas aprovadas: Esteira dividida em duas secções. Essa esteira tem como objetivo levar placas para o último módulo para serem paletizadas. Nessa esteira passariam placas que não precisam mais passar pelo processo de teste e somente precisam ser paletizadas, as secções também têm o objetivo de criar um *buffer*.
- Esteira de rejeito: Uma esteira dividida em três secções. Essa esteira funciona no sentido oposto ao das outras esteiras, pois tem como objetivo levar placas que foram rejeitadas pelo teste para fora da máquina para serem tratadas.

Figura 24 – Esteiras do módulo principal



Fonte: Próprio autor

Todos os componentes elétricos e pneumáticos necessários para gerenciar os controladores e atuadores da solução foram montados em cinco quadros localizados na parte inferior da estrutura principal e mais dois localizados na estrutura do módulo de saída da solução.

A divisão dos componentes em cada quadro do módulo principal ficou da seguinte maneira:

- Quadro de fontes e disjuntores: Neste quadro está toda a parte de potência da solução com todos os disjuntores, fontes para energizar atuadores e sensores, fonte para componentes de segurança e fonte para a iluminação externa do ferramental.
- Quadro de Controle: Neste quadro se encontra o CLP utilizado para o controle das esteiras, torre de iluminação, atuadores de travamento das racks e portas de segurança dos racks.
- Quadro de Drivers: Neste quadro estão todos os drives que comandam os motores de cada secção das esteiras do módulo principal e as fontes necessárias para energizar esses drives e motores.
- Quadro de Rede e segurança: Neste quadro contém um switch para conectar todos os componentes em uma mesma rede. Neste switch estão conectados o CLP, os leitores de *track-id*, os racks e o computador principal onde é processado o software de controle da solução. Ainda neste quadro está o controlador de segurança que controla os sensores das portas, botões de emergência e sensores das portas dos racks, além de se conectar com a segurança do módulo de saída.
- Quadro Pneumático: Neste quadro está a estação de distribuição de ar comprimido, as válvulas que comandam as portas e as válvulas que comandam as travas dos racks.

Figura 25 – Quadros elétrico de drives e Quadro elétrico de rede e segurança



Fonte: Próprio autor

A Figura 25 mostra os quadros elétricos de drives e o quadro elétrico de rede e segurança já montados com todas as ligações internas feitas.

Figura 26 - Quadro elétrico de fontes e disjuntores



Fonte: Próprio autor

A Figura 26 mostra o quadro elétrico frontal que contém as fontes e os disjuntores da automação. A imagem mostra o quadro montado com todas as ligações internas já montadas.

Além disso, nesta montagem também se substituiu a estrutura de alumínio utilizada para simular o ponto de deixa das placas pelos racks que seriam os alvos reais de deixa das placas. A Figura 27 mostra este rack de teste que foi um item comercial comprado pois já é fornecido preparado para automação com gavetas que podem ser abertas remotamente.

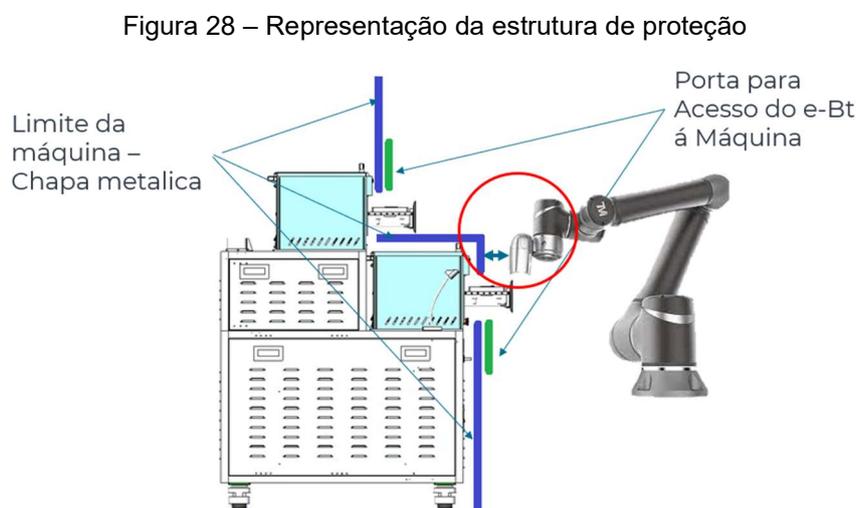
Figura 27 – Racks de Teste



Fonte: Próprio autor

Foi também confeccionada uma estrutura de proteção para garantir a segurança da automação. Esta estrutura é anexada na estrutura principal da máquina e possui portas para que as gavetas possam estar na parte interna da máquina onde o braço robótico possa ter acesso e para que estas portas possam ser fechadas para que, no caso de um dos racks precisar ser retirado para manutenção, o robô possa continuar trabalhando nos outros racks de forma segura sem nenhum risco. Dessa maneira qualquer manutenção nos racks pode ocorrer causando um impacto mínimo na capacidade produtiva.

Na Figura 28 é mostrada uma ilustração de como foi elaborada e montada esta estrutura de proteção.



Fonte: Próprio autor

Foi necessário também fazer adaptações no berço onde a placa estava sendo encaixada para que este berço se aproximasse mais do berço final utilizado por este processo.

Para isso primeiramente se foi feito uma impressão 3D do berço para que ele pudesse ser encaixado na gaveta e a partir daí foram feitas as adaptações necessárias para que fosse possível continuar os testes de visão e encaixe das placas.

A Figura 29 mostra o primeiro protótipo impresso do que seria o berço utilizado para os testes. Este berço inicial precisou ser adaptado para que se tivesse mais contraste se aproximando mais do berço impresso inicialmente.

Figura 29 – Berço impresso da placa



Fonte: Próprio autor

4.4 Programação e Ajustes de automação

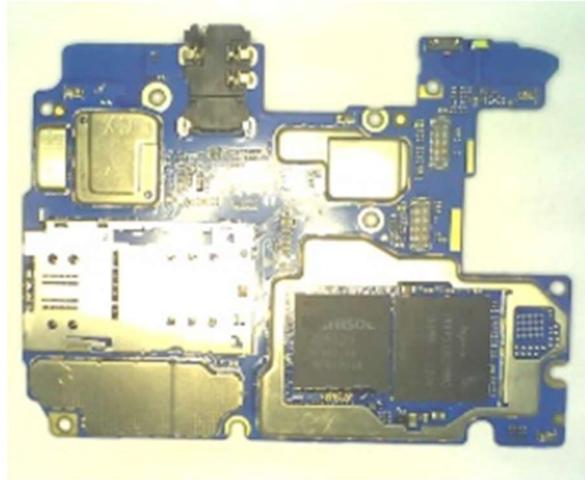
4.4.1 Sistema de visão

4.4.1.1 Testes do sistema de visão

Os testes para determinar a melhor metodologia para o processamento de imagem utilizado para localizar a placa utilizando o sistema de câmera do robô TM-12 da Omron foi inicialmente buscar uma referência dentro da PCB que gerasse maior contraste. Os componentes selecionados após uma análise inicial de algumas PCBs foram as shields metálicas ou outros elementos metálicos.

No primeiro teste se utilizou a própria iluminação disponível no robô tendendo a sua máxima potência para anular os efeitos da iluminação externa e não controlada. Foi desenvolvido um processamento de imagem capaz de localizar as referências da placa mesmo que a placa se movimente. Porém, quando se faz uma rotação de 90 ou 180 graus da placa dentro da cavidade da bandeja, a fonte de iluminação muda de posição em relação a placa e o tratamento de imagem realizado já não consegue identificar as referências. A Figura 30 mostra a imagem sem tratamento onde pode-se verificar uma concentração de luz no centro da placa devido a iluminação não difusa.

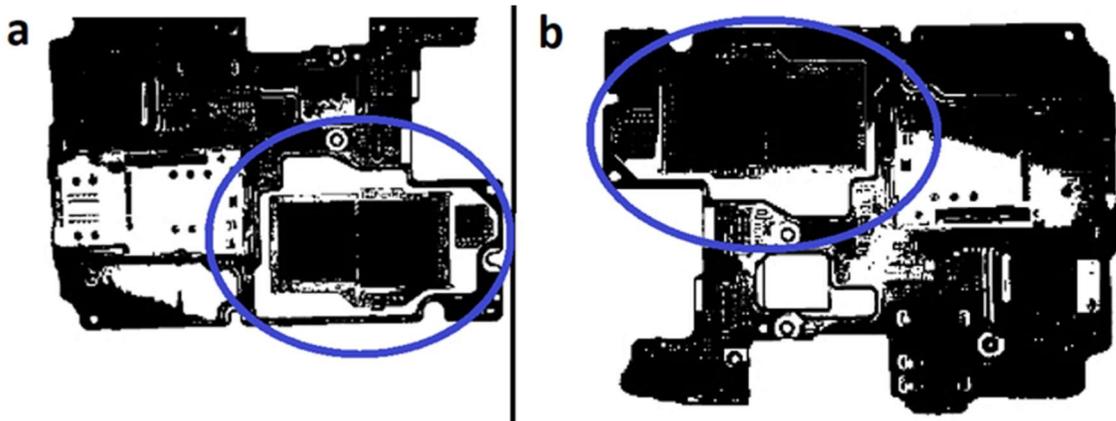
Figura 30 – Placa iluminada pela iluminação do robô



Fonte: Próprio autor

Na Figura 31a é possível ver a referência completamente na imagem, porém movendo a placa 180° como visto na Figura 31b a referência já não pode ser vista completamente devido a mudança da posição da placa em relação a iluminação.

Figura 31 – a) Imagem com a referência buscada bem destacada pela iluminação. b) Parte do contorno da referência selecionada não pode ser mais visto devido a modificação da posição da iluminação



Fonte: Próprio autor

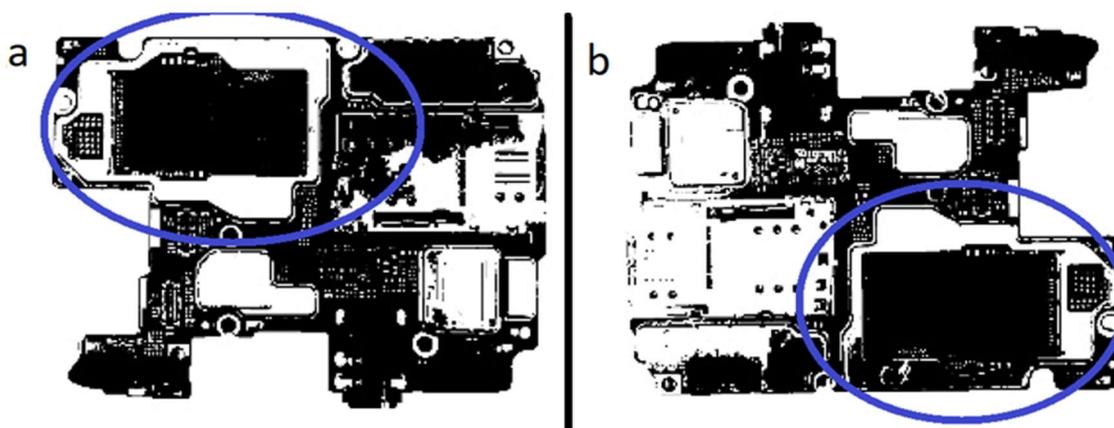
O teste seguinte continuou sendo realizado com a iluminação do robô, porém agora com potências menores e mesmo obtendo resultados melhores de referenciamento da placa mesmo a movendo bastante ainda assim, quando se

fazia uma rotação de 90 ou 180 graus da placa dentro da área de busca desejada a fonte de iluminação muda de posição em relação a placa e o tratamento de imagem realizado já não conseguia identificar as referências.

O próximo teste o foi realizado utilizando um sistema com a iluminação desligada tendo assim somente a iluminação externa do ambiente. O processamento de imagem utilizado foi capaz de localizar as referências da placa mesmo com a placa se movimentando dentro da área de busca, mas apesar de uma melhor capacidade de achar as placas mesmo rotacionadas em 90 ou 180 graus, ainda assim existiu falhas na detecção nesse teste, pois o robô gerou uma sombra da luz externa dependendo da posição da placa dificultando a geração do referenciamento da placa em áreas onde ocorrem essa sombra reduzindo muito o limite da área de busca da placa. Além disso essa metodologia foi descartada por não garantir a constância ou controlar a iluminação do ambiente o que geraria uma variável que no futuro poderia atrapalhar no processamento de imagem.

Na Figura 32a é possível ver a referência completamente na imagem e na Figura 32b a referência já não pode ser vista completamente por causa da sombra gerada pelo braço robótico e pelo ferramental do robô em cima da placa.

Figura 32 – a) Pode-se observar o contorno da referência selecionada. b) Parte do contorno da referência selecionada não pode ser mais visto devido a modificação da posição o que gerou sombra na placa.



Fonte: Próprio autor

Para os testes seguintes foram selecionadas iluminações externas e sistemas de visão diferentes para analisar também o melhor tipo de processamento que poderia ser feito.

Utilizando uma câmera preto e branco e processamento HDR com identificação de referência via contraste, o resultado observado foi um bom comportamento do sistema para identificar referências na placa quando ela foi colocada em diferentes posições e rotações.

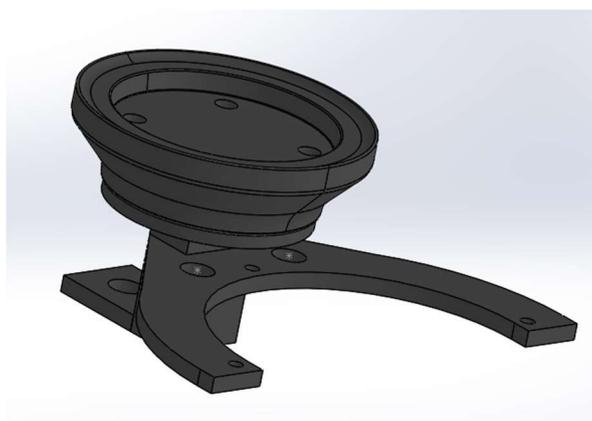
Utilizando uma câmera colorida com o mesmo processamento acima, o sistema identificava as referências das placas com mais dificuldade a depender da posição e rotação que a placa estava em relação a imagem.

Combinando os métodos testado acima com um sistema de iluminação do tipo anel difuso o sistema identificou as referências mesmo com a placa em diferentes posições e rotações.

Dessa maneira o caminho que se concluiu que seria o melhor a se seguir foi colocar no robô um sistema de iluminação do tipo anel difuso e refazer os testes para verificar se com a iluminação externa e com a câmera e processamento interno do robô é possível achar as referências da placa independentemente da posição e rotação que a placa esteja.

Para este teste foi desenvolvida uma ferramenta de suporte Figura 33 para uma iluminação redonda para a garra do robô Figura 34. A ideia é utilizar essa iluminação difusa para tentar eliminar interferências de iluminações externas e sombras geradas pela iluminação integrada do robô que foi testada na atividade anterior.

Figura 33 – Desenho da peça para suporte da iluminação na garra do robô



Fonte: Próprio autor

Figura 34 – Iluminação utilizada no teste e peça impressa para suporte da iluminação na garra do robô



Fonte: Próprio autor

Para determinar a eficácia desse sistema em processar a imagem foram utilizadas as mesmas referências dos testes anteriores: componentes metálicos e shields da placa. Os testes feitos utilizaram os mesmos métodos dos testes anteriores com a única diferença sendo a iluminação utilizada e os parâmetros destes métodos.

Para esse teste utilizou-se a iluminação externa mostrada na Figura 35. Foi utilizado um processamento muito semelhante ao da atividade anterior citada acima e foi possível identificar as referências desejada na placa de maneira satisfatória na maioria das posições testadas.

Porém, nas bordas da imagem ainda é possível identificar pontos onde está iluminação causa um reflexo muito grande como visto na Figura 36. Esse reflexo nos limites da área de busca ainda causa alguma dificuldade no referenciamento da placa em posições muito específicas.

Figura 35 – Sistema de iluminação



Fonte: Próprio autor

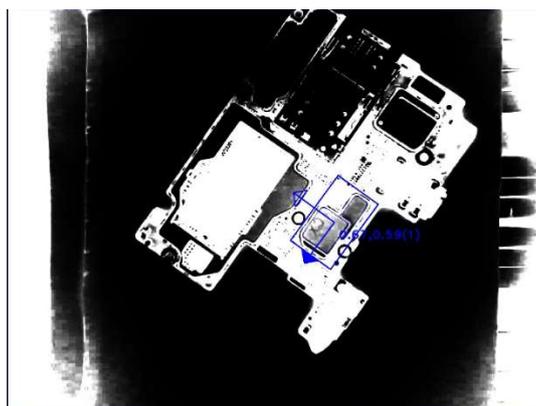
Figura 36 – visão da placa a uma distância de 150mm da câmera.



Fonte: Próprio autor

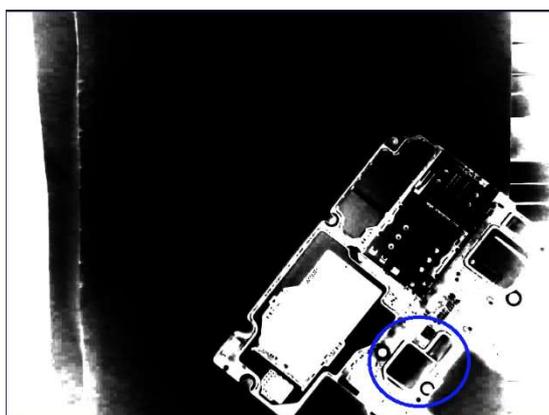
É possível notar que a iluminação incide melhor na placa que em comparação a iluminação do robô. Foi possível identificar o componente destacado em azul na Figura 37 em quase todas as posições utilizando essa com essa configuração.

Figura 37 – Componente referenciado na placa utilizando iluminação difusa e processamento da câmera do robô TM-12



Fonte: Próprio autor

Figura 38 – Sistema não encontrando a referência



Fonte: Próprio autor

Mas quando a placa é colocada um pouco longe do centro o sistema de visão como visto na Figura 38 tem dificuldade de encontrar o componente selecionado.

Um dos possíveis motivos identificados da diferença da iluminação na borda pode estar ocorrendo pela configuração do sistema de iluminação RM140-4Z não estar ideal para a aplicação desejada. Por esse motivo foram realizados mais testes com outro sistema de iluminação, o anel difuso de iluminação mostrado na Figura 39.

Figura 39 – Anel de Iluminação multiangular branca

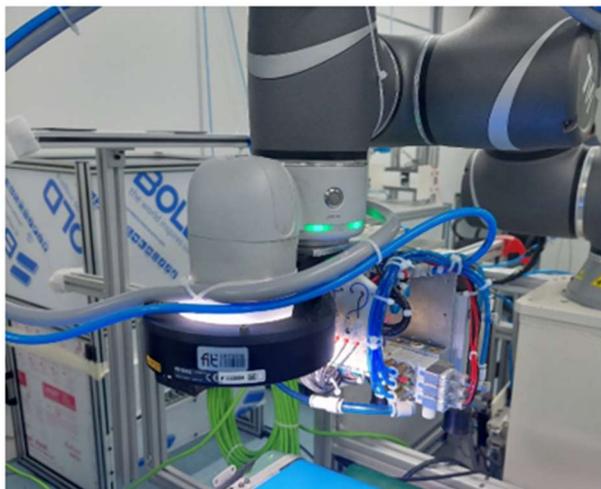


Fonte: Próprio autor

As principais diferenças identificadas entre as duas iluminações externas utilizadas para os testes foram a intensidade e a distância mínima de trabalho. A segunda iluminação possuindo uma distância mínima de trabalho e intensidades maiores que a primeira. Essas características permitiram com que a câmera do robô tivesse uma área de busca da imagem maior e mais consistente.

Após a repetição dos testes com a iluminação CA-DRW13M e reformulação do ferramental para adequá-lo à nova iluminação Figura 40 os resultados obtidos foram muito semelhantes, mas com uma qualidade superior como esperado para a aplicação desejada.

Figura 40 – Garra do robô com o anel de iluminação



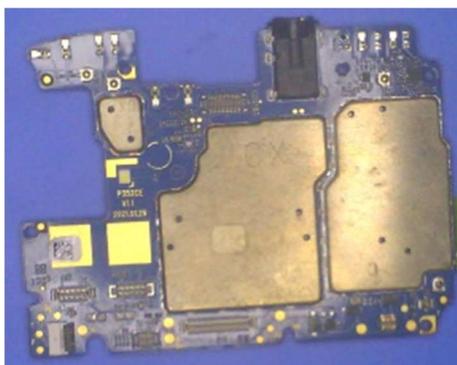
Fonte: Próprio autor

4.4.1.2 Processamento para referenciamento da placa

Após estes testes e a seleção do sistema de iluminação externo ideal para obter o resultado desejado a próxima etapa foi selecionar o tipo e as etapas de processamento de imagem para fazer o referenciamento desejado da pcb. Para referenciamento da placa na entrada do sistema foi selecionado o modo de “*fixed-point*” onde o sistema de visão captura somente uma foto para criar o sistema de coordenadas. Este modo foi selecionado por demandar menos tempo considerando que o robô teria que tirar uma foto sempre que uma nova placa chegue na entrada. Após uma série de testes se determinou que o melhor método para se chegar ao referenciamento desejado foi o seguinte:

- Primeiramente o robô captura uma foto da placa Figura 41

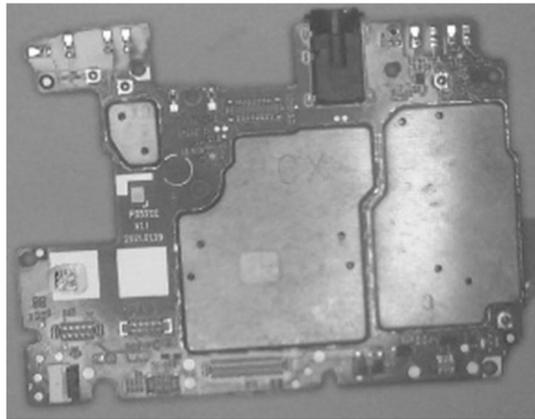
Figura 41 - Foto da placa antes de ser processada



Fonte: Próprio autor

- Essa imagem então passa por um processo de retirada do plano de cor cinza para destacar mais ainda os componentes metálicos que vão ser utilizados para o referenciamento. Figura 42

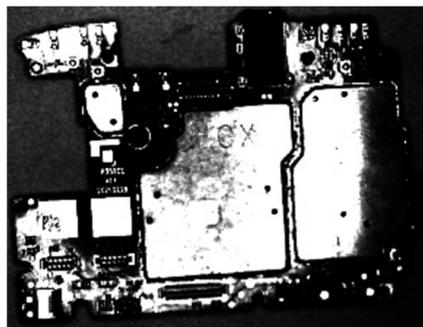
Figura 42 - Foto da placa após extração do plano de cor cinza



Fonte: Próprio autor

- Depois é feito um aprimoramento de contraste para destacar ainda mais esses componentes em relação ao resto da placa. Figura 43

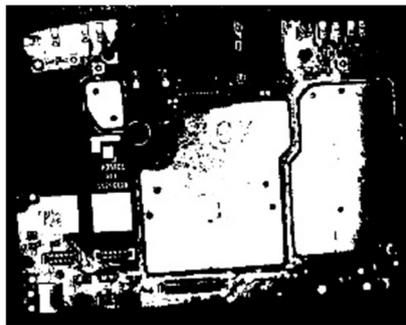
Figura 43 - Foto da placa após aprimoramento do contraste



Fonte: Próprio autor

- Por último é feita uma binarização da imagem. Figura 44

Figura 44 - Foto da placa após binarização



Fonte: Próprio autor

4.4.1.3 Processamento para referenciamento do berço de teste

A seguir foram feitos testes com diversas configurações do sistema de visão do programa de programação do robô com o objetivo de referenciar e calibrar os berços onde a placa será colocada para ser testada.

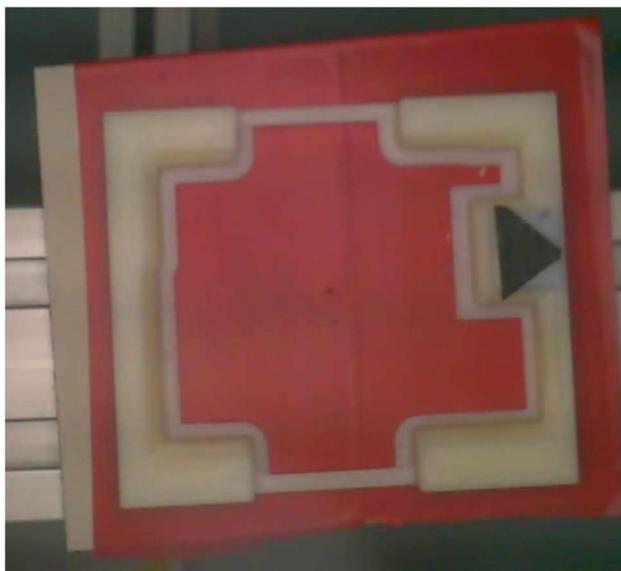
Para os berços de encaixe foram projetados berços impressos em impressora 3D no formato no formato da pcb utilizada nos testes para simular o final com o objetivo de validar o método de calibração.

Concluiu-se que para um melhor processamento de imagem o ideal seria existir um contraste notável entre a forma que se deseja encontrar e o entorno. Para isso, o berço foi impresso vazado e uma fita vermelha foi grudada na base para gerar esse contraste.

Para o processamento do berço foi utilizado o modo “*Servoing*”, esse foi um processo que demanda um pouco mais de tempo de processamento do robô, na escala de milissegundos, mas que não afeta muito o tempo de ciclo por ser um processo que só é necessário na inicialização do sistema. Para o processamento de imagem foram usados os seguintes processos:

- Primeiramente o robô captura uma foto do berço Figura 45

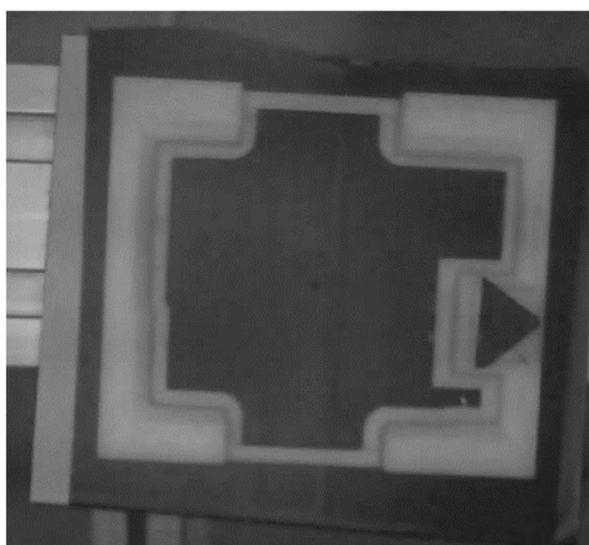
Figura 45 - Foto do berço antes do processamento



Fonte: Próprio autor

- Essa imagem então passa por um processo de retirada do plano de cor verde para destacar mais ainda o fundo vermelho que vai ser utilizados para o referenciamento. Figura 46

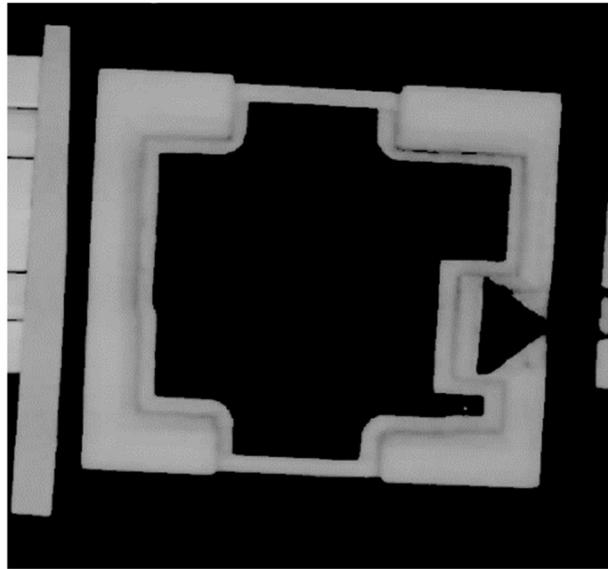
Figura 46 - Foto do berço após o processo extração do plano de cor verde



Fonte: Próprio autor

- Depois é feito um aprimoramento de contraste para destacar ainda mais esse fundo em relação ao resto do berço. Figura 47

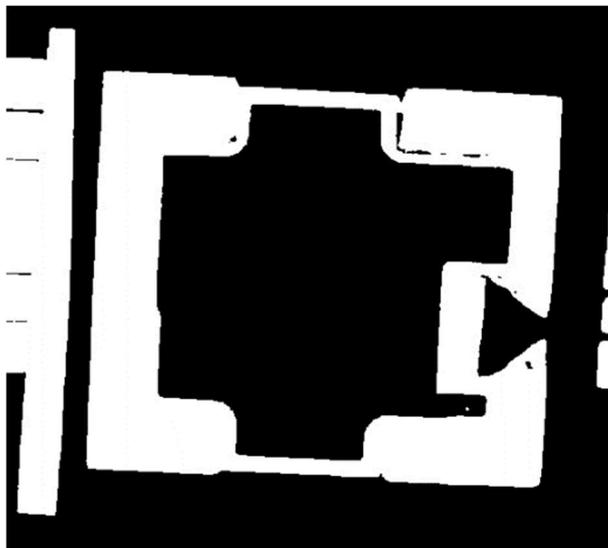
Figura 47 - Foto do berço após o processo de aprimoramento de contraste



Fonte: Próprio autor

- Por último é feita uma binarização da imagem. Figura 48

Figura 48 - Foto do berço após o processo de binarização



Fonte: Próprio autor

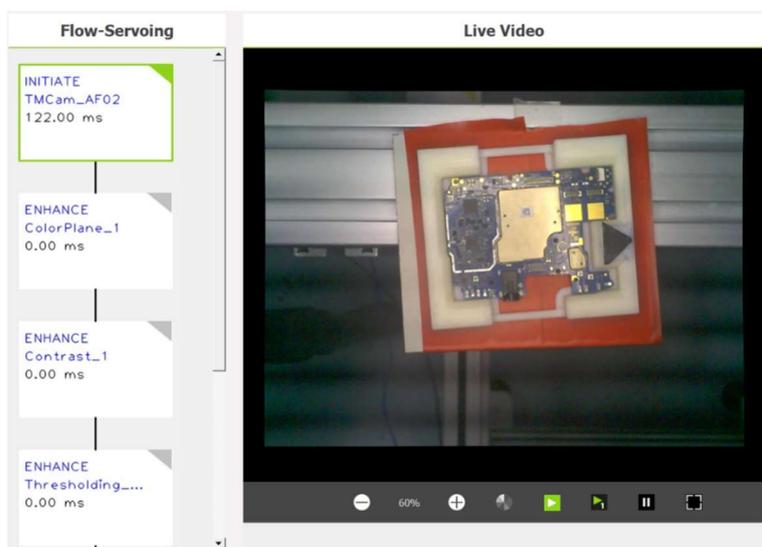
4.4.1.4 Processamento de calibração cruzada

Foi também elaborado um processo com objetivo de relacionar as informações obtidas com o sistema de visão do robô sobre o berço de placas

(local de deixa das placas) e a própria placa (objeto alvo da pega). Esse processo foi nomeado “Calibração cruzada”.

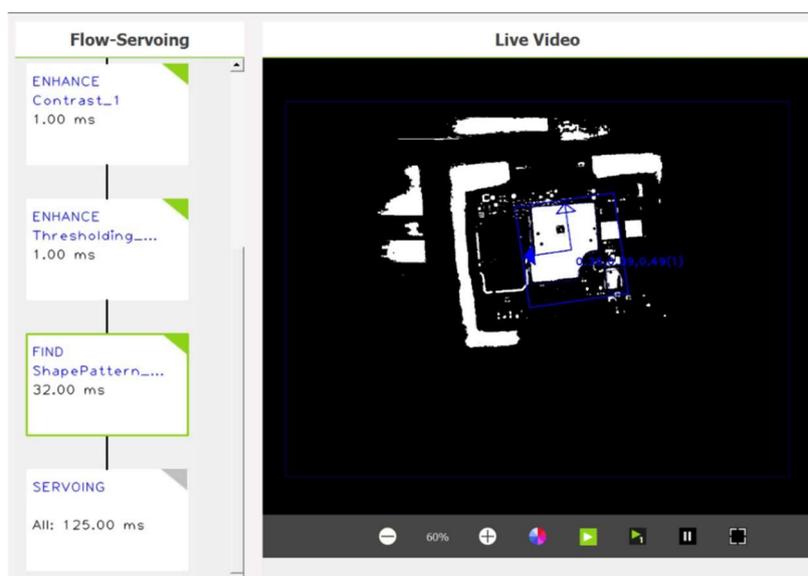
Primeiramente é tirada uma foto da placa colocada no berço Figura 49 onde ela será encaixada criando um sistema de coordenadas relativo ao componente selecionado da placa no sistema de visão. Para isso é usado o mesmo processamento de imagem utilizado para identificar a placa na entrada do sistema como visto na Figura 50.

Figura 49 – Placa no berço antes do processamento do sistema de visão



Fonte: Próprio autor

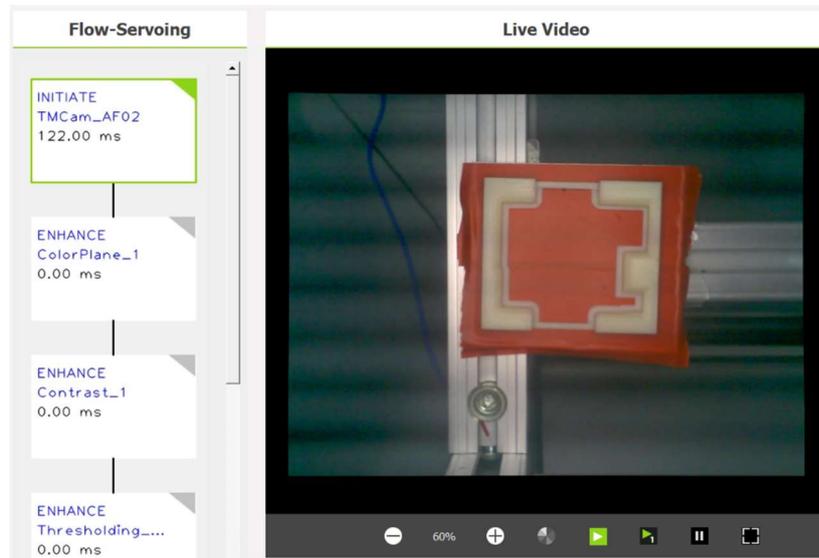
Figura 50 – Foto da placa no berço após o processamento do sistema de visão



Fonte: Próprio autor

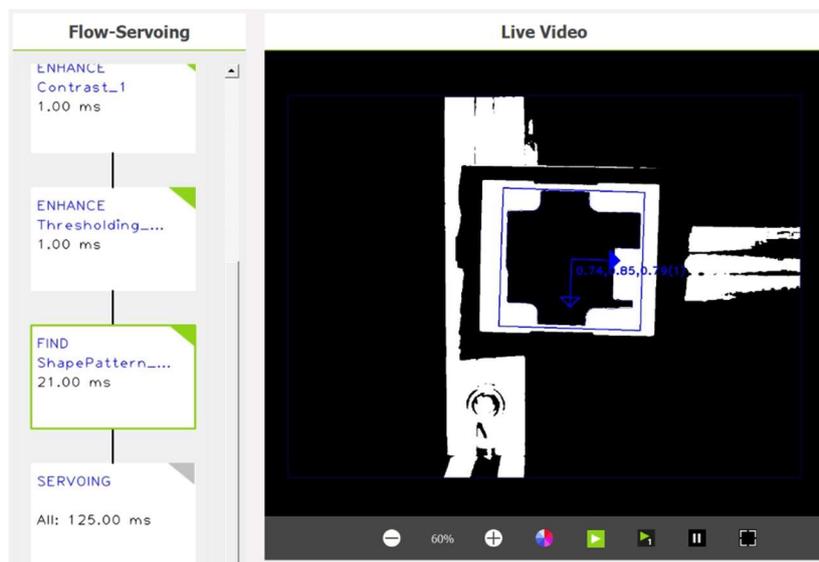
Em seguida uma foto do berço vazio é tirada para se criar um segundo sistema de coordenadas Figura 51, agora em relação ao berço de encaixe. Então é feito o processamento de imagem para criar o sistema de coordenadas desse berço Figura 52

Figura 51 – Foto do berço antes do processamento do sistema de visão



Fonte: Próprio autor

Figura 52 – Foto do berço após o processamento do sistema de visão



Fonte: Próprio autor

Por fim o robô é levado até um ponto da placa seguro para a pega e esse mesmo ponto é criado em relação os dois sistemas de coordenadas servindo assim tanto de ponto de pega quanto de deixa para a placa em berços calibrados com visão. O ponto criado em relação a placa é o utilizado para pega na entrada do sistema e o ponto criado em relação ao sistema de coordenadas do berço é utilizado para a deixa da placa no berço.

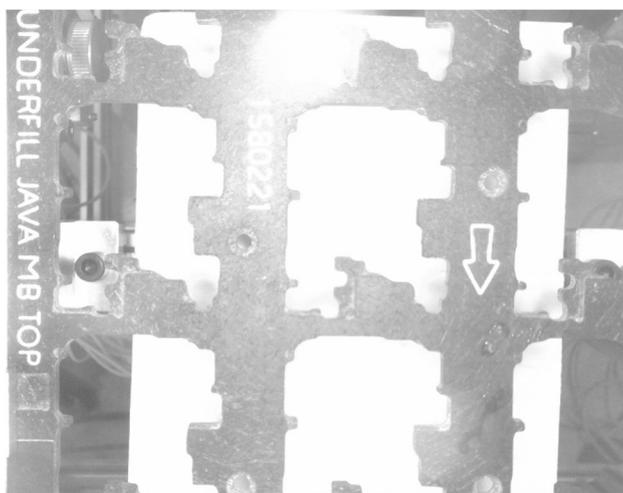
4.4.1.5 Processamento para referenciamento do palete

O último processamento de imagem importante definido para este projeto foi o processamento de imagem para referenciamento do palete de placas. Semelhante ao processo feito para referenciamento do berço de testes esse processamento tem como objetivo referenciar o palete onde a placa será colocada, porém no fim do processo para ser enviada para a estação seguinte. Outra diferença é que esta etapa de deixa da placa no palete é feita após a placa passar pelo sistema de inversor de placas logo está com o lado oposto para cima sendo assim necessário um novo processamento de imagem da placa para que o mesmo método de deixa utilizado no berço seja realizado.

Abaixo estão os passos seguidos para o processamento do palete e para a utilização do método de calibração cruzada.

- Primeiramente o robô captura uma foto do palete Figura 53

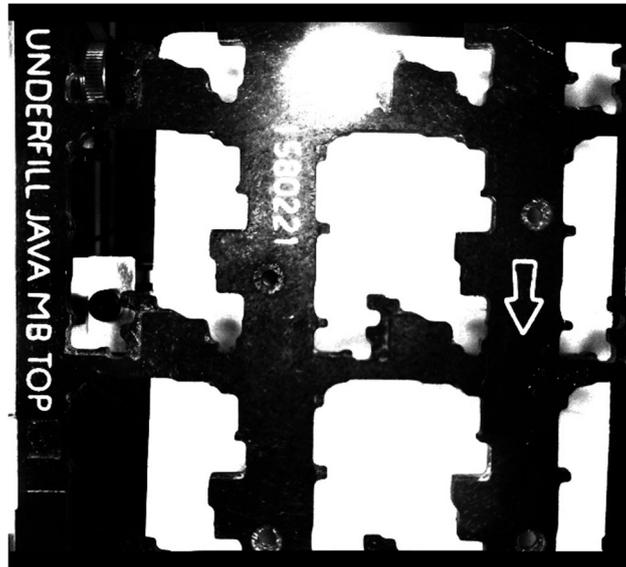
Figura 53 - Foto do palete antes de ser processada



Fonte: Próprio autor

- Depois é feito um aprimoramento de contraste para destacar a cavidade que está sendo usada para o referenciamento. Figura 54

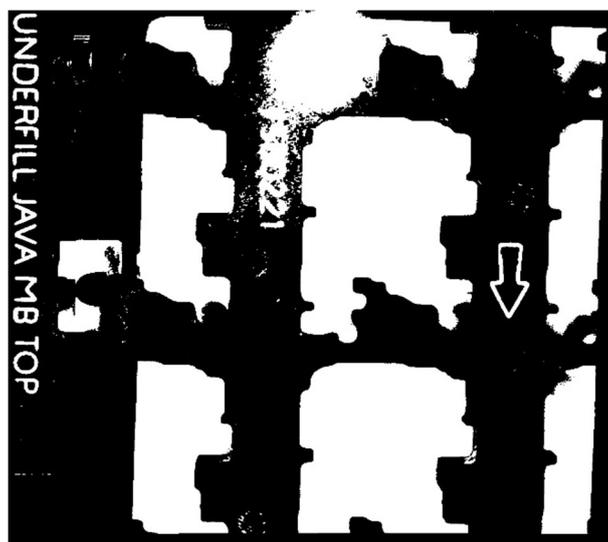
Figura 54 - Foto do palete após aprimoramento do contraste



Fonte: Próprio autor

- Por último é feita uma binarização da imagem. Figura 55

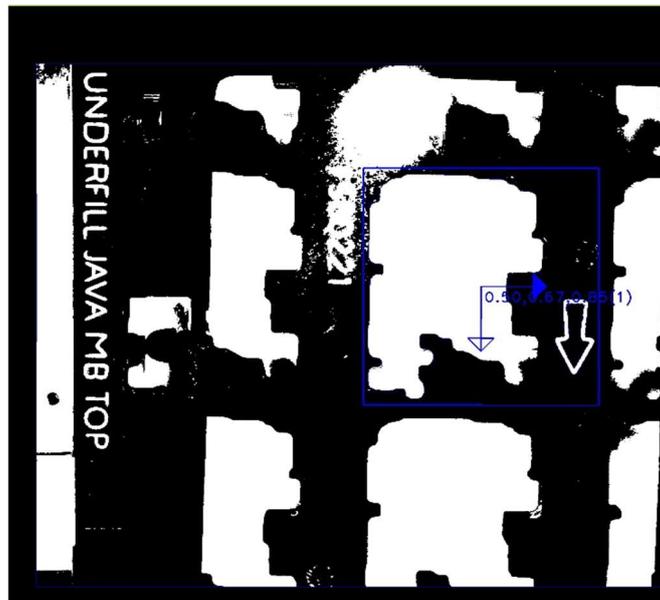
Figura 55 - Foto do palete após binarização



Fonte: Próprio autor

- Dessa maneira é possível achar com sucesso a primeira referência necessária para esse processo Figura 56

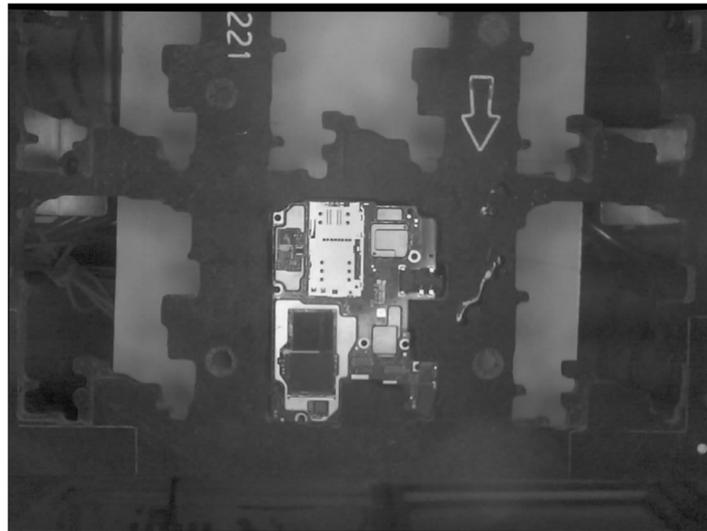
Figura 56 – Referência encontrada no palete



Fonte: Próprio autor

- Com o processamento o paleta criado, uma placa é colocada na cavidade e é feito um processamento de imagem para ela. Figura 57

Figura 57 – Placa na cavidade do paleta



Fonte: Próprio autor

- Após o processamento da placa Figura 58 é gerado um ponto utilizando o sistema de coordenadas do processamento da placa que será utilizado também para a pega da placa no berço de saída

do inversor de placas e nove pontos (um para cada cavidade do palete) no sistema de coordenadas do processamento do palete que serão utilizados para a deixa da placa no palete

Figura 58 – Processamento de imagem referenciando a placa

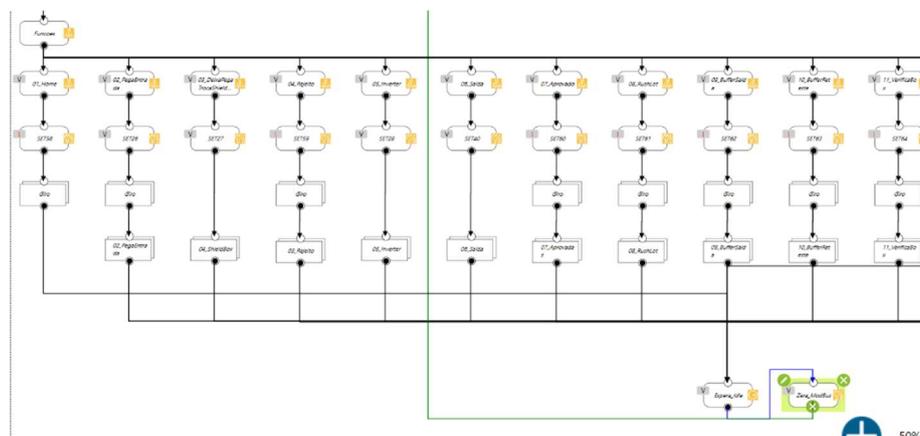


Fonte: Próprio autor

4.4.2 Programação do robô

Para a programação da rotina do robô Omron TM-12 foi utilizado o software TMFlow (Figura 59) nativo desta marca de robôs. A programação englobou toda parte de movimentação e referenciamento por visão do robô para identificar tanto a placa que será manipulada quanto os pontos onde ela será colocada (Testadores, Inversor de placas, palete de saída, buffers etc.). Foram previstos na programação do robô as seguintes funções:

Figura 59 – Exemplo de programação no software TMFlow



Fonte: Próprio autor

- Home: Leva o robô para uma posição de segurança após iniciar o processo ou após ficar muito tempo ocioso.
- Pega Placa Entrada: O robô recebe a informação do controle que existe 1 ou 2 placas posicionadas na entrada do processo e com qual ferramenta cada placa será pega, liga iluminação, utiliza a câmera embarcada para fazer uma foto da placa, se posicionar em um ponto predefinido utilizando a foto como referência para sempre se posicionar no mesmo ponto da placa, avança a ferramenta selecionada para a pega da placa, liga o vácuo e recua a ferramenta finalizando assim a pega. Caso exista uma segunda placa o processo se repete, caso não exista o robô segue para um ponto de segurança e espera pela próxima função.
- Pega/Deixa Placa Box: O robô recebe do controlador informação de qual ou quais testadores e qual ou quais ferramentas vai utilizar para deixar ou pegar placas. Está testador pode se encontrar do lado direito ou do lado esquerdo do robô, por este motivo sempre antes de ir até a testador selecionada o robô tem que estar em um ponto de segurança onde possa girar para qualquer um dos lados e tomar uma decisão baseada no ângulo da junta J1 (junta da base do robô) e no ângulo de destino para poder girar na direção correta. No caso de somente deixa, o robô leva a placa que já está na sua ferramenta até o testador previamente calibrada com visão (função calibra box), avança a ferramenta selecionada pelo controle, solta do vácuo, sobe para uma posição em que pode se movimentar sem colisão e retorna para uma posição de segurança. No caso de somente pega de placa, o robô vai até o testador selecionada com uma ferramenta vazia, avança a ferramenta, aciona o vácuo, sobe a ferramenta com a placa, sobe para uma posição segura e retorna para o ponto seguro. Nessa função é possível fazer de maneira combinada os processos de pega e deixa da placa num mesmo testador fazendo assim uma “troca”. Por causa do projeto da ferramenta, com 3 pontas com ventosas, é possível fazer até 2 deixas, 2 pegas ou 2 trocas.

- **Rejeito:** O robô entra nessa função com 1 ou 2 placas nas suas ferramentas e recebe do controlador a informação em qual das 3 está a placa ou placas que devem ser posicionadas na esteira de rejeito. O robô então vai até o ponto da esteira onde as placas devem ser posicionadas, avança com a ferramenta, solta o vácuo e recua a ferramenta deixando assim a placa na esteira de rejeito, por fim o robô volta para uma posição de segurança. No caso se mais uma placa precisar ser posicionada, o processo se repete. O robô manda um sinal para o controlador para travar a esteira enquanto as placas estiverem sendo colocadas.
- **Pega/Deixa Placa Inversor de placas:** O robô leva placas que já foram testadas nos testadores para o inversor de placas. O sistema de controle envia qual ferramenta vai deixar as placas em quais posições (posição 1 ou 2) e quais ferramentas pegarão quais placas invertidas do outro lado do inversor. O robô vai até o ponto de deixa do inversor previamente calibrado com visão (ver calibra inversor de placas), avança a ferramenta selecionada encaixando a placa no berço do inversor, solta o vácuo, recua a ferramenta e volta para a posição de segurança. Em caso de 2 placas as 2 são posicionadas antes de o robô voltar para a posição de segurança. Caso exista placas para pegar em seguida o robô vai até a posição de pega das placas invertidas também calibrada com visão, avança a ferramenta necessária para pegar a placa, liga o vácuo, recua a ferramenta e sobe para o ponto seguro. Caso 2 placas precisem ser pegadas o processo se repete antes de o robô se posicionar no ponto de segurança.
- **Saída:** Com as placas já invertidas o robô avança até o ponto onde está o módulo de saída em palete do sistema. O controlador envia para esta função quando ferramenta será utilizada para qual posição do palete (posições de 1 a 9) e a informação se aquele palete está calibrado ou não. Caso o palete tenha acabado de ser posicionado no módulo de saída em palete é necessário fazer uma calibração antes de começar a alimentá-lo com placa (ver calibra palete). Com o palete calibrado o robô vai até a posição enviada

pelo controlador com a ferramenta selecionada, avança a ferramenta encaixando a placa na posição do palete, desliga o vácuo, recua a ferramenta e retorna para a posição de segurança. Caso existam 2 placas para serem posicionadas, o processo se repete antes de o robô retornar para a posição de segurança.

- **Aprovado:** O robô vai até a esteira de placas aprovadas. O controle envia para esta função qual placa será pega com qual ferramenta. Está função é semelhante a função de pegar placas na entrada, porém as placas que são colocadas nessa esteira são placas que já passaram pelo processo de teste e não irão para testadores nem inversão de placas e sim direto para a saída.
- **Buffer Saída:** Está função é utilizada para deixar placas no caso de indisponibilidade de palete. A placa é colocada num berço e é retirada e colocada no palete de saída (função Saída) quando existir disponibilidade de palete.
- **Calibra Testador:** Sistema de controle envia para o programa do robô quais racks devem ser calibrados. O robô percorre todas as boxes de todos os racks selecionados pelo sistema de controle com a iluminação ligada e tira fotos para referenciar os pontos onde as placas serão posteriormente colocadas. Essa calibração deve ser feita sempre que um rack for movido por qualquer razão.
- **Calibração Saída:** Sistema de controle indica para o programa do robô que o palete atualmente no módulo de saída de paletes é um palete novo. Então, essa função faz uma foto do palete com o a iluminação ligada para criar um referenciamento para as placas que serão colocadas neste palete. Essa calibração deve ser feita sempre que um novo palete chega no módulo de saída de palete.
- **Calibração de Inversor de placas:** O robô percorre com a iluminação ligada os 2 berços onde a placa é deixada pelo robô (vazios) e os 2 berços onde o robô pega as placas depois de invertidas (com placas) tirando fotos para o referenciamento dos pontos de deixa nos berços e pontos de pega das placas. Essa

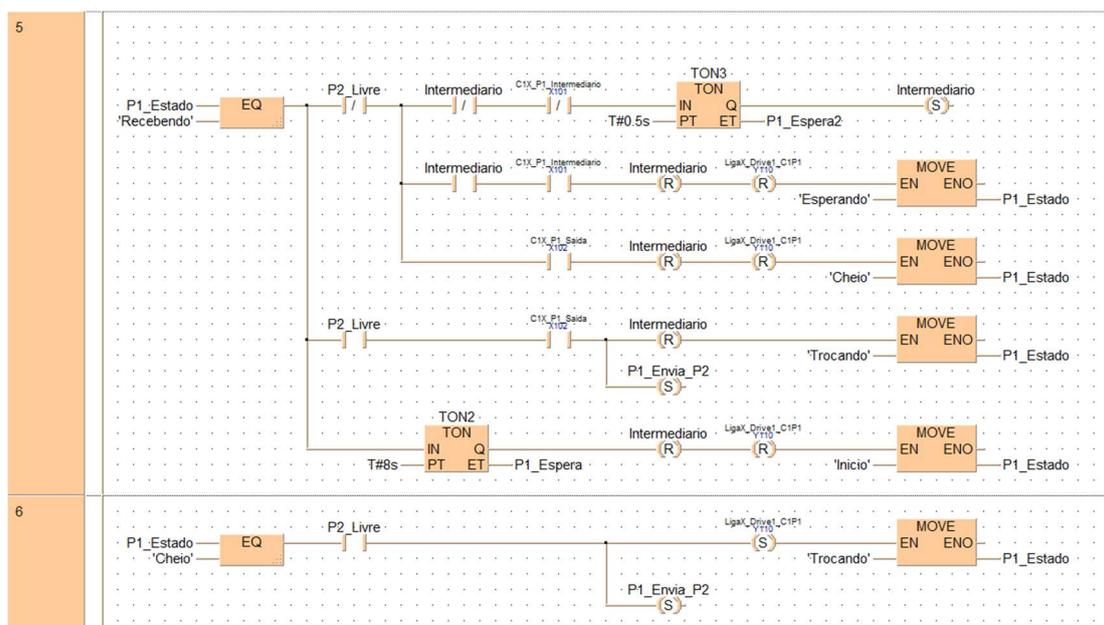
calibração é feita somente em caso de necessidade de retirada do inversor de placas para manutenção ou troca de produto.

- Calibração Cruzada Box: Esta função tem como objetivo utilizar a referência gerada pela foto tirada pelo robô da placa e a referência gerada pela foto do fixture do testador e gerar uma relação entre essas duas fotos, podendo assim utilizar com mais precisão o sistema de visão.
- Calibração Cruzada do Palete: Calibração cruzada utilizada para o paleta.
- Seletor de produto: Função para trocar o produto sendo pilotado pela máquina. Programa do robô recebe do sistema de controle um novo valor representando um produto diferente.

4.4.3 Programação do CLP

Para as esteiras de Entrada, Placas Aprovadas e Rejeito e para o Inversor de placas foi utilizado o programa FPWin Pro 7 Figura 60 para a programação do CLP Panasonic que faz o controle dessas esteiras e inversor de placas. A linguagem utilizada foi LADDER.

Figura 60 – Exemplo de programação em ladder no programa FPWin Pro 7



Fonte: Próprio autor

As funções controladas por esse CLP são:

- Esteira de Entrada de placas: Nesta esteira uma placa é posicionada na entrada e levada pelo conveyor 1 até a posição 1 do conveyor 2 onde pode ser manipulada pelo robô. No caso da entrada de uma segunda placa, esta é levada até a posição 1 do conveyor 2 pelo conveyor 1 colocando assim a primeira placa na posição 2. Se mais placas forem colocadas elas ficam em espera no conveyor 1 até que o conveyor 2 seja liberado.
- Esteira de Placas Aprovadas: Nesta esteira placas que já foram testadas em outros postos (de maneira manual ou em módulos automáticos) passa adiante para serem paletizadas.
- Esteira de placas Rejeitadas: Esta esteira é onde são colocadas as placas que foram rejeitadas pelo teste ou que estão fora da rota do e-BT. O robô coloca uma ou duas placas no conveyor 1 desta esteira que leva elas até o conveyor 2 e até a saída do módulo para serem tratadas.

4.4.4 Comunicação com o sistema supervisório

4.4.4.1 Comunicação com o robô

A programação das rotinas do robô foi feita com base no software *TMFlow* e funciona juntamente com o sistema supervisório em *labview*. Para esta comunicação foi estabelecido um padrão de comunicação através do protocolo *Modbus* e são utilizadas 5 variáveis:

- 9000 – Comando: O supervisório manipula essa variável, quando em 1, o robô deve iniciar a execução da tarefa descrita na variável
- 9001 – Estado
 - 0 – Idle
 - 1 – Start
 - 2 – Reset
 - 3 – Re-arme.

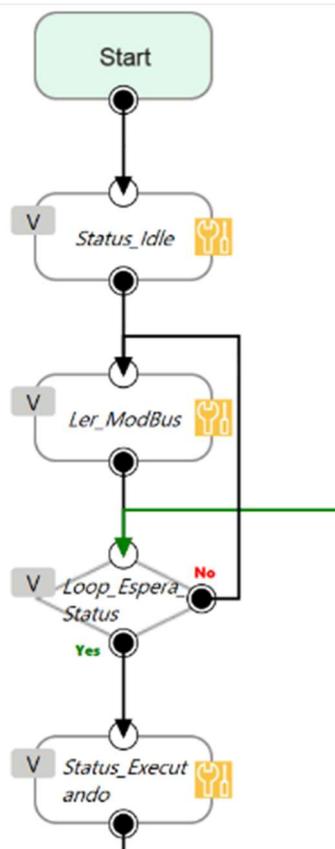
- 9001 – Tarefa_Control: O supervisor manipula essa variável, ela descreve qual tarefa o robô deve executar quando o 9000 – Comando for para 1 – Start
- 9100 – Status_Robô: O Robô manipula essa variável, possui vários significados:
 - 0 – Idle
 - 1 – Robô executando a tarefa
 - 2 – Robô Finalizou a execução da tarefa sem erro.
 - 3 – Robô Finalizou a execução da Tarefa com erro.
 - 4 – Robô em Emergência
- 9101 – Tarefa_Robô: O Robô manipula essa variável, ela descreve qual tarefa o robô está executando ou a última tarefa executada
- 9102 – Status_Erro: O Robô manipula essa variável, ela descreve qual erro o robô tem ou o erro da tarefa (esse último varia por aplicação)
 - 0 – Sem erro
 - 1 - Emergência
 - 2 - Erro Tarefa

Esta comunicação funciona da seguinte maneira: Quando a Variável 9100 – Status_Robô, estiver como 0 e a Variável 9101 – Tarefa_Robô, estiver como 0, o supervisor pode solicitar a execução de uma nova tarefa pelo robô, para tanto:

- O supervisor escreve na variável “9001 – Tarefa_Control” qual tarefa o robô deve executar
- O supervisor monitora a variável “9101 – Tarefa_Robô” até que ela tenha o mesmo valor que a variável “9001 – Tarefa_Control”
- O supervisor escreve na variável “9000 – Comando” o valor 1 Indicando para o robô iniciar o sistema

- O robô escreve na variável “9100 – Status_Robô” o valor 1, indicando que o robô está executando a tarefa desejada
- O robô escreve na variável “9100 – Status_Robô” o valor 2, indicando que o robô finalizou a tarefa sem erro.
- O supervisor escreve na variável “9000 – Comando” o valor 0 indicando para que o robô volte para o estado de pronto para receber um novo start
- O robô escreve na variável “9100 – Status” o valor 0, indicando que o robô está no estado de pronto para receber um novo start

Figura 61 – Início do programa do robô



Fonte: Próprio autor

No loop da

Figura 61 é possível ver pela perspectiva do programa do robô o funcionamento da comunicação. Em Status_Idle o robô determina que a variável 9100 é 0, em Ler_Modbus e Loop_Espera_Status o robô segue lendo no Modbus

as variáveis necessárias para executar cada função (9004 a 9009) e compara as variáveis 9001 e 9101. Assim que estas variáveis são iguais o robô segue e em Status_Executando muda a variável 9100 para 1 e ao concluir a função retorna 2 para esta mesma variável indicando que a função foi concluída com sucesso. Por fim todas as variáveis utilizadas do *Modbus* são resetadas para seus valores iniciais e se reinicia o ciclo.

4.4.4.2 Comunicação com o CLP

Semelhante ao que foi feito para o robô, a comunicação do sistema supervisório com o CLP é feita também através do protocolo de comunicação *Modbus*. No caso do CLP as variáveis do *Modbus* são utilizadas para informar ao supervisório certos estados dos sistemas controlados pelo CLP e para que o supervisório possa informar o CLP de certas ações que precisam ser tomadas.

Para a esteira de entrada de placas as variáveis utilizadas foram:

- Parâmetros de Entrada:
 - R10 – Esvaziar Solicita
 - R11 – Bloqueia Esteira
- Parâmetros de Saída:
 - R50 – Operando
 - R51 - Travado
 - R52 – Esvaziando
 - R53 – Tem uma placa
 - R54 – Tem duas placas

Para a esteira de rejeito as variáveis foram:

- Parâmetros de Entrada:
 - R14 – Esvaziar Solicita
 - R15 – Bloqueia Esteira
- Parâmetros de Saída:
 - R60 – Operando
 - R61 - Travado
 - R62 – Esvaziando
 - R63– Livre

E para o sistema de inversão de placas as variáveis foram:

- Parâmetros de Entrada:
 - R16 – Esvaziar Solicita
 - R17 – Bloqueia Esteira
- Parâmetros de Saída:
 - R64 – Operando
 - R65 - Travado
 - R66 – Girando
 - R67 – Uma placa Entrada
 - R68 – Duas placas Entrada
 - R69 – Uma Placa Saída
 - R70 – Duas Placas Saída

5. Resultados e discussões

O objetivo desse projeto foi a automatização de parte do processo de produção de placas de circuito impresso para telefones celulares. Essa automatização visava a diminuição de custos de produção reduzindo a quantidade de recursos humanos necessários para produção dessas placas, melhoria na qualidade do processo eliminando possíveis falhas operacionais causadas por humanos, diminuição na quantidade de intervenções necessárias por técnicos de manutenção e paradas processo para intervenções por manutenção.

Utilizando ferramentas de desenvolvimento de projeto foi possível desenvolver uma automação viável para validação do conceito e para utilização prática para o objetivo desejado.

Uma das maiores dificuldades encontradas no desenvolvimento do projeto foi a elaboração da metodologia de encaixe das placas no berço de testes e do palete.

Com o objetivo de validar esta metodologia e avaliar o encaixe das placas foi realizado uma bateria de testes de repetição de encaixa.

Na primeira rodada de testes foram feitas 200 repetições do processo de encaixe e se obteve o resultado visto na Figura 64.

Figura 62 - Teste com a primeira calibração



Fonte: Próprio autor

Após o primeiro o primeiro resultado os testes foram repetidos fazendo pequenos ajustes no processamento de imagem e uma nova calibração e os resultados foram os vistos na Figura 65.

Figura 63 - Testes após segunda calibração



Fonte: Próprio autor

Com todos os testes para definição da programação do robô, CLP e processamento de imagem concluídos. O projeto passou para a fase de validação onde todas as definições foram testadas com uma produção controlada com objetivo de identificar possíveis erros e corrigi-los. Nessa fase também foi onde o desempenho do projeto foi testado para avaliar as vantagens da sua aplicação no processo produtivo. Para isso foi desenvolvida uma tabela para registrar todas as peças processadas pela máquina e todas as falhas encontradas durante esse processo.

Tabela 1 - Quantidade de placas produzidas e falhas durante as semanas de validação

	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
Produção	200	356	828	898	909	331
Falhas	9	25	10	23	0	0

Fonte: Próprio autor

Na Tabela 1 foi registrado o número de placas que foram produzidas pela máquina no decorrer das semanas em que foi implementada a etapa de validação de engenharia. É possível observar a quantidade de erros encontrados

zerando nas últimas semanas após esses erros serem identificados e corrigidos seja por correções na programação ou por ajustes no projeto.

Tabela 2 - 5 erros com maior recorrência

Problemas	Recorrência
Erro de encaixe	15
Falha do posicionamento da placa entrada	8
Falha de posicionamento da placa na esteira de rejeito	8
Falha de identificação da placa no berço do inversor	6
Track Id da placa não foi lido pelo leitor	5

Fonte: Próprio autor

A Tabela 2 mostra os cinco erros que foram mais frequentes durante essa fase de validação. Para corrigi-los foram tomadas as seguintes ações:

- Erro de encaixe: Para esse erro duas ações foram tomadas
 - Melhoria do sistema de trava e fixação dos racks de teste
 - Melhoria do processamento de imagem para calibração
- Falha do posicionamento da placa entrada: para esse erro foram tomadas 3 ações
 - Ajuste do sensor para que a placa parasse com mais precisão e repetibilidade
 - Ajuste do ponto onde o robô captura da imagem da placa
 - Elaboração de um pré-processamento da imagem da placa para que o robô a encontrasse com mais precisão
- Falha de posicionamento da placa na esteira de rejeito: Para esse erro duas ações foram tomadas
 - Ajuste do ponto de deixa da placa na esteira de rejeito
 - Ajuste do sensor para que a placa fosse identificada com maior precisão
- Falha de identificação da placa no berço do inversor: Para esse erro uma ação foi tomada
 - Ajuste da posição do sensor do berço
- Track Id da placa não foi lido pelo leitor: Para esse erro uma ação foi tomada
 - Ajuste de parâmetros do leitor

Com o objetivo de validar o processo desenvolvido e avaliar o encaixe das placas foi realizado uma bateria de testes de repetição de encaixa.

Na primeira rodada de testes foram feitas 200 repetições do processo de encaixe e se obteve o resultado visto na Figura 64.

Figura 64 - Teste com a primeira calibração



Fonte: Próprio autor

Após o primeiro o primeiro resultado os testes foram repetidos fazendo pequenos ajustes no processamento de imagem e uma nova calibração e os resultados foram os vistos na Figura 65.

Figura 65 - Testes após segunda calibração



Fonte: Próprio autor

Para validar a metodologia de encaixe das placas no palete também foi realizada uma bateria de testes de repetição de encaixe semelhante ao feito para o encaixe da placa no berço de teste.

Foram feitas 100 repetições do processo de encaixe e se obteve o resultado visto na Figura 66.

Figura 66 – Teste de encaixe da placa no palete



Fonte: Próprio autor

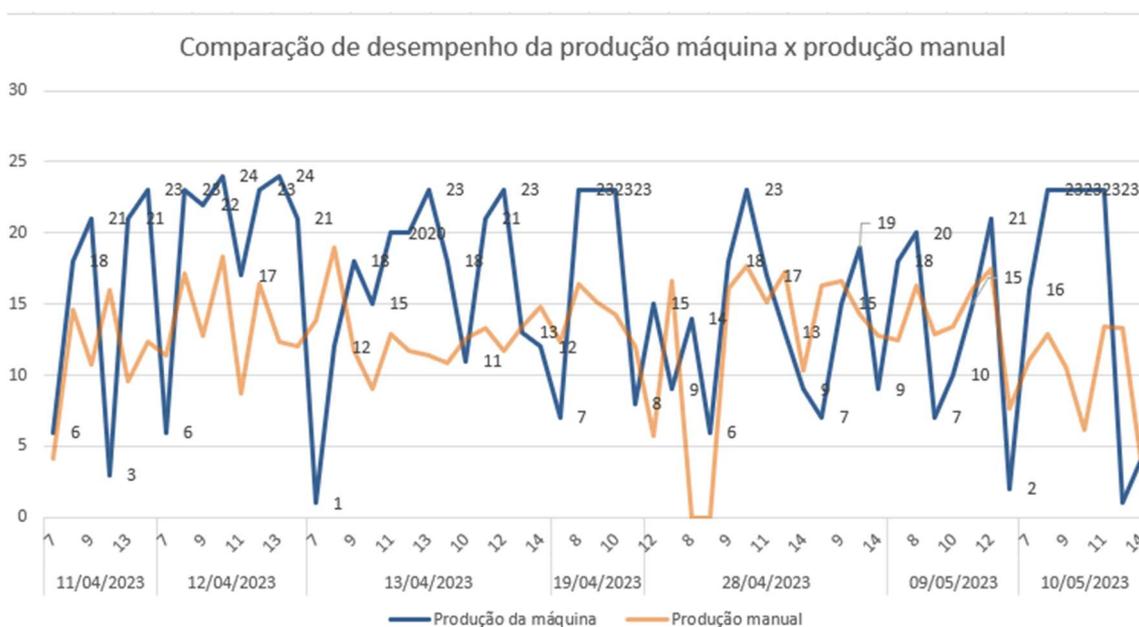
6. Conclusão

Em conclusão, o desenvolvimento deste projeto mostrou um resultado muito positivo, destacando-se integração entre o processamento digital de imagem e a programação de robôs para melhoramento do encaixe de itens em pontos com pouca folga e sem a necessidade de uma grande indexação.

Através de uma abordagem estruturada, foi possível validar e ajustar o sistema para ultrapassar barreiras como o encaixe das placas em diferentes cenários, garantindo que as metas de produtividade e eficiência fossem alcançadas.

Foi obtido ao fim do projeto o desempenho produtivo da máquina. Para ilustrar este desempenho foi elaborado o gráfico presente na Figura 67. Esse gráfico foi gerado utilizando as horas de melhor performance da máquina e mais produção manual da fábrica. Para essa validação foram utilizadas somente dois testadores para teste das placas.

Figura 67 – Comparação de desempenho da produção máquina x produção manual



Fonte: Próprio autor

Nesse gráfico é possível avaliar o desempenho da máquina ultrapassa o desempenho da produção manual em vários momentos. Nesse estudo foi determinado que o robô consegue nas melhores condições chegar a uma

produção média de 15,4 placa por hora por testador enquanto a média da produção manual é somente 12 placas por hora por testador gerando assim um incremento de aproximadamente 28% na produção de placa.

Um dos objetivos principais do projeto foi a diminuição de custos de produção. O custo total do projeto foi de aproximadamente R\$650.000,00 considerando custos dos componentes utilizados para montagem mais um custo previsto para a replicação. Considerando esse valor e a diminuição de recursos humanos causada pela automação o retorno do investimento nesta automação viria após aproximadamente 2,5 anos.

6.1 Trabalhos futuros

Como trabalhos futuros para serem desenvolvidos usando estes como ponto de partida pode-se sugerir os seguintes:

- Estudo do melhor tipo de processamento de imagem para utilização em aplicações que necessitem fazer referenciamento de placas pcb. Nesse projeto foi desenvolvido um método baseado em testes, mas com um estudo mais aprofundado e utilização de outras ferramentas é possível desenvolver uma metodologia mais eficiente para se descobrir com mais rapidez os melhores parâmetros a serem utilizados para chegar no objetivo que se deseja.
- Estudo de métodos de calibração de berços para encaixe de placas pcb para diminuir a necessidade de indexação. O método utilizado nesse projeto pode ser mais desenvolvido e implementado se estudado com mais profundidade focando principalmente na melhor maneira de se relacionar dois sistemas de coordenadas gerados por visão.
- Projeto de automação de outros processos que se comuniquem com este desenvolvido neste trabalho. Outros pontos da fabricação de placas de circuito podem também ser automatizados para aumentar ainda mais a qualidade e eliminar dos recursos humanos a necessidade de realização de tarefas simples e repetitivas.

- Projeto de um sistema de alimentação automático para eliminar a necessidade de inserção manual na automação. Um ponto negativo deste projeto descrito neste trabalho é o fato de que da maneira que foi desenvolvido ainda existe a necessidade de inserção manual uma a uma das placas serem manipuladas na entrada da célula robótica. Sendo assim é sugerido o projeto de uma automação que faça com que esse processo de inserção venha de maneira automática ao inserir em uma máquina a ser desenvolvida uma quantidade maior de placas.

7. Referencial bibliográfico

CRAIG, John J e PRENTICE, Pearson e HALL, Pearson Prentice. **Introduction to Robotics Mechanics and Control Third Edition..** [S.l: s.n.], 2005.

EUSTÁQUIO, José e DE QUEIROZ, Rangel e GOMES, Herman Martins. **Introdução ao Processamento Digital de Imagens..** [S.l: s.n.], [S.d.].

FOROUZAN, Behrouz A. **Comunicação de dados e rede de computadores.** [S.l: s.n.], 2006. v. 3 ed.

FRANCHI, Claiton Moro e DE CAMARGO, Valter Luis Arlindo. **Controladores Controladores Lógicos Lógicos Programáveis Programáveis Sistemas Discretos.** [S.l: s.n.], 2008. v. 1.

FRANK D. PETRUZELLA. **Controladores Lógicos Programáveis.** [S.l: s.n.], 2014. v. 4° Ed.

FRASCÁ, Greicy Echiley e BRUNO, Diego Renan. **AUTOMAÇÃO: introdução a robótica industrial.** Revista Interface Tecnológica, v. 18, n. 2, p. 639–651, 20 Dez 2021. Disponível em: <<https://revista.fatectq.edu.br/interfacetecnologica/article/view/1244>>. Acesso em: 20 ago 2023.

GONZALEZ, Rafael e WOODS, Richard. **Digital Image Processing.** [S.l: s.n.], [S.d.].

LOZANO-PEREZ, Tomas. **Robot Programming.** PROCEEDINGS OF THE IEEE. [S.l: s.n.], 1983.

M D ANTUNES, João Pedro. **Programação de robôs industriais em operações de maquinagem.** 2015. Universidade do Porto, Porto, 2015.

MODBUS ORGANIZATION. **Modbus FAQ.** Disponível em: <<https://modbus.org/faq.php>>. Acesso em: 13 jul 2024.

NOGUEIRA DO NASCIMENTO, Sergio. **FUNDAÇÃO UNIVERSIDADE FEDERAL DE RONDÔNIA-UNIR NÚCLEO DE CIÊNCIAS SOCIAIS APLICADAS-NUCSA PROGRAMA DE PÓS-GRADUAÇÃO MESTRADO EM ADMINISTRAÇÃO...** [S.l: s.n.], 2021.

PAN, Zengxi e colab. **Recent progress on programming methods for industrial robots.** Robotics and Computer-Integrated Manufacturing. [S.l: s.n.], Abr 2012

PEREIRA, Antônio e colab. **UMA RÁPIDA ANÁLISE SOBRE AUTOMAÇÃO INDUSTRIAL.** [S.d.]. Acesso em: 6 mar 2023.

RIBEIRO, Marco Antônio. **Automação Industrial 4 a edição...** [S.l: s.n.], 1997.

ROGGIA, Leandro e CARDOZO FUENTES, Rodrigo. **Automação Industrial.** Brasil: E-Tec, 2016.. Acesso em: 6 mar 2023.

STEVENS, W. Richards. **TCP/IP Illustrated**. Volume 1 ed. [S.l: s.n.], 1994. v. 1.

WENDLING, Marcelo. **Sensores**. Universidade Estadual Paulista: [s.n.], 2010. .
Acesso em: 19 mar 2023.

APÊNDICE A – Link do GitHub com programas do CLP e Cobot TM-12

<https://github.com/alpinto/eBT>