

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL

ALEXANDRE JOSÉ GONÇALVES
JAIR RODRIGUES DA SILVA
JOÃO CARLOS BATISTA

**SISTEMA DIDÁTICO DE AUTOMAÇÃO BASEADO EM
COMPUTADOR PARA SELEÇÃO DE ESFERAS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2015

ALEXANDRE JOSÉ GONÇALVES
JAIR RODRIGUES DA SILVA
JOÃO CARLOS BATISTA

**SISTEMA DIDÁTICO DE AUTOMAÇÃO BASEADO EM
COMPUTADOR PARA SELEÇÃO DE ESFERAS**

Trabalho de Conclusão de Curso de graduação, apresentado a disciplina de Trabalhos de Diplomação do Curso Superior de Tecnologia em Automação Industrial do Departamento Acadêmico de Eletrotécnica – DAELT – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Marcelo Rodrigues

CURITIBA

2015

ALEXANDRE JOSÉ GONÇALVES

JAIR RODRIGUES DA SILVA

JOÃO CARLOS BATISTA

SISTEMA DIDÁTICO DE AUTOMAÇÃO BASEADO EM COMPUTADOR PARA SELEÇÃO DE ESFERAS

Este Trabalho de Diplomação foi julgado e aprovado como requisito parcial para a obtenção do Título de **Tecnólogo em Automação Industrial**, do **Curso Superior de Tecnologia em Automação Industrial**, da **Universidade Tecnológica Federal do Paraná**.

Curitiba, 20 de maio de 2015

Prof. José da Silva Maia, M.Eng.
Coordenador de Curso
Departamento Acadêmico de Eletrotécnica

Prof. Rafael Fontes Souto, Dr.
Responsável pelo Trabalho de Diplomação da Tecnologia
Departamento Acadêmico de Eletrotécnica

BANCA EXAMINADORA

Profa. Lilian Moreira Garcia, Dra..
Universidade Tecnológica Federal do Paraná

Prof. Marcelo Rodrigues, Dr
Universidade Tecnológica Federal do Paraná
Orientador

Prof. José da Silva Maia, M.Eng
Universidade Tecnológica Federal do Paraná

Profa. Luciane Brandalise, Dra.
Universidade Tecnológica Federal do Paraná

A Folha de Aprovação assinada encontra-se na Coordenação do Curso

RESUMO

GONÇALVES, Alexandre José; SILVA, Jair Rodrigues da; BATISTA, João Carlos. Sistema Didático Baseado em Computador para Seleção de Esferas, 2015 65f. Trabalho de Conclusão de Curso - Universidade Tecnológica Federal do Paraná. Curitiba.

O trabalho apresenta um sistema de automação de um módulo didático, composto de sensores indutivos, atuadores e *software* supervisor para leitura e seleção de esferas metálicas com duas dimensões diferentes numa linha de produção. A partir de faixas de tolerância pré-estabelecidas, a estocagem dessa seleção será feita por servomecanismos, imediatamente após essa leitura, distribuindo-as em diferentes recipientes conforme código programado no Arduino. O sistema de supervisão *SCADA (Supervisory Control and Data Acquisition)* foi concebido sob a perspectiva da Tecnologia de Automação Baseada em Computador. Os dados de produção serão exibidos em tempo real, na unidade local e remotamente por dispositivos móveis *PDA (Personal Digital Assistants)*.

Palavras-chave: Módulo Didático. Sensor Indutivo. Atuador. Computador. *SCADA*. Arduino.

ABSTRACT

GONÇALVES, Alexandre José; SILVA, Jair Rodrigues da; BATISTA, João Carlos. Didactic System Based Computer to Spheres Selection, 2015 65f. Trabalho de Conclusão de Curso - Universidade Tecnológica Federal do Paraná. Curitiba.

The work presents an automation system of a didactic module, consisting of inductive sensors, actuators and supervisory software for reading and selection of metallic spheres with two different dimensions in a production line. From pre-set tolerance ranges, the storage of this selection will be made by servomechanisms immediately after this reading, distributing them in different containers according to code programmed in Arduino. The SCADA supervisory system (Supervisory Control and Data Acquisition) was designed from the perspective of Automation Technology Based Computer. The production data is displayed in real time on the local drive and remotely by mobile PDA (Personal Digital Assistants).

Keywords: Didactic module. Inductive sensors. Actuator. Computer. SCADA. Arduino

LISTA DE FIGURAS

Figura 1 - Sistema de Automação baseado em computador.	10
Figura 2 - Conector DB9.	18
Figura 3 - Motor de passo.	20
Figura 4 - Sensor Indutivo.	21
Figura 5 - Nomes/funções dos pinos do PIC 16F877A	24
Figura 6 - Fonte AT adaptada	24
Figura 7 - Descrição do <i>Kit Personalpic</i>	25
Figura 8 - Interface eletrônica.....	27
Figura 9 - Interface Arduino.....	28
Figura 10 - Módulo de Relé.....	28
Figura 11 - Seleccionador de Esferas.....	34
Figura 12 - Tela do supervisorio.....	35
Figura 13 - Tela do Elipse.....	37
Figura 14 - Acionamento do sensor 2 pela esfera menor.....	37
Figura 15 - Acionamento dos sensores 1 e 2 pela esfera maior.	38
Figura 16 - Tags de comunicação.....	39
Figura 17 - Tela do IOKit.	40
Figura 18 - Tela do Proteus.....	40
Figura 19 - Tela ampliada do LCD do Proteus.....	41
Figura 20 - Tela do programa RComSerial.....	42
Figura 21 - Passo 1: Regulagem da Canaleta.....	44
Figura 22 - Passo 2: Regulagem dos Sensores.....	45
Figura 23 - Passo 3: Conexões de Alimentação.....	45
Figura 24 - Passo 4: Conexão Computador/Arduino.....	46
Figura 25 - Passo 5: Inicializar o Programa.....	46
Figura 26 - Passo 6: Tela IHM.....	47
Figura 27 - Passo 7: Alimentação das Esferas.....	47

LISTA DE QUADROS

QUADRO 1 - Módulos disponíveis no Elipse E3.....	16
QUADRO 2 - Descrição dos pinos do Conector DB9.....	18
QUADRO 3 - Características do PIC 16F877A.....	23
QUADRO 4 - Descrição do <i>Kit Personalpic</i>	26
QUADRO 5 - Atividade desempenhada pelo Botão.....	36
QUADRO 6 - Atividades indicadas no display.....	36
QUADRO 7 - Descrição das Tags.....	38

LISTA DE ABREVIATURAS E SIGLAS

ASCII	<i>American Standard Code for Information Interchange</i>
AT	<i>Advanced Technology</i>
CI	Circuito Integrado
SS	<i>Slave Selec</i>
CPU	Unidade central de processamento (do original <i>Central Processing Unit</i>)
EIA	<i>Eletronic Industries Association</i>
IHM	Interface Humano-Máquina (do original <i>Human Machine Interface</i>)
IP	<i>Internet Protocol</i>
LCD	Tela de Cristal Líquido (do original <i>Liquid Crystal Display</i>)
LED	Diodo emissor de Luz (do original <i>Light Emitting Diode</i>)
NA	Normalmente Aberto
NF	Normalmente Fechado
NUPET	Núcleo de Pesquisas Tecnológicas
PDA	Assistente Pessoal Digital (do original <i>Personal Digital Assistants</i>)
PIC	Controlador de Interface Programável (do original <i>Programmable Interface Controller</i>)
RS	<i>Recommended Standard</i>
RTU	<i>Remote Terminal Unit</i>
SCADA	Sistema de Supervisão e Aquisição de Dados (do original <i>Supervisory Control and Data Acquisition</i>)
SPI	<i>Serial Peripheral Interface</i>
TCP	<i>Transmission Control Protocol</i>
USB	<i>Universal Serial Bus</i>
UTFPR	Universidade Tecnológica Federal do Paraná
WI-FI	<i>Wireless Fidelity</i>

SUMÁRIO

1.	INTRODUÇÃO	9
1.1	TEMA	9
1.2	DELIMITAÇÃO DO TEMA	11
1.3	PROPOSTA	12
1.4	OBJETIVOS	12
	1.4.1 Objetivo Geral	12
	1.4.2 Objetivos Específicos	12
1.5	JUSTIFICATIVA	13
1.6	ESTRUTURA DO TRABALHO	13
2.	REFERENCIAL TEÓRICO	15
2.1	AUTOMAÇÃO BASEADA EM COMPUTADOR	15
2.2	SOFTWARE SUPERVISÓRIO	16
2.3	INTERFACE SERIAL RS232 (EIA232)	17
2.4	CONECTOR DB9	17
2.5	DRIVER DE COMUNICAÇÃO ASCII	19
2.6	ATUADORES	20
2.7	SENSORES	21
2.8	SOFTWARE PROTEUS	21
2.9	MIKRO C PRO	22
2.10	MICROCONTROLADOR PIC 16F877A	22
2.11	FONTE DE ALIMENTAÇÃO	24
2.12	KIT PERSONAL PIC	25
2.13	INTERFACE ELETRÔNICA OPTOACOPLADA	26

2.14	DRIVER DE COMUNICAÇÃO MODBUS.....	27
2.15	ARDUINO.....	27
2.16	MÓDULO DE RELÉ.....	28
3.	PROCEDIMENTOS METODOLÓGICOS.....	29
3.1	FLUXOGRAMA.....	30
3.2	DETALHAMENTO DAS ETAPAS DO FLUXOGRAMA.....	31
4.	DESENVOLVIMENTO.....	34
4.1	COMUNICAÇÃO DO SUPERVISÓRIO COM O CI.....	38
4.2	TESTES REALIZADOS.....	41
4.3	ANÁLISE DE RESULTADOS.....	43
4.4	ANÁLISE DIDÁTICA DO PROJETO.....	44
4.5	DIFICULDADES ENCONTRADAS / SUPERADAS.....	48
5.	CONSIDERAÇÕES FINAIS.....	49
5.1	PROPOSTA PARA TRABALHOS FUTUROS.....	50
	REFERÊNCIAS.....	51
	APÊNDICE.....	53
	APÊNDICE A.....	54
	APÊNDICE B.....	55
	ANEXOS.....	60
	ANEXO A.....	61
	ANEXO B.....	62

1. INTRODUÇÃO

Com necessidades cada vez maiores de consumo de produtos manufaturados demandados pela sociedade contemporânea, a automação industrial ganhou importância relevante dentre as inúmeras ferramentas que foram desenvolvidas para que as indústrias pudessem se adequar ao aumento da concorrência. Atuando a partir do nível de exigência do mercado com qualidade e métodos de barateamento do custo final dos produtos, auxiliando nas tarefas pesadas e repetitivas, diminuindo o esforço físico do homem, proporcionando uma redução de desperdícios, o aumento da qualidade e volume de produção.

Com a construção de máquinas cada vez mais avançadas, os computadores foram introduzidos no meio industrial com a ideia de melhoria dos processos fabris, visando uma integração entre o chão de fábrica e o ambiente corporativo, atuando em funções como executar sequenciamento, temporização, contagem, energização/desenergização e manipulação de dados, coletando informações do sistema produtivo e repassando de forma rápida e confiável para que o sistema organizacional possa tomar decisões de melhorias de processos.

A utilização de sistemas digitais de supervisão e controle, proporciona recursos para a manipulação de grandes quantidades de informações e de manobra remota dos equipamentos em tempo real, auxiliando na viabilidade da gestão mais eficaz.

1.1 TEMA

O desenvolvimento de um protótipo que representa em tamanho reduzido, uma situação real de fábrica e que possa ser levado à sala de aula e demonstrado aos alunos para uma explicação didática inicial do tema Automação Industrial.

Servindo como material de apoio prático e de fácil compreensão, apresentando os elementos de um sistema de automação baseado em computador. Incentivando o desenvolvimento das habilidades de investigação dos alunos da instituição, através de experiências voltadas à aprendizagem.

A sala de aula, onde se desenvolve a maior parte das atividades docentes e discentes, constitui um espaço de vivências pedagógicas. É nesse espaço que se concretiza a intervenção do professor - expressão de sua concepção de educação, de ensino e de aprendizagem. Essa intervenção, construída na interação com os alunos, é uma atividade intencionalmente organizada pelo professor para facilitar a aprendizagem do aluno. (ANDRÉ, OLIVEIRA, 1997, pg 121).

A Figura 1 apresenta os principais elementos de um Sistema de Automação Baseado em Computador.

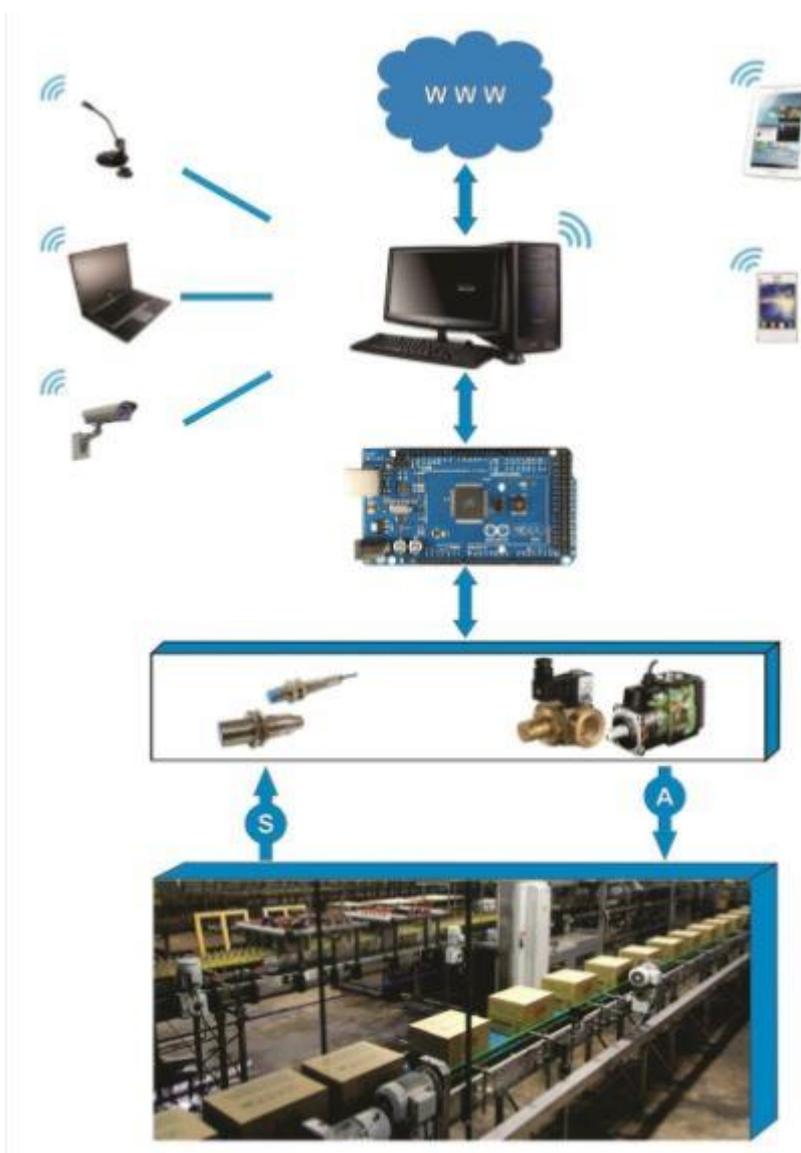


FIGURA 1 - SISTEMA DE AUTOMAÇÃO BASEADO EM COMPUTADOR.
Fonte: Autoria própria, 2014.

Com a utilização do protótipo pode-se observar a ação integrada de Sensores e Atuadores sobre um Processo Produtivo, conectados em uma Interface Arduino. Por meio de uma Porta de Comunicação de Escrita e Leitura de um Computador com essa Interface, a automatização é feita com segurança, confiabilidade e interoperabilidade nas diferentes etapas produtivas.

A operação do Sistema em modo local, em rede cabeada ou sem fio, através de dispositivos móveis *PDA*, amplia cada vez mais a adoção dessa Tecnologia, proporcionando melhores resultados operacionais, financeiros e mercadológicos.

O processamento de Imagens, o comando de voz e o uso de dispositivos móveis permite integrar ainda mais Gestão e Produção, sempre buscando a melhor relação custo-benefício.

Toda essa teoria transformada em material didático que é um termo genérico usado para descrever as fontes que os professores têm para ensinar. Os materiais didáticos podem auxiliar o aprendizado e facilitar a transmissão das informações para os alunos. Esses materiais serão adaptados ao conteúdo que estiver sendo ensinado, aos alunos da turma e ao professor.

1.2 DELIMITAÇÃO DO TEMA

Trabalhar alguns elementos de automação em um módulo didático de simples apresentação, contendo sensores, atuadores, programação em linguagem C e a comunicação envolvendo as informações gravadas no microcontrolador.

Trabalhando integradas no *Software Eclipse*, que mostrará na tela do computador uma demonstração de supervisão em forma de Interface Humano-Máquina (IHM), visualizando o que acontece na linha de produção representada pelo protótipo chamado de Seleccionador de Esferas. Realizar a integração com um sistema supervisorio e tornando-o um conjunto didático para apresentação aos alunos que se interessem pela área tecnológica ajudando na construção do conhecimento passado pela teoria aplicada pelo docente, ilustrando de forma palpável para o discente.

O sistema permitirá o acompanhamento dessa produção em modo local, em rede *TCP-IP* e através de um dispositivo *PDA* em conexão *WI-FI (Wireless Fidelity)*, utilizando a criptografia da plataforma *ELIPSE*.

1.3 PROPOSTA

Parte-se do princípio de utilização do protótipo como ferramenta didática, oferecendo mais uma opção de forma de ensino dentro dos laboratórios da instituição UTFPR (Universidade Tecnológica Federal do Paraná), com a utilização de módulos simples que integrem as tecnologias ministradas nos cursos da área de Automação.

O protótipo também foi desenvolvido com a finalidade de instigar o interesse pelos cursos de Tecnologia e Engenharia, principalmente aos alunos dos primeiros períodos que ainda não possuem uma noção das ferramentas empregadas nas indústrias, mostrando programas de simples compreensão. Problema-Conflito de comunicação entre interface e supervisor.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Construir um protótipo para selecionar esferas metálicas de dois diâmetros diferentes, simulando uma linha de produção, armazenando-as distintamente e sendo integrado por um sistema supervisor para ser apresentado de forma didática a alunos que possuam interesse na área de automação.

1.4.2 Objetivos Específicos

Para a execução de forma eficaz desta proposta, foram desenvolvidas várias etapas, relacionadas:

- ✓ Projetar um equipamento didático para área de automação.
- ✓ Montar o protótipo com sua estrutura mecânica.
- ✓ Programar o Microcontrolador ATMega2560.
- ✓ Integrar sensores e atuadores.
- ✓ Desenvolver uma tela de supervisão na plataforma de um supervisor.
- ✓ Programar um dispositivo *PDA* na plataforma *ANDROID*.
- ✓ Realizar ensaios de funcionamento.

1.5 JUSTIFICATIVA

O protótipo pode ser utilizado dentro da instituição UTFPR como uma ferramenta didática que vai facilitar a compreensão do universo da Automação industrial, poderá ser empregado primeiramente na disciplina “Introdução À Tecnologia de Automação”, ajudando a despertar o interesse do calouro no curso por ele escolhido.

As disciplinas de Projeto Integrador 1 e 2 podem apresentá-lo como um produto final desenvolvido pelos acadêmicos destas disciplinas.

O protótipo e seu desenvolvimento também pode ser exposto em feiras técnicas, para divulgação da área de automação, demonstrando uma aplicação prática de uma parcela da automação, contribuindo para despertar o interesse na área.

1.6 ESTRUTURA DO TRABALHO

No capítulo 1 é dada a introdução do trabalho, explicando sobre o tema abordado e sua relevância de aplicação no auxílio didático de ferramentas técnicas que simulem o ambiente industrial.

No capítulo 2 é desenvolvido o referencial teórico descrevendo os itens utilizados para construção do protótipo bem como os *softwares* necessários para integração dos ambientes que trabalharam em conjunto, com um breve detalhamento da funcionalidade e o modelo de cada item utilizado no trabalho. Também inclui a descrição dos *softwares* aplicados como suporte de trabalho.

O procedimento metodológico é trabalhado no capítulo 3, onde também é construído o fluxograma do projeto indicando os caminhos seguidos com uma sequência lógica de trabalho.

É abrangido no capítulo 4 o desenvolvimento com as explicações de como foi realizado o trabalho, detalhando como as etapas foram sendo construídas. Nele são colocadas figuras que indicam as telas dos programas utilizados, com as suas configurações, programações e telas de trabalho.

Os testes realizados em cada etapa bem como as análises dos resultados obtidos, as dificuldades encontradas e como foram superadas são listadas.

Sendo abrangidas no capítulo 5 as considerações finais e propostas para trabalhos futuros que possam ser agregados no enriquecimento do tema descrito.

2. REFERENCIAL TEÓRICO

Os parágrafos seguintes deste capítulo denotam as estruturas que compõem o Seleccionador de Esferas e os Sistemas de Automação para supervisão de processos industriais, com destaque para os sensores, atuadores, o *Hardware* de interface e o *Software* Elipse.

Um Sistema de Automação Baseado em Computador é aquele que integra as informações para automatizar e apoiar a realização de atividades humanas através do processamento de informações.

. A troca de informações entre o usuário e o sistema produtivo pode ser feita através de interfaces digitais em modo local, em rede ou por dispositivos móveis PDA.

A informação é um componente fundamental nos sistemas de Automação Baseados em Computador. Este último processa e armazena os dados de produção associados a informações de outros sistemas, integrando resultados estratégicos e disponibilizando-os aos usuários previamente cadastrados no domínio da aplicação.

2.1 AUTOMAÇÃO BASEADA EM COMPUTADOR

Na área industrial existe um grande crescimento no uso do computador para medição e controle nas rotinas de trabalho. O emprego de lógicas para manipularem os atuadores, baseando-se em medições e recebimentos de dados.

A utilização dos computadores dentro do chão de fábrica para controle de sistemas dos mais simples até os com níveis maiores de sofisticação vem aumentando muito, devido a fatores como os processadores com melhoria de velocidade, espaço maior para armazenamento de dados e robustez ampliada de seus componentes, diminuindo os custos dos equipamentos (LIPSON; ZALM. 2011).

2.2 SOFTWARE SUPERVISÓRIO

Sistemas de Supervisão e Aquisição de Dados, ou abreviadamente SCADA (*Supervisory Control and Data Acquisition*) também chamado de *software* supervisório, permitem que sejam monitoradas e rastreadas informações de um processo produtivo ou instalação física. Tais informações são coletadas através de equipamentos de aquisição de dados e, em seguida, manipulados, analisados, armazenados e posteriormente apresentados ao usuário (COELHO, 2009).

Segundo Ogata (1993), o *software* supervisório é visto como o conjunto de programas gerados e configurado no *software* básico de supervisão, implementando as estratégias de controle e supervisão com telas gráficas de IHM que facilitam a visualização do contexto atual, a aquisição e tratamento de dados do processo e a gerência de relatório e alarmes.

A partir daí a implementação com as Telas de Operação, Alarme e Histórico do Sistema SCADA, conforme as informações recebidas dos sensores, com o apoio da Empresa *ELIPSE SOFTWARE*.

Existe dentro do *software* do *Elipse E3* uma subdivisão de módulos de trabalho que são exibidos no quadro 1.

MÓDULOS	DESCRIÇÃO DE UTILIZAÇÃO
<i>E3 Server</i>	Servidor de aplicação, executando os processos principais destacando a comunicação em tempo real.
<i>E3 Studio</i>	Ferramenta de configuração das telas de trabalho.
<i>E3 Viewer</i>	Visualizador das aplicações executáveis, assumindo a forma de IHM.
<i>E3 Admin</i>	Responsável pela interface do <i>E3 Server</i> com os outros módulos do <i>E3</i> com o usuário.

QUADRO 1 - MÓDULOS DISPONÍVEIS NO ELIPSE E3.

Fonte: Adaptado do manual do Elipse.

2.3 INTERFACE SERIAL RS232 (EIA232).

O RS (*Recommended Standard*) relata uma padronização de uma interface comum para comunicação de dados entre equipamentos, criada por um comitê conhecido atualmente como EIA (*Electronic Industries Association*).

A comunicação de dados compreendia a troca de dados digitais entre um computador central e terminais de computador remotos, ou entre dois terminais sem o envolvimento do computador. Estes dispositivos poderiam ser conectados através de linha telefônica, e conseqüentemente necessitavam um modem em cada lado para fazer a decodificação dos sinais.

É criado então o padrão RS232 que especifica as tensões, temporizações e funções dos sinais, um protocolo para troca de informações, e as conexões mecânicas.

Das três modificações publicadas pela EIA a mais recente, EIA232E, foi introduzida em 1991. Ao lado da mudança de nome de RS232 para EIA232, algumas linhas de sinais foram renomeadas e várias linhas novas foram definidas.

2.4 CONECTOR DB9

O conector DB9 (inicialmente chamado DE -9) é uma ficha analógica, comportando nove pinos, da família dos conectores D-Subminiaturas (D-Sub ou SubD).

O conector DB9 (figura 2) serve essencialmente para as ligações séries, permitindo a transmissão de dados assíncrona de acordo com a norma RS-232 (RS-232C). Contém duas fileiras paralelas de pinos ou soquetes, usualmente circundados por um envoltório metálico em formato de **D** que provê um anteparo contra interferência eletromagnética e garante um encaixe correto. Os conectores DB9 foram inventados pela empresa Cannon (Robotizando, 2015).

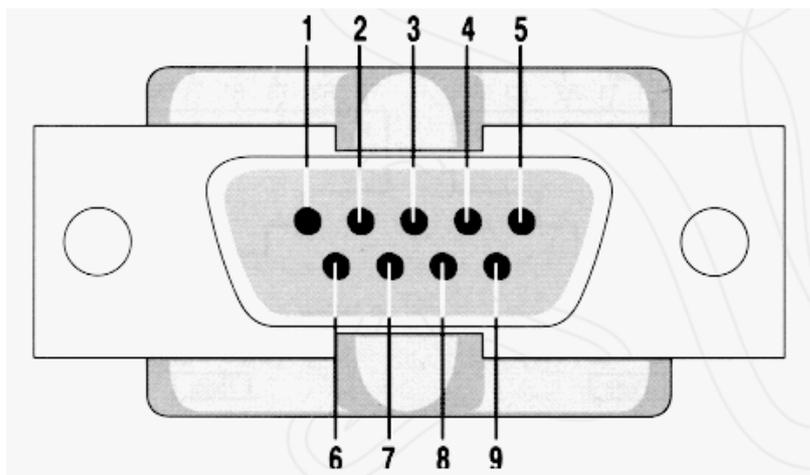


FIGURA 2 - CONECTOR DB9.
Fonte: Robotizando

A transmissão e recepção dos dados por esta porta são feitos bit a bit, que caminham um após o outro por apenas uma via na linha de transmissão formando uma série de pulsos. Tradicionalmente as placas-mãe possuem duas portas seriais RS-232. O padrão RS-232 é descrito como uma interface assíncrona, que transmite os dados somente quando a unidade de recepção está pronta para recebê-los.

O quadro 2 mostra o número dos pinos e função de cada um.

Pin	Nome	Descrição	Sinal
1	CD	Portadora detectada	In
2	RXD	Recepção de dados	In
3	TXD	Transmissão de dados	Out
4	DTR	Terminal de dados pronto	Out
5	GND	Terra	-
6	DSR	Conjunto de dados pronto	In
7	RTS	Pronto para enviar (computador)	Out
8	CTS	Envie os dados (modem)	In
9	RI	Indicador de telefone tocando	In

QUADRO 2 - DESCRIÇÃO DOS PINOS DO CONECTOR DB9.

Fonte: Robotizando,

2.5 DRIVER DE COMUNICAÇÃO ASCII

Para que dois dispositivos possam interagir é necessário um pacote de informações que façam eles se comunicarem numa linguagem comum e esse pacote de informações é chamado de *Driver* de comunicação no formato.dll, que neste projeto foi usado o ASCII.dll (*American Standard Code for Information Interchange*), disponível para download no site da Elipse em: http://www.elipse.com.br/port/download_drivers.aspx. Este *driver* é utilizado para estabelecer a troca de dados simplificada e confiável entre o programa supervisor Elipse (E3 e SCADA) com o Microcontrolador PIC (*Programmable Interface Controller*).

A Tabela ASCII é usada pela maior parte da indústria de computadores para a troca de informações. Cada caracter é representado por um código de 8 bits (um *byte*). Conforme mostrado na tabela 1.

Tabela ASCII (American Standard Code for Information Interchange)

Dec	Hx	Oct	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040	␣	64	40	100	␣	96	60	140	␣	96;`	
1	1	001	SOH	(start of heading)	33	21	041	!	65	41	101	!	97	61	141	!	97;a	
2	2	002	STX	(start of text)	34	22	042	"	66	42	102	"	98	62	142	"	98;b	
3	3	003	ETX	(end of text)	35	23	043	#	67	43	103	#	99	63	143	#	99;c	
4	4	004	EOT	(end of transmission)	36	24	044	\$	68	44	104	\$	100	64	144	\$	100;d	
5	5	005	ENQ	(enquiry)	37	25	045	%	69	45	105	%	101	65	145	%	101;e	
6	6	006	ACK	(acknowledge)	38	26	046	&	70	46	106	&	102	66	146	&	102;f	
7	7	007	BEL	(bell)	39	27	047	'	71	47	107	'	103	67	147	'	103;g	
8	8	010	BS	(backspace)	40	28	050	(72	48	110	(104	68	150	(104;h	
9	9	011	TAB	(horizontal tab)	41	29	051)	73	49	111)	105	69	151)	105;i	
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	74	4A	112	*	106	6A	152	*	106;j	
11	B	013	VT	(vertical tab)	43	2B	053	+	75	4B	113	+	107	6B	153	+	107;k	
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	76	4C	114	,	108	6C	154	,	108;l	
13	D	015	CR	(carriage return)	45	2D	055	-	77	4D	115	-	109	6D	155	-	109;m	
14	E	016	SO	(shift out)	46	2E	056	.	78	4E	116	.	110	6E	156	.	110;n	
15	F	017	SI	(shift in)	47	2F	057	/	79	4F	117	/	111	6F	157	/	111;o	
16	10	020	DLE	(data link escape)	48	30	060	0	80	50	120	0	112	70	160	0	112;p	
17	11	021	DC1	(device control 1)	49	31	061	1	81	51	121	1	113	71	161	1	113;q	
18	12	022	DC2	(device control 2)	50	32	062	2	82	52	122	2	114	72	162	2	114;r	
19	13	023	DC3	(device control 3)	51	33	063	3	83	53	123	3	115	73	163	3	115;s	
20	14	024	DC4	(device control 4)	52	34	064	4	84	54	124	4	116	74	164	4	116;t	
21	15	025	NAK	(negative acknowledge)	53	35	065	5	85	55	125	5	117	75	165	5	117;u	
22	16	026	SYN	(synchronous idle)	54	36	066	6	86	56	126	6	118	76	166	6	118;v	
23	17	027	ETB	(end of trans. block)	55	37	067	7	87	57	127	7	119	77	167	7	119;w	
24	18	030	CAN	(cancel)	56	38	070	8	88	58	130	8	120	78	170	8	120;x	
25	19	031	EM	(end of medium)	57	39	071	9	89	59	131	9	121	79	171	9	121;y	
26	1A	032	SUB	(substitute)	58	3A	072	:	90	5A	132	:	122	7A	172	:	122;z	
27	1B	033	ESC	(escape)	59	3B	073	;	91	5B	133	;	123	7B	173	;	123;{	
28	1C	034	FS	(file separator)	60	3C	074	<	92	5C	134	<	124	7C	174	<	124;	
29	1D	035	GS	(group separator)	61	3D	075	=	93	5D	135	=	125	7D	175	=	125;}	
30	1E	036	RS	(record separator)	62	3E	076	>	94	5E	136	>	126	7E	176	>	126;~	
31	1F	037	US	(unit separator)	63	3F	077	?	95	5F	137	?	127	7F	177	?	127;DEL	

TABELA 1 - TABELA ASCII POR LOOKUP TABLES

Fonte: ASCIItable,

2.6 ATUADORES

Thomazini (2010) afirma que os atuadores são dispositivos que alteram uma variável controlada. Eles atuam no sistema controlado a partir do recebimento de um comando de outro dispositivo. São exemplos de atuadores, motores, cilindros hidráulicos e válvulas. No caso deste projeto utilizamos como atuador o motor de passo.

Os Motores de Passo são dispositivos eletromecânicos que convertem pulsos elétricos em movimentos mecânicos que geram variações angulares discretas. O rotor ou eixo de um motor de passo rotaciona em pequenos incrementos angulares, denominados “passos”, quando pulsos elétricos são aplicados em uma determinada sequência nos terminais deste.

A rotação de tipo de atuador é diretamente relacionada aos impulsos elétricos recebidos, bem como a sequência a qual tais pulsos são aplicados reflete diretamente na direção a qual o motor gira. A velocidade de rotação é dada pela frequência de pulsos recebidos e o número de pulsos aplicados é que determina a amplitude do ângulo rotacionado pelo eixo, (BRITES e SANTOS, 2008).

Neste trabalho o atuador utilizado foi o PM55_048, ver Anexo A (Figura 03).

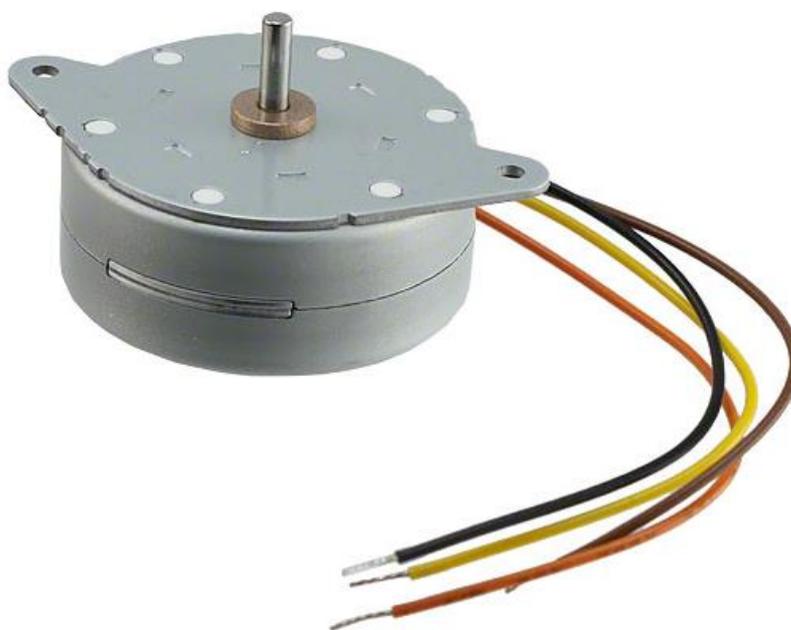


FIGURA 3 - MOTOR DE PASSO.
Fonte: Digikey

2.7 SENSORES

Os sensores são dispositivos sensíveis a alguma forma de energia no ambiente físico. Os sensores digitais podem assumir apenas dois valores no seu sinal de saída ao longo do tempo e que podem ser interpretados como 0 ou 1. (THOMAZINI, 2010).

O Sensor indutivo mostrado na figura 4, ver Anexo B, é um dispositivo eletrônico que devido a proximidade de objetos metálicos responde de maneira específica, esses dispositivos exploram o princípio da impedância de uma bobina de indução. A partir da aproximação de um objeto metálico pelo campo magnético da bobina do sensor indutivo, é habilitada a passagem da corrente elétrica para o circuito . (<https://comatreleco.com.br/sensor-indutivo/>).

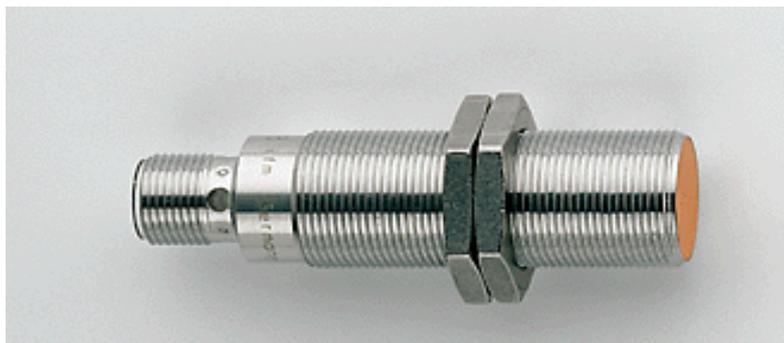


FIGURA 4 - SENSOR INDUTIVO.
Fonte: IFM

2.8 SOFTWARE PROTEUS

O Proteus é uma ferramenta útil para a simulação e a construção de circuitos elétricos. Nele estão disponíveis vários componentes, incluindo microcontroladores, com a possibilidade de observar o funcionamento de forma virtual. Pode ser utilizada tanto para circuitos analógicos quanto para circuitos digitais ou híbridos. No Proteus ainda são disponibilizadas ferramentas de laboratório como osciloscópios, multímetros, geradores de sinais entre outros, permitindo o aprendizado da operação dos principais instrumentos essenciais em uma bancada de eletrônica.

O uso do Proteus foi essencial neste trabalho, para a execução de testes de funcionamento e aprendizado, pois foram usados vários exemplos até a absorção e adaptação com as plataformas utilizadas. Seu uso serviu também para a verificação da confecção do circuito e corrigir erros de programação.

A versão demo está disponível para download no site da Labcenter(<http://www.labcenter.com/index.cfm>).

2.9 MIKRO C PRO

O mikroC PRO é um compilador desenvolvido pela empresa MikroElektronika. Consiste num sistema integrado de desenvolvimento (IDE) para o sistema operacional Windows e suporta toda a linha de microcontroladores PIC (séries PIC12, PIC16, PIC18, PIC24, dsPIC30 e dsPIC33). Ferramentas que auxiliam no desenvolvimento de aplicações, tais como: emuladores de Terminal Serial, GLCD gráficos, displays de 7 segmentos, EEPROM, Terminal USB (*Universal Serial Bus*) e TCP/IP, etc. O MikroC também oferece a opção de encontrar erros de código na própria IDE.

2.10 MICROCONTROLADOR PIC 16F877A

Um Microcontrolador é um circuito integrado que possui diferentes dispositivos e periféricos internos, como um processador, conexões de entrada e saída e memória. Através de sua programação, um Microcontrolador é capaz de controlar os sinais elétricos de suas saídas e ler valores de tensão em seus pinos estabelecidos como entrada. Considerando a variedade de opções de micro controladores existentes no mercado, quando da realização de projetos deve-se elencar o que vai ser preciso ou necessário deste componente, para então escolher um Microcontrolador específico e adequado para a aplicação.

Algumas características e periféricos a serem observados são a quantidade de portas analógicas, quantidade de portas digitais, memória interna de programas e dados, velocidade de processamento, tipo de alimentação e periféricos como comunicação serial, USB, conversores analógico-digitais, entre outros, que podem ou não estar embutidos diretamente no Microcontrolador (MARTINS, 2005).

Foi utilizado nesse trabalho o Microcontrolador fabricado pela Microchip pertencente à família PIC com a nomenclatura 16F877A. Para todos micro controladores define-se uma arquitetura básica, com os recursos mínimos. Quando é adicionado algum recurso extra dentro dessa arquitetura, um novo membro da

família é criado. Em todas essas famílias, há a preocupação de manter-se a compatibilidade com essa arquitetura básica (ZELENOVSKY, MENDONÇA, 2005).

No quadro 3 observa-se as características básicas do PIC 16F877A:

- 33 pinos de I/O;
- 8k de memória de programa FLASH;
- 368 bytes de memória RAM;
- 256 bytes de memória EEPROM;
- Velocidade máxima de trabalho 20Mhz;
- Watchdog timer (WDT);
- 3 Timers;
- 8 canais A/D de 10 bits;
- 1 USART síncrona/assíncrona;
- Porta Paralela escrava (PSP);
- Porta serial síncrona SSP, SPI e I2C;

QUADRO 3 - CARACTERÍSTICAS DO PIC 16F877A

Fonte: Rogercom.com

A figura 5 mostra todos os pinos do PIC 16F877A, com seus respectivos nomes/funções. Os pinos (RA0 a RA5) estão associados à porta 'A', (RE0, RE1 e RE2) à porta "E", (RC0 a RC7) à porta "C", (RD0 a RD7) à porta "D". E por último, os pinos (RB0 a RB7) associados à porta "B".

Cada um destes pinos pode ser usado como entrada ou saída e são definidos na programação. Observe que a maioria dos pinos tem mais de uma função. Como exemplo o pino 10 que tem as funções de entrada/saída digital (RE2) e de selecionar um chip SPI (CS); ou a função de um canal Analógico/Digital(AN7).

1	MCLR/Vpp	PGD/RB7	40
2	RA0/AN0	PGC/RB6	39
3	RA1/AN1	RB5	38
4	RA2/AN2	RB4	37
5	RA3/AN3	PGM/RB3	36
6	RA4/T0CKI	RB2	35
7	RA5/SS/AN4	RB1	34
8	RE0/RD/AN5	INT/RB0	33
9	RE1/WR/AN6	Vdd	32
10	RE2/CS/AN7	Vss	31
11	Vdd	PSP7/RD7	30
12	Vss	PSP6/RD6	29
13	OSC1/CLKIN	PSP5/RD5	28
14	OSC2/CLKOUT	PSP4/RD4	27
15	RC0/T1OSO	RX/RC7	26
16	RC1/T1OSI	TX/RC6	25
17	RC2/CCP1	SD0/RC5	24
18	RC3/SCK	SD1/RC4	23
19	RD0/PSP0	PSP3/RD3	22
20	RD1/PSP1	PSP2/RD2	21

FIGURA 5 - NOMES/FUNÇÕES DOS PINOS DO PIC 16F877A
Fonte: rogercom.com

2.11 FONTE DE ALIMENTAÇÃO

Para o funcionamento do projeto, uma fonte de computador AT (*Advanced Technology*) mostrada na figura 6, foi convertida em fonte de bancada, para se obter as tensões usadas para o funcionamento dos componentes que integram o protótipo, obtendo a tensão de 24 Volts usada para alimentação dos sensores.



FIGURA 6 - FONTE AT ADAPTADA
Fonte: Autoria Própria, 2014

2.12 KIT PERSONAL PIC

O kit *Personalpic* é fabricado pela empresa Microchip, tendo como objetivo principal servir de apoio aos indivíduos que se utilizam de sistemas micro controlados. Este kit é baseado no microcontrolador PIC16F877A que possui 40 pinos, dos quais 33 podem ser configurados como Entradas/Saídas digitais, oito canais de conversão Analógico/Digital, 3 *timers*, entre outros.

A figura 7 mostra o *Kit Personalpic*:

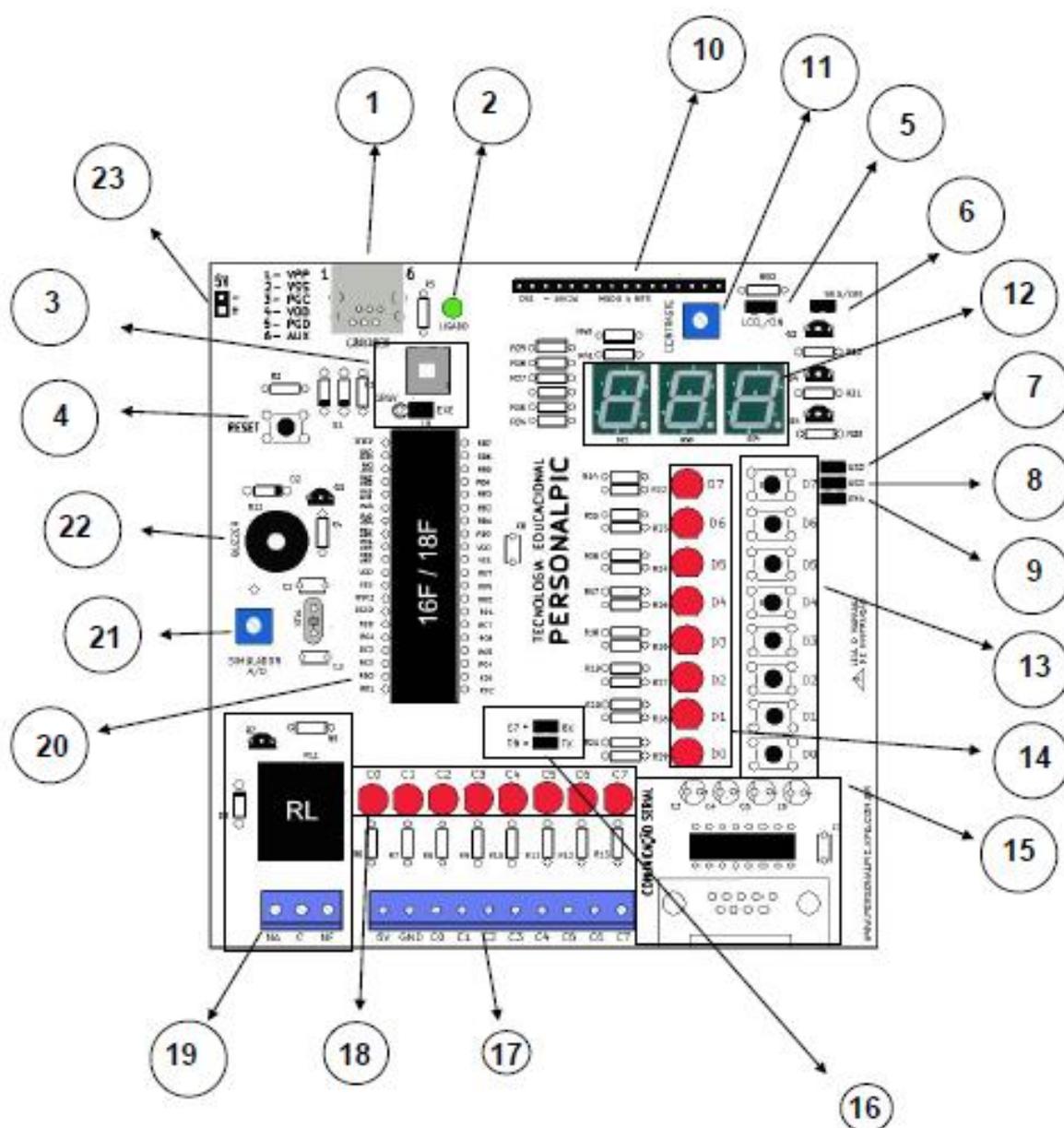


FIGURA 7 - DESCRIÇÃO DO *KIT PERSONALPIC*
 Fonte: Manual do Usuário do *Personalpic*.

O quadro 4 descreve os itens mostrados da figura 7.

ITEM	DESCRIÇÃO	FUNÇÃO
1	Conector Alimentação/Gravador	Conector alimentação (5V) e gravação
2	LED	"LIGADO" indica placa energizada
3	Chave de gravação	Seleciona modo gravação ou modo execução
4	RESET	Reinício do programa simulado
5	Jumper LCD/ON	Ativa Display LCD 16x2
6	Jumper SEG/ON	Ativa Display's de 7 segmentos (DS2,DS3,DS4)
7	Jumper DS2	Ativa Display individualmente de 7 segmentos DS2
8	Jumper DS3	Ativa Display individualmente de 7 segmentos DS3
9	Jumper DS4	Ativa Display individualmente de 7 segmentos DS4
10	Conector Display 16x2	Conexão de Display de Carácter 16x2A
11	Trimpot	Ajuste contraste LCD16x2 – DS1
12	Display's de 7 segmentos	DS2,DS3,DS4
13	Chave de simulação	Simula acionamento (nível lógico 1) PortD-Entrada
14	LED's	Simula acionamento (nível lógico 1) PortD-Saída
15	Conector RS232	Comunicação serial
16	Jumper's	Inverte Rx/C7 , Tx/C6
17	Conector KRE portC IN/OUT	Adaptação de circuitos externos
18	LED's	Simula acionamento (nível lógico 1) PortC-Saída
19	Relê	Acionamento de Potência
20	Soquete para Microcontrolador	PIC (família 16F/18F)
21	Trimpot	Simulação A/D
22	Buzzer	Simulação Sonora (Beeps)
23	Conector de Alimentação	Alimentação Externa 5V

QUADRO 4 - DESCRIÇÃO DO KIT *PERSONALPIC*.
Fonte: Manual do Usuário do *Personalpic*

2.13 INTERFACE ELETRÔNICA OPTOACOPLADA

A Interface Eletrônica Optoacoplada, mostrada na figura 8, foi desenvolvida no NUPET (Núcleo de Pesquisas Tecnológicas) da UTFPR - Câmpus Reitoria para utilização em Sistemas de Automação Baseada em Computador.

Contém cinco entradas digitais e oito saídas NA (normalmente aberta) e quatro NF (normalmente fechada), a comunicação com o computador é feita através da porta paralela.

Possui uma alimentação de 127 Volts, tensão alternada, e na placa já possui uma fonte de 24 Volts, tensão contínua, integrada.

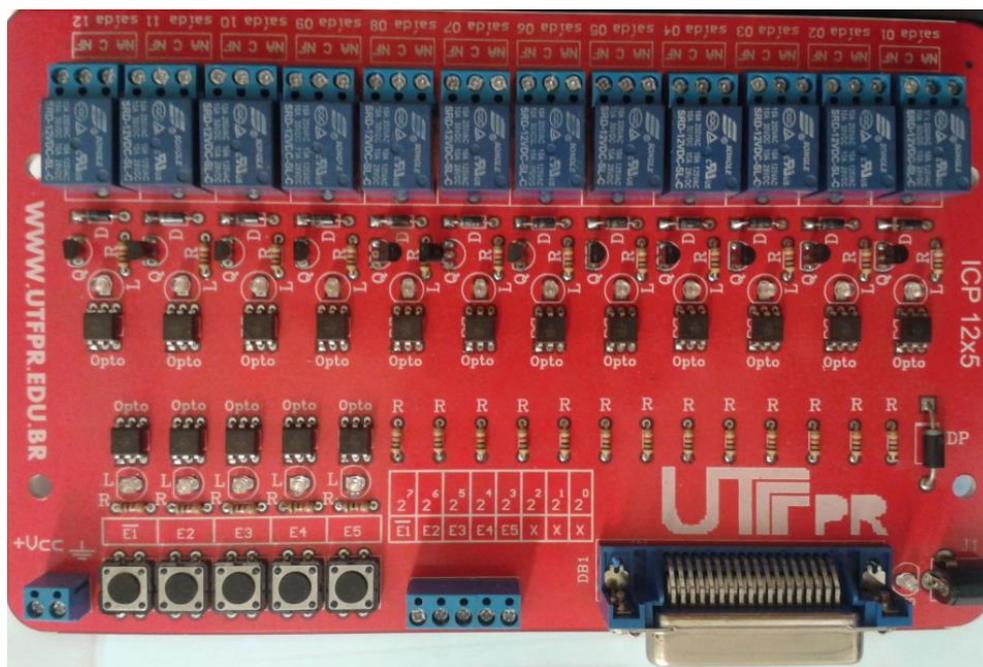


FIGURA 8 - INTERFACE ELETRÔNICA
 FONTE: AUTORIA PRÓPRIA, 2014

2.14 DRIVER DE COMUNICAÇÃO MODBUS

Para estabelecer a comunicação com o módulo Arduino, será utilizado o driver *Modicon Modbus Master (ASC/RTU/TCP)*, disponível para *download* no site da *Elipse Software*.

2.15 ARDUINO

O Arduino (Figura 9) é mais conhecido por seu *hardware*, mas também precisa de um *software* para programação. Tanto o *hardware* e o *software* são chamados de "Arduino." A combinação permite que você crie projetos para controlar o mundo físico. O *software* é livre, de código aberto e multi-plataforma.

As placas são de baixo custo, variando seu preço para compra de R\$ 70,00 até R\$ 500,00, ou você pode construir a sua própria placa, os projetos de *hardware* também são *open source*.

De acordo com Michael Margolis, o *download* do *software* pode ser feito de forma gratuita e os desenhos da placa estão disponíveis sob uma licença aberta, e é livre para adaptá-lo às suas necessidades.

O ambiente Arduino foi projetado para ser de fácil manuseio para iniciantes com ou sem experiência em eletrônica e ainda é usado em muitos programas educacionais em todo o mundo (MARGOLIS, 2011).

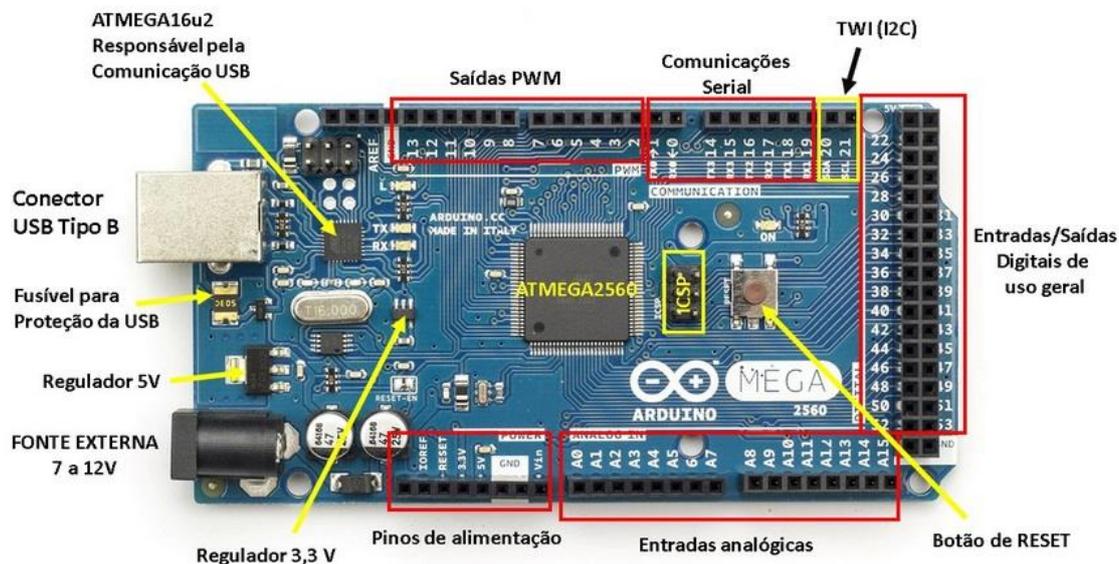


FIGURA 9 - INTERFACE ARDUINO
 FONTE: WWW.ARDUINO.CC

2.16 MÓDULO DE RELÉ

O módulo de relé (Figura 10), composto de borne básico e relé miniatura plugável, foi utilizado para interfaceamento devido os sensores de detecção necessitarem alimentação de 24 Volts e o sinal necessário na entrada do *kit* Arduino ser de 5 Volts.



FIGURA 10 - MÓDULO DE RELÉ
 Fonte: www.phoenixcontact.com

3. PROCEDIMENTOS METODOLÓGICOS

A proposta original deste TCC era utilizar a placa de interface “JI”, que foi desenvolvida nos laboratórios da UTFPR e se comunica com o Computador via porta paralela.

Foi criada a lógica na tela de supervisão do *Eclipse* e instalado a interface de comunicação, porém constatou-se que o tempo de comunicação da porta paralela era muito lento e não atendia as necessidades do projeto que gera um pulso rápido nos sensores.

Em consenso o grupo decidiu mudar o sistema de controle para o PIC que possui comunicação via RS232. O trabalho começou do zero, pois houve a necessidade reescrever toda a lógica no PIC e no *elipse* apenas algumas associações das tags de comunicação.

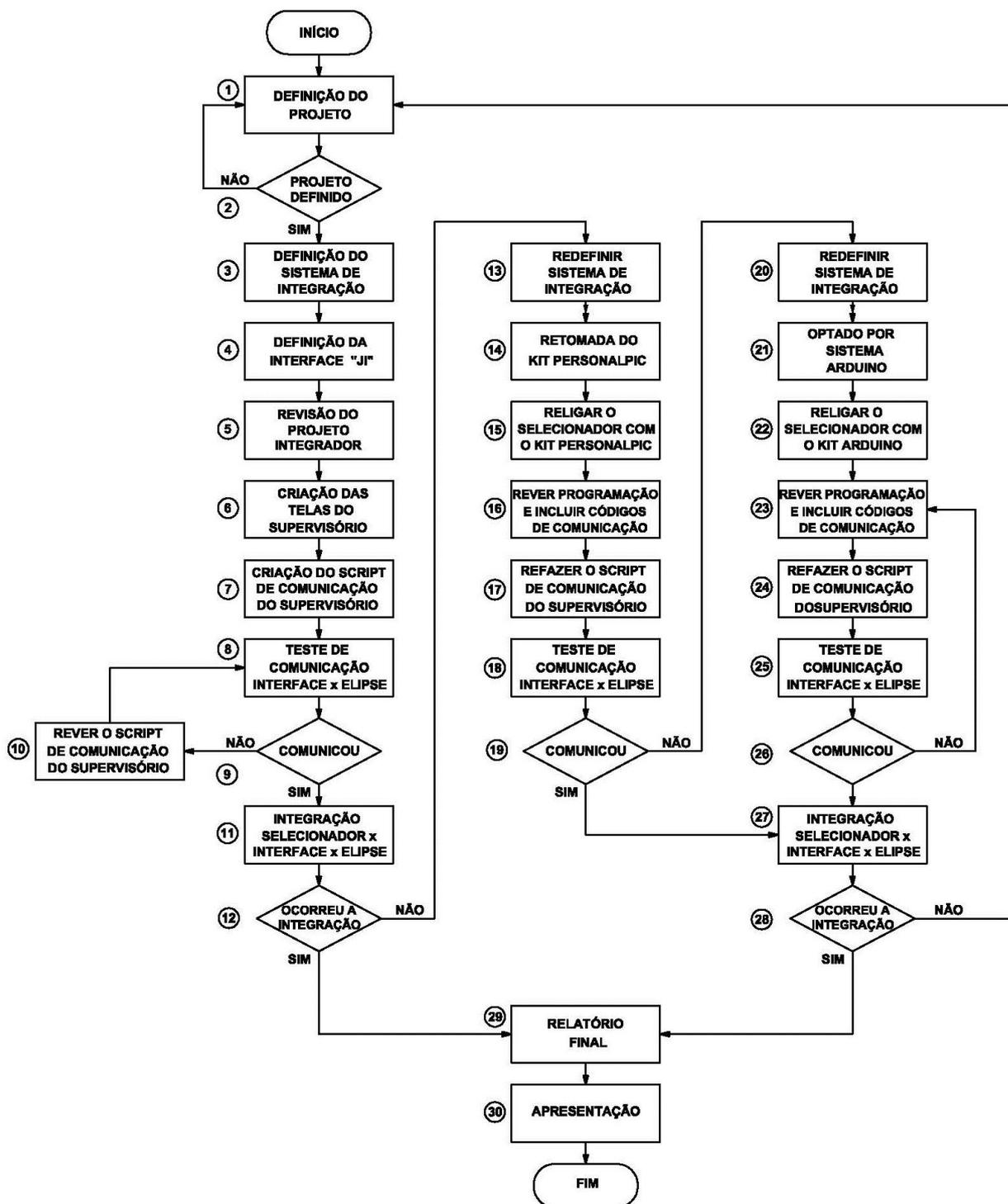
Porém não foi obtido sucesso na comunicação. Solicitado a opinião de um programador com experiência em PIC, o qual afirmou que o código estava correto e que o problema deveria no na placa ou no cabo serial.

Como o grupo não conseguiu identificar onde estava a falha, decidiu-se utilizar o arduino.

Para esta nova configuração, no *Eclipse* foi preciso mudar pouca coisa, mas o código para o arduino foi feito todo novo e a comunicação realizada com o protocolo modbus. Desta vez deu tudo certo, o arduino se comunicou e o projeto foi então finalizado.

O fluxograma a seguir apresenta um resumo das três etapas do projeto e na sequência é descrito com mais detalhes as atividades desenvolvidas em cada uma delas.

3.1 FLUXOGRAMA



3.2 DETALHAMENTO DAS ETAPAS DO FLUXOGRAMA

1. O início do trabalho foi baseado no protótipo apresentado na disciplina Projeto Integrador 2 (ET-54E), do curso de Tecnologia em Automação Industrial, chamado de selecionador de esferas SE800-16/20, com a ideia de criar uma IHM para utilizá-lo como material didático, com uma integração conjunta a um sistema supervisorio para apresentação acadêmica.

2. A ideia foi elaborada e apresentada para o professor escolhido para ser o coordenador, sendo aceita como forma de Trabalho de conclusão de curso.

3. A definição para utilizar como software supervisorio o programa Elipse E3, foi por ser um software disponível no mercado e ter a possibilidade de uso em modo demo, sem a necessidade de aquisição de licença, portanto sem custo financeiro.

4. Para utilizar como interface de integração optou-se pela Interface Eletrônica optoacoplada, desenvolvida por um professor da Instituição UTFPR e denominada de interface "JI" que faz comunicação com o PC via porta paralela.

5. Com as primeiras definições estabelecidas, iniciou-se a revisão e teste de funcionamento adequado do protótipo, para que pudesse ser utilizado novamente, agora para o presente trabalho de conclusão de curso.

6. Seguiu-se então para a criação das telas gráficas de IHM, onde o operador pode visualizar os contadores, a condição de operação do equipamento e atuar no mesmo, através das teclas de emergência e *reset*.

7. Foi preciso realizar a programação via *Script* no sistema supervisorio, configurando o sistema conforme a necessidade exigida para execução do trabalho determinado.

8. Utilizando o *driver IO Genéric*, o teste de comunicação entre a interface optoacoplada e o supervisorio foi realizado forçando manualmente o sinal do sensor.

9. Obtendo um resultado positivo entre o sinal fornecido e a sua recepção nesse teste aplicado.

10. Caso a comunicação não fosse satisfatória, deveria ser revisto o script, mas não houve a necessidade.

11. A integração do protótipo Selecionador de Esferas com a interface "JI" e também com o sistema supervisorio *Elipse* foi feita para a realização de testes práticos de comunicação.

12. Mas constatou-se que o sinal gerado pela passagem da esfera no sensor é um pulso muito rápido não sendo compatível com o tempo de comunicação da porta paralela.

13. Houve a necessidade de rever o projeto e chegar a um acordo comum, para redefinir outro sistema de integração que permitisse reconhecer o sinal do pulso emitido pelo sensor indutivo. A equipe se reuniu para analisar os resultados obtidos e estudar alternativas viáveis para resolução do problema.

14. Optou-se pela retomada do *kit Personalpic*, pela equipe estar familiarizada, pois o mesmo já havia sido utilizado originalmente no Projeto Integrador 2 e possuir comunicação serial via RS232.

15. Iniciou-se a religação do *kit Personalpic* ao protótipo do Seleccionador de Esferas.

16. Houve a necessidade de rever a programação e incluir códigos de comunicação para permitir o envio e recebimento de mensagens do microcontrolador ao supervisor e do supervisor ao microcontrolador.

17. Com a modificação do sistema de integração, foi refeita a programação do sistema supervisor, para execução do trabalho com o microcontrolador.

18. Utilizou-se o driver de comunicação ASCII, iniciou-se o teste de comunicação entre o *Elipse* e o *kit Personalpic*, o qual deveria integrar o funcionamento do protótipo Seleccionador de Esferas com o acionamento dos sensores e validar através da tela do IHM, representando o que acontecia fisicamente no supervisor.

19. Foi nesse ponto que o próximo problema aconteceu, onde não se conseguiu fazer a comunicação entre os meios envolvidos e o objetivo não foi alcançado nesta etapa do projeto.

20. Após várias tentativas de funcionamento num período de aproximadamente oito semanas, testou-se o meio físico como também os códigos do programa, os testes individuais, tanto no PIC quanto no *Elipse* funcionam, mas não se completou a comunicação. A equipe se reuniu-se para analisar os resultados obtidos e estudar alternativas viáveis para resolução do problema.

21. Devido a comunicação RS232 entre o microcontrolador e o supervisor não ter obtido resultado satisfatório de funcionamento a equipe optou por substituir o *kit Personalpic* pelo Arduino mega, por também ser um sistema prático e didático.

22. Foi retirado o *kit Personalpic* do protótipo do Seleccionador de Esferas e feita a ligação do Kit Arduino mega.

23. Novamente houve a necessidade de revisão do projeto com a redefinição do sistema de integração juntamente com os códigos necessários para o emprego do CI ATmega2560 o qual é utilizado no Arduino.

24. Foi realizado novamente a modificação do Script no sistema supervisorio, para execução do trabalho com o CI ATmega2560.

25. Utilizando o *driver* de comunicação *Modbus*, devidamente configurado no supervisorio, foi realizado o teste de comunicação entre o supervisorio e o sistema de integração do Arduino mega, forçando manualmente o sinal que seria gerado pelo sensor.

26. Então o resultado obtido foi positivo no teste aplicado e possibilitou assim concluir o estágio de comunicação.

27. A etapa seguinte foi a de integração entre o protótipo Seleccionador de Esferas, o sistema supervisorio Elipse E3 e a interface de integração o kit Arduino ATmega.

28. A integração entre os sistemas empregados no trabalho foi alcançada com êxito e os testes práticos realizados para verificar os ajustes necessários para conclusão dessa etapa.

29. Com a parte de trabalho prático finalizada, foi possível concluir o relatório escrito do trabalho de conclusão de curso, o qual foi trabalhado paralelamente com as outras etapas do projeto realizado.

30. Chegando por fim o trabalho de construção do material ilustrativo do projeto e tornando-o apto para apresentação à banca de professores, com intuito da avaliação final do projeto exposto.

4. DESENVOLVIMENTO

Este Trabalho de Conclusão de Curso toma como base o Seleccionador de Esferas, mostrado na Figura 11, apresentado na Disciplina Projeto Integrador 2, o qual funcionava didaticamente separando, quantificando e armazenando esferas de diâmetros diferentes, utilizando a Tecnologia de Programação Embarcada em Microcontrolador.

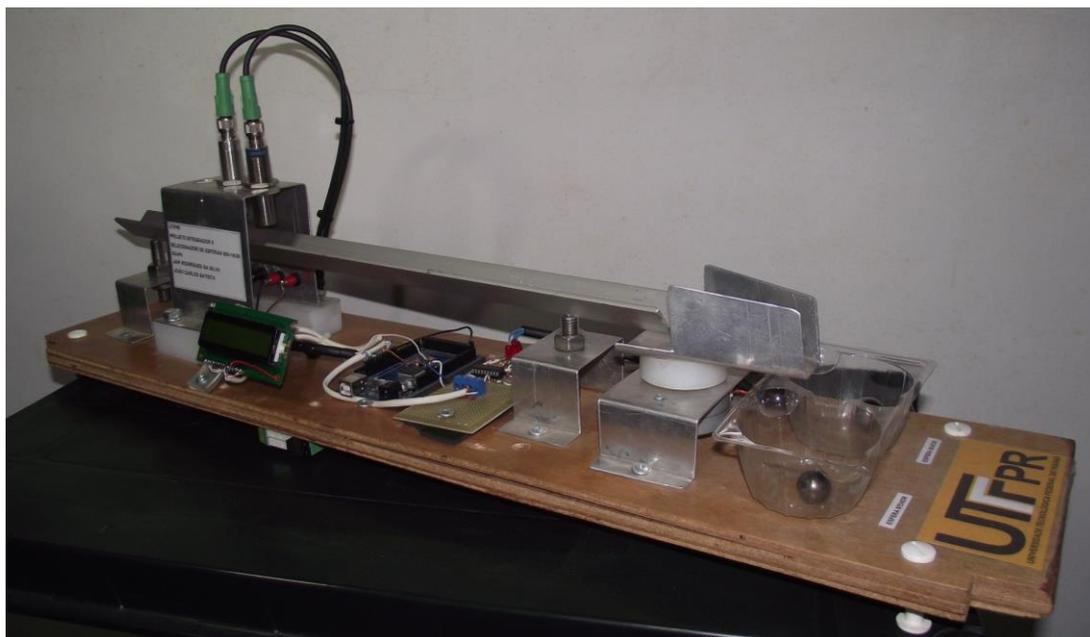


FIGURA 11 - SELECCIONADOR DE ESFERAS
Fonte: Autoria Própria, 2014.

Com base no protótipo desenvolvido sobre uma base de compensado, peças dobradas em alumínio foram montadas para servirem de suporte estrutural dos sensores e do motor de passo como atuador.

Foram feitas as Telas de Supervisão e Alarme do Sistema *SCADA*, conforme as informações recebidas dos sensores. O PDA foi programado de forma colaborativa em um dispositivo celular, na plataforma *ANDROID*, para receber os dados de processo e será operado por uma rede *WI-FI* sob um *IP (Internet Protocol)* dedicado.

A escolha do sistema *Elipse E3* foi devido estar disponível no mercado em modo Demo. Existe dentro do software do *E3* uma subdivisão de módulos de trabalho os quais estão descritos no Quadro 1 da página 17.

O projeto de programação foi desenvolvido em linguagem C, ver Apêndice B, a qual tem uma matéria específica de programação ao longo do curso de Tecnologia. A programação com a lógica desenvolvida para o projeto é transferida para o *software* Arduino.

Com o kit Arduino conectado ao computador, faz-se o upload do programa para o kit Arduino mega.

Foi utilizado o software Eclipse E3 como ferramenta de visualização do processo de seleção de esferas. O leiaute da página de apresentação do trabalho, mostrado na Figura 12, ilustra a tela de supervisão do sistema.



FIGURA 12 - TELA DO SUPERVISÓRIO
Fonte: Autoria Própria, 2014.

Através de desenhos feitos com um programa de modelagem em 3D, ver Apêndice A, foram sobrepostas três figuras com posições de trabalho distintas. Estas figuras foram utilizadas para criar a tela de supervisão no programa *Eclipse* de forma que, quando o selecionador estiver aguardando uma esfera, a figura na posição central fica visível sobrepondo as demais figuras.

Quando o selecionador identifica uma esfera que deverá ir para a direita, a figura com essa indicação sobrepõe as demais e quando o selecionador identifica a esfera que deverá ir para a esquerda, a figura correspondente sobrepõe as demais. Assim cria-se a ilusão de movimento para a tela.

Após a sobreposição das telas, foram criados botões, *displays*, textos e outros itens que serão utilizados para desenvolver funções específicas durante a operação do selecionador de esferas e no quadro 5 é apresentado o botão indicado no IHM e a atividade desempenhada pelo acionamento do mesmo.

BOTÃO	ATIVIDADE DESEMPENHADA PELO ACIONAMENTO
Simulação	Habilita a simulação de acionamento do selecionador
Direita	Simula o acionamento do Selecionador para o lado direito
Esquerda	Simula o acionamento do Selecionador para o lado esquerdo
Emergência	Para o funcionamento do sistema, indicando Falha
Reset	Zera todos os contadores e retira a condição de Falha do sistema

QUADRO 5 - ATIVIDADE DESEMPENHADA PELO BOTÃO
Fonte: Autoria Própria, 2015

Os *displays* demonstrados na tela principal têm suas atividades correspondentes a cada um deles indicadas no quadro 6.

DISPLAY	ATIVIDADE INDICADA
Esfera Menor	Armazena a quantidade de Esferas de diâmetro menor selecionadas
Esfera Maior	Armazena a quantidade de Esferas de diâmetro maior selecionadas
Total	Armazena o total de Esferas dos dois diâmetros selecionadas
Texto	Libera a colocação da próxima Esfera e indica qual das Esferas está passando pelo sensor indutivo no momento, se a menor ou a maior

QUADRO 6 - ATIVIDADES INDICADAS NO DISPLAY
Fonte: Autoria Própria, 2015

No projeto são usados três contadores, conforme mostrado na figura 11. O primeiro para as esferas menores, o segundo para as maiores e finalmente o terceiro contador indica a quantidade total de esferas que foram selecionadas e para zerar as contagens foi colocado um botão de *reset*.

A demonstração com a rotina do *script* do programa de funcionamento da tela do botão Direito mostrada na figura 13, descreve-se a lógica de funcionamento dependendo do sinal que recebe e comando que executa.

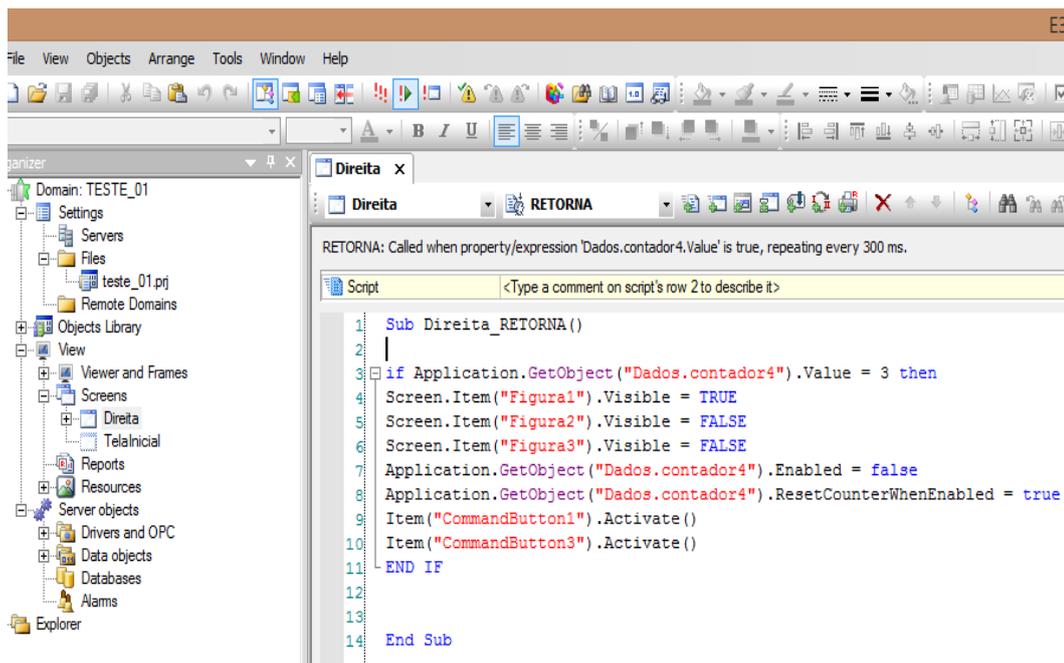


FIGURA 13 - TELA DO ELIPSE
Fonte: Os autores, 2014.

Com o equipamento energizado o funcionamento se dá com o movimento de passagem das esferas e para demonstrar melhor essa lógica a figura 14 mostra a passagem da esfera menor com $\varnothing 16$ mm de diâmetro pelos sensores, fazendo a sensibilização apenas do sensor 2 dando início a lógica empregada a este evento.

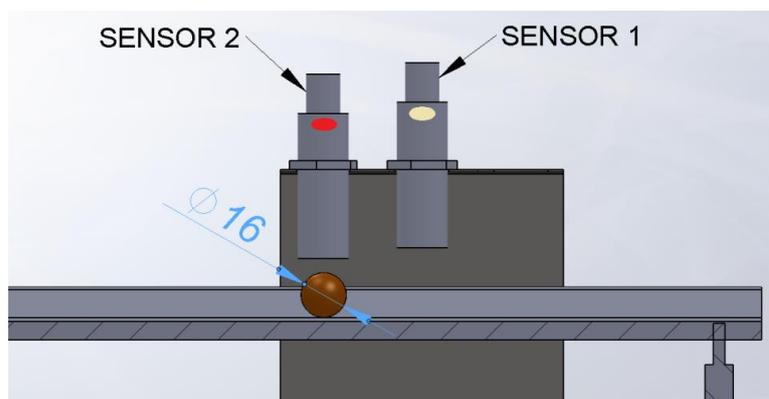


FIGURA 14 - ACIONAMENTO DO SENSOR 2 PELA ESFERA MENOR.
Fonte: Autoria Própria, 2014.

A figura 15 mostra a passagem da esfera maior de $\varnothing 20$ mm, sensibilizando o sensor 1 que inibe temporariamente o sensor 2, que mesmo sendo sensibilizado não envia o sinal para o Arduino dando sequência a lógica programada para o outro evento.

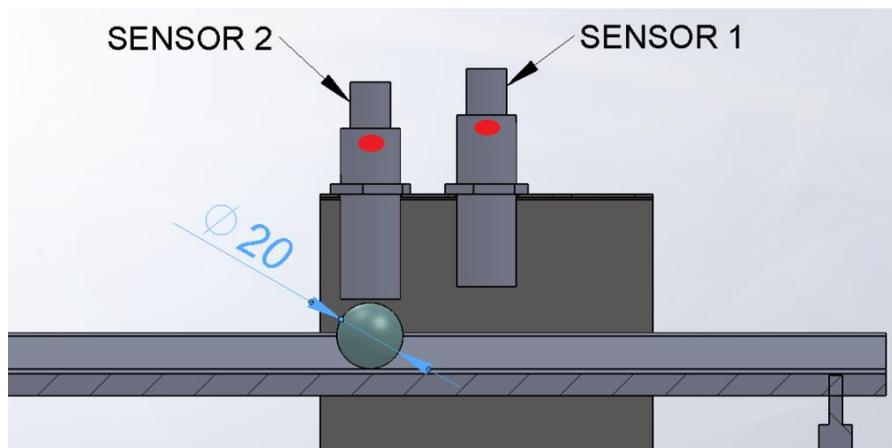


FIGURA 15 - ACIONAMENTO DOS SENSORES 1 E 2 PELA ESFERA MAIOR.
Fonte: Autoria Própria, 2014.

4.1 COMUNICAÇÃO DO SUPERVISÓRIO COM O CI

Para a comunicação do *Software* de supervisão e o CI ATmega2560 foi utilizado o protocolo de comunicação *Modbus*. A utilização do driver de comunicação MODBUS.dll que é um *driver* genérico criado pela *Elipse Software*.

Foram criadas quatro *Tags* (comandos) de comunicação diferentes para a aplicação neste projeto listadas no quadro 7.

TAG	DESCRIÇÃO
Primeira	ENVIA_1,2,3,4
Segunda	LEITURA_30
Terceira	LEITURA_21,22,23,24
Quarta	ENVIA_30

QUADRO 7 - DESCRIÇÃO DAS TAGS
Fonte: Autoria Própria, 2015.

A *Tag* "ENVIA_1,2,3,4" foi utilizada para o sistema supervisório enviar um número referente a posição em que o selecionador se encontra, a segunda "LEITURA_30" foi utilizada para receber os dados gerados pelo Arduino, conforme o acionamento da tecla emergência na tela IHM, a terceira "LEITURA_21,22,23,24" foi utilizada para receber a informação da leitura do sensor e a quarta "ENVIA_30" foi utilizada para enviar os dados gerados pelo supervisório para o *kit* Arduino, conforme o acionamento da tecla *Reset* na tela IHM.

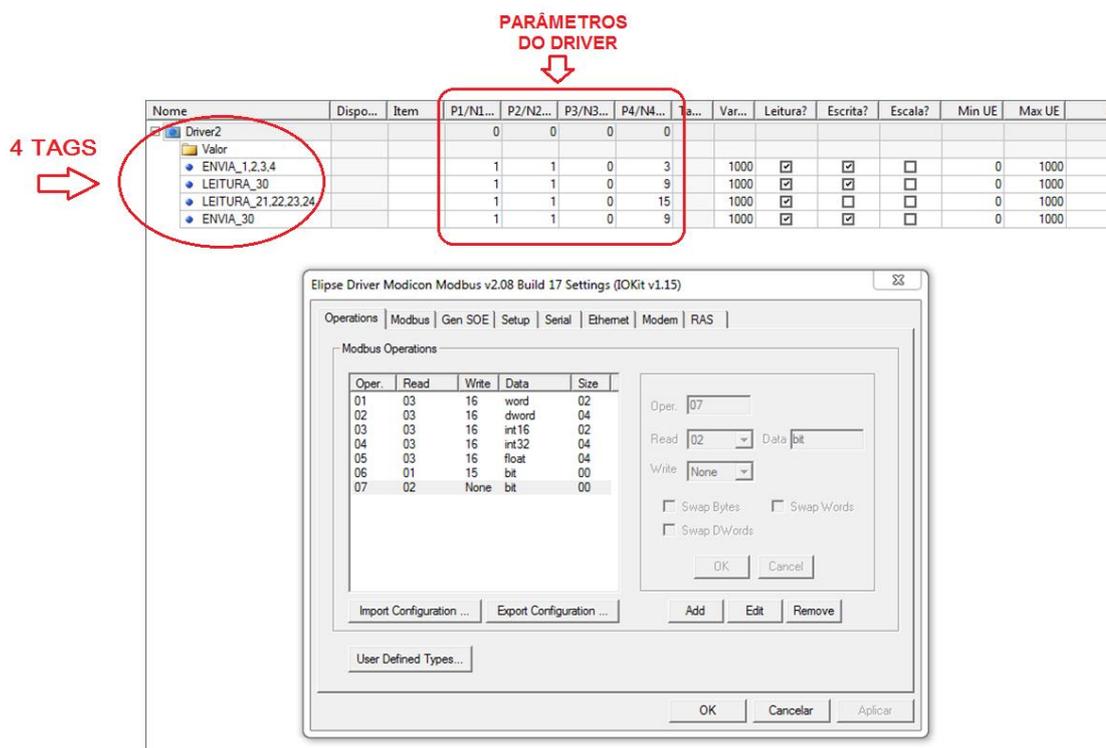


FIGURA 16 - TAGS DE COMUNICAÇÃO
 Fonte: Autoria Própria, 2014.

Cada *Tag* de Comunicação possui quatro parâmetros (N1, N2, N3, N4) cujos preenchimentos variam de acordo com cada *Driver* de comunicação, sendo necessária a consulta do Manual do *Driver* para que as configurações sejam realizadas corretamente. Os parâmetros que foram necessários para a configuração das quatro *Tags* utilizadas foram os parâmetros N1, N2 e N4. O parâmetro N3 não recebe nenhuma configuração

Nesse projeto, se tratando do envio de caracteres em *word*, foi escolhido o tipo de dado 's' (*String*). O *IOkit* é um recurso compartilhado do *Elipse E3* que proporciona as configurações devidas do *Driver* de comunicação. Com ele é possível escolher o formato em que serão enviados ou recebidos os dados, e o número de *bits* envolvidos nessa comunicação. Assim como a porta de comunicação, *baudrate*(*velocidade de comunicação 9600*), *stopbits*, *paridade* e *databits*, como é mostrado na Fig17.

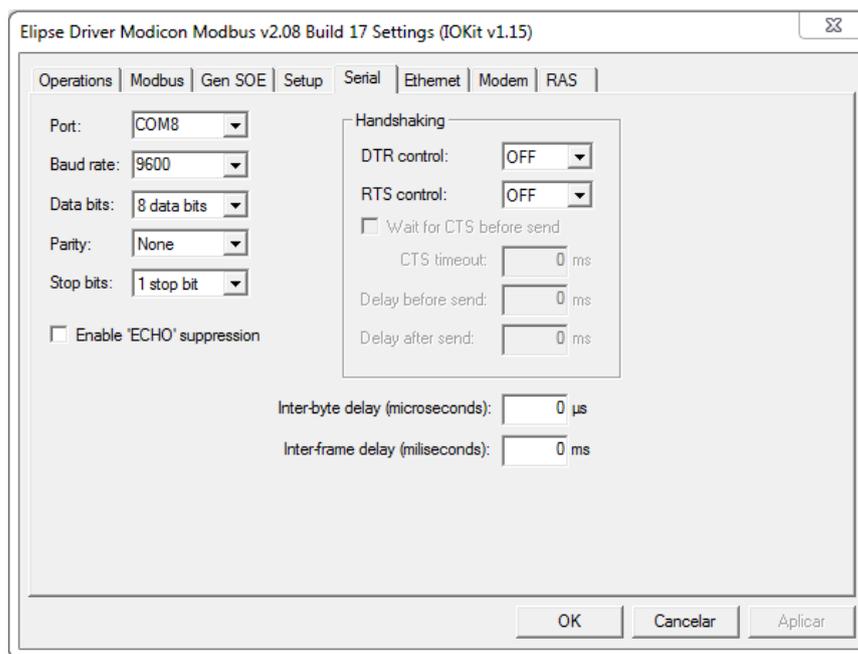


FIGURA 17 - TELA DO IOKIT.
Fonte: Autoria Própria, 2014.

Uma ferramenta importante no auxílio do desenvolvimento do projeto eletrônico é o *software Proteus Design Suite*, que desenha circuitos com a utilização de caixa de ferramentas que contém vários dispositivos eletrônicos e faz a simulação do funcionamento de circuitos eletrônicos digitais e analógicos. Na figura18 é mostrado como ficou o circuito do projeto do PIC no *software Proteus*.

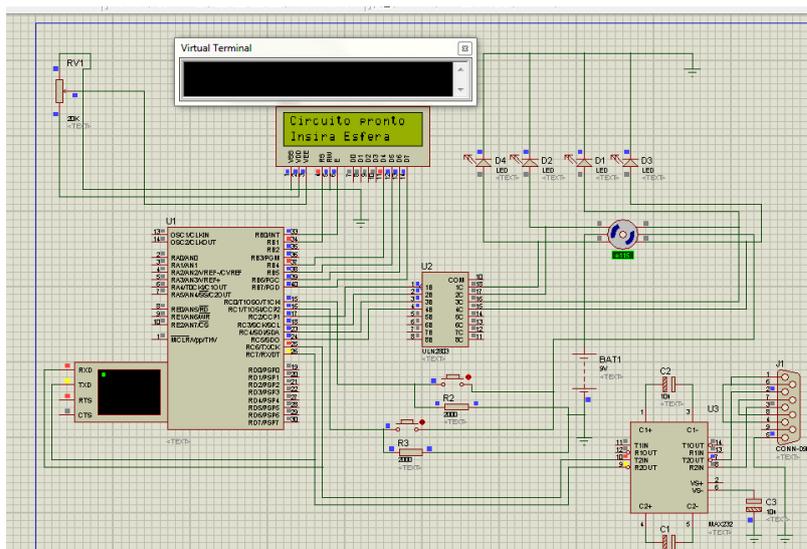


FIGURA 18 - TELA DO PROTEUS
Fonte: Autoria Própria, 2014.

Para uma melhor visualização do projeto a figura 19 mostra a tela do LCD (*Liquid Crystal Display*) com as indicações feitas para o circuito pronto para funcionamento indicando que pode ser inserida próxima esfera no equipamento, tendo em seguida a demonstração do LCD indicando que a esfera menor passou pelos sensores e foi direcionada para a caixa esquerda e tendo a indicação E no terminal virtual e também a indicação da esfera maior sensibilizando os sensores indutivos e direcionando a caixa para direita e indicando a letra D no terminal virtual.



FIGURA 19 - TELA AMPLIADA DO LCD DO PROTEUS
Fonte: Autoria Própria, 2014.

4.2 TESTES REALIZADOS

À medida que as tarefas eram realizadas fazia-se o respectivo teste de funcionalidade da etapa trabalhada e assim foram realizados vários testes para dar prosseguimento à próxima etapa de cada procedimento.

Teste de acionamento da placa de Interface Eletrônica Optoacoplada com recebimento de pulsos fornecidos pelos sensores indutivos, onde foi detectado deficiência no tempo de resposta da placa, impossibilitando a sua utilização no projeto.

Teste de funcionamento de gravação da linguagem “C” no PIC, conforme a necessidade de validação das etapas implementadas.

Através da utilização de um programa chamado RComSerial mostrado na figura 20, foi possível realizar teste de comunicação da porta COM. Foram

interligados os pinos 2 e 3 que são RX e TX do cabo adaptador *Trendnet* que faz a conversão RS232 com a USB e o resultado do teste foi positivo mostrando a transferência de comunicação, pois a mensagem escrita no campo designado para TX era enviada e recebida no campo RX.



FIGURA 20 - TELA DO PROGRAMA RCOMSERIAL
Fonte: www.rogercom.com

Na entrada RS232 do *kit Personalpic* os pinos correspondentes a RX e TX, foram interligados para fazer o teste de comunicação para que a rotina de falha feita pela mudança da letra "D" (Direita) pela "F" (Falha) e quando acionado os dois sensores o corpo direcionador do protótipo virava para a direita e realizava o ciclo completo, habilitando para o próximo passo.

Quando apenas o segundo sensor fosse acionado o corpo direcionador, girava para a esquerda e voltava para o centro bloqueando o sistema, dando o resultado esperado naquele teste.

Para testar-se a comunicação entre a interface do Arduino e o sistema de supervisão estava ocorrendo, a maneira utilizada foi forçando o sinal manualmente com 5 Volts, simulando a atuação do sensor na entrada respectiva a ele e analisando a resposta obtida na tela do computador.

E o teste mais importante do projeto, foi o de integração do sistema físico com o computador tendo o *Elipse* como base de integração e a interface Arduino. Com a

passagem das esferas pelos sensores acionando o atuador para dar o movimento no Seleccionador de Esferas e a sua reprodução no IHM, com a simulação de movimento bem como a representação no *display* representando o LCD de indicação da atividade que acontecia no exato momento.

4.3 ANÁLISE DE RESULTADOS

Inicialmente o projeto visava utilizar a porta paralela conectada a placa de interface “JI”, que é um circuito desenvolvido nos laboratórios da UTFPR. Foi feita toda a programação necessária no *software* supervisor e realizado os testes de comunicação onde foram obtidos sucessos, porém quando houve a interligação do projeto com a interface ocorreu um problema devido a velocidade de comunicação da porta ser muito lenta, não atendendo a necessidade do projeto.

Após a constatação destes resultados, a impossibilidade da comunicação com a porta paralela, foi definido a substituição da placa “JI” pelo PIC 16F877A que fazia parte do projeto inicial. Sendo preciso implementar o código de comunicação com a interface RS 232, alterando o código do programa no PIC para abrir a comunicação RS 232, alterando também a programação no supervisor para que recebesse os sinais do microcontrolador.

Depois de ajustada a programação do PIC e do supervisor foram iniciados os testes os quais não obtiveram sucesso na comunicação. Decidiu-se fazer então o teste individual de cada parte do sistema. Testada a programação do PIC interligando RX com TX na saída do cabo de comunicação, foi verificado que o mesmo estava escrevendo e lendo, e o mesmo foi feito com o supervisor, porém a escrita de um não chegava ao outro, foi feito o teste com o *driver* de comunicação RS 232, a pedido do suporte técnico do supervisor, que também não surtiu efeito.

Depois destas ultimas tentativas, resolveu-se substituir o *kit Personalpic* pelo kit Arduino e refazer a programação em “C”, agora com a IDE do Arduino para a comunicação entre o CI e o supervisor. Foram feitas as alterações necessárias no *script* do supervisor para possibilitar a troca de informações com o Arduino. Testes em bancada foram realizados, forçando o sinal que posteriormente seria gerado pelos sensores e observou-se resultado positivo. Conectado o kit Arduino com o protótipo do seleccionador e realizado os testes finais com as esferas passando pelos

sensores, foi possível observar a correta comunicação entre os sistemas supervisor e o *kit* Arduino.

4.4 ANÁLISE DIDÁTICA DO PROJETO

Para auxiliar na preparação e montagem do projeto para apresentação é necessário a sequência de passos a serem seguidos pelo responsável da apresentação, sendo criado assim um tutorial de apresentação.

Passo 1: Ajustar a regulagem da canaleta guia, deixando um leve declive em sentido ao corpo selecionador, a fim de que as esferas deslizem quando do seu depósito na canaleta. Essa regulagem será feita através dos parafusos e porcas situados nas extremidades da canaleta mostrados na figura 21.

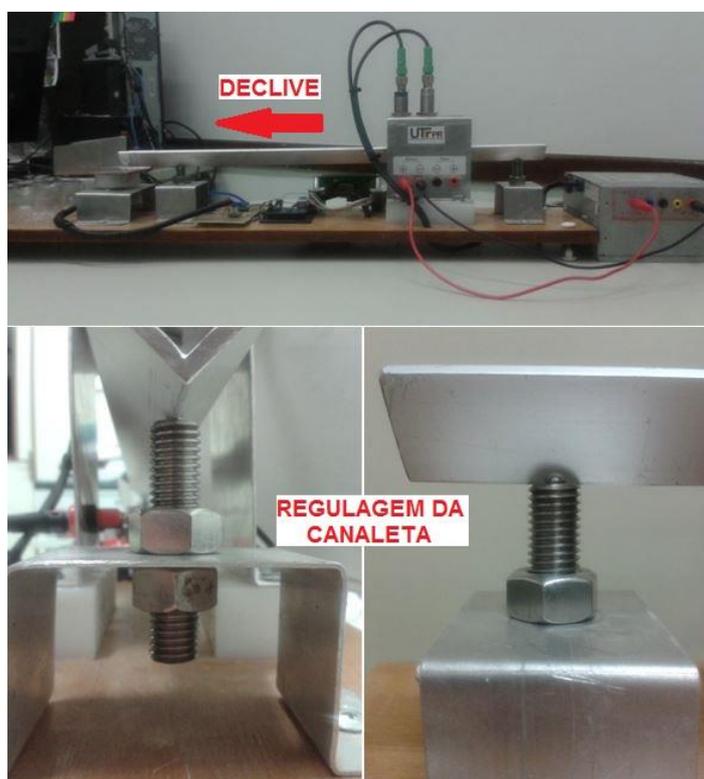


FIGURA 21 - PASSO 1: REGULAGEM DA CANALETA
Fonte: Autoria Própria 2015

Passo 2: Regular as alturas dos sensores indutivos indicados na figura 22, através da rosca existente no corpo dos mesmos, iniciando pelo sensor 2, deixando a uma altura em que a esfera de \varnothing 20mm passe tangenciando por ele sem que haja nenhuma interferência. Passa-se então para o sensor1, regulando com uma distância de afastamento de aproximadamente 3 mm, com o propósito de que a

esfera de $\varnothing 16\text{mm}$ passe sensibilizando apenas o sensor 2, devido a distância ser maior que o alcance do range determinado no *datasheet*.

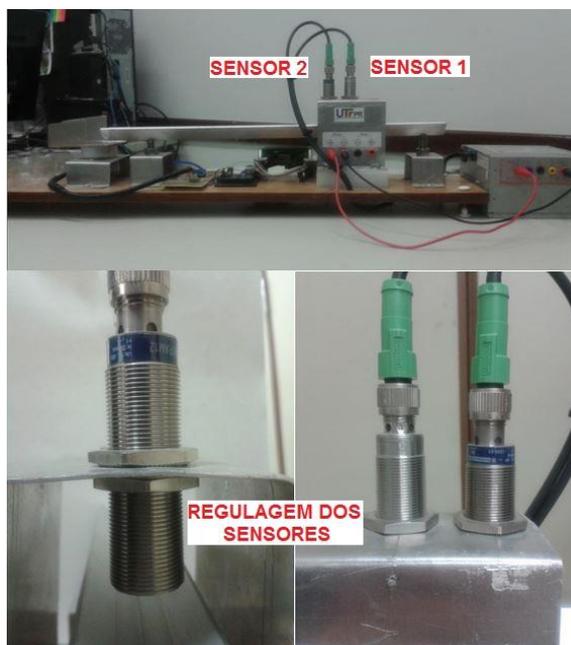


FIGURA 22 - PASSO 2: REGULAGEM DOS SENSORES
Fonte: Autoria Própria, 2015

Passo 3: Conectar os cabos de alimentação da fonte AT às conexões de alimentação do protótipo, os quais já possuem as devidas marcações de tensão correspondentes a 24 Volts para energizar os sensores indutivos, (figura 23).

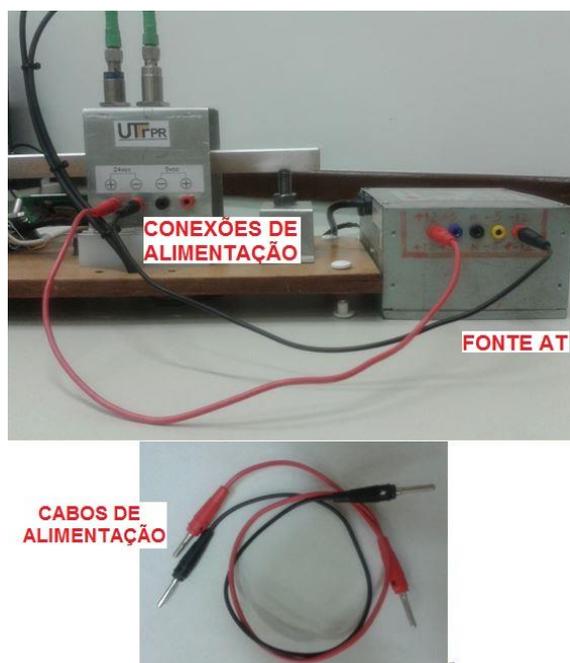


FIGURA 23 - PASSO 3: CONEXÕES DE ALIMENTAÇÃO
Fonte: Autoria Própria, 2015

Passo 4: A figura 24 demonstra como conectar o cabo de comunicação serial do *kit* Arduino com o computador na entrada USB.



FIGURA 24 - PASSO 4: CONEXÃO COMPUTADOR/ARDUINO
Fonte: Autoria Própria, 2015

Passo 5: Inicializar o programa *E3 Studio*, abrir o projeto do Seleccionador de esferas e executar o aplicativo (*Saves and Runs the Domains*), demonstrado na figura 25.

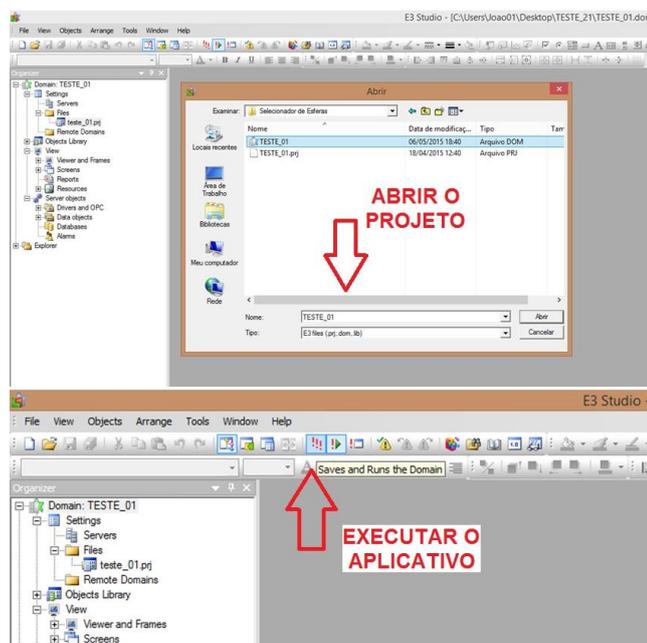


FIGURA 25 - PASSO 5: INICIALIZAR O PROGRAMA
Fonte: Autoria Própria, 2015

Passo 6: Será aberta a tela principal representando o IHM, para visualizar as informações e ter a possibilidade de simulação através dos botões indicados na figura 26.



FIGURA 26 - PASSO 6: TELA IHM
Fonte: Autoria Própria. 2015

Passo 7: Iniciar o processo de alimentação manual, inserindo as esferas uma por vez a partir da liberação indicada na tela do IHM ou do LCD, como mostra a figura 25.



FIGURA 27 - PASSO 7: ALIMENTAÇÃO DAS ESFERAS
Fonte: Autoria Própria, 2015

4.5 DIFICULDADES ENCONTRADAS / SUPERADAS

A primeira dificuldade encontrada foi no teste do projeto com a interface “JI” pois a resposta da placa com o recebimento do sinal enviado pelo sensor indutivo não era rápida o suficiente para armazenar a informação da passagem das esferas, não conseguindo assim trabalhar o sinal para que pudesse ser enviado para o sistema supervisor.

Optou-se então em voltar a utilizar o *kit personalpic* com o microcontrolador, pois já tínhamos a garantia do funcionamento do protótipo com o PIC16877A.

Foi então alterado o protocolo de comunicação o que gerou outro problema, toda a programação feita até então não servia, foi preciso refazê-la para utilizar o PIC, o que possibilitou aprofundar um pouco mais os conhecimentos em linguagem “C” e principalmente o protocolo UART que é muito utilizado quando se trata de comunicação via porta serial.

E foi justamente essa comunicação que falhou, embora toda programação estivesse correta, não foi possível enviar e receber dados pela serial.

Após essa falha optou-se em utilizar kit Arduino com o CI ATmega 2560, esta alteração exigiu uma reescrita quase que total do código, pois o sistema de comunicação da linguagem “C” do ATmega difere da escrita do PIC16F877A.

5. CONSIDERAÇÕES FINAIS

Ocorreram na prática problemas de comunicação e de integração entre os sistemas escolhidos inicialmente, existiu a necessidade da troca das interfaces por duas vezes, para que o projeto tivesse o desempenho esperado.

Durante o desenvolvimento do projeto foi possível a aplicação de conhecimentos adquiridos durante o curso com a integração de algumas tecnologias tais como programação em “C”, microcontroladores, sensores, *softwares* e eletrônica.

A partir da escolha do *kit* Arduino e a modificação dos *scripts* de comunicação, os resultados práticos de integração dos sistemas envolvidos no projeto aconteceram finalmente na prática, resultando em um projeto que pode ser apresentado em todas as suas etapas.

No supervisor desenvolvido tem-se a possibilidade de monitorar o sistema Seleccionador de Esferas, visualizando na tela do computador a quantidade de esferas de cada um dos dois diâmetros selecionáveis, a quantidade total de esferas que foram selecionadas.

Pode-se interferir no processo de seleção via IHM ao acionar o botão de falha para assim bloquear o circuito e *resetar* para tornar o sistema novamente operável. A simulação via tela de IHM, pode ser realizada com o acionamento do botão “simulação” e fazer a simulação completa do ciclo.

Obtendo um projeto montado em tamanho reduzido, com ligações elétricas e ajustes simples de serem realizados, com a possibilidade de transporte para locais onde poderá ser apresentado com enfoque didático, simulando uma linha industrial automatizada com supervisão.

A apresentação poderá ser feita em escolas para alunos que estejam concluindo o ensino médio, calouros iniciando o curso superior dentro das instituições de ensino e feiras tecnológicas, comprovando sua aplicabilidade como recurso simples ou como componente de um módulo de instrução didático.

5.1 PROPOSTA PARA TRABALHOS FUTUROS

Para um aperfeiçoamento do projeto, propõe-se que sejam feitas apresentações do módulo em sala de aula na disciplina de Introdução a Tecnologia de Automação, com a elaboração de um questionário de avaliação da interatividade do material didático com o aluno, para receber um *feedback* da real ajuda na aplicação didática do trabalho exposto.

Desenvolver novos módulos com outras lógicas de programação, inserir novas interfaces e implementar sensores e atuadores diferentes dos aplicados neste trabalho de conclusão de curso.

REFERÊNCIAS

ANDRÉ, Marli E. D. A. de. OLIVEIRA, Maria R. N. S.: **Alternativas no Ensino de Didática** 12ª ed. Campinas. Editora Papirus,1997.

Arduino, Disponível em: www.arduino.cc. Acesso em 08 de março de 2015.

ASCII, Manual do Driver, Disponível em:
<http://www.elipse.com.br/drivers.aspx?fltr1=1&fltr2=0&index=16&idioma=1>

ASCIITABLE, Disponível em <http://www.asciitable.com> Acesso em 15 de fevereiro de 2015.

BRITES, Felipe Gonçalves; SANTOS, Vinícius Puga de Almeida: **Motor de Passo PTEle**. Dissertação (Programa de Educação Tutorial) – Universidade Federal Fluminense, 2008.

CAETANO, André G.L.S. **Sistemas de Supervisão de Chão-de-Fábrica: Uma Contribuição para Implantação em Indústrias de Usinagem**. 2000. 110 f. Dissertação (Mestrado em Engenharia Mecânica) - Escola de Engenharia de São Carlos da Universidade de São Paulo, 2000.

COELHO, Marcelo S.: **Sistemas Supervisórios**; disponível em <
<http://www.ebah.com.br/content/ABAAAUMcAH/sistemas-supervisorios>>; acessado em 12/02/2015

Digikey Disponível em <http://www.digikey.com/product-detail/en/PM55L-048-HHD0/P14344-ND/2417068> Acesso em 20 de fevereiro de 2015

E3, Manual de Referências de Scripts do E3, Disponível em:
http://www.elipse.com.br/downloads.aspx?id_produto=1&versao=3.5&doc=true&idioma=1

IDOETA, Ivan V.; CAPUANO, Francisco Gabriel. Elementos de eletrônica digital. 18. ed. São Paulo: Érica, 1992.

IFM Disponível em <http://www.ifm.com/products/br/ds/IGS208.htm> Acesso em 20 de fevereiro de 2015

Labcenter, Disponível em:
http://www.labcenter.com/download/prodemo_autodl_general.cfm.
Acesso em 05 de março de 2015.

LIPSON, Phillip; ZALM Geert van der, Bosch Rexroth Corp, (2011) . **Inside Machines: PC versus PLC: Comparing control options**. Disponível em:
<http://www.controleng.com/single-article/inside-machines-pc-versus-plc->

comparing-control-options/9bf8690c6f23b11370bec90b52cb15c9.html Acesso em 28 Novembro 2014]

Manual do Usuário do Personalpic. Disponível em www.personalpic.xpg.com.br Acesso em 20 de fevereiro de 2015

Manual ELIPSE. Disponível em <http://www.elipse.com.br/port/index.aspx> Acesso em 07junho 2014.

MARGOLIS, Michael. **Arduino Cookbook**. Editora: O'Reilly Media, Inc., 2011.

MARTINS, N.A. **Sistemas Microcontrolados - Uma Abordagem com o Microcontrolador PIC 16f84**. São PAulo: Novatec Editora. 2005.

Módulo de Relé. Disponível em: www.phoenixcontact.com. Acesso em 14 de março de 2015.

OGATA, Katsuhiko.:**Engenharia do Controle Moderno**. Editora: Prentice-Hall, 1993.

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. Rio de Janeiro: Editora Brasport, 1999.

Robotizando, Disponível em:<http://www.robotizando.com.br/pinagem_rs232.php > Acesso em 20 de fevereiro de 2015.

Rogercom.com.,Disponível em <http://www.rogercom.com/CursoOnlineUSB/ModuloUnicoAula015.htm> Acesso em 20 de fevereiro de 2015

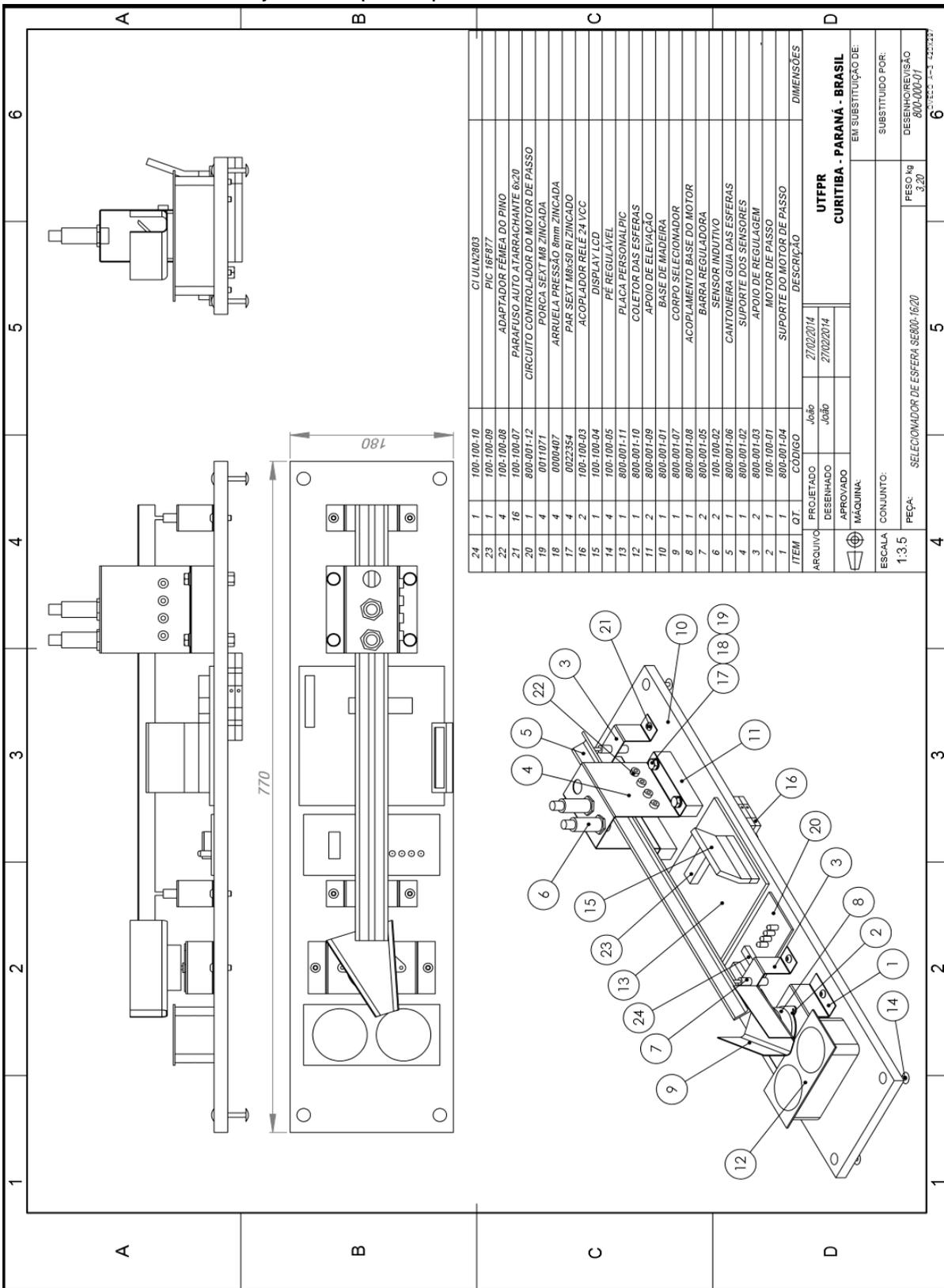
THOMAZINI, Daniel; **Sensores industriais: fundamentos e aplicações**. São Paulo: Érica, 2010

ZELENOVSKY, Ricardo. MENDONÇA, Alexandre.: **Microcontroladores: Programação e Projeto com a Família 8051**, MZ Editora, 2005

APÊNDICE

APÊNDICE A

Desenho de conjunto do protótipo Seleccionador de Esferas SE800-16/20



APÊNDICE B

Linguagem de Programação

```
#include <SimpleModbusSlave.h> //http://code.google.com/p/simple-modbus/
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); //Configura os pinos do Arduino para se
const int HOLDING_REGS_SIZE = 22; // I use 22 registers, first 11 for sending
//and next 11 registers for receiving info from Mach3
unsigned int holdingRegs[HOLDING_REGS_SIZE]; // function 3 and 16 register are

int rcd[11]; //envia informação para o ecrã
int mcd[11]; //recebe informação do ecrã
const int ledPin13 = 13;
const int ledPin12 = 12;
const int ledPin11 = 11;
const int ledPin10 = 10;
const int sensorPIN8 = 8;
const int sensorPIN9 = 9;
int ledState = HIGH;
bool direita = false;
bool esquerda = false;
bool aux01 = false;
int aux02 = 2;
int sensorState1 = 0;
int sensorState2 = 0;

void setup() {

  pinMode(ledPin13, OUTPUT);
  pinMode(ledPin12, OUTPUT);
  pinMode(ledPin11, OUTPUT);
  pinMode(ledPin10, OUTPUT);
  pinMode(sensorPIN8, INPUT); // sensor 01 do projeto
  pinMode(sensorPIN9, INPUT);
  // with Arduino Micro I used modified library files and this line:
  modbus_configure(&Serial, 9600, SERIAL_8N2, 1, 2, HOLDING_REGS_SIZE, holdingRegs);
  //lcd.begin(16, 2);
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Insira a Esfera !");
  lcd.setCursor(0, 10);
}

void loop() {
  sensorState1 = digitalRead(sensorPIN8);
  sensorState2 = digitalRead(sensorPIN9);
  //lcd.begin(16, 2);
  modbus_update();
  // read the sensor:
  int inByte = holdingRegs[2];

  if (sensorState1 == HIGH && holdingRegs[8] == 30) { //AVANÇA PARA ESQUERDA
```

```

    esquerda=true;
    holdingRegs[2] = 1 ;
}
if (sensorState2 ==HIGH && esquerda ==false && holdingRegs[8] == 30) {
    holdingRegs[2] = 3 ;
}
if (holdingRegs[2] == 2 ) {///AVANÇA PARA DIREITA
    holdingRegs[2] = 2 ;
}
if (holdingRegs[2] == 4 ) {///AVANÇA PARA DIREITA
    holdingRegs[2] = 4 ;
}
if (holdingRegs[8] == 40 && holdingRegs[2] < 5 ) {
    holdingRegs[2] = 5 ;
}

}

switch (inByte) {
    case 1:
        delay(400);
        ledState =HIGH;
        digitalWrite(ledPin13, ledState);
        delay(100);
        ledState =LOW;
        digitalWrite(ledPin13, ledState);
        delay(100);
        ledState =HIGH;
        digitalWrite(ledPin12, ledState);
        delay(100);
        ledState =LOW;
        digitalWrite(ledPin12, ledState);
        delay(100);
        ledState =HIGH;
        digitalWrite(ledPin11, ledState);
        delay(100);
        ledState =LOW;
        digitalWrite(ledPin11, ledState);
        delay(100);
        ledState =HIGH;
        digitalWrite(ledPin10, ledState);
        delay(100);
        ledState =LOW;
        digitalWrite(ledPin10, ledState);
        delay(100);
        lcd.clear();
        lcd.begin(16, 2);
        // Print a message to the LCD.
        lcd.print("  Esfera Maior ");
        lcd.setCursor(4, 10);

```

```

// print the number of seconds since reset:
// lcd.print(millis() / 1000);
lcd.print("Esquerda");
holdingRegs[14] = 21;
holdingRegs[2] = 0 ;
esquerda=false;
break;
case 2:////RETORNA PARA CENTRO
  delay(100);
  ledState =HIGH;
  digitalWrite(ledPin10, ledState);
  delay(100);
  ledState =LOW;
  digitalWrite(ledPin10, ledState);
  delay(100);
  ledState =HIGH;
  digitalWrite(ledPin11, ledState);
  delay(100);
  ledState =LOW;
  digitalWrite(ledPin11, ledState);
  delay(100);
  ledState =HIGH;
  digitalWrite(ledPin12, ledState);
  delay(100);
  ledState =LOW;
  digitalWrite(ledPin12, ledState);
  delay(100);
  ledState =HIGH;
  digitalWrite(ledPin13, ledState);
  delay(100);
  ledState =LOW;
  digitalWrite(ledPin13, ledState);
  delay(100);
  lcd.clear();
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Insira a Esfera !");
  lcd.setCursor(0, 10);
  holdingRegs[14] = 22;
  holdingRegs[6] = holdingRegs[6] * 0;
  holdingRegs[2] = 0 ;
  break;
// MOVE MOTOR PARA LADO DIREITO.
case 3:
  delay(100);
  ledState =HIGH;
  digitalWrite(ledPin10, ledState);
  delay(100);
  ledState =LOW;

```

```

digitalWrite(ledPin10, ledState);
delay(100);
ledState =HIGH;
digitalWrite(ledPin11, ledState);
delay(100);
ledState =LOW;
digitalWrite(ledPin11, ledState);
delay(100);
ledState =HIGH;
digitalWrite(ledPin12, ledState);
delay(300);
ledState =LOW;
digitalWrite(ledPin12, ledState);
delay(100);
ledState =HIGH;
digitalWrite(ledPin13, ledState);
delay(100);
ledState =LOW;
digitalWrite(ledPin13, ledState);
delay(100);
lcd.clear();
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("  Esfera Menor ");
lcd.setCursor(4, 10);
// print the number of seconds since reset:
// lcd.print(millis() / 1000);
lcd.print("Direita");
holdingRegs[14] = 23;
holdingRegs[2] = 0 ;
break;
case 4:// RETORNA PARA CENTRO
delay(100);
ledState =HIGH;
digitalWrite(ledPin13, ledState);
delay(100);
ledState =LOW;
digitalWrite(ledPin13, ledState);
delay(100);
ledState =HIGH;
digitalWrite(ledPin12, ledState);
delay(100);
ledState =LOW;
digitalWrite(ledPin12, ledState);
delay(100);
ledState =HIGH;
digitalWrite(ledPin11, ledState);
delay(100);
ledState =LOW;

```

```

digitalWrite(ledPin11, ledState);
delay(100);
ledState =HIGH;
digitalWrite(ledPin10, ledState);
delay(100);
ledState =LOW;
digitalWrite(ledPin10, ledState);
delay(100);
lcd.clear();
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Insira a Esfera !");
lcd.setCursor(0, 10);
holdingRegs[14] = 22;
holdingRegs[6] = holdingRegs[6] * 0;
holdingRegs[2] = 0 ;
break;
case 5:
delay(500);
lcd.clear();
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Sistema em Falha");
lcd.setCursor(4, 10);
// print the number of seconds since reset:
// lcd.print(millis() / 1000);
lcd.print("Resetar");
holdingRegs[2] = 6 ;
aux01 =true;
break;
default:
if ( holdingRegs[8] == 30) {///AVANÇA PARA ESQUERDA
aux02 = aux02 ;
}
}
}

```

ANEXOS

ANEXO A

Datasheet do motor de passo PM55L-048



PM55L-048



PM Motor
PM Type

Reference Characteristics

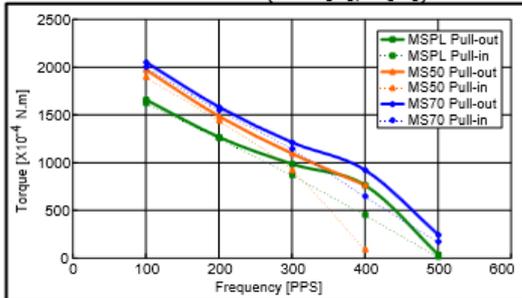
Motor Size	PM55L-048	
Number of Steps per Rotation	48(7.5°/Step)	
Drive Method	2-2 PHASE	
Drive Circuit	UNIPOLAR CONST. VOLT.	BIPOLAR CHOPPER
Drive Voltage	24[V]	24[V]
Current/Phase		800[mA]
Coil Resistance/Phase	30[Ω]	5.5[Ω]
Drive IC	25C3346	UDN2916B-V
Magnet Material	Ferrite plastic magnet (MSPL) Polar anisotropy ferrite sintered magnet (MS50) Nd-Fe-B bonded magnet (MS70)	
Insulation Resistance	100M[Ω] MIN	
Dielectric Strength	AC 500[V] 1[min]	
Class of Insulation	CLASS E	
Operating Temp.	-10[°C] ~ 50[°C]	
Storage Temp.	-30[°C] ~ 80[°C]	
Operating Hum.	20[%] RH ~ 90[%] RH	

Applications

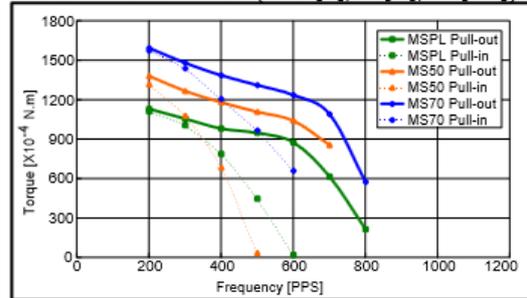
- OA Equipment : Printers / Scanners
- Industrial equipment : Flow control valves
- Toys : Slot machines
- Home automation appliances : Sewing machines

Torque Characteristics

PM55L-048 UNI-CONST. V (at 24[V],30[Ω])

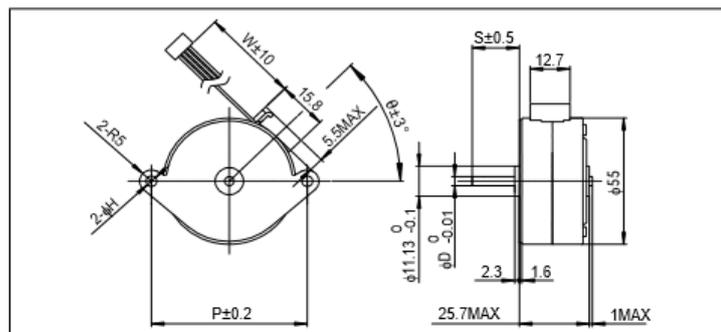


PM55L-048 BI-CHOPPER (at 24[V],5.5[Ω],800[mA])



These torque values are reference only. Heat radiation conditions and temperature rise effect by duty are different on each equipment, therefore please select motors after considering the heat conditions in the actual equipment.

Dimensions



If you would like to know this Dimensions(D,S,W,θ), Please see Standard Dimensions in our Home Page.

ANEXO B

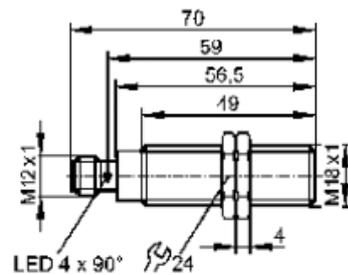
Datasheet do sensor indutivo IGS 208

efector100



IGS208

Inductive sensors



Made in Germany

Product characteristics	
Inductive sensor	
Metal thread M18 x 1	
Connector	
Operation as 3-wire or 2-wire possible	
Increased sensing range	
Optical setting aid (2 LED)	
gold-plated contacts	
Sensing range 8 mm [f]	
flush mountable	

Electrical data	
Electrical design	3-wire DC PNP; 2-wire DC PNP/NPN
Operating voltage [V]	10...30 DC
Current consumption [mA]	< 12 (24 V)
Protection class	II
Reverse polarity protection	yes

Outputs	
Output function	normally open
Voltage drop [V]	< 2.8
Minimum load current [mA]	2 *)
Leakage current [mA]	< 0.5 *)
Current rating [mA]	100
Short-circuit protection	pulsed
Overload protection	yes

Switching frequency [Hz]	400	
Range		
Sensing range [mm]	8	
Real sensing range [mm] (Sr)	8 ± 10 %	
Operating distance [mm]	0...6.5	
Accuracy / deviations		
Correction factors	mild steel = 1 / stainless steel approx. 0.7 / brass approx. 0.5 / Al approx. 0.4 / Cu approx. 0.3	
Hysteresis [% of Sr]	3...15	
Switch-point drift [% of Sr]	-10...10	
Environment		
Ambient temperature [°C]	-25...70	
Protection	IP 67	
Tests / approvals		
EMC	EN 61000-4-2 ESD:	4 kV CD / 8 kV AD
	EN 61000-4-3 HF radiated:	10 V/m (80...1000 MHz)
	EN 61000-4-4 Burst:	2 kV
	EN 61000-4-6 HF conducted:	10 V (0.15...80 MHz)
	EN 55011:	class B
MTTF [Years]	1364	
Mechanical data		
Mounting	flush mountable	
Housing materials	brass white bronze coated; active face: PBT (Pocan)	
Weight [kg]	0.055	
Displays / operating elements		
Output status indication	LED	yellow
Setting aid	LED	red
Electrical connection		
Connection	M12 connector; gold-plated contacts	
Wiring		