

UNIVERSIDADE DE SÃO PAULO

Fábio Jacober Werlang

Aplicação de conceitos de indústria 4.0 em tarefas com robôs industriais

SÃO CARLOS
2018

FÁBIO JACOB WERLANG

Aplicação de conceitos de indústria 4.0 em tarefas com robôs industriais

Trabalho de conclusão de curso
apresentado ao Departamento de
Engenharia Elétrica e de Computação
para obtenção do título de Bacharelado
em Engenharia Elétrica

Área de Concentração: Robótica em
Nuvem

Orientador: Prof. Dr. Glaucio Augusto
de Paula Caurin

SÃO CARLOS
2018

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

W489a	Werlang, Fabio Jacober Aplicação de conceitos de indústria 4.0 em tarefas com robôs industriais / Fabio Jacober Werlang; orientador Glauco Augusto Caurin. São Carlos, 2018. Monografia (Graduação em Engenharia Elétrica com ênfase em Sistemas de Energia e Automação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2018. 1. Indústria 4.0. 2. Automação. 3. Internet das coisas. 4. Computação em nuvem. 5. Sistemas embarcados. I. Título.
-------	---

FOLHA DE APROVAÇÃO

Nome: Fabio Jacober Werlang

Título: "Aplicação de conceitos de indústria 4.0 em tarefas com robôs industriais"

Trabalho de Conclusão de Curso defendido e aprovado
em 26 / 11 / 2018,

com NOTA 60 (Seis, Zero), pela Comissão Julgadora:

Prof. Titular Glauco Augusto de Paula Caurin - Orientador - SAA/EESC/USP

Prof. Associado Evandro Luis Linhari Rodrigues - SEL/EESC/USP

Mestre Henrique Borges Garcia - Doutorando - SEM/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

Resumo

O setor industrial e a pesquisa sobre indústria 4.0 tem crescido muito nos últimos anos, visando a integração de dispositivos em todas as instâncias fabris. Contudo, muitas plataformas dependem do hardware implementado em chão de fábrica para fazer sua integração com a nuvem. Tendo em vista que a maior parcela dos robôs industriais implementados na indústria não possui comunicação com softwares de internet das coisas, este trabalho busca desenvolver um equipamento capaz de manter a comunicação em tempo real do robô, garantindo sua execução de tarefa enquanto insere os dados obtidos em sistemas de computação em nuvem. Diante disso, o hardware na aplicação é capaz de manter baixos os tempos de comunicação, sem gastar processamento com planejamento e tratamento dos dados obtidos. A sua implementação se dará integrando a placa de desenvolvimento, da fabricante Toradex, com RTOS (sistema operacional em tempo real) para controle do robô Kuka modelo KR16. O software desenvolvido funciona corretamente em computadores, mas seu funcionamento em sistema embarcado ainda precisa de ajustes para completar seu funcionamento com o robô. Com o desenvolvimento desse projeto pode-se concluir que é possível fazer a integração de robôs ou equipamentos à rede e colher os benefícios desta usando uma estrutura de implementação rápida e eficiente.

Palavras chave: Indústria 4.0; Automação; Internet das coisas; Computação em nuvem, Sistemas embarcados

Abstract

The industrial sector and research on industry 4.0 has grown greatly in recent years, aiming the integration of devices in all manufacturing instances. However, many platforms rely on the hardware deployed on the shop floor to integrate with the cloud. Since most of the industrial robots implemented in the industry do not have communication with internet of things software, this work seeks to develop an equipment capable of maintaining real-time communication of the robot, ensuring its execution of task, while inserting the data obtained in cloud computing systems. With this, the hardware in the application is able to keep down the communication times, without spending processing with planning and processing of the obtained data. Its implementation will be implemented by the development board of the manufacturer Toradex, with RTOS (real-time operating system) to control the robot Kuka model KR16. The software developed works correctly on computers, but its operation on the embedded system still needs adjustments to complete its integration with the robot. With the development of this project it can be concluded that it is possible to integrate robots or equipment into the network and reap the benefits of this integration using a fast and efficient implementation structure.

Keywords: Industry 4.0; Automation; Internet of things; Cloud computing; embedded systems

Lista de Figuras

Figura 1: Pilares da indústria 4.0	12
Figura 2: Funcionamento da indústria 4.0 no ciclo de um produto.	15
Figura 3: Piramide de complexidade de coordenar linha de produção.	16
Figura 4: Arquitetura dos cloud robots.	20
Figura 5: Logo do RTOS analisados.....	22
Figura 6: Análise do modo de operação. a) PLC + Computador. b) Sistema embarcado.	23
Figura 7: Kuka KR16 no Laboratório de Mecatrônica da EESC-USP.....	26
Figura 8: Colibri Evaluation Board V3.2B.....	27
Figura 9: Colibri T20 512MB V1.2A.	28
Figura 10: Orçamento Toradex.....	28

Lista de abreviaturas e siglas

CPS	Cyber Physical System
EUR	Euro
GHz	GIGA-HERTZ
IoS	Internet Of Service
IoT	Internet das Coisas
IP	Protocolo de Internet
IPv4	Protocolo de Internet Versão 4
OTG	On The Go
PIB	Produto Interno Bruto
PLC	Controlador Lógico Programável
RAM	Random Access Memory
RSI	Robot Sensor Interface
RTOS	Real Time Operating Systems
SDK	Software Development Kit
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
USB	Universal Serial Bus
XML	Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO	9
2. OBJETIVO	11
2.1 Objetivo Geral	11
2.2 Objetivo Específico.....	11
3. REVISÃO BIBLIOGRÁFICA	12
3.1 Indústria 4.0.....	12
3.2 Sistemas em nuvem	17
3.3 Sistemas Embarcados	22
4. MATERIAIS e MÉTODOS	26
4.1 Equipamentos	26
4.1.1 Manipulador Robótico	26
4.1.2 Placa de desenvolvimento	27
4.1.3 Computador	29
4.2.1 Configuração da Placa de desenvolvimento	29
4.2.3 Comunicação com o robô	31
4.2.4 Desenvolvimento do Software	32
5. RESULTADOS	35
6. CONCLUSÃO	36
REFERÊNCIAS BIBLIOGRÁFICAS.....	37
Apêndice A - XML recebido do robô pelo computador	40
Apêndice B - XML recebido do robô pela placa de desenvolvimento.....	41
Apêndice C - XML enviado para o robô pelo computador.....	42
Apêndice D - XML enviado para o robô pela placa de desenvolvimento	43
Apêndice E - XML modelo de recebimento.....	44
Apêndice F - XML modelo para envio.....	44
Apêndice G – Fluxograma do código	45

1. INTRODUÇÃO

A indústria, nos últimos séculos, passou por transformações na sua forma de atuação gerando inovação e profundas mudanças sociais e econômicas. Para Azevedo (2017), a Primeira Revolução Industrial aconteceu entre 1760 e 1840, sendo marcada pela construção de ferrovias e de máquinas a vapor, bem como as locomotivas. A Segunda Revolução Industrial, que iniciou a partir da segunda metade do século XIX em busca do aprimoramento contínuo, se utilizou da descoberta de energias fósseis e do eletromagnetismo, utilizando a eletricidade como alternativa do vapor. A utilização de tais recursos impulsionou a aceleração do ritmo industrial, descobrindo-se assim, o potencial de reduzir os custos e o tempo de fabricação de produtos, os quais poderiam ser consumidos em escala cada vez maior. Um dos exemplos mais evidentes dessa revolução é a introdução da linha de montagem, tendo como referência tradicional o automóvel Ford T.

A Terceira Revolução Industrial teve início em meados da década de 1970. Para Atzori et al. (2010), Kaitan e McCalley (2015) citados por Liao et al. (2018) essa revolução ficou conhecida pelo seu uso intenso de eletrônicos e da Tecnologia da Informação (TI) para se atingir o objetivo de automação da manufatura. Na área industrial, os sistemas *Supervisory Control and Data Acquisition* (SCADA - Controle Supervisório de Aquisição de Dados), em conjunto com o controlador lógico programável (PLC), constituem o seu maior expoente, sendo que tais sistemas atualmente convergem para sistemas abertos e, em alguns casos, conectados à rede corporativa ou, até mesmo, à internet. A utilização de recursos de telecomunicação e o avanço tecnológico possibilitaram o acesso remoto, compartilhamento, integração e, conseqüentemente, o processamento de dados à distância. A integração com os diversos sistemas visa ao aumento da produtividade e da eficiência na tomada de decisões.

Atualmente, segundo Schwab (2016), vive-se o limiar de uma Quarta Revolução Industrial, a chamada Revolução Industrial baseada no conceito de sistemas Cyber Physical Systems (CPS - ciberfísicos). Além dos sistemas ciberfísicos, surgem novas tecnologias que, através do seu uso, promovem o rompimento do atual paradigma, destacando-se: Internet das Coisas (IoT), análise de *big data* e computação em nuvem que visam atender aos requerimentos de um sistema de manufatura avançada, promovendo a integração de toda uma cadeia de suprimentos.

Atualmente, de acordo com o artigo da RTInsights (2017), 62% dos fabricantes americanos dependem de papel e caneta para acompanhar os estágios do processo de

fabricação, o que é considerado muito ineficiente e suscetível a erros humanos. Devido a iminência das fábricas estarem mais conectadas espera-se que esse número caia para 25% até 2022.

Nas indústrias, para Clickatell (2017), uma maior eficiência significa maiores lucros, se atualmente a fabricação de produtos é atrasada devido a demora da comunicação entre as etapas produtivas, podemos dizer que existe ociosidade. Fábricas conectadas manterão isso a um mínimo absoluto através do monitoramento constante de equipamentos, além de permitir que as falhas ou avarias do equipamento sejam antecipadas. Em um mundo competitivo e capitalista, em especial a área industrial, a otimização de recursos e aumento na eficiência da fábrica são vitais para sua prosperidade.

Este projeto tem por finalidade desenvolver uma estrutura, que possa ser adaptada a diferentes robôs e equipamentos, capaz de conectar robôs a internet ou a uma rede de fábrica e também viabilizar que sejam implementadas aplicações na nuvem.

2. OBJETIVO

2.1 Objetivo Geral

A proposta deste trabalho é integrar robôs da geração passada às novas tecnologias sem a necessidade de um alto investimento financeiro, instalando um dispositivo ao robô que o permita enviar e receber dados à nuvem em que outros equipamentos da fábrica estão conectados e, a partir dessa comunicação, viabilizar um novo estudo que permite o robô a tomar ações independentemente ou ser controlado remotamente por um operador.

2.2 Objetivo Específico

- Estudar na literatura arquiteturas usadas na indústria 4.0 para a integração de máquinas e robôs.
- Devolver uma estrutura genérica, para servir de base para que seja possível sua replicação em outros robôs e equipamentos.
- Elaborar uma comunicação estável e segura utilizando protocolo UDP entre o hardware dedicado e robô.
- Evoluir o *software* em linguagem C# para uma versão compatível com o hardware dedicado. O hardware também será responsável por todo o processamento e manipulação dos dados do sistema.
- Executar *cross-compile* do notebook ao hardware dedicado.

3. REVISÃO BIBLIOGRÁFICA

3.1 Indústria 4.0

A Indústria 4.0 é marcada por processos de automação e digitalização altamente desenvolvidos, além do uso de tecnologias eletrônicas e de TI na fabricação de produtos e serviços (figura 1). Segundo Lu (2017), a integração e análise em massa de dados em tempo real otimizam recursos no processo de fabricação e alcançam um melhor desempenho. IoT, big data e computação na nuvem são as principais tecnologias. Em particular a computação na nuvem fornece dados e serviços poderosos e precisos para a indústria 4.0 por integrar redes industriais de IoT.

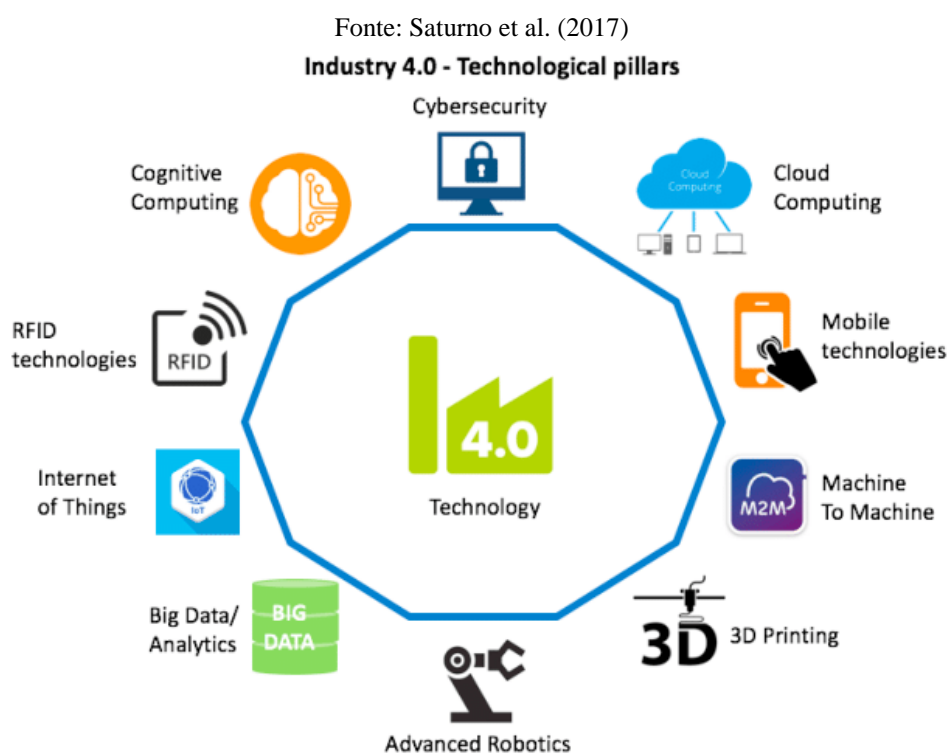


Figura 1: Pilares da indústria 4.0

Para Gubbi et al. (2013), o IoT é sobre uma evolução da internet conectando equipamentos entre-si. Esta nova evolução é caracterizada pelo uso intensivo de tecnologias digitais com o intuito de não mais apenas extrair informações do ambiente. Com o uso de diversos tipos de sensores sua evolução é focada em atuar com o mundo físico, executando operações de comando, controle e atuação, permitindo até mesmo a fabricação de produtos

personalizados de forma rápida e ágil, gerando otimização em tempo real da produção e da cadeia de suprimentos.

O desenvolvimento da IoT está associado ao desenvolvimento da Indústria 4.0 e para LU (2017), IoT é a tendência e direção para a nova revolução industrial, ao conectar humanos e máquinas, ela transfere e integra dados entre organizações externas e internas. Ao facilitar a informação, a IoT melhora a eficiência e a eficácia do desenvolvimento e gerenciamento de conhecimento na indústria 4.0 e, com isso, mudarão o relacionamento entre clientes, produtores e fornecedores. As decisões de produção não serão dominadas por fabricantes e varejistas. Em vez disso, a IoT e a Indústria 4.0 tornam os clientes mais envolvidos nas decisões sobre qualidade e personalização de produtos.

A Indústria 4.0 é caracterizada por uma conexão sem precedentes via internet ou outros meios distribuídos e os chamados Sistemas ciberfísicos (CPS), que podem ser considerados sistemas que unem o mundo físico e o mundo virtual (AKANMU; ANUMBA, 2015 apud HOFMANN; RÜSCH, 2017). Mais precisamente, sistemas ciber-físicos são integrações de computação com processos físicos. Computadores e redes incorporados monitoram e controlam os processos físicos, geralmente com *feedback loops* onde os processos físicos afetam os cálculos e vice. No contexto de fabricação, isso significa que as informações relacionadas ao chão-de-fábrica e ao espaço computacional virtual são altamente sincronizadas, permitindo um novo grau de controle, vigilância, transparência e eficiência no processo de produção.

É importante entender, segundo Hofmann e Rüsch (2017) que não apenas os processos de produção, mas também os papéis dos funcionários devem mudar drasticamente. Esperam-se que os funcionários tenham mais responsabilidade, sejam tomadores de decisão e assumam tarefas de supervisão, em vez de dirigir empilhadeiras, por exemplo. No mesmo contexto, alguns críticos apontaram recentemente que a natureza automatizada e auto-reguladora da fábrica inteligente pode causar uma grave destruição de emprego. No entanto, quase nenhum estudo confiável apóia esse medo.

Como um CPS complexo, a IoT integra vários dispositivos equipados com recursos de detecção, identificação, processamento, comunicação e rede. Um sistema IoT consiste em Redes Sem Fio Industriais (IWN) e Internet das Coisas (IoT), incluindo máquinas, equipamentos e redes à nuvem (LU, 2017). Um sistema IoT é capaz de oferecer produtos específicos e personalizados permitindo, por exemplo, que os usuários possam personalizar produtos através de páginas da web. Em seguida, os servidores da Web transmitem dados para a nuvem industrial e as instalações por meio de redes com ou sem fio. Com base nos

dados recebidos, o fabricante integrará o design, otimizará, gerenciará e monitorará o processo de produção para produzir produtos com eficiência. Com a ajuda da auto-otimização e mecanismo autônomo de tomada de decisão, máquinas e equipamentos irão colaborar para melhorar o desempenho (ROBLEK; MESKO; KRAPEZ, 2016 apud LU, 2017). Como a fabricação e o fornecimento são dinâmicos, o ciclo de vida de um produto também é variável e, de acordo com as mudanças, a descentralização, a auto-otimização e a automação podem ajudar esse processo dinâmico de forma mais eficiente e eficaz.

O desenvolvimento da indústria é um processo integrado de complexidade e agilidade entre humanos e máquinas. A indústria 4.0 aumenta a digitalização da manufatura com CPS, na qual redes conectadas de humanos e robôs interagem e trabalham em conjunto com informações compartilhadas e analisadas, apoiadas por big data e computação em nuvem ao longo de cadeias de valor industriais inteiras. A Indústria 4.0 aumenta a eficiência de custo e tempo e melhora a qualidade do produto, associada às tecnologias, métodos e ferramentas. Como resultado, ela acelerará a indústria para alcançar níveis sem precedentes de eficiência operacional e crescimento da produtividade (LU, 2017).

No ambiente fabril, sensores sempre foram parte essencial da fábrica, compondo a parte de automação, controle de temperatura e mais. A integração generalizada desses sensores e de outros equipamentos gera quantidades massivas de vários dados. Esses sensores e equipamentos serão conectados à nuvem que, por sua vez, está encarregada do processamento desses dados, podendo se utilizar de softwares de inteligência artificial (IA) para gerar os resultados desejados desses dados. Dessa forma, os hardwares de fábrica não ficarão mais sobrecarregados com o processamento de dados e também aumenta a flexibilidade de fazer alterações no seu processamento de dados (LU, 2017).

A indústria 4.0 torna as fábricas mais inteligentes, flexíveis e dinâmicas ao equipar a produção com sensores, atuadores e sistemas autônomos. Assim, máquinas e equipamentos atingirão altos níveis de otimização e automação (figura 2). Além disso, o processo de fabricação tem a capacidade de cumprir padrões e requisitos mais complexos e exigentes nos produtos (ROBLEK; MESKO; KRAPEZ, 2016 apud LU, 2017).

Fonte: Pisching (2018)



Figura 2: Funcionamento da indústria 4.0 no ciclo de um produto.

A adaptabilidade, a eficiência dos recursos e a integração dos processos de oferta e demanda são aprimorados na Indústria 4.0, portanto, fábricas, produção, cidades e possíveis equipamentos e objetos inteligentes tornam-se inteligentes (VARGHESE; TANDUR, 2014, p. 634-638 apud LU, 2017). O termo “inteligente” é usado para se referir a aplicações da Indústria 4.0 na literatura.

Para Brettel et al. (2014), com a revolução e as indústrias cada vez mais seguindo filosofia da produção enxuta, objetivando estoques reduzidos e produtos mais personalizados, junto disso a redução desperdícios ao longo da cadeia produtiva. A internet, na indústria 4.0 é identificada como uma ferramenta poderosa para auxiliar nessa transformação.

Além das vantagens citadas, mais dispositivos conectados à rede reduz a complexidade de coordenar uma linha produção que, por sua vez, aumenta o nível de automação e, conseqüentemente, demanda menos operações humanas (figura 3).

Fonte: Brettel et al. (2014)

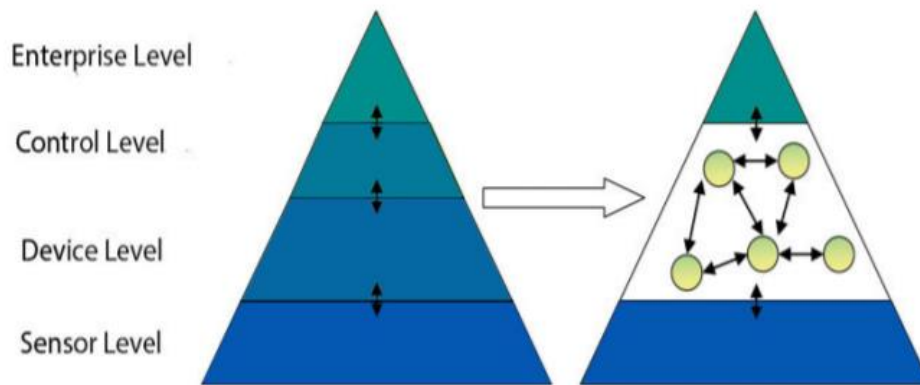


Figura 3: Pirâmide de complexidade de coordenar linha de produção.

Para Kagermann et al. (2012), espera-se que nações industriais fortes, como a Alemanha, continuarão sendo bem sucedidas apenas se conseguirem participar ativamente da iniciativa Indústria 4.0. As oportunidades e benefícios que se espera vir junto com a Indústria 4.0 são variadas e podem resultar, por exemplo, em produção em massa altamente flexível, coordenação em tempo real da produção e maior valor agregado ao produto, possibilitando também o surgimento de serviços e modelos de negócios inteiramente novos.

Segundo Dujin (2014) o setor industrial desempenha um papel crucial na Europa, servindo como um motor essencial do crescimento econômico e na criação de empregos, representando 75% de todas as exportações e 80% de todas as inovações. No entanto, a paisagem industrial europeia é dupla. A Europa perdeu cerca de 10% de sua participação industrial nos últimos 20 anos, enquanto os países emergentes conseguiram dobrar sua participação, passando a corresponder a 40% da produção global. Há alguns anos, a Alemanha começou a pensar em iniciativas para manter e até fomentar seu papel de "precursora" no setor industrial.

Eventualmente, para Liao (2018), o termo Indústria 4.0 foi introduzido publicamente na Feira de Hanôver, Alemanha, em 2011, representando parte da estratégia de alta tecnologia da Alemanha para preparar e fortalecer o setor industrial com relação aos requisitos de produção futura.

Poucos anos depois, o termo indústria 4.0 já trazia consigo grandes expectativas, Rubmann, et al. (2015) fez uma estimativa do potencial da transformação da indústria 4.0, com ganhos em 4 áreas que as indústrias de manufaturas Alemãs se beneficiarão.

1. Produtividade: Nos próximos 5 - 10 anos a indústria 4.0 vai estar presente em mais companhias, aumentando a produtividade da manufatura das indústrias alemãs entre EUR 90 - 150 bilhões. Excluindo o custo do material, que varia entre 15% a 25%. Considerando isso espera-se atingir ganhos entre 5% e 8% na produtividade. Esses aumentos variam entre cada indústria, a indústria de componentes de manufatura são as que mais se beneficiam, com ganhos entre 20% e 30%, por exemplo a indústria automotiva pode esperar ganhos entre 10% e 20%.
2. Aumento do Lucro: Vai levar a um aumento no lucro de EUR 30 bilhões de lucro, ou aproximadamente 1% do PIB Alemão.
3. Empregos: Levará ao aumento de 6% de empregos nos próximos 10 anos. Mas novas habilidades serão necessárias, trabalhos com baixo nível técnico reduzirão ao mesmo tempo que haverá aumento de vagas nas áreas de Tecnologia de Informação e mecatrônica.
4. Investimentos: A adaptação das indústrias na incorporação da indústria 4.0 custará em torno de EUR 250 bilhões nos próximos 10 anos.

3.2 Sistemas em nuvem

Existem várias definições sobre a *cloud manufacturing* (CBM), mas somente algumas tem ampla aceitação, Wu (2014) comenta a respeito de 3 definições entendidas por CBM.

1. É uma solução orientada para computação e serviços, um modelo de fabricação desenvolvido a partir de modelos de manufatura como por exemplo, *application service provider* (aplicação do provedor de serviço), *lean manufacturing* (fabricação ágil), *web manufacturing* (fabricação em rede), *manufacturing grid* (fabricação em grade) e tecnologias sob o suporte da computação em nuvem, Internet das coisas (IoT), virtualização e tecnologias de computação.

2. É um modelo para habilitar, conforme necessidade, acesso à rede que possui amplos recursos para a demanda formando uma rede de compartilhamento de recursos de fabricação configuráveis (por exemplo, ferramentas de software, equipamentos de fabricação e capacidades de fabricação) que podem ser rapidamente configuradas e liberadas com o mínimo esforço de gerenciamento ou esforço do fornecedor.

3. Refere-se a um modelo de desenvolvimento de produtos orientado a serviços no qual os consumidores podem configurar produtos ou serviços, bem como reconfigurar sistemas de fabricação através de Infraestrutura como Serviço (IaaS), Plataforma como serviço (PaaS), Hardware como serviço (HaaS) e Software como serviço (SaaS) em resposta rápida necessidade de mudanças do cliente. CBM é caracterizado por ser *on demand*, podendo ser acessado de qualquer lugar pela rede e também possui escalabilidade rápida.

Já segundo Tao et al. (2011) ela se desenvolveu a partir de modelos de manufatura avançada e tecnologias de informação corporativas sob o suporte da computação em nuvem, IoT, virtualização e tecnologias orientadas a serviços e avançadas tecnologias de computação. Seu objetivo é realizar o compartilhamento e a circulação total de informações e infraestrutura, permitindo a sua alta utilização e uso sob demanda dos recursos de fabricação, fornecendo serviços de fabricação seguros e confiáveis, de alta qualidade, baratos e sob demanda para todo o ciclo de vida do produto na fábrica.

Para Thames (2016), o CBM pode ser descrito como um modelo de fabricação em rede pois pode possuir demanda compartilhada de recursos de manufatura e distribuídos para formar linhas de produção ciber-física temporárias e reconfiguráveis que aumentam a eficiência, reduzem os custos do ciclo de vida do produto e permitem a melhor alocação de recursos em resposta à tarefa gerada pelo cliente de demanda variável

Comparado com os modelos de fabricação existentes, o CBM possui as seguintes características típicas.

1. É orientada a serviços e requisitos. A maioria dos modelos de manufatura é orientada a recursos ou pedidos, enquanto o CBM é um modelo de manufatura orientado a serviços e requisitos. No CBM, os recursos e a capacidade de fabricação são virtualizados e encapsulados no serviço de fabricação de nuvem, e podem ser usados por todos os usuários de acordo com suas necessidades por meio da Internet;
2. Dinâmica. Os recursos e serviços no CBM são diversos e dinâmicos, e as soluções para uma tarefa de manufatura também necessitam ser dinâmicas;
3. Iniciativa. Em um sistema CBM, os serviços de fabricação em nuvem e as tarefas de fabricação (ou requisitos) estão ativos; eles podem automaticamente encontrar e combinar um com o outro;
4. Fisicamente distribuído e logicamente centralizado. O recurso e a capacidade de fabricação física do CBM são localizados em locais diferentes e são controlados por diferentes pessoas ou organizações;

5. Grupo de fabricação baseada em inovação. Qualquer pessoa, instituição ou empresa pode participar e contribuir com seus recursos de fabricação, habilidades e conhecimento para uma plataforma de serviços CBM. Enquanto isso, qualquer empresa pode realizar suas ações de fabricação com base nesses recursos de fabricação, habilidades e conhecimento. Semelhante à Wikipedia, o CBM é um modelo de fabricação baseado em inovação em grupo;
6. Limiar inferior e terceirização. Dentro de um ambiente CBM, uma empresa não precisa possuir todo o ambiente de fabricação de hardware (como oficina, equipamentos, infraestruturas de TI e pessoal) ou a capacidade de fabricação de software (como design, fabricação, gerenciamento e capacidade de vendas). Pode alugar ou invocar os recursos e habilidades e serviços na plataforma CBM.

A nuvem também tem o potencial de trazer grandes melhorias para os robôs e a automação, surgindo os termos *Cloud Robot* e *Cloud Automation*, para Kehoe (2015) esses sistemas podem ser amplamente definidos da seguinte maneira: Qualquer robô ou sistema de automação que se baseie em dados ou código da nuvem para suportar sua operação, ou seja, onde nem toda a detecção, computação e memória são integradas em um único sistema independente. Um exemplo é o carro autônomo do Google que indexa mapas e imagens coletadas atualizadas por satélite, e utiliza *street view*, e *crowdsourcing* da nuvem para melhorar a precisão da localização.

Hu, Tay e Wen (2012) ilustram como deve funcionar a arquitetura de comunicação dos *cloud robots* na figura 4. Segundo eles, sua arquitetura é organizada em duas partes complementares, sendo a comunicação entre robôs (M2M) e a comunicação dos robôs com a nuvem (M2C). Na comunicação M2M eles se comunicam por redes sem fio formando uma computação colaborativa trazendo diversas vantagens.

Fonte: Hu, Tay e Wen (2012).

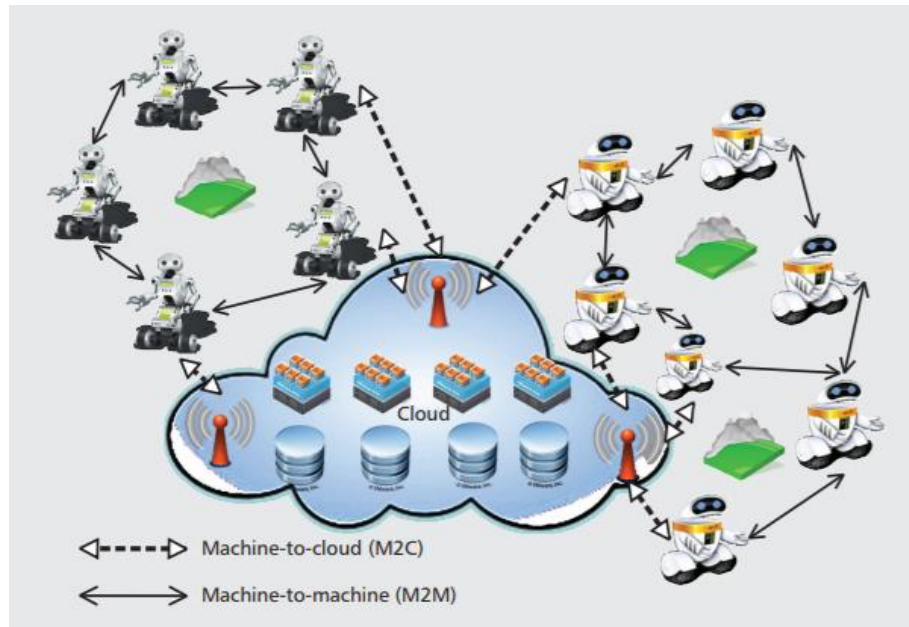


Figura 4: Arquitetura dos *cloud robots*.

A primeira é que os robôs dentro da rede podem repartir o processamento de funções que compõe uma tarefa a fim de reduzir seu tempo de execução, dessa forma a capacidade de computação dos robôs são somadas, em segundo a informação pode ser compartilhada entre os robôs para que a tomada de decisão seja feita após a análise de vários robôs, dando maior segurança. Por último, ela também permite que robôs que estão fora do alcance do acesso da nuvem receber ou enviar informações a ela, pois se algum robô da rede M2M está conectado à nuvem este pode fazer o encaminhamento das informações.

Na comunicação M2C a nuvem fornece diversos recursos de computação e armazenamento compartilhados que podem ser alocados dependendo da demanda em tempo real. Este modelo de computação elástica permite ao grupo de robôs da rede descarregar

tarefas para execução remota, livrando os robôs em fábrica de se ocupar com esse processamento.

Além disso, os benefícios de um grande volume de armazenamento fornecido pela nuvem centralizada são dois. Em primeiro lugar ele pode unificar um grande volume de informações sobre o ambiente, que pode ser organizado em um formato utilizável por robôs. Em segundo lugar, ela pode fornecer uma extensa biblioteca de habilidades ou comportamentos que estão relacionados com os requisitos da tarefa e complexidades, tornando possível que os robôs aprendam com a história de todos robôs conectados à nuvem que já fizeram tarefas semelhantes.

Futuras aplicações robóticas se beneficiarão da *cloud robotics*. Hu, Tay e Wen (2012) listam as seguintes vantagens que obterão sobre redes tradicionais.

1. Capacidade de transferir tarefas com uso intensivo de computação para a nuvem. Os robôs só precisam manter sensores, atuadores e nível de processamento básico para permitir ações em tempo real. Como o software da nuvem pode ser atualizado independentemente de atualizações no robô a sua configuração é facilitada;
2. Acesso a grandes quantidades de dados. Os robôs podem adquirir informação e conhecimento para executar tarefas através de bancos de dados na nuvem sem ter que lidar com a criação e manutenção de tais dados;
3. Acesso ao conhecimento compartilhado e novas habilidades. A nuvem fornece um meio para os robôs compartilharem informações, aprender novas habilidades e trocar conhecimentos uns dos outros. A nuvem pode hospedar um banco de dados ou biblioteca de habilidades capazes de mapear diferentes requisitos de tarefas e complexidades ambientais. O projeto *RoboEarth* está tentando transformar isso em uma realidade.

3.3 Sistemas Embarcados

Conforme dito por Caurin e Aroca (2009), os sistemas operacionais em tempo real (RTOS) são especialmente projetados para atender a rigorosas restrições de tempo. Em várias situações, os RTOS estão presentes em sistemas embarcados. No mercado internacional, existem mais de uma centena de opções de RTOS para escolher, ao mesmo tempo que existem opções similares gratuitas como o sistema operacional Linux. Desta forma, a decisão sobre qual sistema usar pode ser um fator chave para o sucesso ou fracasso de um projeto, e a análise deve ser feita com imparcialidade e critérios adequados. Segundo Sohal (2001) citado por Caurin e Aroca (2009), os fatores mais importantes dos sistemas em tempo real são o pior tempo de resposta de uma tarefa e o pior tempo de resposta de uma interrupção. Tempo de resposta é o tempo total gasto entre o instante que é feita uma requisição ao sistema e o instante em que ela é atendida.

Para ter métricas a respeito do RTOS e assim ter informações para escolher a mais adequada a necessidade da aplicação, Caurin e Aroca (2009) analisaram 6 RTOS populares, sendo eles: Windows CE 6.0, QNX Neutrino, Real Time Application Interface (RTAI), Micrium μ C/OS-II, Linux, Wind River VxWorks, (figura 5) e 1 sistema operacional comum Windows XP para comparações.



Figura 5: Logo do RTOS analisados.

Todos os sistemas operacionais são analisados em 3 parâmetros: tempo de resposta, latência e *jitter*. O resultado pode ser visto na tabela 1.

Latência é a medida do tempo que um dado gasta para ir de um ponto ao outro e *jitter* é medida da variação desse tempo analisando vários envios de dados, ou seja, o tempo de latência não é estático e cada dado enviado terá um tempo único para chegar ao destino, já o

jitter mede a variação desses tempos de latência. A latência muitas vezes não impossibilita aplicações, como em áudio e vídeo, porém o *jitter* pode impossibilitar aplicação, uma vez que dados podem chegar fora de ordem.

Fonte: Caurin e Aroca (2009)

	Win XP	Win CE	Neutrino	μ C/OS-II	Linux	RTAI	VxWorks
A	200 μ s	20 μ s	20 μ s	1,92 μ s	13,89 μ s	5 μ s	3,85 μ s
B	848 μ s	99 μ s	35,2 μ s	3,2 μ s	98 μ s	11,4 μ s	13,4 μ s
C	700 μ s	88,8 μ s	32 μ s	2,32 μ s	77,6 μ s	7.01 μ s	10,4 μ s

Tabela 1 : Comparativo entre os sistemas operacionais. A: tempo de resposta, B: Latência e C: *Jitter*

Percebeu-se que, com exceção do Windows XP, que não é um RTOS, todos os sistemas estudados atendem os parâmetros temporais de maneira satisfatória. Os sistemas bem consolidados QNX Neutrino e VxWorks da Wind River realmente mostraram determinismo e confiabilidade, embora dois sistemas mais recentes que não são difundidos também tenham características promissoras: Windows CE e RTAI Linux. O Windows CE Embedded foi escrito especialmente para aplicativos críticos e, durante os testes, ele se comportou como um sistema robusto, poderoso e flexível.

Moreira, Fonseca e Tavares (2009) estudaram uma alternativa ao antigo modo de operação dos sistemas que usam controlador lógico programável (PLC) combinado com computador (figura 6.a). A sua substituição por um sistema embarcado com sistema operacional Windows CE (figura 6.b), sistemas com capacidade em tempo real são usados para superar os problemas de tempo de desenvolvimento e sistema de controle caro.

Fonte: Moreira, Fonseca e Tavares (2009)

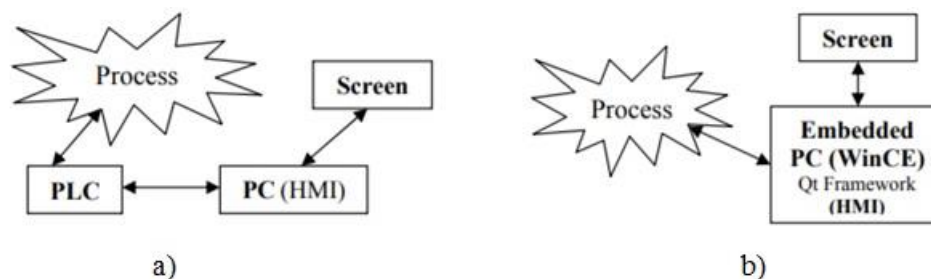


Figura 6: Análise do modo de operação. a) PLC + Computador. b) Sistema embarcado.

Segundo Jaqueson (1999), citados por Moreira, Fonseca e Tavares (2009), com a crescente adoção do Windows CE no ambiente industrial, devido à sua capacidade de interface em tempo real e fácil de usar, surgiram novas oportunidades para reduzir o tempo de desenvolvimento do sistema e, conseqüentemente, o custo do mesmo. Como os requisitos mínimos são necessários para executar o Windows CE e com características como: ativação instantânea, baixa ISR (Rotina de serviço de interrupção), 255 níveis de prioridade e baixa latência de thread, ela se torna perfeita para quase todos os processos industriais. O uso dos níveis de prioridade em um sistema multitarefa preventivo requer uma atribuição cuidadosa aos diferentes encadeamentos para evitar o impasse do aplicativo, já que o planejador fornece ao processador o encadeamento com prioridade mais alta.

Como dito anteriormente, as células industriais são normalmente controladas por um PLC e monitoramento de dados realizado por um PC. No trabalho em questão o objetivo do autor é controlar uma aparafusadora automática, constituída por três eixos e um parafuso elétrico.

As principais vantagens deste sistema são: solução mais barata que a abordagem PLC e computador. Só exige a necessidade de aprender uma linguagem de programação, no caso do autor, C ++ com Qt Framework, que reduz o tempo de desenvolvimento, se beneficiando do uso uma linguagem de alto nível, ao contrário das linguagem de programação de PLC. Este sistema pode satisfazer os requisitos de tempo da maioria das aplicações industriais, nomeadamente para este tipo de células de trabalho. É possível desenvolver diferentes módulos baseados em Circuito Inter-integrado (I2C) para atender às necessidades de outros aplicativos, tornando-se um sistema muito flexível e expansível. O sistema apresenta diferentes conexões, incluindo integração com pequenas telas sensíveis ao toque que o torna adequado para o controle do painel. A integração de rede é fácil porque a placa possui uma porta Ethernet.

Para Suseela., Kumar e Garividi (2005), os sistemas embarcados são invenções alimentadas pela ideia de fazer pré-programas para executar uma faixa restrita dedicada de funções como parte de grandes sistemas. Os sistemas embarcados geralmente têm baixo custo e estão facilmente disponíveis na prateleira para a maioria das aplicações. Eles costumam ter baixos riscos de projeto, já que é fácil verificar o design usando ferramentas que alimentam o crescimento de sistemas embarcados. Os sistemas embarcados receberam uma injeção importante como resultado de três desenvolvimentos: o primeiro foi o desenvolvimento de plataformas de tempo de execução padrão, como o java, que permitiram seu uso em inúmeras formas inimagináveis no passado. O segundo foi a união de sistemas embarcados e da

Internet, que possibilitaram a rede de vários sistemas embarcados para operar como parte de um grande sistema entre redes, e o terceiro foi o surgimento de vários ambientes de software integrados, o que simplificou as implementações desses aplicativos.

Quase todos os carros que saem da linha de produção atualmente utilizam tecnologia incorporada de uma forma ou de outra. Esses sistemas podem ser usados para implementar recursos que vão desde o ajuste da suspensão até as condições da estrada e o conteúdo de octano no combustível, até sistemas de freios antitravamento (ABS) e sistemas de segurança. Sistemas embarcados também podem tornar veículos autônomos uma realidade.

Conforme publicado em um artigo por IntervalZero RTOS (2018), usando-se do RTOS interval zero, líderes da indústria de tecnologia de montagem em superfície (SMT) Japonesa obtiveram resultados significativos e mensurados, sendo os principais resultados:

- Redução de 25% ou mais no custo geral com infra-estrutura de computadores;
- Aumento de 99.5% para 99.98% na assertividade da tarefa *pick and place* (pegar e colocar);
- Reduzir em 50% tempo de preparar a máquina para operação, devido a plataforma similar ao Windows;
- Economia de 5 mil dólares anuais por máquina, devido a simplicidade de configuração e manutenção;
- Capacidade de executar duas tarefas na mesma plataforma reduzindo os custos por mais 25%;
- Capacidade de dobrar sua produção.

4. MATERIAIS e MÉTODOS

4.1 Equipamentos

Foi utilizado um manipulador robótico, uma placa de desenvolvimento e um computador. A descrição de cada um segue nas seções de 4.1.1 a 4.1.3.

4.1.1 Manipulador Robótico

O modelo KR16 da empresa Kuka Robotics GmbH (figura 7) é um robô manipulador para baixas cargas, até 16 kg, desenvolvido para aplicações industriais e, portanto, com controle de posição validado. O robô oferece diferentes funcionalidades integradas para facilitar a sua implementação em diversas áreas como montagem, soldagem, entre outros. Possui 6 graus de liberdade com uma repetibilidade de $\pm 0,05$ mm de posicionamento do ponto de referência na extremidade (Tool Center Point – TCP) e um envelope de trabalho de até 1610 mm, descontando o alcance adicional da ferramenta a ser inserida (KUKA ROBOTER GMBH, 2004).



Figura 7: Kuka KR16 no Laboratório de Mecatrônica da EESC-USP.

Este robô possui controlador KR C2 com o software KSS 5.4 (Kuka System Software). Para interface este robô utiliza Windows XP Embedded, mas como software de tempo real, o VxWorks, da empresa americana Wind River.

4.1.2 Placa de desenvolvimento

A placa de desenvolvimento utilizado neste projeto é composta de duas partes, a placa de suporte e o computador em módulo.

A placa de suporte utilizada é a Colibri Evaluation Board V3.2B (figura 8) , com destaque para as portas de entrada e saída utilizados:

- Ethernet: Para a conexão entre a placa de desenvolvimento e o robô;
- USB 2.0 On-The-Go (OTG): Para a conexão entre a placa de desenvolvimento e o computador;
- Colibri SODIMM: Para a conexão da placa com o computador em módulo.

Fonte: www.toradex.com/products/carrier-board/colibri-evaluation-board

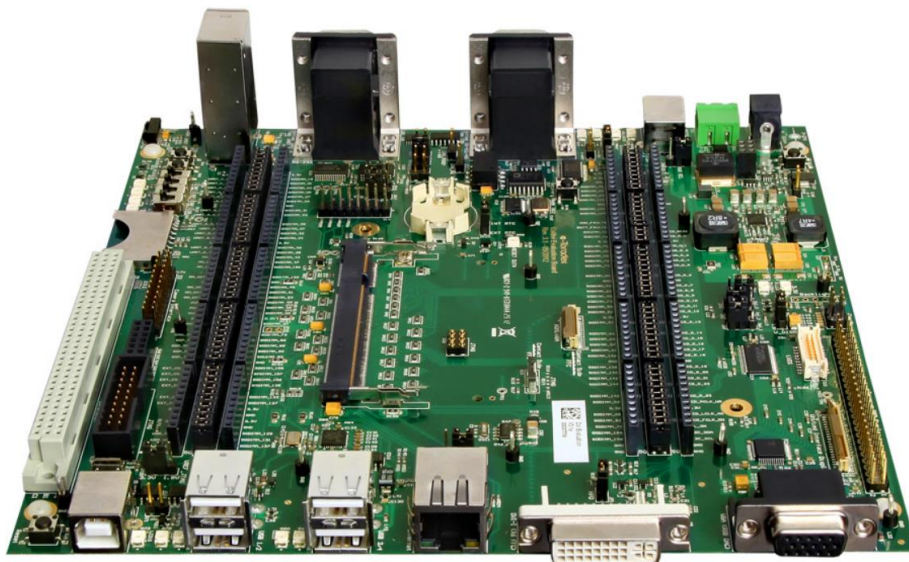


Figura 8: Colibri Evaluation Board V3.2B.

O computador em módulo usado é o Colibri T20 512MB V1.2A (figura 9), e suas características são: Processador e interface gráfica NVIDIA Tegra 2 dual-core Cortex-A9 1.0 GHz, memória RAM DDR2 de 512MB e memória nand flash de 1GB. Funcionando com o

sistema operacional do Windows Embedded Compact 7.0 e a sua plataforma de execução é o .NET compact Framework v3.5 (.NET CF v3.5).

Fonte: www.toradex.com/computer-on-modules/colibri-arm-family/nvidia-tegra-2



Figura 9: Colibri T20 512MB V1.2A.

A placa de desenvolvimento, composta da placa de suporte Colibri Evaluation Board e o computador em módulo usado é o Colibri T20 podem ser compradas diretamente pelo site da toradex por US\$ 367,00 que é equivalente a R\$ 1370,19 na cotação de 09/11/2018 (figura 10). Seu preço é relativamente baixo se pensado ao potencial de benefícios que sua implementação pode trazer para as fábricas.

Fonte: www.toradex.com/pt-br/order-now

No.	Código do Produto	Nome	Quantidade	Total
01.	00221200	Colibri T20 512MB IT	1	US\$ 166,50
02.	01013201	Colibri Evaluation Board	1	US\$ 200,50
Sub-Total				US\$ 367,00
Sub-Total		US\$ 367,00		
Custo de embalagem e frete		US\$ 60,00		
Total estimado do pedido		US\$ 427,00		

Figura 10: Orçamento Toradex.

4.1.3 Computador

O computador de desenvolvimento utilizado neste projeto é um notebook modelo Dell 15R special edition 7520, com 8 GB de memória RAM, sistema operacional Microsoft Windows 8.1 e processador Intel Core I7-3632QM 2,2GHz.

4.2 Métodos

A sua implementação foi feita da seguinte forma: foi evoluído um software em Visual Studio 2008 no computador para gerar um programa executável na placa de desenvolvimento com sistema operacional Windows CE 7, o qual foi diretamente conectado ao robô, possibilitando ao programa enviar comandos de ação ao robô e também ler todos os seus sensores.

4.2.1 Configuração da Placa de desenvolvimento

A comunicação entre os equipamentos foi feita por um cabo USB-A x USB-B (comumente usados em impressoras), o USB-A no computador e o USB-B na placa de desenvolvimento. O Windows CE da placa de desenvolvimento permite o funcionamento do *Remote Display* V2, que serve para exibir a tela do Windows CE em funcionamento na placa de desenvolvimento, e permite também usar o mouse e teclado para controlar o Windows CE pelo próprio computador.

Para executar o *Remote Display*, foi preciso fazer o download dele no computador, que pode ser baixado gratuitamente em: <https://docs.toradex.com/102838-remotedisplay-v2.zip>.

Foi necessário extrair o arquivo “zip” baixado acima e, já com o cabo entre o computador e a placa de desenvolvimento conectado, executar o arquivo "CeRHost.exe", fruto da extração do arquivo anterior. Nesse momento foi possível observar a tela do Windows CE com livre acesso aos arquivos da placa de desenvolvimento.

Para atingir o objetivo deste trabalho foi fundamental que o computador gerasse um código executável na placa de desenvolvimento, processo que permitiu a produção de um código executável em uma plataforma diferente da qual estava sendo executado, conhecido como compilador cruzado.

Para fazer essa compilação cruzada, ou seja, o computador gerar um arquivo executável na placa de desenvolvimento, foram feitas algumas configurações no computador:

1. Instalação do Visual Studio 2008;
2. Instalação de 2 updates do Visual Studio disponíveis no próprio site da microsoft

em:

- <http://www.microsoft.com/en-us/download/details.aspx?id=11935>
- <http://www.microsoft.com/en-us/download/details.aspx?id=27729>

3. Instalação do Kit de desenvolvimento de software mais conhecido como (SDK), o qual é um conjunto de ferramentas que permite a criação de software para um dispositivo específico. O fabricante da placa de desenvolvimento é a Toradex, e o SDK usado e disponibilizado por eles para especificamente essa placa, pode ser baixado gratuitamente em: <https://docs.toradex.com/102481>.

4.2.2 Configuração do Robô

O meio de comunicação do robô com a placa de desenvolvimento foi feito via rede Ethernet, para essa conexão apenas a configuração do número do Protocolo de Internet (IP) e a máscara de sub-rede foram suficientes para que os equipamentos pudessem se comunicar e trocar informações.

Para isso foi necessário que ambos os equipamentos tivessem o IP versão 4 (IPv4) fixo, e a mesma máscara de sub-rede. O IPv4 é composto de 4 dígitos separados por “.”, uma parte do Número IP (1, 2 ou 3 dos 4 números) é a identificação da rede, a outra parte é a identificação da máquina dentro da rede. O que define quantos dos quatro números fazem parte da identificação da rede e quantos fazem parte da identificação da máquina é a máscara de sub-rede, que também é composta por 4 números separados por “.”.

Nesse caso, é configurado o IP do robô como 192.168.10.1 e sua mascara de sub-rede como 255.255.255.0. Os três primeiros números da máscara de sub-rede (subnet) iguais a 255 indicam que os três primeiros números representam a identificação da rede e o último número é a identificação do equipamento dentro da rede. Dessa forma o número IP de todos os equipamentos da rede deve começar com 192.168.10, restando apenas o último dígito para identificar novos equipamentos, devendo ele ser único na rede e diferente de 0 e 255 devido a exigências do protocolo.

Atendendo esses requisitos foi definido o IPv4 da placa de desenvolvimento como 192.168.10.5 e a mesma mascarará de sub-rede 255.255.255.0.

A comunicação do robô foi executada utilizando a biblioteca RSI 2.3 (Robot Sensor Interface) responsável pela comunicação de dispositivos externos com o sistema de controle do robô (KUKA ROBOTER GMBH, 2009). Esta biblioteca permitiu a comunicação do robô com dados enviados via rede Ethernet. Os dados do robô foram compilados e comunicados em formato de texto XML (eXtensible Markup Language). Após o envio, o sistema do robô aguardou um pacote XML vindo do computador com os dados de comando, permitindo assim, garantir a troca de informações entre computador e robô.

4.2.3 Comunicação com o robô

A biblioteca RSI 2.3 (*Robot Sensor Interface*) é responsável pela comunicação de dispositivos externos com o sistema de controle do robô (KUKA ROBOTER GMBH, 2009). Os dados do robô e sensor de força foram compilados e comunicados em formato de texto XML. Após o envio, o sistema do robô aguardou um pacote XML vindo do computador com os dados de comando, permitindo assim, garantir a troca de informações entre computador e robô. Os pacotes XML fornecidos pela própria Kuka Robotics GmbH (KUKA ROBOTER GMBH, 2009) foram adaptados por Soares (2017) e usado neste trabalho e estão representados no Apêndice E e Apêndice F.

A RSI possui em sua configuração uma forma de verificar a qualidade de comunicação, tanto no tempo de envio e resposta, como na perda de pacotes, a fim de prover uma segurança e estabilidade na comunicação. O tempo para a troca de pacotes exigida pelo sistema de tempo real do controlador é de 12 ms e o limite de pacotes perdidos podem ser configurados pelo usuário.

O protocolo selecionado para efetuar a comunicação com dispositivos externos e a RSI 2.3 é o UDP (User Datagram Protocol). Trata-se de um padrão de comunicação da camada de transporte do modelo OSI (Open System Interconnection) adotado pelos meios de comunicação atuais, possuem algumas singularidades favoráveis para comunicação que exigem uma velocidade alta no fluxo de dados. Por não possuir uma forma de verificação se os dados fornecidos à rede foram recebidos pelo endereço, se torna uma forma mais rápida de comunicação, entretanto é necessário inserir uma forma de verificação (handshaking) para evitar as perdas de pacotes.

Os termos delay e IPOC são as formas de conferir a reciprocidade e segurança da conexão. O primeiro realiza uma contagem de pacotes perdidos e o segundo é um número

para identificação de cada pacote e, por se tratar de uma comunicação em tempo real, é obrigatório realizar comunicação com um período máximo de 12ms.

A estrutura de dados XML de comunicação, no sentido manipulador robótico ao computador (Apêndice E), contempla informações sobre a situação atual, dados como posição, tanto nas coordenadas X, Y e Z (RIst e RSol) como os ângulos de cada junta (EIst), força (FTC) e saídas digitais (DigOut), e também um número único IPOC em cada envio.

Como resposta, a placa de desenvolvimento deve enviar uma mensagem com uma estrutura de dados XML de comunicação (Apêndice F), com comandos de correção na posição (Rkorr e EKorr), entradas digitais (DigIn) e com o mesmo número da IPOC da mensagem recebida.

4.2.4 Desenvolvimento do Software

O software de processamento foi desenvolvido em linguagem C# compatível com Windows 8.1 e com o Visual Studio 2008, recomendado pela Toradex, fabricante da placa de desenvolvimento.

A placa de desenvolvimento efetua todo o processamento necessário para o fluxo de dados. A plataforma com capacidade de comunicação UDP e serial é essencial para a comunicação de todos os dispositivos disponíveis e necessários, por este motivo o C# foi selecionado, já que possui bibliotecas para suprir as necessidades. O projeto é constituído de 4 módulos:

- Programs.cs - Responsável pela interação com o usuário;
- kuka.cs - Responsável por estabelecer a comunicação com o robô e também manipular os arquivos XML a enviar e recebidos;
- Position.cs - Responsável por trocar a medida de ângulos negativos por positivos, e também algumas funções auxiliares como tratar os valores de posição e formatar dados que serão inseridos no XML;
- Impedance.cs - Responsável pelo controle do impedância do robô.

O fluxograma do código pode ser visto no apêndice G, com o seguinte desenvolvimento, para entender seu funcionamento do software, no módulo program.cs é declarado classe da variável do robô e, em seguida, são definidos 3 parâmetros que irão reger

a comunicação. O primeiro parâmetro é a porta de comunicação a ser usada, que deve ser a mesma configurada no arquivo XML do robô, no caso foi usada a porta 2777. O segundo parâmetro é definido da máxima velocidade cartesiana do robô, e o terceiro é definido a máxima velocidade rotacional do robô.

Posteriormente, é chamada a função para iniciar a comunicação, chamada no código de *StartCommunication()*, implementada no módulo *kuka.cs*, dentro da função é criado um novo thread que é iniciado para atender a função *Connection()* responsável por manter, em loop infinito, o envio e recebimento de mensagens com o robô e dessa forma estabelecendo a comunicação com o robô. Enquanto isso, o thread original retorna ao módulo *program.cs* onde irá atender as rotinas de tarefas do robô implementada.

O primeiro comando dado no thread responsável pela comunicação com o robô, é função *Connection()*, nessa função primeiramente é definido o cliente da comunicação UDP e a porta pela qual ela se comunicará, definida no início, em seguida é carregado o arquivo de XML modelo (Apêndice F) em um objeto, e então o thread entre em um loop infinito de recebimento e envio de mensagem.

Dentro do loop infinito, o primeiro comando é o de armazenamento dos dados do robô pelo cliente UDP criado, e o de montar em uma mensagem completa com os dados, essa mensagem é a que pode ser vista no Apêndice A e B, após isso é verificado se a mensagem veio completa, se sim, começa a sua interpretação. A interpretação da mensagem é feita da seguinte forma, a variável *Rlst*, composta de 6 parâmetros, sendo X, Y, Z referente a sua posição em coordenadas cartesianas e os parâmetros A, B, C referente a sua rotação em torno de cada eixo da coordenada cartesiana, é armazenada como a posição inicial do robô.

No próximo passo é checado se o robô está em movimento ou ainda há alguma requisição de tarefa, quando não existem ambas: começa a montagem da mensagem a ser enviada para o robô (Apêndice D e E). A mensagem a ser enviada é montada em três etapas, primeiro é copiado para a mensagem o mesmo IPOC da mensagem recebida, a segunda é inserir a correção da posição do robô, no caso chamada de *Rkorr*, como não foi executado nenhuma tarefa é a posição desejada para ele se mover portanto também não há valor a ser corrigido é seu valor é nulo (0.000), a terceira é inserir a sua saída digital, em seguida a mensagem é enviada para o kuka via UDP, e então retorna loop retomando o código.

Quando é dado um comando de enviar posição ao robô, ou seja, quando é executada a função *SendPosition(parâmetro)* o seu funcionamento é descrito da seguinte forma: o parâmetro, passado junto com a função, é composto das seis variáveis que definem a posição do controlador robótico X, Y, Z, A, B, C. A função *SendPosition()* foi implementada no

módulo kuka.cs e funciona assim: ela calcula a diferença da posição enviada com a sua posição atual (Rlst) do robô, essa diferença é a correção que ele precisa fazer para atingir a posição desejada, mas antes de carregá-la na variável Rkorr para ser enviada ao robô, é preciso dividir a correção em correções menores para que seja respeitado o limite de velocidade cartesiana e rotacional máximas definidas quando se iniciou a comunicação com o robô, isso é feito pela função SpeedControlABS(). Sendo assim, para atingir a posição desejada, cada correção menor é calculada e carregada na variável Rkorr, dessa forma são garantidos os limites de velocidade e o robô chega a posição desejada após o envio de todas as correções menores.

5. RESULTADOS

Foi possível controlar o robô quando executado pelo computador, as mensagens XML enviadas podem ser vistas no Apêndice C, entretanto, o mesmo código funcionando na placa de desenvolvimento, as mensagens XML enviadas podem ser vistas no Apêndice D, apresentou comportamento inesperado não mantendo comunicação com o robô. Na interface do robô é vista a seguinte mensagem de erro, “SEN: RSI execution error <execute> - RSI stopped” e a comunicação é interrompida. Por se tratar do mesmo código, era de se esperar igual comportamento quando executado pelo computador ou pela placa de desenvolvimento. Diante disso, foi feito contato com o suporte da fabricante da placa, Toradex, o qual informou que não possui muitos recursos em Windows CE para poder auxiliar.

Examinando o problema, verificou-se que o tratamento dos dados recebidos do robô pelo computador (apêndice A) via XML são iguais aos recebidos pela placa de desenvolvimento (apêndice B), e também os enviados pelo computador ao robô (apêndice C) estavam iguais aos enviados pela placa de desenvolvimento ao robô para executar a mesma tarefa (apêndice D). Como não foram observadas divergências no processamento das mensagens XML, pode-se notar que o código de processamento e montagem da mensagem funciona corretamente.

6. CONCLUSÃO

Esse trabalho visou apresentar um estudo sobre indústria 4.0 e as suas ferramentas, mostrando a sua importância, os benefícios da implementação e as expectativas dessa revolução para os próximos anos. Após a apresentação desses tópicos o trabalho focou no desenvolvimento de uma alternativa de relativamente baixo custo às fábricas e empresas para fazer a integração dos seus equipamentos rede.

Como proposto, pode-se concluir que é possível fazer a integração de robôs e equipamentos a rede e futuramente colher os benefícios desta por um custo relativamente baixo. Apesar do software não ter funcionado como esperado na placa de desenvolvimento, o mesmo funciona corretamente no computador, sugerindo que o problema possa estar no sistema operacional da placa de desenvolvimento ou em seu ambiente de execução .NET CF v3.5.

Como sugestão para trabalhos futuros, aponto a necessidade de corrigir o problema de comunicação com a placa de desenvolvimento. Primeiramente, sugiro desenvolver uma plataforma que permita o envio de comandos ao robô remotamente, a fim de que o mesmo chegue primeiro a placa de desenvolvimento e essa, por sua vez, envie o comando ao robô. Outra proposta de trabalho, é a realização de projetos utilizando a Inteligência Artificial (IA) na nuvem como, por exemplo, um robô projetado para apertar parafusos fazer uma requisição de compra de mais parafusos quando eles estão chegando ao fim.

REFERÊNCIAS BIBLIOGRÁFICAS

- AZEVEDO, M. T. **Transformação Digital na Indústria: Indústria 4.0 e a Rede de Água Inteligente no Brasil**. 2017. 177 f. Tese para obtenção do título de doutor (Ciências) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2017.
- SCHWAB, K. **The Fourth Industrial Revolution**. VINT research report, p. 1-39, 2016.
- GUBBI, J. et al. **Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions**. 2013. Disponível em <https://arxiv.org/ftp/arxiv/papers/1207/1207.0203.pdf> >. Acesso em: 03 nov. 2018.
- BRETTEL, M. et al. **How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective**. 2013. Disponível em <http://waset.org/publications/9997144>>. Acesso em: 03 nov. 2018.
- KAGERMANN, H. et al. **Bericht der Promotorengruppe KOMMUNIKATION Wirtschaft und Wissenschaft begleiten die Hightech-Strategie Forschungsunion Im Fokus: Das Zukunftsprojekt Industrie 4.0**. 2012. Disponível em https://www.bmbf.de/pub/hts/kommunikation_bericht_2012-1.pdf>. Acesso em: 03 nov. 2018.
- DUJIN, A. et al. , **Think Act. Industry 4.0 – The New Industrial Revolution: How Europe Will Succeed**. 2014. Disponível em https://www.rolandberger.com/en/Publications/pub_industry_4_0_the_new_industrial_revolution.html>. Acesso em: 03 nov. 2018.
- LIAO, Y. et al. **The impact of the fourth industrial revolution: a cross-country/region comparison**. 2018. Disponível em <http://www.scielo.br/pdf/prod/v28/0103-6513-prod-28-e20180061.pdf> >. Acesso em: 03 nov. 2018.
- RUBMANN, M et al. **Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries**. 2015. Disponível em http://www.inovasyon.org/pdf/bcg.perspectives_Industry.4.0_2015.pdf>. Acesso em: 03 nov. 2018.
- KUKA ROBOTER GMBH. **KR 6 ; KR 16 ; KR 6 K ; KR 16 K Description**. 2004. 2004.
- KUKA ROBOTER GMBH. **Kuka.RobotSensorInterface 2.3**. V1. ed. [s.l: s.n.].
- LU, Y. **A survey on technologies, applications and open research issues**. 2017. Disponível em <https://www.sciencedirect.com/science/article/pii/S2452414X17300043?via%3Dihub>>. Acesso em: 03 nov. 2018.

HOFMANN, E.; RÜSCH, M. **Industry 4.0 and the current status as well as future prospects on logistics.** 2017. Disponível em <https://www.sciencedirect.com/science/article/pii/S0166361517301902?via%3Dihub>.

Acesso em: 03 nov. 2018.

MOREIRA, A.; FONSECA, J.; TAVARES, A., **WinCE-based Embedded System for Control of an Industrial Screw Machine.** 2009. Disponível em <https://repositorium.sdum.uminho.pt/bitstream/1822/9707/1/2900-pf-007269.pdf>

Acesso em: 03 nov. 2018.

TAO, F.; ZHANG, L.; VENKATESH, V. C.; LUO Y.; CHENG.; **Cloud manufacturing: a computing and service oriented manufacturing model.** 2011. Disponível em <http://www.ntu.edu.sg/home/wptay/MyPapers/Journals/HuTayWen%20-%20Cloud%20robotics%20architectures%20challenges%20and%20applications.pdf> >

Acesso em 08 nov. 2018.

THAMES, L.; SCHAEFER, D.; **Software-Defined Cloud Manufacturing for Industry 4.0.** 2016. Disponível em https://ac.els-cdn.com/S2212827116307910/1-s2.0-S2212827116307910-main.pdf?_tid=92990269-0532-4ea1-99f5-1ce63dece6a3&acdnat=1541893994_886fa41db974214d656741db4af615d0 >

Acesso em 08 nov. 2018.

AROCA, R.; CAURIN, G., **A Real Time Operating Systems (RTOS) Comparison.** 2009. Disponível em <https://sites.google.com/site/rafaelaroca/artigos/ArocaWSO2009.pdf>

Acesso em: 03 nov. 2018.

SUSEELA, A.; KUMAR V.; GARIVIDI, S., **Embedded Systems in Real Time Applications, Design & Architecture.** 2005. Disponível em https://dl.acm.org/ft_gateway.cfm?id=1086450&type=pdf Acesso em: 03 nov. 2018.

INTERVAL ZERO, **Best Practices for Real-Time Machine Automation Performance in IoT & Industry 4.0.** 2018. Disponível em <https://www.intervalzero.com/assets/Best-Practices-for-Real-Time-Machine-Automation-Performance-in-IoT-Industry-4.0.pdf>

Acesso em: 03 nov. 2018.

CLICKATELL, **How IoT connected factories are changing manufacturing for the better.** 2017. Disponível em <https://www.clickatell.com/articles/technology/iot-connected-factories-manufacturing/> Acesso em: 09 nov. 2018.

RTINSIGHTS, **Manufacturers Will Have Fully Connected Factories by 2022**. 2017. Disponível em <<https://www.rtinsights.com/manufacturers-will-have-fully-connected-factories-by-2022/>> Acesso em 09 nov. 2018.

SOARES, J. V. R. **Análise e implementação de técnicas de robótica colaborativa para tarefa de montagem de mobiliário aeronáutico**. 2017. 92 f. Tese para obtenção do título de Bacharelado em Engenharia Elétrica - Escola de Engenharia de São Carlos Universidade de São Paulo, São Paulo, 2017.

WU, D.; **Cloud-Based Manufacturing: Old Wine in New Bottles?**. 2014. Disponível em <<https://www.sciencedirect.com/science/article/pii/S2212827114002674/pdf?md5=730bc2ff54baa7eed4b56f400b54589f&pid=1-s2.0-S2212827114002674-main.pdf>> Acesso em 09 nov. 2018.

HU, G.; TAY, W. P.; WEN, Y.; **Cloud Robotics: Architecture, Challenges and Applications**. 2012. Disponível em <<http://www.ntu.edu.sg/home/wptay/MyPapers/Journals/HuTayWen%20-%20Cloud%20robotics%20architectures%20challenges%20and%20applications.pdf>> Acesso em 09 nov. 2018.

KEHOE, B.; **A Survey of Research on Cloud Robotics and Automation**. 2015. Disponível em <<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7006734>> Acesso em 11 nov. 2018.

PISCHING, Marcos André. **Arquitetura para descoberta de equipamentos em processos de manufatura com foco na indústria 4.0**. 2017. Tese (Doutorado em Engenharia de Controle e Automação Mecânica) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2017. doi:10.11606/T.3.2018.tde-05032018-133720. Acesso em 11 nov. 2018.

Apêndice A - XML recebido do robô pelo computador

```
IT IS RECEIVING THESE INFORMATION FROM THE KUKA
<Rob Type="KUKA">
<RIst X="929.9320" Y="-0.0533" Z="1105.4889" A="179.9986" B="-0.0001" C="-180.00
00"/>
<RSo1 X="929.9319" Y="-0.0533" Z="1105.4868" A="179.9977" B="-0.0001" C="180.000
0"/>
<AIPos A1="0.0000" A2="-90.0000" A3="89.9998" A4="0.0000" A5="90.0002" A6="0.001
0" />
<ASPos A1="-0.0002" A2="-90.0000" A3="89.9998" A4="0.0000" A5="90.0002" A6="0.00
10"/>
<EIPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<ESPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<MACur A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000"/>
<MECur E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<Delay D="52" />
<Tech C11="0.0000" C12="0.0000" C13="0.0000" C14="0.0000" C15="0.0000" C16="0.00
00" C17="0.0000" C18="0.0000" C19="0.0000" C110="0.0000" />
<DiL>not linked:1</DiL>
<Digout o1="not linked:2" o2="not linked:3" o3="not linked:4" />
<ST_Source>not linked:5</ST_Source>
<FTC Fx="-89.7465" Fy="-94.1336" Fz="188.2684" Mx="-6.1433" My="0.2595" Mz="1.44
53" />
<IPOC>4218277478</IPOC>
</Rob>
```

Apêndice B - XML recebido do robô pela placa de desenvolvimento

```
IT IS RECEIVING THESE INFORMATION FROM THE KUKA
<Rob Type="KUKA">
<RIst X="929.9312" Y="-0.0533" Z="1105.4868" A="179.9976" B="-0.0002" C="-180.00
00"/>
<RSol X="929.9319" Y="-0.0533" Z="1105.4868" A="179.9977" B="-0.0001" C="180.000
0"/>
<AIPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" A5="90.0002" A6="-0.00
01" />
<ASPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" A5="90.0000" A6="0.000
0"/>
<EIPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<ESPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<MACur A1="-0.0146" A2="0.8877" A3="-0.8279" A4="0.0891" A5="-0.0007" A6="-0.128
4"/>
<MECur E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<Delay D="95" />
<Tech C11="0.0000" C12="0.0000" C13="0.0000" C14="0.0000" C15="0.0000" C16="0.00
00" C17="0.0000" C18="0.0000" C19="0.0000" C110="0.0000" />
<DiL>not linked:1</DiL>
<Digout o1="not linked:2" o2="not linked:3" o3="not linked:4" />
<ST_Source>not linked:5</ST_Source>
<FTC Fx="-89.8989" Fy="-93.8486" Fz="188.2753" Mx="-6.1321" My="0.2433" Mz="1.44
50" />
<IPOC>4217923208</IPOC>
</Rob>
```

Apêndice C - XML enviado para o robô pelo computador

```
esperando comunicacao
<Sen Type="ImFree">
  <Estr>Laboratorio de Manipulacao Robotica EESC USP</Estr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715653</IPOC>
</Sen>
<Sen Type="ImFree">
  <Estr>Laboratorio de Manipulacao Robotica EESC USP</Estr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715659</IPOC>
</Sen>
<Sen Type="ImFree">
  <Estr>Laboratorio de Manipulacao Robotica EESC USP</Estr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715665</IPOC>
</Sen>
Comunicacao OK
<Sen Type="ImFree">
  <Estr>Laboratorio de Manipulacao Robotica EESC USP</Estr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715671</IPOC>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715671</IPOC>
</Sen>
<Sen Type="ImFree">
  <Estr>Laboratorio de Manipulacao Robotica EESC USP</Estr>
  <RKorr X="0.0007" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715677</IPOC>
</Sen>
<Sen Type="ImFree">
  <Estr>Laboratorio de Manipulacao Robotica EESC USP</Estr>
  <RKorr X="0.5562" Y="0.5556" Z="0.5556" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4208715683</IPOC>
</Sen>
Voltando para posicao inicial
```

Apêndice D - XML enviado para o robô pela placa de desenvolvimento

```
esperando comunicacao
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  </>
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  </>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4210007123</IPOC>
</Sen>
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  </>
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  </>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4210007129</IPOC>
</Sen>
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  </>
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  </>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4210007135</IPOC>
</Sen>
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  </>
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  </>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4210007141</IPOC>
</Sen>
Comunicacao OK
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  </>
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  </>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4210007147</IPOC>
</Sen>
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0007" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  </>
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  </>
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC>4210007165</IPOC>
</Sen>
Voltando para posicao inicial
■
```

Apêndice E - XML modelo de recebimento

```
<Rob Type="KUKA">
<RIst X="930.0542" Y="0.1778" Z="971.7679" A="179.9992" B="0.0011" C="-180.0000"/>
<RSol X="930.0553" Y="0.1778" Z="971.7679" A="179.9993" B="0.0013" C="180.0000"/>
<AIPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" A5="90.0002" A6="-0.0001" />
<ASPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" A5="90.0000" A6="0.0000"/>
<EIPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<ESPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<MACur A1="0.0313" A2="0.9014" A3="-0.7944" A4="-0.0630" A5="0.0129" A6="-0.0430"/>
<MECur E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>
<Delay D="0" />
<FTC Fx="0.0148" Fy="0.0241" Fz="-0.0560" Mx="0.0035" My="-0.0032" Mz="-0.0034" />
<IPOC></IPOC>
</Rob>
```

Apêndice F - XML modelo para envio

```
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <DigOut D1="0" D2="0" D3="0" />
  <IPOC></IPOC>
</Sen>
```

Apêndice G – Fluxograma do código

