

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE ENGENHARIAS DA MOBILIDADE
CURSO DE ENGENHARIA MECATRÔNICA

WILLIAM DE PAULI

SISTEMA SEPARADOR DE ITENS UTILIZANDO RFID

Joinville, 2015

WILLIAM DE PAULI

SISTEMA SEPARADOR DE ITENS UTILIZANDO RFID

Trabalho apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Curso de Engenharia Mecatrônica da Universidade Federal de Santa Catarina, Campus de Joinville.

Orientador: Dr. Anderson Wedderhoff Spengler

Joinville, 2015

SISTEMA SEPARADOR DE ITENS UTILIZANDO RFID

WILLIAM DE PAULI

Esta monografia foi julgada e aprovada como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Centro de Engenharias da Mobilidade, da Universidade Federal de Santa Catarina, Campus de Joinville.

Joinville (SC), 02 de julho de 2015.

Dr. Xisto Lucas Travassos Júnior

Coordenador do Curso

COMISSÃO EXAMINADORA:

Dr. Anderson Wedderhoff Spengler

Orientador

Dr. Gierry Waltrich

Membro

Dr. Wellington Roque

Membro

Dedico aos meus pais Salete e José de Pauli.

Agradeço primeiramente a Deus por dar-me força para ultrapassar todos meus obstáculos, e impedindo qualquer pensamento de desistência ou fraqueza.

A minha família que esteve sempre presente em minha vida, principalmente aos meus amados pais Salete e José, por estarem sempre comigo em minhas conquistas até hoje, sempre motivando-me ou acolhendo-me em momentos de desconforto. Minhas irmãs Cyndi e Muriel, por toda ajuda oferecida a mim, desde coisas simples da vida até auxílio na vida acadêmica

As minhas amigas feitas durante a faculdade, meus colegas da Engenharia Mecatrônica e meus companheiros do futebol de toda sexta-feira, que estiveram sempre ao meu lado, me auxiliando e superando qualquer obstáculo comigo.

Aos amigos, que de algum jeito, auxiliaram-me na realização do meu trabalho de conclusão de curso, oferecendo disposição e interesse para conclusão do projeto.

Aos professores da Universidade Federal de Santa Catarina – Campus Joinville, principalmente ao meu orientador Dr. Anderson W. Spengler, por toda paciência e tempo investido em mim, solucionando minhas dúvidas e meus erros ao decorrer das disciplinas.

“O importante é não parar de questionar. A curiosidade tem sua própria razão de existir”.

(Albert Einstein)

RESUMO

A logística é assunto muito discutido nos últimos anos por ser questão primordial para empresas que buscam mais velocidade e menos custo em suas linhas de produção. Assim, há uma grande procura por sistemas automatizados para auxiliar nas produções industriais. A construção de um sistema separador de itens utilizando RFID se destina a setores industriais com o objetivo de tornar o processo de separação de itens mais rápido, confiável e de baixo custo. A ferramenta de identificação por rádio frequência é uma tendência que está crescendo muito na última década, vindo à tona e substituindo outros meios de identificação. O processo de construção inicia pela parte estrutural, estrutura da esteira, passando pela parte de hardware, como kits e componentes eletrônicos, e finalizando ao software, que compreende a programação dos mesmos para possibilitar as devidas funções atribuídas ao sistema separador de itens. O sistema entra em funcionamento no momento que passar um item com seu cartão de identificação próximo ao sensor, e assim, estabelecer comunicação sem fio, utilizando ZigBee com o LabVIEW, para acesso ao banco de dados identificando qual destino do produto e, assim, realizar as devidas separações no instante que o mesmo chegar próximo as hastes seletoras.

Palavras-Chave: RFID, Logística, ZigBee, Separação de itens, LabVIEW.

ABSTRACT

Logistics has been a widely discussed subject over the last few years because it is a primary issue for companies seeking more speed and less cost in their production lines. This way, there is a great search for automated systems to assist industrial production. The construction of a RFID item separation system focuses on areas of the industry that need a fast, reliable, and low cost separation system. The radio frequency identification tool is a tendency that has been growing over the last decade, already substituting some means of identification. The construction process begins with the structural part, conveyor frame, through the hardware part, with electronic components, and finalizing with the software, which comprises its programming to enable the necessary functions assigned to the items selector system. The system starts at the moment an item with its ID card pass near the sensor and then establishes a wireless communication, utilizing ZigBee with LabVIEW, for access the database identify the product's destination and perform the separation when the item reaches the selector rods.

Keywords: RFID, Logistics, ZigBee, Separation of items, LabVIEW.

LISTA DE ILUSTRAÇÕES

- Figura 1** – Comparação entre diferentes protocolos em relação à taxa de transmissão de dados e alcance.
- Figura 2** – Fluxograma do funcionamento do projeto.
- Figura 3** – Kit utilizado no sistema - MSP430F5529LP.
- Figura 4** – Algumas topologias de rede do protocolo ZigBee.
- Figura 5** – Módulo ZigBee.
- Figura 6** – RFID Starter kit.
- Figura 7** – Servo-motor Tower Pro 9g.
- Figura 8** – Fonte de tensão utilizada para alimentação do sistema.
- Figura 9** – Esteira confeccionada para a seleção de itens.
- Figura 10** – Software utilizado para programação e utilização dos ZigBee's.
- Figura 11** – Exemplo de aplicação usando LabVIEW.
- Figura 12** – Exemplo de aplicação do CCS.
- Figura 13** – Banco de dados usando o MO Access.
- Figura 14** – Descrição do vetor de informação de uma tag.
- Figura 15** – Programação e configuração para acesso ao banco de dados e definição de local – Estrutura 1.
- Figura 16** – Programação e configuração para acesso ao banco de dados e definição de local – Estrutura 2.
- Figura 17** – Programação e configuração para acesso ao banco de dados e definição de local – Estrutura 3.
- Figura 18** – Interface criada para a comunicação usando LabVIEW.
- Figura 19** – Esquema de ligação do projeto.
- Figura 20** – Projeto finalizado.

SUMÁRIO

1. INTRODUÇÃO	11
2. CONTEXTUALIZAÇÃO.....	13
3. MATERIAIS E MÉTODOS.....	16
3.1. MSP430F5529 LaunchPad	19
3.2. ZigBee	20
3.3. KIT RFID INICIANTE.....	21
3.4. SERVO-MOTOR.....	22
3.5. FONTE DE ALIMENTAÇÃO CONTROLÁVEL	22
3.6. PROTÓTIPO	23
3.7. X-CTU.....	23
3.8. LABVIEW	24
3.9. CODE COMPOSER STUDIO (CCS).....	25
3.10. MICROSOFT OFFICE ACCESS.....	26
3.11. ANÁLISES E CONFIGURAÇÕES	27
CONCLUSÃO	35
REFERÊNCIAS	36
ANEXOS	38
ANEXO 1 – CÓDIGO BASE PARA COMUNICAÇÃO UART.....	38
ANEXO 2 – CÓDIGO BASE PARA ACIONAMENTO DOS SERVO-MOTORES	39
APÊNDICES.....	41
APÊNDICE A – TABELA DE CUSTO DOS DISPOSITIVOS DO PROJETO	41
APÊNDICE B - IMPLEMENTAÇÃO FINAL.....	42

1. INTRODUÇÃO

Em busca de maior flexibilidade na gestão das empresas, a logística tem sido grande aliado de empreendedores que buscam inovação de seus sistemas. Em um artigo escrito na Faculdade de Tecnologia de Santa Catarina, JACINTO et al. (2008) explicam que esse desenvolvimento social e econômico faz com que os gestores busquem a inovação, visando sempre a qualidade, competitividade, satisfação e retenção desses consumidores. Esse fato tem gerado conseqüentemente, maior desenvolvimento empresarial.

Tendo em vista que a maior preocupação de implementações de novos sistemas em empresas giram em torno de custos, é necessário efetuar levantamento de dados que vise custo benefício, pois para as companhias que dependem de um sistema de seleção, investir na melhoria da logística de sua empresa é algo primordial.

Com a demanda de sistemas organizacionais nas indústrias e empresas de entrega e distribuição de itens, a separação de produtos com rapidez e exatidão tornou-se um aspecto comercial muito importante para consumidores em geral. Assim, a automatização de sistemas vem à tona tornando a mão de obra humana desnecessária, alterando as variáveis tempo, dinheiro e confiabilidade.

Destarte propomos neste trabalho a implantação de uma esteira separadora automática utilizando sistema de reconhecimento via rádio frequência que pretende facilitar e acelerar sistemas de empresas que tem em sua rotina a finalidade de separação e destinação de itens. Além disso, a esteira será de extrema valia para empresas de pequeno ou grande porte onde admitem precisão em seus resultados.

O sistema é baseado na utilização da ferramenta de identificação por rádio frequência (RFID). Cada item terá um cartão de identificação (TAG). Quando aproximado ao sensor, a checagem ocorrerá em um banco de dados através do software LabVIEW, assim, buscando informações sobre o local que o item deve ser separado com a ação de hastes atuadoras. Ao momento de inexistência de TAG's

em itens ou no banco de dados, o mesmo não será separado, seguindo para análise e reconfiguração.

A metodologia do projeto será dividida em três blocos. A primeira etapa consiste na construção e elaboração da esteira. A segunda etapa consiste na aquisição de equipamentos como plataformas eletrônicas, componentes eletrônicos e motores que serão utilizadas para o funcionamento do projeto. E finalmente, a programação em plataformas eletrônicas para controle e comunicação de todo o sistema. Também realizando criação de um banco de dados que tem como função a comunicação sem fio utilizando ZigBee com a placa controladora, fazendo a checagem e verificação dos itens a serem selecionados para atingir os seguintes objetivos.

Objetivo Geral:

- Realizar montagem e implantar um sistema de separação de itens em uma esteira utilizando sistema de identificação por rádio frequência (RFID).

Objetivos Específicos:

- Promover o funcionamento e desenvolver comunicação sem fio entre o banco de dados e a placa controladora.
- Empregar um sistema de banco de dados ao protótipo para verificar e checar informações de distintos itens.
- Identificar e separar itens com confiabilidade e agilidade.
- Facilitar a seleção de itens em empresas ou redes de distribuição.

2. CONTEXTUALIZAÇÃO

A construção de esteiras separadoras de itens vem à tona para facilitar o transporte de mercadorias, tornando mais ágil e confiável a devida destinação de determinados itens.

Para Rodrigues, Medeiros e Bittencourt “A esteira é a melhor opção para ter uma maior velocidade no transporte de materiais em uma célula de manufatura, agilizando a produção.” (2012, p. 1). É notável que empresas de diferentes portes procurem sistemas ágeis, confiáveis e de baixo custo para ser aplicada em seu sistema fabril, por isso a motivação de diversos projetos com fins destinados a linhas de produção estarem aparecendo nos últimos anos. As esteiras transportadoras são largamente utilizadas nas indústrias devido as suas características como mobilidade, rapidez e custo no transporte de diferentes produtos (VIDAL, 2013).

Outro tópico importante citado pelos pesquisadores é o custo da construção do projeto, pois é de grande interesse para indústrias projetos de baixo custo para aplicação e desenvolvimento em seus setores.

Bueno e Costa (2011) mostram que o principal motivo da implementação de uma esteira didática é que ela será capaz de reduzir consideravelmente os custos totais de um sistema de automação convencional, tornando a empresa mais moderna e autossustentável.

É de grande valia construção de projetos com baixo custo agregado para diferentes finalidades. Para o projeto da esteira separadora de itens utilizando RFID foi escolhidos equipamentos de baixo custo, como kit's de placas microcontroladas de baixo valor comercial e peças adquiridas no laboratório ou em sucatas eletrônicas.

Rodrigues, Medeiros e Bittencourt (2012) afirmam que o projeto de uma esteira transportadora didática é um modelo de custo reduzido, iniciando a construção do protótipo em um ambiente virtual e construindo com materiais reciclados e de equipamentos eletrônicos desativados, construindo um projeto barato.

Para a indústria, quando discute-se de custos é um tópico muito delicado, e necessita ser analisado. Produzir um sistema separador de itens de baixo custo e com características já citadas é uma tarefa desafiadora, pois deve ser mantida sua função de modo a não causar erros relacionados a hardware ou software do projeto.

O desenvolvimento de uma esteira como protótipo de um sistema simples de automação, utilizando peças em madeira para a construção da base e estrutura foi a ideia de Bueno e Costa (2011) para construir um sistema de baixo custo e sendo aplicado em automação industrial. A mesma ideia de redução de custos foi adotada ao projeto de uma esteira seletora de Viecelli e Arnaut (2009) que foi desenvolvido para obter economia de energia elétrica, ou seja, apenas ocasionará o funcionamento quando existir algum objeto sob a esteira, senão será desligada até o momento de inserção de itens.

Em projetos mecatrônicos é comum o aparecimento de tecnologias que estão em alta no mercado, como sistemas sem fio (WIFI, Rádio Frequência, Bluetooth e ZigBee), sistemas infravermelhos e sistemas de capturas de imagens.

O sistema de identificação via rádio frequência é uma tecnologia existente desde a Segunda Guerra Mundial, mas com o tempo foi reconhecida em empresas para melhorar seu desempenho e agilizar processos principalmente destinados a indústria (JUNIOR, 2007).

O sistema RFID ganha espaço no mercado a partir do ano 2000, onde os custos de implementações e produções de TAG's foram reduzidos, tornando esta ferramenta mais atraente para setores industriais comparadas com outras tecnologias existentes, afirma Kawano e Cugnasca (2014).

Fazendo um comparativo com outras tecnologias, como o sistema de código de barras, o RFID supera em alguns aspectos oferecendo um controle preciso, redução em custos e maior agilidade, além de poder ser aplicado em ambientes onde não serão possíveis outras tecnologias. (FRANCO et al., 2009).

O funcionamento do sistema de rádio frequência é simples, o leitor é responsável por emitir energia através de um campo eletromagnético (rádio frequência), que por sua vez é recebida e respondida pela tag com o conteúdo de sua memória.

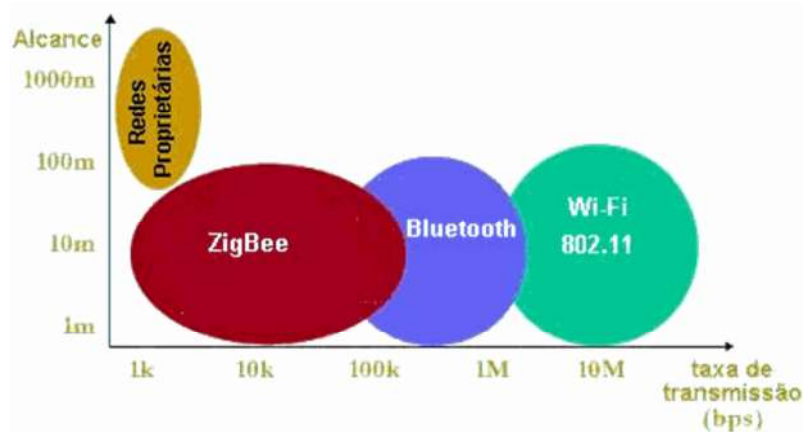
Sobre o ZigBee, Della Lucia (2014) afirma que este protocolo é um padrão de comunicação sem fio de baixa potência, padronizado pela ZigBee Alliance. Ele foi constituído da ideia do protocolo IEEE 802.15.4 (desenvolvido para redes de

sensores sem fio) onde tem a finalidade de transmitir ou receber dados a uma longa distância.

Neste protocolo há opção muito interessante quando o assunto é consumo de energia. A função *sleep*, é utilizada para o baixíssimo consumo de energia durante a utilização de um sistema em geral, esta função permite a atividade apenas quando necessário ou solicitado, após, retorna ao modo novamente para economia de energia.

Na figura 01 é apresentado um comparativo entre protocolos muito utilizados hoje em dia.

Figura 01 – Comparação entre diferentes protocolos em relação à taxa de transmissão de dados e alcance.



Fonte: <http://www.projetoderedes.com.br/artigos/imagens/image32.gif>

É possível verificar que o protocolo ZigBee, juntamente com o Bluetooth e WIFI, apresentam alcance considerável de transmissão e recepção, mas o protocolo ZigBee tem a menor taxa de transmissão. Assim, é de grande valia o estudo e o levantamento de dados do projeto para a escolha de qual protocolo utilizar.

Por motivo da facilidade de aplicação, custo e grande acolhimento no mercado foi selecionado o protocolo ZigBee em relação aos outros, notando que estava totalmente apto ao projeto de seleção de itens.

3. MATERIAIS E MÉTODOS

Esta pesquisa é definida como do tipo experimental. Como afirma Gil (2010), estes tipos de pesquisas experimentais devem seguir e apresentar algumas propriedades como, manipulação, em que o pesquisador deve fazer algo para ser manipulado, e controle, em que deve ser adicionado um ou mais tipos de controle ao sistema experimental.

O local do desenvolvimento do projeto com seus devidos testes aconteceram no Laboratório de Ensino de Circuitos Elétricos e Eletrônica e Laboratório de Sistemas Embarcados da Universidade Federal de Santa Catarina – Campus Joinville, com o intuito de promover a construção de um sistema separador de itens utilizando RFID.

Para determinar as funções do projeto, foi escolhida uma placa microcontrolada, leitores, dispositivos e componentes eletrônicos já presentes no mercado. A finalidade é implementar a comunicação entre eles para realizar devidas funções na esteira. O intuito é apresentar um processo de inovação apostando em sistemas de baixo custo ao alcance do consumidor, sendo aplicado a diversos setores, tanto comercial quanto industrial.

O processo do sistema é dividido em três momentos. Primeiro, leitura da TAG, quando o item a ser colocado encima da esteira passar pelo leitor, assim irá ler sua identificação; Segundo, comunicação com o banco de dados, haverá este recurso para realizar análise do destino do item, utilizado comunicação sem fio com ZigBee's; e por fim a separação do item, ao ser lido e analisado o item, chegará ao final da esteira sendo separado para a direita, esquerda ou não separado (seguindo reto), quando a identificação não constar no banco de dados ou no caso de inexistência de cartão no item, indo para reconfiguração.

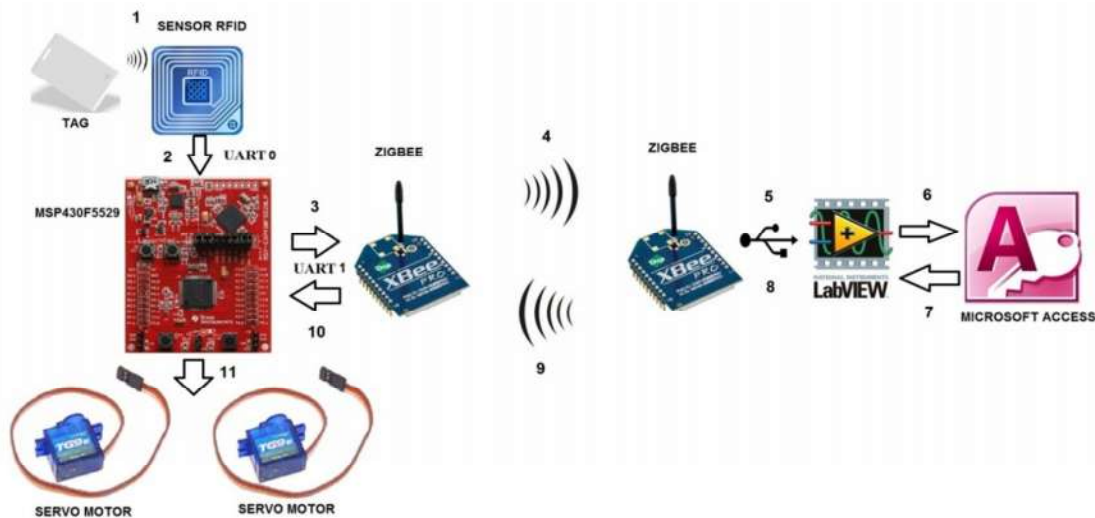
A montagem do sistema separador de itens inicia na estrutura mecânica do protótipo, onde tem a finalidade de construção uma esteira. É utilizado madeira do tipo MDF (Medium Density Fiberboard) para as laterais, onde há dois eixos

circulares de metal unindo-as, com função de rolar a lona da esteira. Assim, é fixado um motor de corrente contínua ao eixo para causar rotação e, portando, o funcionamento da mesma. Ao meio da esteira haverá um leitor, ou seja, onde o módulo RFID será acoplado. Ao final da esteira os servo-motores serão conectados um em cada lado, atuando como hastes separadoras. As outras placas usadas ao projeto não necessitam estar sobre a esteira, então serão fixadas próximas da mesma.

O projeto tem como objetivo a separação de itens utilizando RFID, realizando e comprovando conceitos adquiridos durante o curso de Engenharia Mecatrônica. Iniciando o funcionamento com o item sobre o protótipo de esteira, realizando a leitura de sua identificação e assim, fazendo comunicação sem fio com um banco de dados criado em um computador, então, ocorrendo à busca do item neste banco de dados e retornando o local que o mesmo deve ser separado.

Para facilitar a descrição do funcionamento do sistema, foi elaborado um fluxograma do projeto. Na figura 02 é apresentada a sequência crescente do funcionamento seguido pelo sistema, desde a leitura da tag até a atuação dos servo-motores.

Figura 02 – Fluxograma do funcionamento do projeto.



Fonte: Autor, 2015.

O funcionamento inicia com a leitura do cartão RFID (TAG), utilizando o sensor RFID (etapa um). Ao momento da leitura é captado o código deste cartão. Este código é, normalmente, de 32 bytes fornecidos então ao MSP430 via comunicação *UART0* (Universal asynchronous receiver/transmitter) pelo pino *Tx* do módulo RFID com um *baud rate*, isto é, velocidade da taxa de transmissão de dados, selecionado de 9600 bps (etapa dois).

O kit RFID utilizado apresenta dois cartões passivos de identificação, que trabalham numa faixa frequência de 125 kHz. É usando esta sequência de caracteres que acontecerá a comparação com o banco de dados para a decisão de separação do item (que serão explicadas nas próximas etapas).

No Anexo 1 é apresentado o código base usado para a aquisição do sinal vindo do sensor RFID utilizando a comunicação *UART0*. Este código foi retirado dos exemplos fornecidos pelo CCS usado para induzir a aquisição e manipulação do código via *UART0*.

A etapa três e quatro consiste o envio da mensagem, com um *baud rate* selecionado de 115200 bps via *UART1*, do MSP430 para o Zigbee, em que este mesmo recebe a informação e a envia para outro ZigBee via comunicação sem fio.

O segundo dispositivo ZigBee (junto ao computador) recebe esta informação e, como apresentado no fluxograma da figura 02, repassa ao LabVIEW via *UART* com *baud rate* de 115200 bps também, sendo usado um módulo XBee para a comunicação via USB, este usado como meio de troca de informações e definindo a etapa cinco. Nesta etapa o LabVIEW tem a função de interface do MSP430 com o sistema de banco de dados criado usando a ferramenta do pacote Office, o Access.

A partir deste momento inicia o retorno do local (1, 2 ou nada) definido para o microcontrolador, ou seja, ocorrerá o sistema inverso; O LabVIEW envia via *UART* ao ZigBee, que transmite para o outro dispositivo usado e assim, chegando ao MSP430. Chegando ao microcontrolador é, novamente, trabalhada a variável definida como *local*.

Assim, inicia a atuação das hastes seletoras. É possível notar no código do Apêndice B, que os servo-motores estão em 90° (estado inicial) ou 0° (atuando). Nele são apresentados três momentos que poderão ocorrer, como separação para o lado direito, separação para o lado esquerdo e sem separação, onde seguirá reto na esteira para análise e/ou reconfiguração.

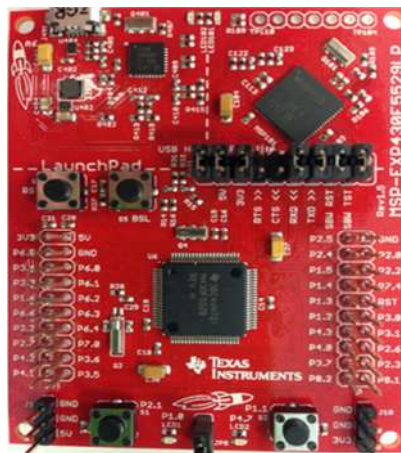
No projeto consta a utilização de uma placa microcontroladora, um kit MSP430F5529 Launch Pad, dois módulos ZigBee's (XBP09-DM e XBP09-DP), um kit RFID iniciante (leitor RFID ID-12, módulo RFID para conexão e duas tag's cartão distintas), dois micro servo-motores Tower Pro 9g acoplados a hastes servindo como um sistema de cancela para a separação de itens e um motor de limpador de para-brisa automotivo (12v corrente contínua) acoplado a um eixo para acionar a esteira.

Para tais escolhas de dispositivos do projeto, os quesitos para esta aquisição foram como, por exemplo, o MSP430F5529LP foi escolhido pelo motivo da familiaridade com este tipo de microcontrolador durante algumas disciplinas cursadas do curso de Engenharia Mecatrônica; o Kit RFID foi escolhido pela facilidade de uso e valor comercial; já o ZigBee foi utilizado pela facilidade de configuração e seu reconhecimento no setor industrial.

3.1. MSP430F5529 LaunchPad

O MSP430F5529LP (figura 03) é um kit usado para realizar emulações para programação e depuração de software de um modo facilitado, contém um microcontrolador MSP430F5529, além de botões e led's para adicionar rapidamente uma interface com o usuário, o kit consta com alguns tipos de comunicação, *timers*, sinais PWM, interrupções entre outras tarefas podendo ser utilizados para várias aplicações. Terá a finalidade de fazer a comunicação com o LabVIEW e com os servo-motores para a atuação.

Figura 03 – Kit utilizado no sistema - MSP430F5529LP.

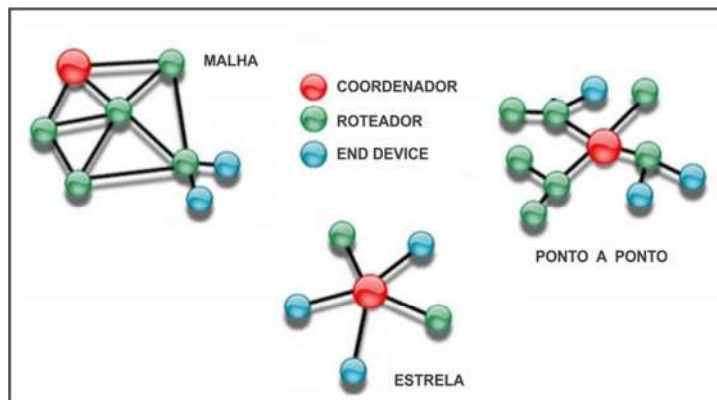


Fonte: http://www.deyisupport.com/cfs-file.ashx/__key/communityserver-discussions-components-files/55/6740.IMG_5F00_0070_5B00_1_5D00_.JPG

3.2. ZigBee

Uma das grandes vantagens do ZigBee é o fato de suportar redes em malha (figura 04), havendo vários caminhos possíveis. Dentro das topologias de rede podemos citar rede em estrela, árvore, malha e par. Independentemente do tipo de rede utilizada, o protocolo ZigBee permite até 65535 dispositivos por cada nó coordenado.

Figura 04 – Algumas topologias de rede do protocolo ZigBee.



Fonte: http://www.taskblog.com.br/wp-content/uploads/2013/04/ZigBee_interna.jpg

É possível montar redes apenas com um coordenador e dispositivos finais (end device), um coordenador e roteadores e redes com um coordenador, roteadores e dispositivos finais. Esta escolha de topologia é definida pela necessidade específica de cada projeto.

Deste modo, os dados são transmitidos para outros nós intermediários que fazem o redirecionamento da informação até que esta chegue ao destinatário. Na figura 05 é apresentado o módulo Zigbee utilizado para as funções de troca de informações no projeto seletor de itens.

Figura 05 – Módulo ZigBee.



Fonte: <http://www.embeddedstar.com/postimages/2010/Digi-XBee-PRO-ZB.jpg>

3.3. KIT RFID INICIANTE

O kit RFID (figura 06) é uma placa simples de usar. Nele será usado apenas o pino VCC, sendo alimentado com 5 volts; GND, comum da placa e TX, para transmissão e dados. Ao momento de aproximação de tag's passivas com uma frequência de 125 kHz ao RFID, apresentará um identificador de 32 bytes, uma sequência de caracteres. Cada cartão terá seu identificador.

Figura 06 – RFID Starter Kit.



Fonte:

http://www.labdegaragem.org/loja/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/i/m/image_306.jpg

3.4. SERVO-MOTOR

Os servo-motores (figura 07) são usados em várias aplicações quando deseja-se movimentar algo de forma precisa e controlada. Sua principal característica é a sua capacidade de movimentar sua haste até uma dada posição, ou seja, recebem um sinal de controle, verificam a posição atual e atuam no sistema indo para a posição desejada.

Figura 07 – Servo-motor Turnigy TG9.



Fonte: [http://cdn.hobbyking.com/hobbyking/store/catalog/9549\(2\).jpg](http://cdn.hobbyking.com/hobbyking/store/catalog/9549(2).jpg)

Os servo-motores são alimentados com tensões de 5v e recebem um sinal de formato PWM (Pulse Width Modulation), com sua largura de pulso variando de 0 a 1.

3.5. FONTE DE ALIMENTAÇÃO CONTROLÁVEL

Para alimentação da placa microcontroladora e demais componentes eletrônicos foi usado uma fonte de tensão modelo ICEL MANAUS PS-4000 0-30v 3A (figura 08), fornecendo uma tensão de 5 volts ao MSP430, e assim sucessivamente aos demais componentes dependentes do mesmo.

Figura 08 – Fonte de tensão utilizada para alimentação do sistema.



Fonte: Adaptado de

http://www.casamirandaeletron.com.br/media/catalog/product/cache/1/image/640x468/17f82f742ffe127f42dca9de82fb58b1/f/o/fonte_de_alimenta_o_-_ps-4000_-_icel_n_o_tem.jpg

3.6. PROTÓTIPO

Como mencionado, a esteira (figura 09) foi construída com madeira do tipo MDF, separadas com dois eixos retirados de sucatas eletrônicas (aparelhos de Fax) e lona de esteira ergométrica. Junto a ela foi acoplado um motor de limpador de para-brisa 12 volts de corrente contínua para forçar o acionamento da mesma.

Figura 09 – Esteira confeccionada para a seleção de itens.



Fonte: Autor.

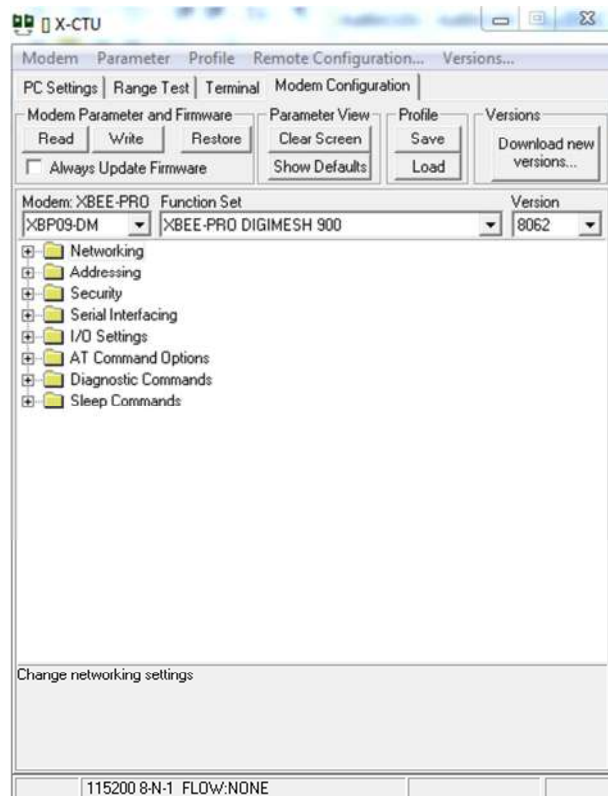
3.7. X-CTU

Para a programação e configuração dos ZigBee's foi utilizado o software da DIGI International, o X-CTU.

O X-CTU é um software de plataforma livre projetado para permitir a interação com os módulos ZigBee's. Nele consta uma interface simples utilização, incluindo novas ferramentas que tornam mais fácil a configuração dos dispositivos.

Há uma ampla diversidade de configurações disponíveis no programa, como apresentado na figura 10.

Figura 10 – Software utilizado para programação e configuração dos ZigBee's.



Fonte: Autor, 2015.

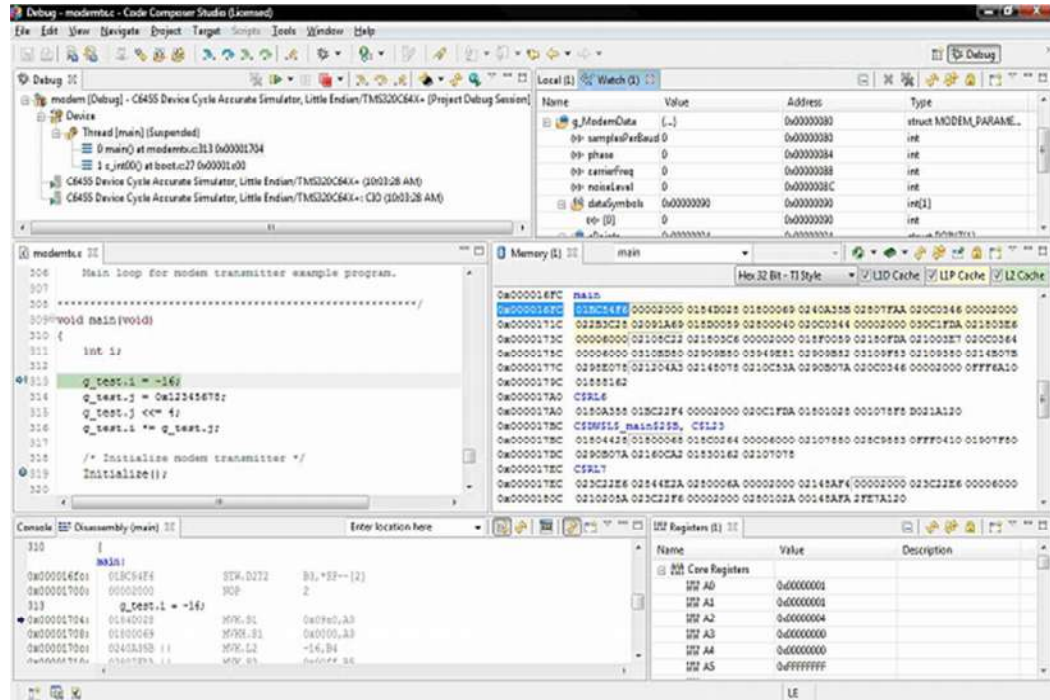
3.8. LABVIEW

Usando como interface para a comunicação do microcontrolador com o banco de dados criado, foi utilizado o software LabVIEW.

LabVIEW é o software base da plataforma de projeto da National Instruments. Sua utilização é ideal para desenvolvimento de vários sistemas de controle ou medição. Nele, aplicam-se muitas ferramentas e que os engenheiros precisam para o desenvolvimento de projetos. O LabVIEW é um ambiente de desenvolvimento voltado a uma ampla utilização em diversas aplicações. A

Para facilitar a avaliação da implementação, o CCS fornece, na interface de compilação, uma análise linha a linha durante a depuração, facilitando a procura por erros ou apenas para melhor entendimento, como mostrado na figura 12.

Figura 12 – Exemplo de aplicação do CCS.



Fonte:

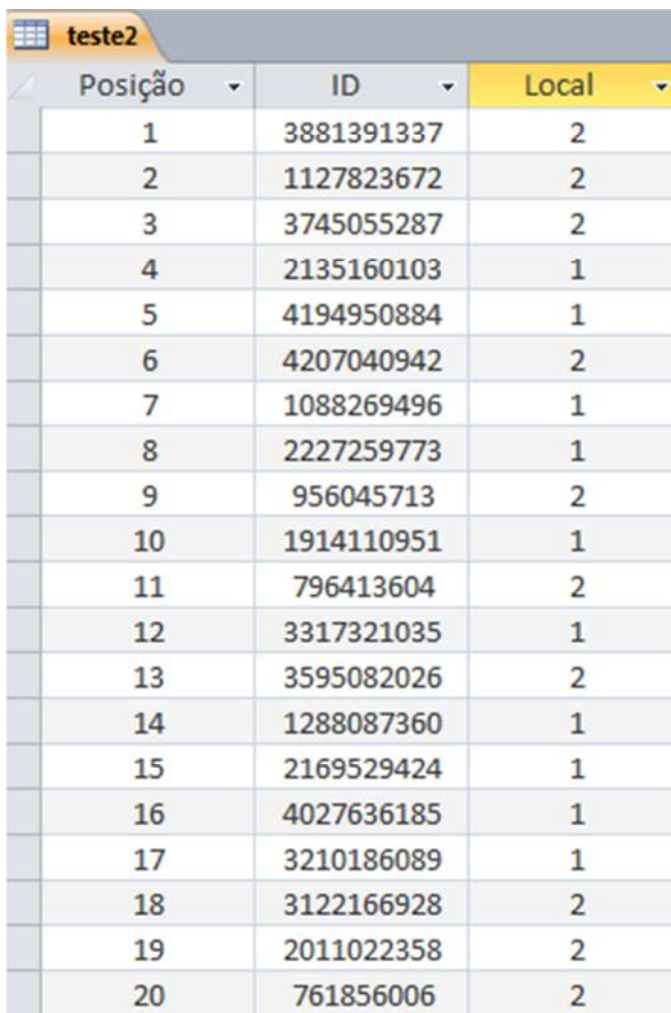
http://www.element14.com/community/servlet/JiveServlet/showImage/145724/ccs_screen_hot_1_sm.jpg

3.10. MICROSOFT OFFICE ACCESS

Para criação de um banco de dados, que será utilizado para conferir as tag's cadastradas, utilizaremos a ferramenta da Microsoft Office, o Access.

O Access é um sistema de gerenciamento de banco de dados muito conhecido da Microsoft. Nele é permitido criações e desenvolvimento de sistemas de banco de dados de modo simples e rápido. Na figura 13 é apresentada uma parte do banco de dados criado para o protótipo, nele apenas foram inseridos os dois cartões de identificação do kit RFID (formato inteiro) juntamente com mais de um milhão de itens aleatórios e seus devidos locais (locais para o qual deverá ser separado), assim tendo uma perspectiva real de um banco de dados.

Figura 13 - Banco de dados usando o MO Access.



Posição	ID	Local
1	3881391337	2
2	1127823672	2
3	3745055287	2
4	2135160103	1
5	4194950884	1
6	4207040942	2
7	1088269496	1
8	2227259773	1
9	956045713	2
10	1914110951	1
11	796413604	2
12	3317321035	1
13	3595082026	2
14	1288087360	1
15	2169529424	1
16	4027636185	1
17	3210186089	1
18	3122166928	2
19	2011022358	2
20	761856006	2

Fonte: Autor, 2015.

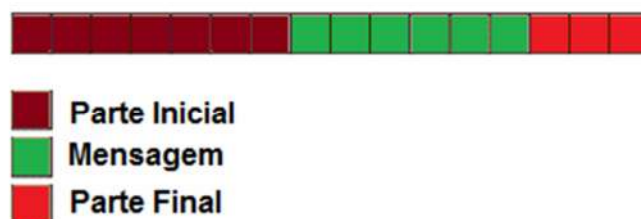
3.11. ANÁLISES E CONFIGURAÇÕES

No levantamento de dados, foi notado que cada cartão apresenta três partes em sua sequência de bytes, sendo o primeiro e o terceiro idênticos em todas as tag's, e o segundo estado sendo a mensagem em si. Assim, para evitar problemas relacionados a esta leitura e também para diminuir o tamanho de mensagem na troca de informações, foram feitas manipulações com as mensagens lidas, como apresentado no Apêndice B.

A sequência de bytes completa foi salva em um vetor para, então, facilitar o manuseio das checagens. A primeira etapa é a análise da primeira e terceira parte,

somente se os dois forem idênticos aos vetores bases criados seguirá a segunda etapa que consiste no envio da mensagem. A figura 14 apresenta detalhadamente o vetor e os estados da informação.

Figura 14 – Descrição do vetor de informação de uma tag.



Fonte: Autor, 2015.

No Apêndice B é apresentado o código final do projeto. Podemos notar que foram feitas muitas manipulações com a mensagem recebida no MSP430. O primeiro passo foi uma série de checagens com o intuito de assegurar uma alta confiabilidade na troca de mensagens e na separação dos itens.

Como mencionado, a mensagem contida ao meio do vetor é a identificação de cada cartão, sendo assim, apenas sendo a informação enviada ao ZigBee fixado no protótipo.

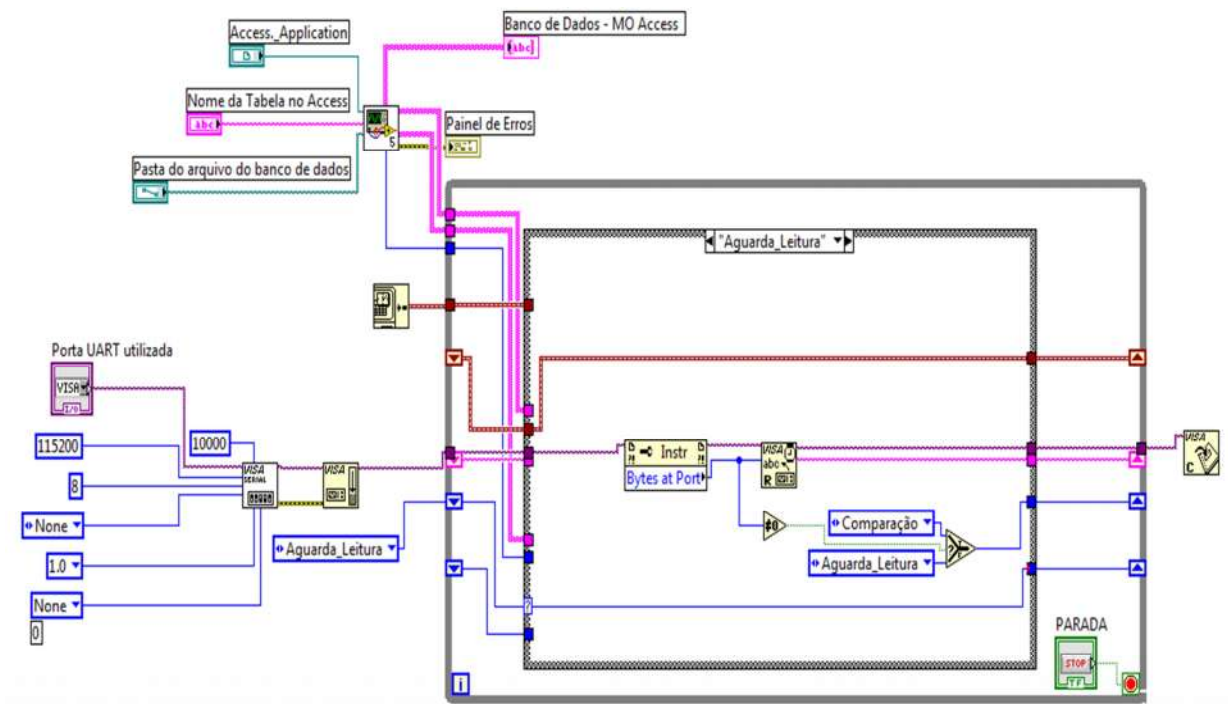
É notado que a parte inicial é composta por 7 bytes e a parte final por 3 bytes. Estas duas partes serão iguais independente do cartão de identificação, sendo apenas diferente a mensagem de cada cartão, composto por 6 bytes.

Para a configuração dos ZigBee's foi usado o software livre X-CTU (figura 09). Nele foi possível alterar algumas opções importantes para o correto funcionamento. Primeiro foi necessário a configuração da opção *ID – Modem VID*, onde para realizar o pareamento entre os dois dispositivos é necessário que os dois apresentem o mesmo *ID*, sendo usado no projeto *1234*. Segundo passo é a configuração do *HP – Hopping Channel*, ou seja, é necessário que os dois dispositivos estejam em mesmo canal para possibilitar a troca de informações, o canal usado foi o *1*. Terceiro passo é a definição de qual dispositivo será usado como *Coordinator (coordenador)* e *End Device (dispositivo final)*; assim foi definido o dispositivo anexado ao MSP430 como dispositivo final e o outro, junto ao computador, como coordenador. O quarto passo é a desativação da ferramenta *RTS FLOW CONTROL*, situado na opção *D6 – DIO6 Configuration*, esta ferramenta tem

por função avisar o recebimento de informação e sendo desnecessária para nossa aplicação. Para finalizar foi criado um sistema de segurança à rede, foi ativado na opção *Security* adicionando uma senha hexadecimal para a mesma, sendo *ABCDE12345*.

A programação inicial do LabVIEW foi o acesso e busca de dados em uma tabela do Access. Para isso, foi configurado com as ferramentas necessárias para este acesso (figura 15) ao banco de dados, buscando a tabela criada, chamada *teste2*, como mostrada na figura 13. Então, foi configurada a comunicação *UART*, ou seja, definido para recebimento da informação vinda do ZigBee.

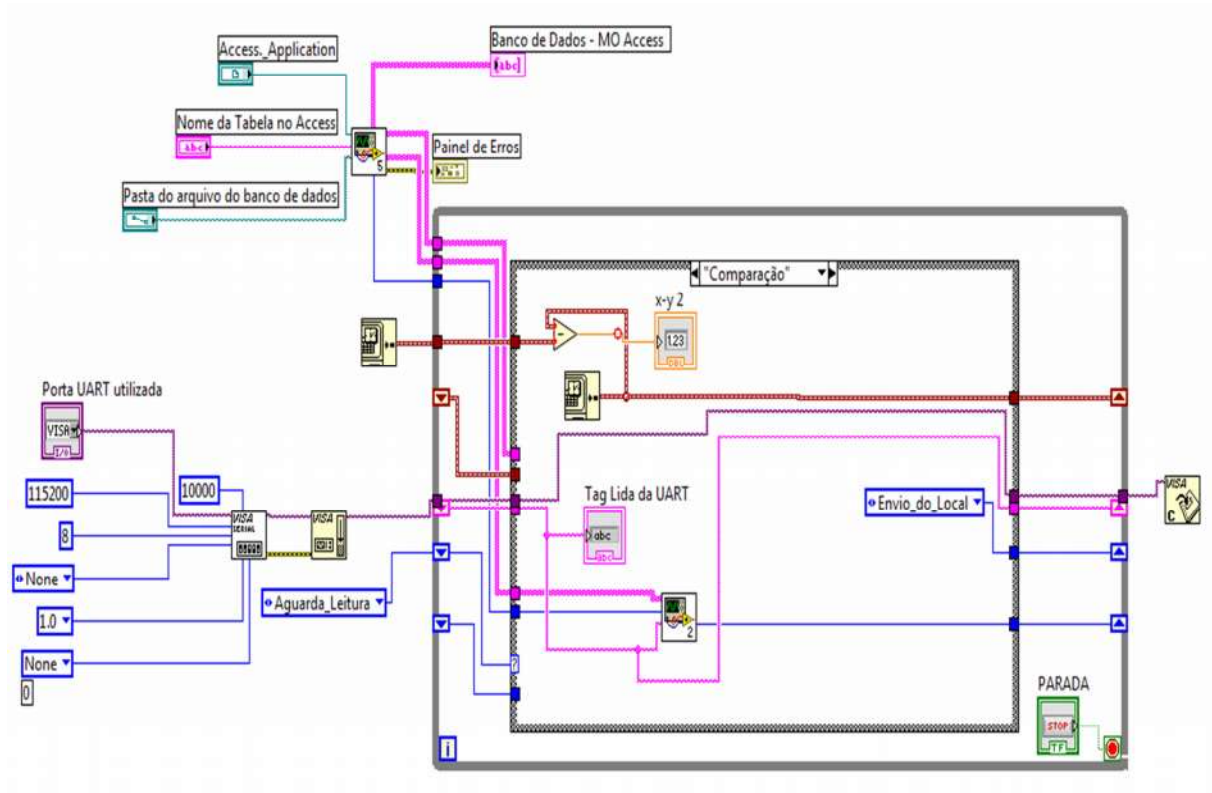
Figura 15 – Programação e configuração para acesso ao banco de dados e definição de local – Estrutura 1.



Fonte: Autor, 2015.

Para o correto funcionamento foi criado estruturas na programação do LabVIEW. Na figura 15 é apresentada a estrutura inicial, *Aguardando_Leitura*, que tem a função de manter sempre em funcionamento (*loop*) até ao momento de realizar uma leitura do leitor RFID, assim passando para a próxima estrutura. Na figura 16 é mostrada a segunda estrutura, definida como *Comparação*.

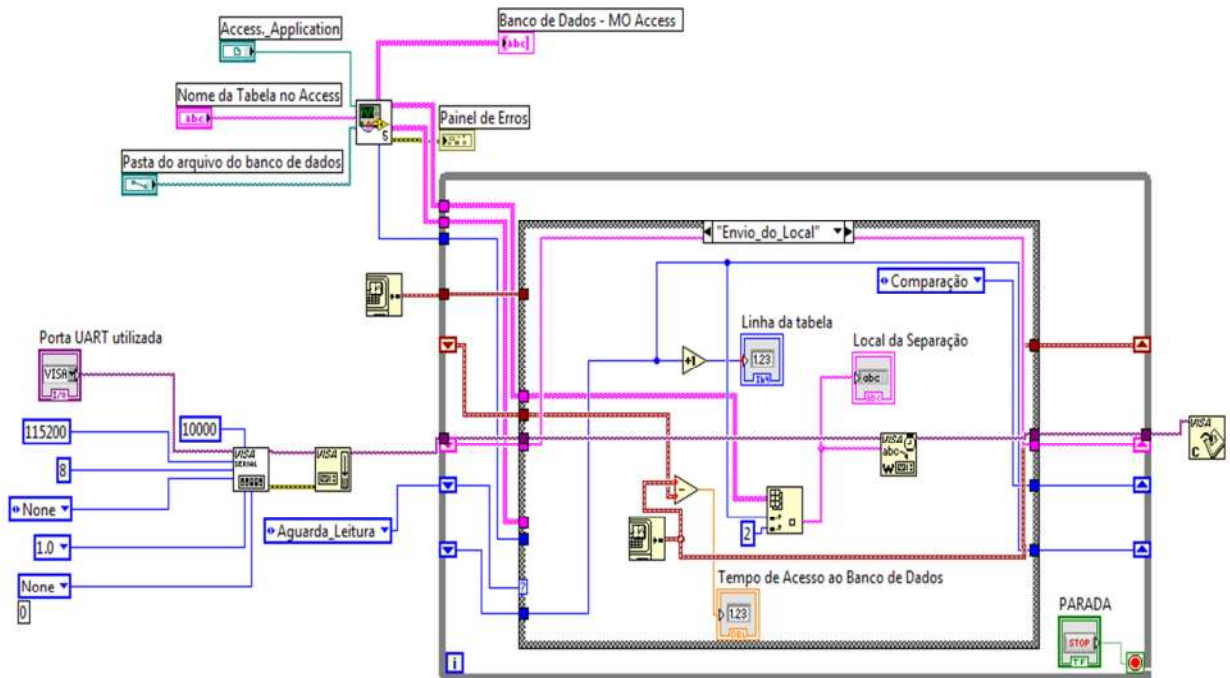
Figura 16 – Programação e configuração para acesso ao banco de dados e definição de local – Estrutura 2.



Fonte: Autor, 2015.

Nesta etapa é feita a comparação da informação lida pelo leitor RFID e recebida pelo LabVIEW com as informações do banco de dados. É realizada uma varredura no banco de dados e ao momento de igualdade é, automaticamente, lido o seu devido local, no caso 1 ou 2. E assim, é definida a terceira estrutura, chamada *Envio_do_Local*, como mostra a figura 17.

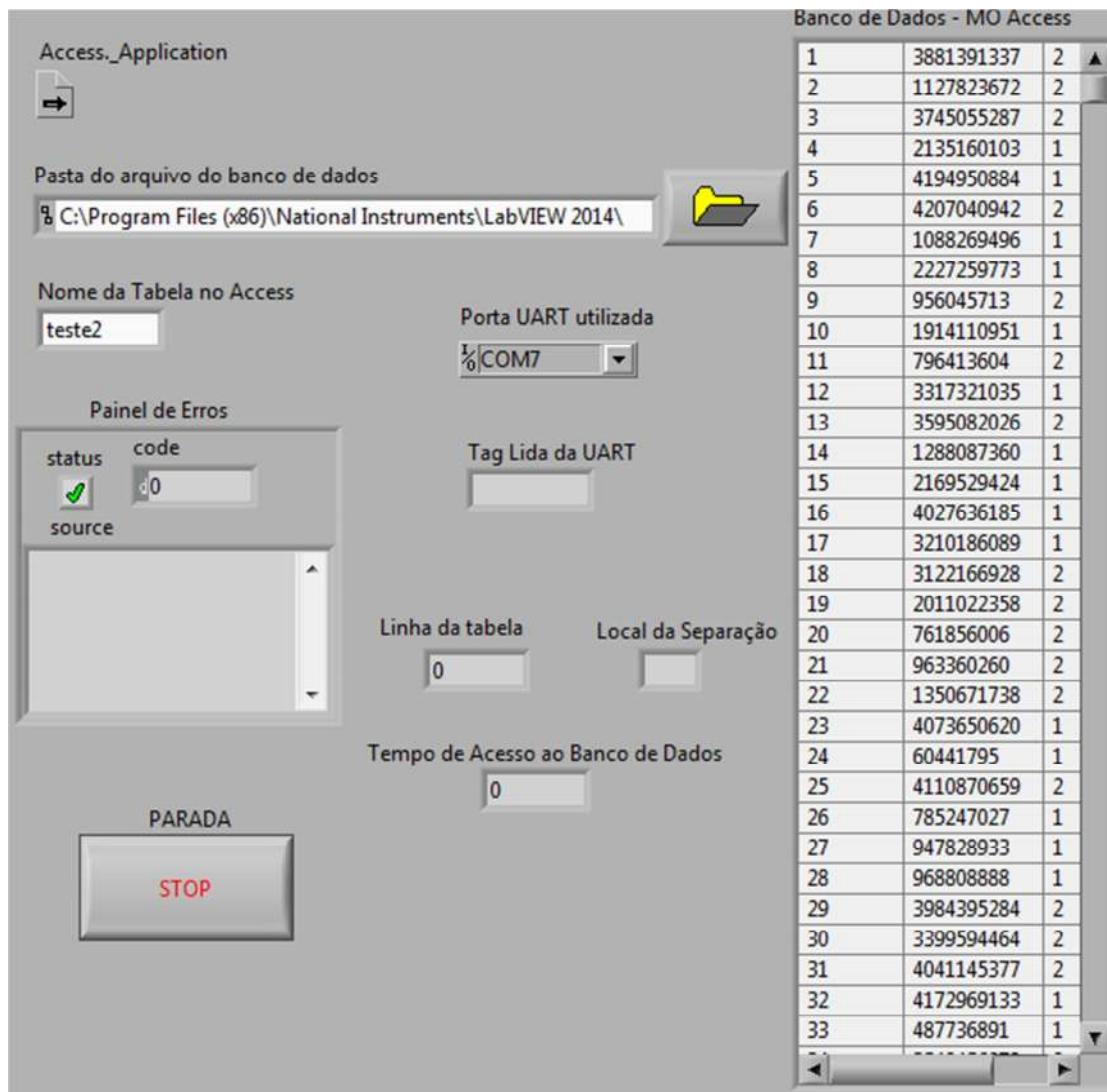
Figura 17 – Programação e configuração para acesso ao banco de dados e definição de local – Estrutura 3.



Fonte: Autor, 2015.

Finalmente a etapa da apresentação do local a ser separado. Na figura 17 é encontrado o valor do local que deve ser separado e assim enviado ao MSP430. A interface definida é mostrada na figura 18.

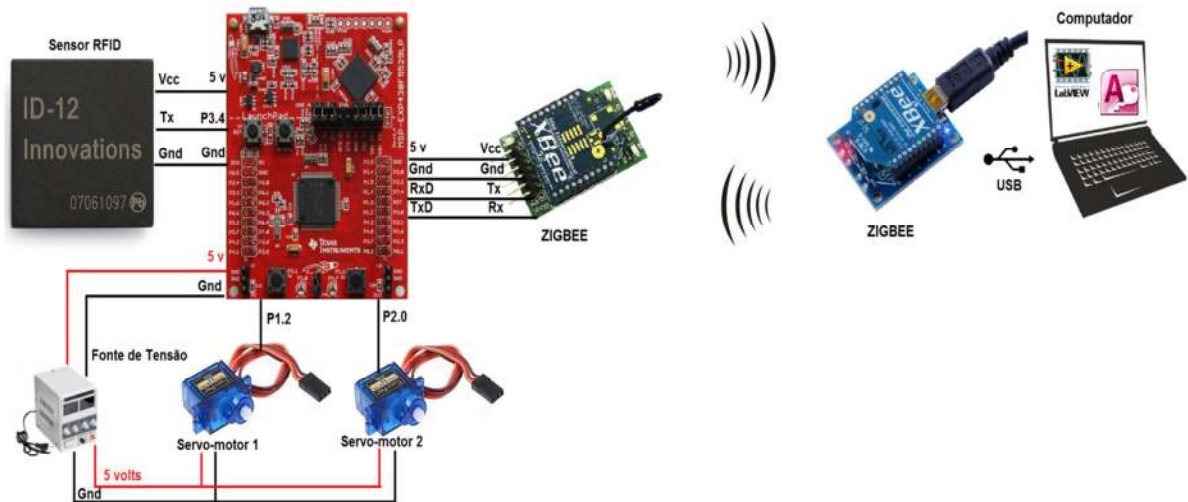
Figura 18 – Interface criada para a comunicação usando LabVIEW.



Fonte: Autor, 2015.

Para a montagem da parte eletrônica do projeto foi definido alguns pinos e alimentação externa para tornar um sistema autônomo. A figura 19 apresenta o esquema de ligação do mesmo, apresentando seus pinos e devidas ligações.

Figura 19 – Esquema de ligação do projeto.



Fonte: Autor, 2015.

Um tópico importante a ser citado é o aparecimento de erros por motivo de inexistência de cartão de identificação, falha na leitura, problemas externos ou outros motivos que podem atrapalhar a separação de itens. Além das checagens ocorridas com os cartões já citadas, é apresentado na programação final no Apêndice B que ao momento de não retorno do local, em torno de 2 segundos, o sistema será reinicializado, sendo possível uma nova leitura e continuação do processo.

Para aquisição do tempo de acesso ao banco de dados, foi implementado uma rotina no LabVIEW para calcular o tempo desta procura por itens. O banco de dados compreende cerca de 1,048 milhões de itens, ou seja, sua identificação poderá estar no começo, meio ou fim desta lista, assim, pôde-se avaliar que o tempo da varredura de todo o banco de dados foi de 0,47 segundos, assim denominando o tempo máximo de leitura. Este tempo é assunto muito discutido em empresas, pois muitas visam produções em menor tempo possível, sendo assim, esta rápida busca de informações e de separações viriam como aspecto positivo para as mesmas.

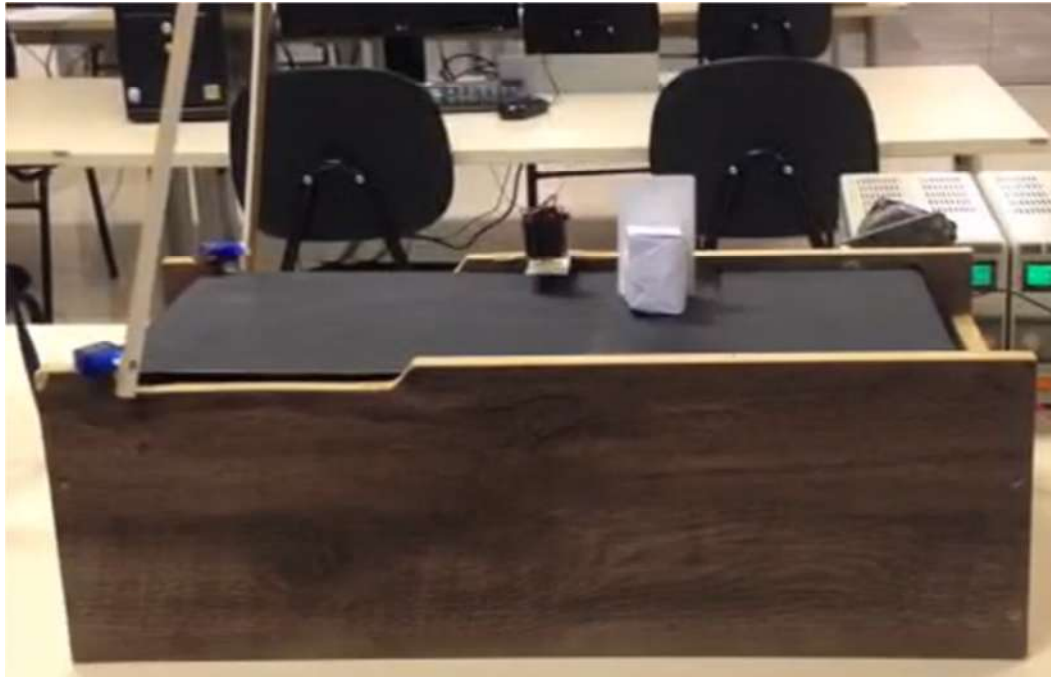
Sendo custos o assunto discutido, foi elaborada uma tabela relacionando todos os componentes eletrônicos essenciais para o sistema e seus respectivos valores, a mesma é apresentada no Apêndice A.

Outro tópico interessante é a intensidade do sinal da rede ZigBee. Realizado testes de bancada foi notado que o sistema trabalhando em um ambiente sem

obstáculos pôde trocar informações em uma distância de na base de 150 metros. Já para o outro caso, com obstáculos, estipulando um ambiente fabril, foi em torno de 60 metros entre os dispositivos ZigBee's. Assim, sendo um interessante valor para muitas aplicações neste ramo.

O resultado da montagem do sistema final é mostrado na figura 20.

Figura 20 – Projeto finalizado.



Fonte: Autor, 2015.

CONCLUSÃO

É notado que o avanço de meios de comunicação vem crescendo muito nas últimas décadas, principalmente quando o assunto é comunicação sem fio. É grande a procura por sistemas automatizados utilizando protocolos de comunicação sem fio pela facilidade de operação e rápidas trocas de informações em lugares distintos.

O sistema baseou-se em um separador de itens utilizando a ferramenta RFID e o protocolo ZigBee para comunicação sem fio com um banco de dados criado no Microsoft Office Access, todo este sistema sendo aplicado a um protótipo de esteira, que por fim, pode vir a ser aplicado em área fabris onde deseja-se realizar separação de itens com confiabilidade e menor custo.

É de grande valia citar o baixo tempo de resposta do sistema, ou seja, o tempo desde a leitura do cartão de identificação, a busca no banco de dados e a resposta do local que deve ocorrer à separação. É muito importante trabalhar com tempos pequenos para poder acelerar um dado método de separação. Outro tópico importante que deve ser citado é as rotinas de segurança criadas e implementadas para o momento de falhas inesperadas durante o funcionamento do sistema, falhas que podem ser por software ou hardware.

Como trabalhos futuros é de grande interesse implementar o controle de velocidade da esteira, ou seja, acionar a esteira utilizando sistemas de conversores CC-CC, assim variando o tempo de separação de itens. Outra tarefa interessante é diminuir o tempo de busca ao banco de dados, mas isso implica em trabalhar com dispositivos eletrônicos mais eficientes. Usar comunicação WIFI seria um grande desafio aplicado ao projeto, substituir o protocolo ZigBee pelo protocolo WIFI agregaria muito valor a ideia do sistema.

Ao desenvolver este projeto, foi percebido o amplo grau de aprendizado e o aprimoramento do conhecimento em determinadas áreas da mecatrônica, pelo motivo de este projeto unir vários conceitos abordando eletrônica analógica, microcontroladores, programação em linguagem C e sistemas de comunicação.

REFERÊNCIAS

BUENO, Luciano; COSTA, Marco Aurélio. **Automação de uma esteira seletora por meio de CFTV**. Monografia apresentada como requisito para o título de Tecnólogo no Curso Superior de Tecnologia em Eletrônica – Modalidade Automação de Processos Industriais da Universidade Tecnológica Federal do Paraná - UTFPR. Curitiba. 2011. 68 f. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/924/1/CT_COMET_2011_2_02.pdf>. Acesso em: 04.nov.2014.

DELLA LUCIA, Felipe Lorenzo. **Desenvolvimento de um sistema de gestão de iluminação pública através de redes de sensores e atuadores sem fio**. Tese apresentada como requisito para o título de Doutor em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas – UNICAMP. Campinas, 2014.

FRANCO, Rebeca Such Tobias; COSTA, Yandre Maldonado; COSTA, Gomes da; UBER, Flávio Rogério; GONÇALVES, Paulo César. Tecnologia RFID: Um estudo de caso aplicado a uma indústria de bebidas de Maringá - PR. **Diálogos & Saberes**, Mandaguari, v. 5, n. 1, p. 9-18, 2009. Disponível em: <<http://seer.fafiman.br/index.php/dialogosesaberes/article/viewFile/39/19>>. Acesso em: 04.nov.2014. Só se coloca até 3 nomes. Mais de 3 se coloca só o 1º nome e depois et al.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 5 ed.. São Paulo: Editora Atlas, 2010.

JACINTO, Juliano; HEIL, Luciano; SOUZA, Márcio Fernandes; RODRIGUES, Sidnei. **Logística: O endereçamento como ferramenta fundamental na armazenagem e estocagem**. Faculdade de Tecnologia de Santa Catarina. Disponível em: <http://www.bm.edu.br/fatesc.edu.br/wp-content/blogs.dir/3/files/pdf/tccs/o_endereçamento_como_ferramenta_fundamental_na_armazenagem.pdf>. Acesso em: 15.set.2014.

JUNIOR, Joel Andrelo. **RFID: identificação por radiofrequência**. Monografia apresentada como Trabalho de Conclusão do Curso de Tecnólogo em Informática para Gestão de Negócios da Faculdade de Tecnologia de Praia Grande. Praia Grande. 2007. Disponível em: <<http://br.monografias.com/trabalhos3/rfid-identificacao-radiofrequencia/rfid-identificacao-radiofrequencia3.shtml>>. Acesso em: 04.nov.2014

KAWANO, Bruno Rógora; CUGNASCA, Carlos Eduardo. **RFID nas universidades brasileiras**. Escola Politécnica da USP. São Paulo. 2014. Disponível em: <<http://brasil.rfidjournal.com/artigos/vision?11984>>. Acesso em: 04.nov.2014.

MARQUES, Carlos Alberto; JUNIOR, Valdir Furlan; MUNIZ, Jorge; CHAVES, Carlos Alberto; URIAS, Ângelo. A tecnologia de identificadores de rádio frequência (RFID) na logística exploratória numa empresa de usinado para o setor aeroespacial. **GEPROS – Gestão da Produção, Operações e Sistemas**. São Paulo, ano 4, n. 2, p. 109-122, abr./jun. 2009. Disponível em:

<revista.feb.unesp.br/index.php/gepros/article/download/750/229>. Acesso em: 04.nov.2014.

MITCHELL, Michael. **MSP430 Launchpad+PWM**. Disponível em: <<http://mitchtech.net/msp430-launchpad-pwm/>>. Acesso em: 20.abr.2015.

RODRIGUES, Renato A.; MEDEIROS, Justino; BITTENCOURT, Andrea. Desenvolvimento de uma esteira transportadora didática. In: **Anais do VII CONNEPI – Congresso Norte Nordeste de Pesquisa e Inovação**. Palmas - TO, 2012. Disponível em: <<http://propi.ifto.edu.br/ocs/index.php/connepi/vii/paper/viewFile/3430/2460>>. Acesso em: 04.nov.2014.

SALEIRO, Mário; EY, Emanuel. **ZigBee – uma abordagem prática**. Universidade do Algarve – Escola Superior de Tecnologia. S.d. Disponível em: <http://lusorobotica.com/ficheiros/Introducao_ao_Zigbee_-_por_msaleiro.pdf>. Acesso em: 06.mai.2015.

VIDAL, Luciano. **TCC Automação**. Disponível em: <<http://www.lucianovidal.com.br/tcc-automacao>>. Acesso em: 04.nov.2014.

VIECELLI, Bruno Henrique; CARDOSO, Felipe; ARNAUT, Paulo Einstein Melges. **Automação Esteira Seletora**. Trabalho apresentado como Projeto na disciplina de Microprocessadores II do Curso de Engenharia da Computação. Pontifícia Universidade Católica do Paraná – PUC. Curitiba. 2009. Disponível em: <http://engcomp.org/sites/default/files/proj-files/PI_MicroII_V2.pdf>. Acesso em: 04.nov.2014.

ANEXOS

ANEXO 1 – CÓDIGO BASE PARA COMUNICAÇÃO UART

```

#include "msp430.h"

void main(void)
{
    volatile unsigned int i;

    WDTCTL = WDTPW + WDTHOLD;           // Stop WDT
    P1SEL |= BIT3+BIT4;                 // P1.3,1.4 = USART0 TXD/RXD
    do
    {
        IFG1 &= ~OFIFG;                // Clear OSCFault flag
        for (i = 0x47FF; i > 0; i--);  // Time for flag to set
    }
    while ((IFG1 & OFIFG));             // OSCFault flag still set?

    ME1 |= UTXE0 + URXE0;               // Enable USART0 TXD/RXD
    U0CTL |= CHAR;                      // 8-bit character
    U0TCTL |= SSEL1;                    // UCLK= SMCLK
    U0BR0 = 10;                         // 1MHz 115200
    U0BR1 = 0x00;                       // 1MHz 115200
    U0MCTL = 0x00;                      // 1MHz 115200 modulation
    U0CTL &= ~SWRST;                    // Initialize USART state machine
    IE1 |= URXIE0;                      // Enable USART0 RX interrupt
    P1SEL2 |= BIT0;                     // Set SMCLK at P1.0

    __bis_SR_register(LPM0_bits + GIE); // Enter LPM0 w/ interrupt
}

#if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
#pragma vector=USART0RX_VECTOR
__interrupt void USART0_RX (void)
#elif defined(__GNUC__)
void __attribute__((interrupt(USART0RX_VECTOR))) USART0_RX (void)
#else
#error Compiler not supported!
#endif
{
    while (!(IFG1 & UTXIFG0));          // USART0 TX buffer ready?
    TXBUF0 = RXBUF0;                   // RXBUF0 to TXBUF0
}

```

ANEXO 2 – CÓDIGO BASE PARA ACIONAMENTO DOS SERVO-MOTORES

```
#include "msp430g2553.h" // make sure you change the header to suit your particular device.

#define MCU_CLOCK          1000000

#define PWM_FREQUENCY      46    // In Hertz, ideally 50Hz.

#define SERVO_STEPS        180   // Maximum amount of steps in degrees (180 is
common)

#define SERVO_MIN          650   // The minimum duty cycle for this servo

#define SERVO_MAX          2700  // The maximum duty cycle

unsigned int PWM_Period    = (MCU_CLOCK / PWM_FREQUENCY);    //PWM Period

unsigned int PWM_Duty      = 0;    // %

void main (void){

    unsigned int servo_stepval, servo_stepnow;

    unsigned int servo_lut[ SERVO_STEPS+1 ];

    unsigned int i;

    // Calculate the step value and define the current step, defaults to minimum.

    servo_stepval = ( (SERVO_MAX - SERVO_MIN) / SERVO_STEPS );

    servo_stepnow = SERVO_MIN;

    // Fill up the LUT

    for (i = 0; i<= SERVO_STEPS; i++) {

        servo_stepnow += servo_stepval;

        servo_lut[i] = servo_stepnow;

    }

    // Setup the PWM, etc.

    WDTCTL      = WDTPW + WDTHOLD;    // Kill watchdog timer

    TACCTL1     = OUTMOD_7;           // TACCR1 reset/set

    TACTL = TASSEL_2 + MC_1;    // SMCLK, upmode

    TACCR0      = PWM_Period-1;    // PWM Period

    TACCR1      = PWM_Duty;        // TACCR1 PWM Duty Cycle

    P1DIR |= BIT2;                // P1.2 = output
```

```

P1SEL |= BIT2;          // P1.2 = TA1 output

// Main loop
while (1){
    // Go to 0°
    TACCR1 = servo_lut[0];
    __delay_cycles(1000000);

    // Go to 45°
    TACCR1 = servo_lut[45];
    __delay_cycles(1000000);

    // Go to 90°
    TACCR1 = servo_lut[90];
    __delay_cycles(1000000);

    // Go to 180°
    TACCR1 = servo_lut[179];
    __delay_cycles(1000000);

    // Move forward toward the maximum step value
    for (i = 0; i < SERVO_STEPS; i++)
    {
        TACCR1 = servo_lut[i];
        __delay_cycles(20000);
    }

    // Move backward toward the minimum step value
    for (i = SERVO_STEPS; i >= 0; i--) {
        TACCR1 = servo_lut[i];
        __delay_cycles(20000);
    }
}
}

```


APÊNDICES**APÊNDICE A – TABELA DE CUSTO DOS DISPOSITIVOS DO PROJETO**

Tabela 01 – Custos dos equipamentos eletrônicos.

QUANTIDADE	ITEM	VALOR (unidade)
1	Kit RFID iniciante	R\$ 229,00
1	MSP430F5529 LaunchPad	R\$ 38,00
2	Módulo ZigBee	R\$ 160,00
2	Shield XBee	R\$ 39,00
TOTAL:		R\$ 466,00

Fonte: <http://labdegaragem.com/>, <https://www.robocore.net/>.

APÊNDICE B – IMPLEMENTAÇÃO FINAL

```

#include <msp430.h> // Importa a biblioteca da família MSP430

extern char vetor[15]={0}; //vetor para guardar a mensagem recebida do sensor RFID
extern int flag=0; // Variável Global.
extern int i=0; // Variável Global para contagem.

#define MCU_CLOCK 1000000 //1 MHz
#define PWM_FREQUENCY 46 // In Hertz, ideally 50Hz.
#define SERVO_STEPS 180 // Maximum amount of steps in degrees
#define SERVO_MIN 650 // The minimum duty cycle for this servo
#define SERVO_MAX 2700 // The maximum duty cycle
unsigned int PWM_Period = (MCU_CLOCK / PWM_FREQUENCY); // PWM Period
unsigned int PWM_Duty = 0; // %

int main(void)
{
int j, k, l, codigo_valido, aguardando_resposta=0, contador=0; // Declaração de variáveis
int local = 0;
char ref1[7] = {0x02,0x37,0x31,0x30,0x30,0x32,0x35}; // Dados do início da mensagem
char ref2[3] = {0x0D,0x0A,0x03}; // Dados do final da mensagem

unsigned int servo_stepval, servo_stepnow; //Variáveis para utilização do servo-motor
unsigned int servo_lut[SERVO_STEPS+1], q; //Variáveis para utilização do servo-motor

WDTCTL = WDTPW + WDTHOLD; // Stop WDT
P3SEL |= BIT3 + BIT4; // P3.3,4 = USCI_A0 - Declaração dos pinos da UART 0
P4SEL |= BIT4 + BIT5; // P4.4,5 = USCI_A1 - Declaração dos pinos da UART 1

// Inicialização da Uart 0
UCA0CTL1 |= UCSWRST; // **Put state machine in reset**
UCA0CTL1 |= UCSSEL_2; // SMCLK
UCA0BR0 = 104; // 1MHz 9600 baudrate
UCA0BR1 = 0; // 1MHz 9600 baudrate
UCA0MCTL |= UCBRS_1 + UCBRF_0; // Modulation UCBRSx=1, UCBRFx=0
UCA0CTL1 &= ~UCSWRST; // **Initialize USCI state machine**

// Inicialização da Uart 1
UCA1CTL1 |= UCSWRST; // **Put state machine in reset**
UCA1CTL1 |= UCSSEL_2; // SMCLK
UCA1BR0 = 9; // 1MHz 115200
UCA1BR1 = 0; // 1MHz 115200
UCA1MCTL |= UCBRS_1 + UCBRF_0; // Modulation UCBRSx=1, UCBRFx=0
UCA1CTL1 &= ~UCSWRST; // **Initialize USCI state machine**

UCA0IE |= UCRXIE; // Enable USCI_A1 RX interrupt

__bis_SR_register(GIE); // Enter LPM0, interrupts enabled
__no_operation(); // For debugger

// Setup the PWM, servo 1
TA0CCTL1 = OUTMOD_7; // TACCR1 reset/set
TA0CTL = TASSEL_2 + MC_1; // SMCLK, upmode
TA0CCR0 = PWM_Period-1; // PWM Period
TA0CCR1 = PWM_Duty; // TACCR1 PWM Duty Cycle

```

```

    // Setup the PWM, servo 2
    TA1CCTL1 = OUTMOD_7;           // TACCR1 reset/set
    TA1CTL   = TASSEL_2 + MC_1;   // SMCLK, upmode
    TA1CCR0  = PWM_Period-1;     // PWM Period
    TA1CCR1  = PWM_Duty;         // TACCR1 PWM Duty Cycle

    P1DIR |= BIT2;                // P1.2 = output    Servo 1
    P1SEL |= BIT2;                // P1.2 = TA0.1 output

    P2DIR |= BIT0;                // P2.0 = output    Servo 2
    P2SEL |= BIT0;                // P2.0 = TA1.1 output

    // Calculate the step value and define the current step, defaults to
    minimum. servo_stepval = ( (SERVO_MAX - SERVO_MIN) / SERVO_STEPS );
    servo_stepnow = SERVO_MIN;

    // Fill up the LUT
    for (q = 0; q <= SERVO_STEPS; q++)
    {
        servo_stepnow += servo_stepval;
        servo_lut[q] = servo_stepnow;
    }

    TA1CCR1 = servo_lut[90];      // Iniciar servo motor 1 em posição de 90º
    TA0CCR1 = servo_lut[90];      // Iniciar servo motor 2 em posição de 90º

    while(1)
    {
    if (aguardando_resposta == 0) // Enquanto não retorna resposta do Labview
    {

        if (flag==1) // Há informação nova então...
        {
            codigo_valido=1;
            for(j=0; j<7; j++) // Analisando a parte inicial do vetor.
            {
                if (vetor[j]!=ref1[j]) //Se haver algo diferente...
                {
                    codigo_valido=0; // Falha na autenticação
                    while (!(UCA1IFG&UCTXIFG)); // delay
                    UCA1TXBUF = vetor[j]; // Imprime a mensagem errada.
                    j=10; // break
                }
            }
            if (codigo_valido==1) // Se validar parte inicial...
            {
                for(k=13; k<16; k++) // Analisando a parte final...
                {
                    if (vetor[k]!=ref2[k-13]) // Se haver algo diferente...
                    {
                        codigo_valido=0; // Falha na autenticação
                        while (!(UCA1IFG&UCTXIFG)); // delay
                        UCA1TXBUF = vetor[k]; // Imprime a mensagem errada.
                        k=20; // break
                    }
                }
            }
        }
    }
}

```

```

if (codigo_valido==1) // Se Validar parte inicial e final...
{for (l=7; l<13; l++) // Analisando ao meio do vetor.
{
    while (!(UCA1IFG & UCTXIFG)); // Delay
    UCA1TXBUF = vetor[l]; // Envia a Mensagem ao Labview
    aguardando_resposta = 1; // Está aguardando resposta do Local
}

    while(UCA1STAT & UCBSY);
}
flag=0; // Reiniciando as variáveis
codigo_valido=1; // Reiniciando as variáveis
__delay_cycles(2000000); // Delay de 2 segundos
}
else
{

if(UCA1IFG & UCRXIFG)
    local = UCA1RXBUF; //Recebendo o local de separação e atribuindo a 'Local'.

    switch (local)
    {
        case '1': // Se for local 1...
            __delay_cycles(100000); //delay de 1 seg
            TA0CCR1 = servo_lut[15]; // Atuação do servo 2, para 15º
            __delay_cycles(7000000); // +- 7 seg
            TA0CCR1 = servo_lut[90]; // reinicia hastes para posição inicial
            local = 's'; // Atribuindo 's' para a variável 'Local'
            break;
        case '2': // Se for local 2...
            __delay_cycles(100000); // delay de 1 seg
            TA1CCR1 = servo_lut[170]; // Atuação do servo 1, para 170º
            __delay_cycles(7000000); // +- 7 seg
            TA1CCR1 = servo_lut[90]; // reinicia hastes para posição inicial
            local = 's'; // Atribuindo 's' para a variável 'Local'
            break;
        default: // Caso de inexistência de tag.
            contador++;
            TA1CCR1 = servo_lut[90]; // Atuação do servo 1, para 90º
            TA0CCR1 = servo_lut[90]; // Atuação do servo 2, para 90º
            break;
    }

    if((local == 's') || (contador >= 4000)) {
        UCA1IFG = ~UCRXIFG; // limpar bit de novo dado recebido.
        aguardando_resposta=0; // Reinicia a flag para entrar no loop novamente
        contador = 0; // Reinicia a variável 'contador'
        local = 0; // Reinicia a variável 'local'
    }
    __delay_cycles(1000); //delay 1ms
}
}
}
//Interrupção
#if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
#pragma vector=USCI_A0_VECTOR
__interrupt void USCI_A0_ISR(void)
#elif defined(__GNUC__)
void __attribute__ ((interrupt(USCI_A0_VECTOR))) USCI_A0_ISR (void)

```

```
#else
#error Compiler not supported!
#endif
{
  switch(__even_in_range(UCA0IV,4))
  {
    case 0:break; // Vector 0 - no interrupt
    case 2: // Vector 2 - RXIFG
      while (!(UCA1IFG & UCTXIFG)); // USCI_A1 TX buffer ready? - Delay

      vetor[i] = UCA0RXBUF; // Enviando a mensagem ao vetor criado, UART A0
      i++; // Incremento do contador

      if(i==16) // Quando chegar ao final da leitura ...
      {
        i=0; // Reinicia o contador
        flag=1; // Há Informação nova
      }
      break;
    case 4:break; // Vector 4 - TXIFG
    default: break;
  }
}
```