

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE MANUTENÇÃO INDUSTRIAL
CURSO SUPERIOR DE TECNOLOGIA EM MANUTENÇÃO INDUSTRIAL**

**ANDRIO ELIAS DUTRA
FABIO JOSE ZIMMERMANN**

**APRIMORAMENTO E REFORMA DE UMA LAVADOURA DE
LOUÇAS INDUSTRIAL**

TRABALHO DE CONCLUSÃO DE CURSO

MEDIANEIRA - PR

2014

ANDRIO ELIAS DUTRA
FABIO JOSE ZIMMERMANN

**APRIMORAMENTO E REFORMA DE UMA LAVADOURA DE
LOUÇAS INDUSTRIAL**

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Tecnólogo, do curso de Tecnologia em Manutenção Industrial, da Universidade Tecnológica Federal do Paraná - UTFPR - Campus Medianeira.

Orientador: Prof. Luis Paulo Zanolla Boschetti

MEDIANEIRA - PR

2014



TERMO DE APROVAÇÃO

APRIMORAMENTO E REFORMA DE UMA LAVADOURA DE LOUÇAS INDUSTRIAIS

por

ANDRIO ELIAS DUTRA

FABIO JOSE ZIMMERMANN

Este Trabalho de Conclusão de Curso foi apresentado as 21:00 h do dia 13 de fevereiro de 2014 como requisito parcial para a obtenção do título de Tecnólogo em Manutenção Industrial, da Universidade Tecnológica Federal do Paraná. Os acadêmicos foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Luis Paulo Zanolla Boschetti
UTFPR – *Câmpus* Medianeira
Prof. Orientador

Prof. Almiro Weiss
UTFPR – *Câmpus* Medianeira
(Convidado)

Prof. Ivair Marchetti
UTFPR – *Câmpus* Medianeira
(Convidado)

Prof. Yuri Ferruzzi
UTFPR – *Câmpus* Medianeira
(Responsável pelas atividades de TCC)

A Folha de Aprovação assinada encontra-se na coordenação do Curso de Tecnologia em Manutenção Industrial.

Dediquemos este trabalho aos nossos pais, pelo apoio e confiança que depositam em nos a cada dia que passa e por estar sempre nos apoiando nas decisões e percalços da vida. Dediquemos à todos as pessoas que nos auxiliaram, permitindo assim atingir o sucesso do estudo.

AGRADECIMENTOS

À Deus pelo dom da vida, pela fé e perseverança para vencer os obstáculos.

Aos nossos pais, pela orientação, dedicação e incentivo nessa fase do curso de graduação e durante toda minha vida.

Ao nosso orientador, o professor Luis Paulo Zanolla Boschetti, que foi um dos responsáveis por despertar o interesse pelo Arduino.

Agradecemos à empresa Hospital e Maternidade Nossa Senhora da Luz pelo espaço concedido para execução do projeto.

Agradecemos aos pesquisadores e professores do curso de Tecnologia em Manutenção Industrial e professores da UTFPR, *Campus Medianeira*.

Enfim, somos **gratos** a todos que contribuíram de forma direta ou indireta para realização desta trabalho de conclusão de curso.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário”.

(Albert Einstein)

RESUMO

ZIMMERMANN, Fabio Jose e DUTRA, Andrio Elias. Aprimoramento e Reforma de uma Lavadora de Louças Industrial. Trabalho de Conclusão de Curso - Tecnologia em Manutenção Industrial, Universidade Tecnológica Federal do Paraná. Medianeira, 2014.

O trabalho consiste no aprimoramento e reforma de uma lavadora de louças industrial. A lavadora é composta por duas motobombas que são responsáveis pela lavagem e enxágüe das louças, por resistências para aquecimento da água, por sensores e botões de operação. Além da reforma, o trabalho mostra o projeto elétrico e a execução das ligações elétricas de cada motor, das resistências e dos botões, tanto da parte de potência, quanto da parte de sinais, e também a utilização do sistema Arduino para automatizar a máquina e seu programa escrito em linguagem C.

Palavras-chave: Lavadora, Arduino, Linguagem C.

ABSTRACT

ZIMMERMANN, Fabio and Jose Dutra, Andrio Elias. Improvement and Reform of an Industrial Washer Crockery. 2014. Completion of course work - in Industrial Maintenance Technology, Federal Technological University of Paraná. Mediatrix 2014.

The work is the improvement and renovation of an industrial dishwasher a washing. The washer consists of two pumps that are responsible for washing and rinsing the dishes, for resistance to heat water for sensors and operation buttons. Besides the reform, the work shows the design and implementation of electrical wiring for each motor, resistors and buttons of both the power, and the part of signals, and also the use of the Arduino system to automate the machine and its program written in C language.

Keywords: Washer, Arduino, C Language.

LISTA DE FIGURAS

Figura 1 - Lavadora louças Industrial	10
Figura 2 - Antigo Quadro Comando (Detalhe dos temporizadores.....	12
Figura 3 - Hardware Arduino	15
Figura 4 - Placa Auxiliar Relés	17
Figura 5 - Esquema Placa de Relés	18
Figura 6 - Software Arduino	19
Figura 7 - Novo Quadro Comando	20
Figura 8 - Ligação Placa Relés	20
Figura 9 - Ligação Comando Elétrico	21

SUMÁRIO

1 INTRODUÇÃO	10
2 JUSTIFICATIVA DO PROBLEMA	12
3 OBJETIVOS	14
3.1 OBJETIVO GERAL	14
3.2 OBJETIVO ESPECÍFICO	14
4 FUNDAMENTAÇÃO TEÓRICA	15
4.1 O ARDUINO UNO	15
4.2 PLACA AUXILIAR DE RELÉS	17
4.3 SOFTWARE DO ARDUINO	19
5 MATERIAIS E MÉTODOS	20
6 RESULTADOS E DISCUSSÕES	21
6.1 RECURSOS DE SOFTWARE/CÓDIGOS UTILIZADOS NO SISTEMA	21
7 CONCLUSÃO	27
REFERÊNCIAS	28
ANEXO A - Datasheet BC 337	29
ANEXO B - Datasheet diodo IN4007	30
ANEXO C - Datasheet relé Metaltex	31

1 INTRODUÇÃO

O trabalho de conclusão de curso a seguir descrito tem como objetivo a aplicação prática de conceitos e fundamentos técnicos estudados durante a jornada acadêmica do curso superior de Tecnologia em Manutenção Industrial, propiciando assim uma formação acadêmica de maneira consistente, pois cria para o **aluno** certas experiências e situações exigidas frente a possíveis problemas técnicos e de resoluções práticas que porventura venham a existir e preparando-o assim para certas atividades e funções que possa vir a enfrentar em indústrias e ou empresas.

Este projeto tem como meta a reforma e automação, utilizando o sistema Arduino, de uma lavadora de louças industrial, como mostra a figura 1. A lavadora de louças industrial tem a finalidade de lavar as louças utilizadas no ambiente hospitalar com rapidez e eficiência. Para que o processo de lavagem seja eficaz, há a necessidade de que o sistema não apresente falhas, tanto no que diz respeito ao controle de temperatura da água, volume da água e controle de produtos químicos como o detergente e o secante.



Figura 1 - Lavadora Louças Industrial

Esse equipamento estava desativado pois o sistema que ela possuía tinha muitas falhas no processo e dava muita manutenção e o fabricante Brasinox da lavadora, modelo BA5 não dispunha de peças de reposição pois não fabricam mais esse modelo.

O projeto foi realizado em parceria com a empresa Hospital e Maternidade Nossa Senhora da Luz.

2 JUSTIFICATIVA DO PROBLEMA

No ambiente hospitalar atual, a automação é cada vez mais aceita, visto que sistemas automatizados bem elaborados trazem vários benefícios.

A utilização correta de processos automatizados traz maior confiabilidade ao equipamento, ocasionando menos paradas inesperadas e um melhor funcionamento do sistema.

A máquina estava com um temporizador que fazia a contagem do tempo de lavagem e outro temporizador que fazia o tempo de enxágüe, como mostra a figura 2, tinha uma resistência de 1000 watts para aquecimento da água com detergente para a lavagem e outra de 2000 watts para aquecimento só de água para enxágüe, também tinha um sensor de nível de água que foi adaptado, tinha também uma motobomba para encher a maquina de água e outra motobomba para enxágüe. Ela não possuía um sistema de proteção contra a falta de água o que ocasionava a queima das resistências e tinha um termostato que fazia o controle da temperatura da água e a carcaça estava toda danificada por vibração e vazamentos. A dosagem de produtos químicos era feita manualmente sem controle preciso ocasionando um gasto excessivo.



Figura 2 - Antigo Quadro Comando (Detalhe dos temporizadores)

Com a reforma e automação, permaneceu as motobombas de água, e será instalado o sistema Arduino. As resistências permaneceram as mesmas e os tempos de lavagem, enxágüe, nível de água e o sistema de proteção contra falta de água é realizado pelo Arduino. O programa de funcionamento do Arduino conta os tempos para lavagem e enxágüe tendo quatro entradas que são: nível de água, emergência, botão de ligar e botão de operar e cinco saídas que são: acionamento da motobomba lavagem, acionamento motobomba enxágüe, válvula e detergente, resistências e led indicativo que a máquina está ligada e mais um led para indicação que a máquina está operando. Com a instalação do Arduino, reduzirá a probabilidade de paradas inesperadas pelo desgaste dos componentes velhos. Eliminará a queima das resistências por falta de água, sendo que a máquina não vai funcionar se ela não estiver no nível máximo. Terá um sistema de proteção na porta que quando o operador abrir a porta o processo todo para simulando uma emergência.

3 OBJETIVOS

3.1 OBJETIVO GERAL

O objetivo geral deste projeto é reformar e automatizar o processo de uma lavadora de louças industrial, através de alterações que serão feitas no sistema de acionamento.

3.2 OBJETIVO ESPECÍFICO

- Reforma para eliminação de ferrugem e sujeira;
- Instalação de um sistema Arduino Uno para o controle do processo;
- Foram mantidas as motobombas e termostato para ajuste da temperatura das resistências;
- Instalado um sensor para nível máximo;
- Chave de emergência na porta aumentando a segurança;
- Instalado duas manoplas para o operador selecionar o processo desejado;
- Lâmpadas de indicação de funcionamento da máquina.

4 FUNDAMENTAÇÃO TEÓRICA

4.1 O ARDUINO UNO

O Arduino Uno, como mostra a figura 3, surgiu em 2005, na Itália, com um professor chamado Massimo Banzi, o qual tinha objetivo de ensinar eletrônica e programação de computadores a seus alunos de design, para que eles usassem em seus projetos de arte, interatividade e robótica. Porém, ensinar eletrônica e programação para pessoas que não são da área não é uma tarefa simples, e outra dificuldade era a inexistência de placas poderosas e baratas no mercado.

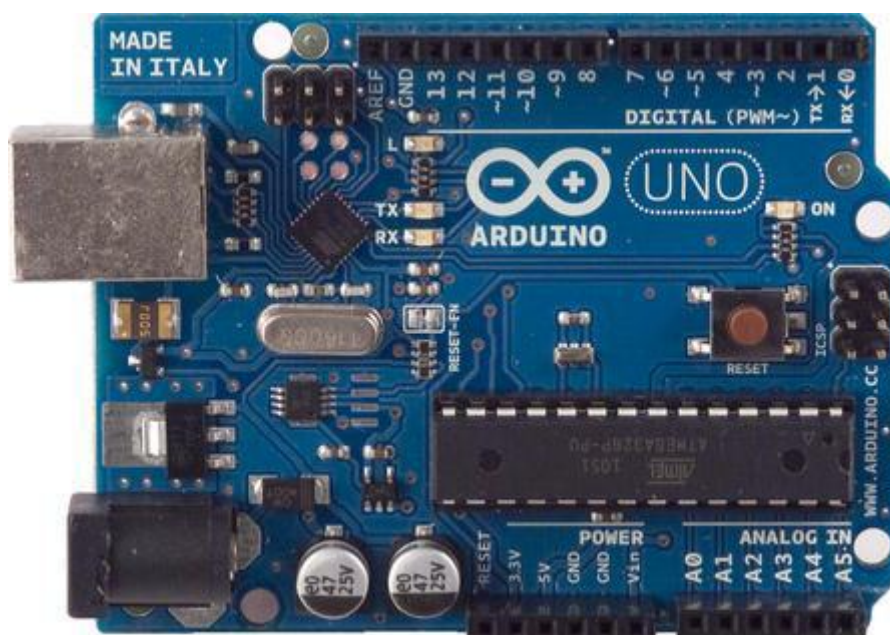


Figura 3 - Hardware Arduino

Foi pensando nisso que Massimo e David Cuartielles decidiram criar sua placa própria, com a ajuda do aluno de Massimo, David Mellis, que ficou responsável por criar a linguagem de programação do Arduino. Várias pessoas conseguiram utilizar o Arduino e fazer coisas incríveis, surgindo assim essa febre mundial da eletrônica (MINAKAWA *et al*, 2012, p. 3).

O Arduino é uma placa de controle de entrada de dados (IN), como sensores, e saída de dados (OUT), como motores e leds, com cristal oscilador de 16 Mhz, um regulador de tensão de 5 V, botão de reset, plugue de alimentação, pinos conectores, e alguns LEDs para facilitar a verificação do funcionamento (MINAKAWA *et al.*, 2012, p. 3).

Este *hardware* se divide em vários modelos. Dentre os principais pode-se citar:

Arduino Uno: É uma placa com micro controlador Atmega328. Possui 14 entradas/saídas digitais, 6 entradas analógicas, um cristal oscilador de 16MHz, conexão USB, uma entrada para fonte, soquetes para ICSP, e um botão de reset. A placa contém todos os recursos necessários para ser utilizado como micro controlador. O Uno seleciona automaticamente a fonte de alimentação (USB ou fonte externa) (ARDUINO, 2013).

O Arduino ainda possibilita a implantação de placas auxiliares, que são placas que visam estender suas funcionalidades. Por exemplo, a placa de reles que oferece a interface necessária para conectar o Arduino ao comando da máquina.

Não há obrigatoriedade em utilizar placas auxiliares no projeto, pois há liberdade para fazer os seus próprios circuitos eletrônicos e se comunicar com qualquer equipamento. Mas, a idéia destas placas é adicionar funcionalidades a seu projeto, sem a necessidade de se investir em projetos complexos, desenvolvimento e fabricação.

Segundo David Mellis da revista Info Online (GREGO, 2013), uma das vantagens do Arduino por ser *Open Source* é a possibilidade de adaptar o modelo de negócios e o fornecimento a diferentes situações. Outras empresas podem vender kits para a montagem de dispositivos compatíveis com Arduino, por exemplo. Também podem redesenhar os produtos para trabalhar com componentes que são mais baratos e fáceis de conseguir em seus países.

Um exemplo de produto derivado das placas Arduino que atende a um uso específico é a ArduPilot, placa de código aberto para navegação autônoma em aeronaves (MELLIS, 2009).

Dentre as principais vantagens de se utilizar o Arduino, pode-se citar:

- Possibilidade de integração com diversas plataformas;
- Baixo custo (algo em torno de R\$ 65,00);
- Drivers e IDE's executam em Windows, Mac e Linux;
- Execução leve e rápida;
- Possibilidade de transferência de código via USB (dispensando portas seriais);

4.2 PLACA AUXILIAR DE RELÉS

Para o desenvolvimento do protótipo houve a necessidade de desenvolver uma placa auxiliar com relés como mostra a figura 4.

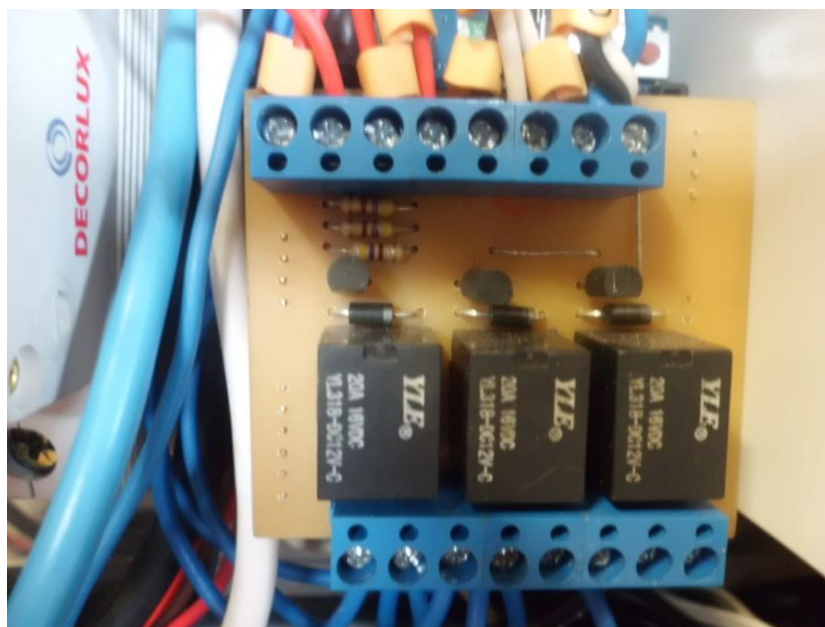


Figura 4 - Placa Auxiliar de Relés

Essa placa de relés foi conectada no Arduino e tem a função de fazer a comutação para o acionamento das contatoras.

Os relés comuns da Metaltex, como mostra o datasheet em anexo, possuem sensibilidades que variam entre 10 e 100mA, com tensão de acionamento de 12volts. Essa corrente, relativamente elevada para sua operação, faz com que eles não possam ser utilizados diretamente no Arduino. Normalmente, o que se faz é utilizar nesses casos uma etapa de amplificação como um transistor (REVISTA SABER ELETRÔNICA, 2011).

O transistor BC337, como mostra datasheet em anexo, tem um ganho de corrente da ordem de 100, onde as saídas do Arduino não são suficientemente potentes para excitar diretamente um relé. No entanto, em muitos projetos experimentais precisa-se usar relés, e quando isso ocorre, ficamos na dependência de um circuito excitador de bom ganho.

O Arduino manda um pulso de 5volts que passa pelo resistor de base do transistor chaveando o transistor para o GND e excitando a bobina do relé fazendo com que abra ou feche os contatos internos do relé como mostra a figura 5. O diodo IN4007, como mostra datasheet em anexo, em paralelo com o relé tem a função de proteção.

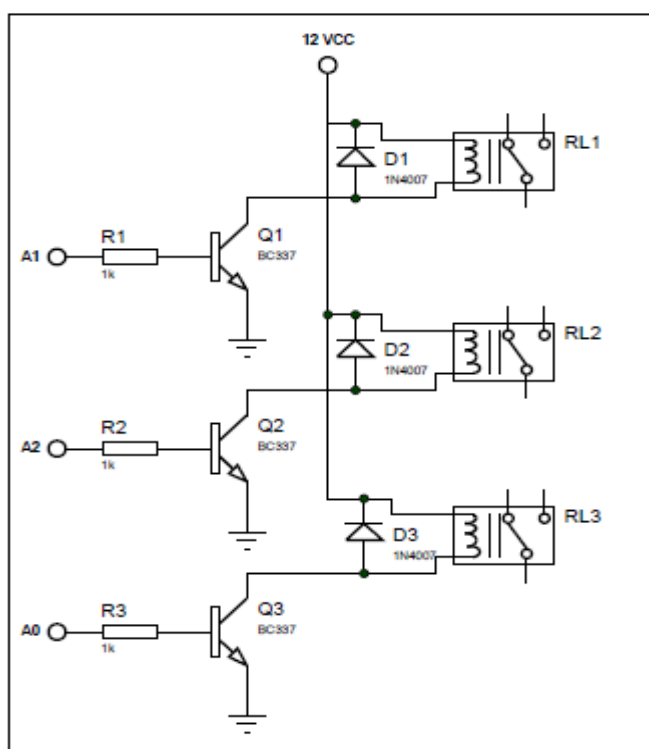


Figura 5 - Esquema Placa de Relés

4.3 SOFTWARE DO ARDUINO

O ambiente de desenvolvimento do Arduino torna fácil escrever o código e enviar para a placa de entrada e saída. Ele funciona em Windows, Mac OS X e Linux. O ambiente de programação é escrito em Java e baseado em *softwares* de código livre (MULTILOGICA, 2013).

As principais bibliotecas são escritos em C e C++. A plataforma Arduino simplifica o processo de desenvolver *software* para micro controladores, alocando os detalhes importantes e complexos de programação desses componentes em um pacote *easy-to-use* (fácil de utilizar), oferecendo grande vantagem a professores, estudantes e amadores interessados em sistemas físicos (BENTES, 2011).

Atualmente, o *software* Arduino se encontra na versão 1.0.5 Beta. A versão utilizada para o exemplo foi a 1.0.4 como mostra a figura 6. O *software* é gratuito e pode ser realizado o download no portal oficial do Arduino (<http://www.arduino.cc>).

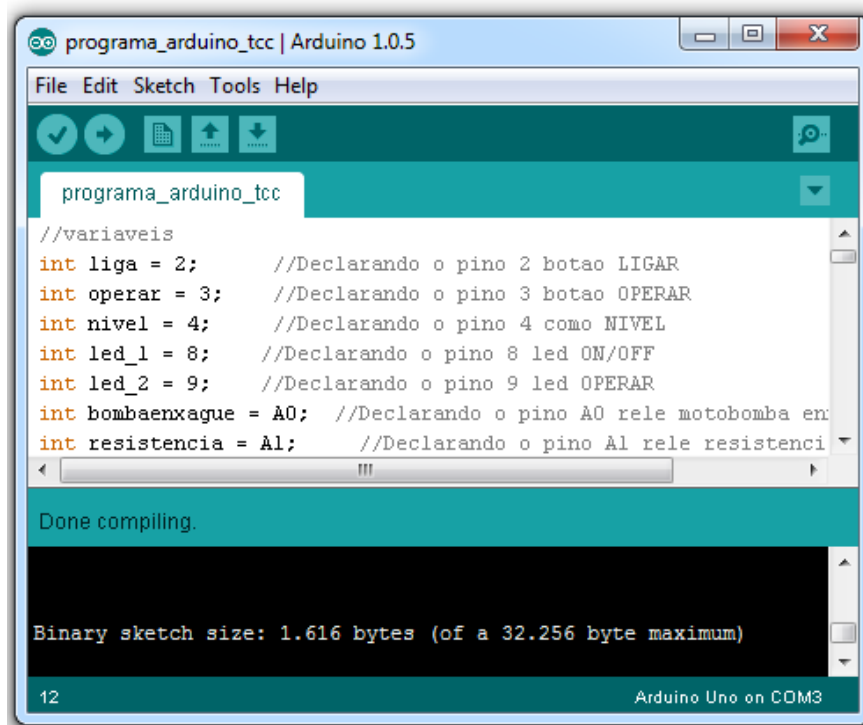


Figura 6 - Software Arduino

5 MATERIAIS E MÉTODOS

Inicialmente foi definido o escopo do sistema: uma aplicação para o hardware Arduino para controle de elementos elétricos. O escopo do trabalho foi apresentar o funcionamento do Arduino, sua linguagem de programação e esquema de ligação do Arduino como mostra a figura 7 e do comando elétrico como mostra a figura 8.

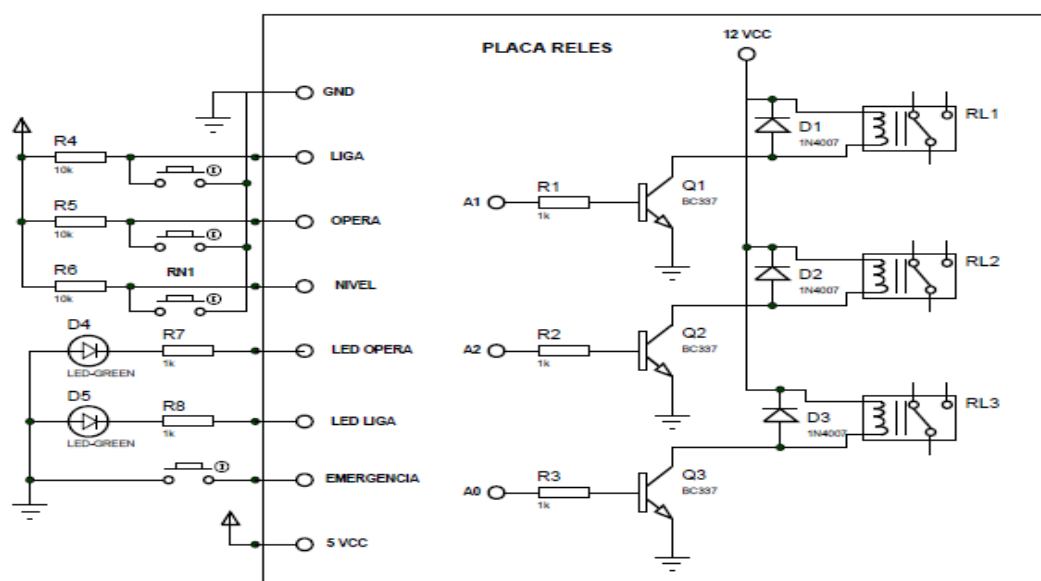


Figura 7 - Ligação Placa Reles

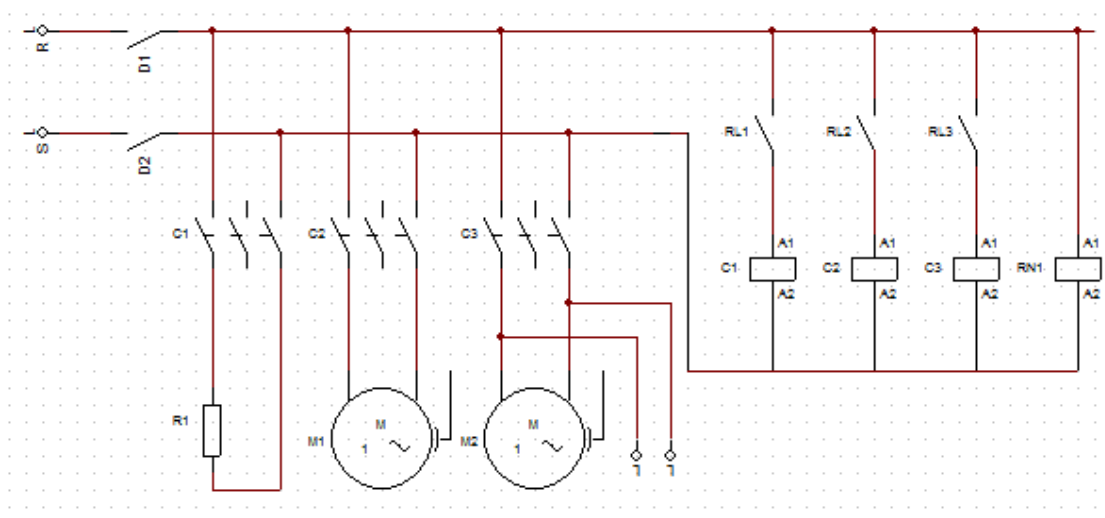


Figura 8 - Ligação Comando Elétrico

6 RESULTADOS E DISCUSSÕES

A seguir serão apresentados os resultados obtidos com o presente trabalho, sendo apresentados o software/códigos que permitiram o desenvolvimento do aplicativo Arduino.

Também, como mostra a figura 9, o novo quadro de comando redimensionado.

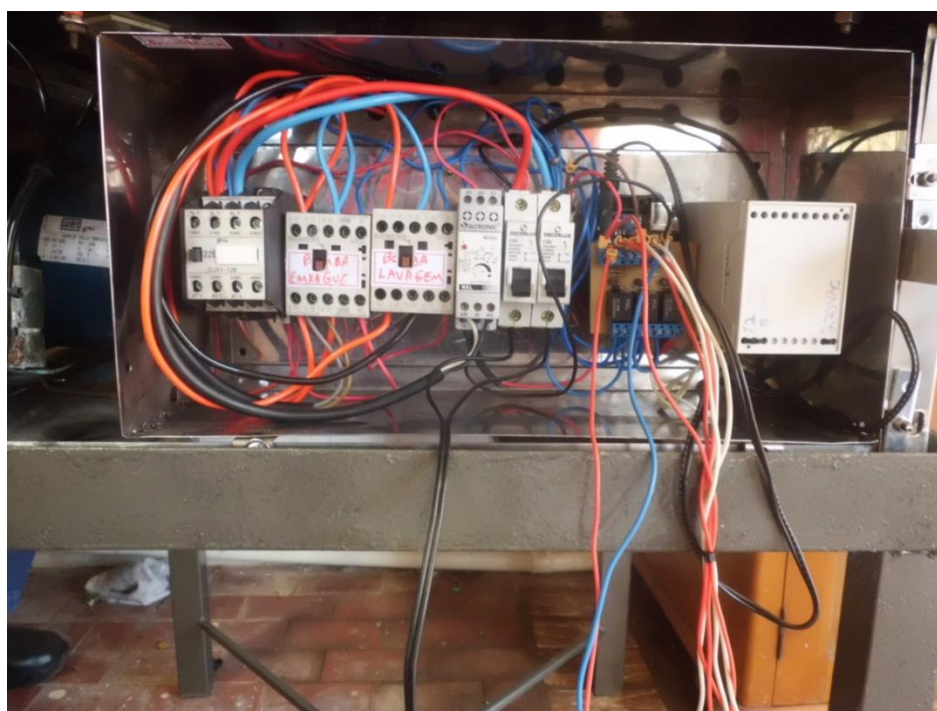


Figura 9 - Novo Quadro Comando

6.1 RECURSOS DE SOFTWARE/CÓDIGOS UTILIZADOS NO SISTEMA

O ambiente open-source Arduino torna fácil escrever código e enviá-lo à placa i / o. Ele roda em Windows, Mac OS X e Linux. O ambiente é escrito em Java e baseado em Processing, avr-gcc e outros softwares de código aberto.

O software/código é mostrado no quadro 01, escrito em linguagem C.

```

int liga = 2;           //Declarando o pino 2 botão LIGAR
int operar = 3;        //Declarando o pino 3 botão OPERAR
int nivel = 4;         //Declarando o pino 4 como NIVEL
int led_1 = 8;         //Declarando o pino 8 led ON/OFF
int led_2 = 9;         //Declarando o pino 9 led OPERAR
int bombaenxague = A0; //Declarando o pino A0 rele motobomba enxágüe
int resistencia = A1;  //Declarando o pino A1 rele resistência
int bombalavagem = A2; //Declarando o pino A2 rele motobomba lavagem
void setup() {
  pinMode(liga, INPUT);    //declara pino como entrada, LIGA
  pinMode(operar, INPUT); //declara pino como entrada, OPERAR
  pinMode(nivel, INPUT);  //declara pino como entrada, NIVEL
  pinMode(led_1, OUTPUT); // Declarando o pino ON/OFF
  pinMode(led_2, OUTPUT); // Declarando o pino OPERAR
  pinMode(bombaenxague, OUTPUT); //Declarando o pino motobomba enxágüe
  pinMode(resistencia, OUTPUT);
  pinMode(bombalavagem, OUTPUT);
}
void loop() {
  int lig = digitalRead(liga);    // le o valor na entrada
  int opera = digitalRead(operar);
  int nive = digitalRead(nivel);
  if (lig == LOW) {               // se pressionar botão LIGAR
    digitalWrite(led_1, HIGH);
    if (nive == LOW){             //quando chega ao nível
      digitalWrite(resistencia, HIGH); //aciona contatora resistência
      digitalWrite(bombaenxague, LOW);
      if (opera == LOW){
        digitalWrite(led_2, LOW);
        digitalWrite(bombalavagem, HIGH);
        delay(60000); //tempo para lavagem 60 segundos
        digitalWrite(bombaenxague, HIGH);
        delay(30000); //tempo para enxágüe 30 segundos
      }
    }
    else {
      digitalWrite(bombaenxague, LOW);
      digitalWrite(led_2, HIGH);
      digitalWrite(bombalavagem, LOW);
      delay(100);
    }
  }
  else {
    digitalWrite(resistencia, LOW);
    digitalWrite(led_1, HIGH); // led ligado
    digitalWrite(bombaenxague, HIGH);
    digitalWrite(bombalavagem, LOW);
    digitalWrite(led_2, HIGH); // led ligado
    delay(500);
    digitalWrite(led_2, LOW); // led desligado
    delay(500);
  }
}
else{
  digitalWrite(led_1, LOW);
  digitalWrite(led_2, LOW);
  digitalWrite(bombaenxague, LOW);
  digitalWrite(resistencia, LOW);
  digitalWrite(bombalavagem, LOW);
}
}
}

```

Quadro 01 - Código Programa

O Arduino começa pela declaração das variáveis utilizadas no decorrer do programa exibida no quadro 02.

```
int liga = 2;    //Declarando o pino 2 botão LIGAR
int operar = 3; //Declarando o pino 3 botão OPERAR
int nivel = 4;  //Declarando o pino 4 como NIVEL
int led_1 = 8;  //Declarando o pino 8 led ON/OFF
int led_2 = 9;  //Declarando o pino 9 led OPERAR
int bombaenxague = A0; //Declarando o pino A0 rele motobomba enxágüe
int resistencia = A1;   //Declarando o pino A1 rele resistência
int bombalavagem = A2; //Declarando o pino A2 rele motobomba lavagem
```

Quadro 02 - Variáveis

Após a configuração das variáveis, o método setup() é responsável por definir os pinos do Arduino. Note no Quadro 03 que os pinos foram definidos como Input e Output sendo 3 entradas e 5 saídas.

```
void setup() {
  pinMode(liga, INPUT);    //declara pino como entrada, LIGA
  pinMode(operar, INPUT); //declara pino como entrada, OPERAR
  pinMode(nivel, INPUT);  //declara pino como entrada, NIVEL
  pinMode(led_1, OUTPUT); // Declarando o pino saída ON/OFF
  pinMode(led_2, OUTPUT); // Declarando o pino saída OPERAR
  pinMode(bombaenxague, OUTPUT); //Declarando o pino saída
                                  //motobomba enxágüe
  pinMode(resistencia, OUTPUT); //Declarando o pino saída resistência
  pinMode(bombalavagem, OUTPUT); //Declarando o pino saída
                                  //motobomba lavagem
}
```

Quadro 03 - Definição do tipo dos pinos no Arduino

Logo em seguida, o método loop() faz laços consecutivos, para que possa avaliar as informações recebidas, alterar o status de suas variáveis e enviar respostas a placa auxiliar de relés. O Quadro 04 apresenta a declaração do método loop() e a verificação do estado das entradas.


```
void loop() {  
  int lig = digitalRead(liga);      // le o valor na entrada do botão ligar  
  int opera = digitalRead(operar); // le o valor na entrada do botão operar  
  int nive = digitalRead(nivel);   // le o valor na entrada do nível
```

Quadro 04 - Método loop()

Após a verificação das entradas, o Arduino captura qual parâmetro foi repassado. O Quadro 05 apresenta a verificação das entradas e acionamento das saídas.

Como podemos ver no Quadro 05, no primeiro *"if"* o programa faz a verificação se o botão "LIGAR" está acionado. Se eles estiver acionado o programa faz a leitura seguindo o *"{"* que aciona o led de indicação de ON/OFF e espera o sinal do segundo *"if"* que faz a verificação do nível e se o botão não estiver acionado o programa pula para o ultimo *"e/se"* que faz os desligamentos dos leds ON/OFF e OPERAR, das duas motobombas e da resistência.

Enquanto a máquina enche de água o programa pula para o penúltimo *"e/se"* onde ele mantém desligado o relé referente as resistências, aciona led ON/OFF, aciona rele referente a motobomba enxágüe, mantém desligado o relé referente a motobomba de lavagem e fica piscando led OPERAR. Ao chegar no nível o programa executa o segundo *"if"* que aciona o relé referente a resistência para aquecimento da água e desliga relé referente a motobomba enxágüe onde para de encher de água a máquina e mantém acionado o led OPERAR sinalizando que a máquina esta pronta para operar.

```

if (lig == LOW) {           // se precionar botão LIGAR
  digitalWrite(led_1, HIGH); //aciona led de indicação de on/off
  if (nive == LOW){        //quando chega ao nível
    digitalWrite(resistencia, HIGH); //aciona contatora resistêcia
    digitalWrite(bombaenxague, LOW); //desliga motobomba enxágüe
    if (opera == LOW){     //se pressionar botão operar
      digitalWrite(led_2, LOW); //desliga led OPERAR
      digitalWrite(bombalavagem, HIGH); //aciona motobomba lavagem
      delay(60000);        //tempo para lavagem 60 segundos
      digitalWrite(bombaenxague, HIGH); //aciona motobomba enxágüe
      delay(30000);        //tempo para enxágüe 30 segundos
    }
  }
  else                      //enquanto não pressionar o botão OPERAR
  {
    digitalWrite(bombaenxague, LOW); //desliga motobomba enxágüe
    digitalWrite(led_2,HIGH);        //desliga led IOPEARAR
    digitalWrite(bombalavagem, LOW); //desliga motobomba lavagem
    delay(100);
  }
}
else {                      //enquanto não estiver no nível
  digitalWrite(resistencia, LOW); //desliga Resistencia
  digitalWrite(led_1,HIGH);        // aciona led on/off
  digitalWrite(bombaenxague, HIGH); //aciona motobomba enxágüe
  digitalWrite(bombalavagem, LOW); //desliga motobomba lavagem
  digitalWrite(led_2,HIGH);        // aciona led OPERAR
  delay(500);                      //tempo para led piscar
  digitalWrite(led_2,LOW);         // desliga led OPERAR
  delay(500);                      //tempo para led piscar
}
}
else{                        //enquanto não LIGAR
  digitalWrite(led_1, LOW);        //desliga led ON/OFF
  digitalWrite(led_2, LOW);        //desliga led OPERAR
  digitalWrite(bombaenxague, LOW); //desliga motobomba enxágüe
  digitalWrite(resistencia, LOW);  //desliga resistêcia
  digitalWrite(bombalavagem, LOW); //desliga motobomba lavagem
}
}
}

```

Quadro 05 - Verificação e acionamento das entradas e saídas do Arduino

Ao pressionar o botão OPERAR o programa executa o terceiro e ultimo "if" que desliga o led OPERAR e aciona o relé referente a motobomba de lavagem e conta um tempo de 60 segundos. Apos esse tempo de 60 segundos aciona a o relé da outra motobomba de enxágüe e conta mais 30 segundos e apos esses 90 segundos desliga tudo e volta a ligar o led de OPERAR sinalizando que a máquina esta pronta para uma nova operação.

Se enquanto a máquina estiver enchendo de água ate chegar ao nível ou estiver operando, se abrir a porta ou desligar o botão LIGAR o Arduino faz o reset de sua placa sinalizando que a máquina esta em emergência, Ao fechar a porta ou ligar ela pela botão LIGAR, o programa volta ao seu estado inicial, ou seja, no primeiro "if" como mostra o Quadro 05.

7 CONCLUSÃO

O trabalho elaborado e concluído da reforma e aprimoramento de uma lavadora de louças industrial foi projetado a partir de uma problema existente no Hospital e Maternidade Nossa Senhora da Luz, onde existia essa máquina a muitos anos abandonada por falta de peças e que com muito esforço foi reformado e instalado um sistema, como o Arduino, para sua automação sendo viabilizado a parte financeira que para automatizar não se gastou mais que R\$200,00 (duzentos reais), bem como da operação prática e simples que pudesse minimizar esta situação, tendo em vista que apenas um funcionário faz a operação da máquina agilizando o processo de lavagem de louças e com isso tendo mais tempo para outros serviços. Visto que em um mês já se paga o gasto do investimento. A instalação do sistema não implica numa imposição, mas sim numa opção a mais, uma vez que permite que o Hospital economize na compra de uma máquina nova que custa e torno de R\$12000,00 (doze mil reais).

O trabalho foi bem sucedido e agregou novos conhecimentos.

REFERÊNCIAS

ARDUINO. Disponível em <<http://arduino.cc/>>. Acesso em 05/12/2013.

BENTES, Leandro Maurício Araújo. Arduino: **hardware e software open-source**. Disponível em <<http://www.hardware.com.br/artigos/arduino/>>. Acesso em 28/12/2013.

DATASHEET BC 337. Disponível em <http://www.datasheetcatalog.com/datasheets_pdf/B/C/3/3/BC337.shtml>. Acesso em 28/01/2014.

DATASHEET DIODO IN4007. Disponível em <http://www.datasheetcatalog.net/pt/datasheets_pdf/1/N/4/0/1N4007.shtml>. Acesso em 28/01/2014.

DATASHEET RELÉ METALTEX. Disponível em <<http://www.metaltex.com.br/downloads/A.pdf>>. Acesso em 28/01/2014.

GREGO, Maurício. **O hardware em 'código aberto'**. Disponível em <<http://info.abril.com.br/professional/tendencias/hardware-livre-leve-e-solto.shtml?2>>. Acesso em 05/12/2013.


MELLIS, David. Disponível em <<http://info.abril.com.br/professional/tendencias/hardware-livre-leve-e-solto.shtml?2>>. Acesso em 05/12/2013.

MINAKAWA, Riccieli; SANTOS, Wellington Oliveira dos; SILVA, Lucas Tsutsui da; TIOSSO, Luiz Eduardo. Introdução ao Arduino. Disponível em 45 <http://destacom.ufms.br/mediawiki/images/9/9f/Arduino_Destacom.pdf>. Acesso em 06/12/2013.

MULTILOGICA. **Download do software do Arduino**. Disponível em <<http://multilogica-shop.com/Download>>. Acesso em 28/12/2013.

REVISTA SABER ELETRÔNICA. **Relé Eletrônico Multi-uso**. Disponível em <<http://www.mecatronicaatual.com.br/educacao/986-rel-eletrnico-multi-uso>>. Acesso em 24/01/2014.

ANEXO A - Datasheet BC 337

FAIRCHILD SEMICONDUCTOR*		BC337/338				
Switching and Amplifier Applications		 <p style="text-align: center;">TO-92 1. Collector 2. Base 3. Emitter</p>				
<ul style="list-style-type: none"> • Suitable for AF-Driver stages and low power output stages • Complement to BC327/BC328 						
NPN Epitaxial Silicon Transistor						
Absolute Maximum Ratings $T_J=25^{\circ}\text{C}$ unless otherwise noted						
Symbol	Parameter	Value	Units			
V_{CE5}	Collector-Emitter Voltage					
	: BC337	50	V			
	: BC338	30	V			
V_{CE0}	Collector-Emitter Voltage					
	: BC337	45	V			
	: BC338	25	V			
V_{EBO}	Emitter-Base Voltage	5	V			
I_C	Collector Current (DC)	800	mA			
P_C	Collector Power Dissipation	625	mW			
T_J	Junction Temperature	150	$^{\circ}\text{C}$			
T_{STG}	Storage Temperature	-55 ~ 150	$^{\circ}\text{C}$			
Electrical Characteristics $T_J=25^{\circ}\text{C}$ unless otherwise noted						
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
BV_{CEO}	Collector-Emitter Breakdown Voltage	$I_C=10\text{mA}, I_B=0$				
	: BC337		45			V
	: BC338		25			V
BV_{CES}	Collector-Emitter Breakdown Voltage	$I_C=0.1\text{mA}, V_{BE}=0$				
	: BC337		50			V
	: BC338		30			V
BV_{EBO}	Emitter-Base Breakdown Voltage	$I_E=0.1\text{mA}, I_C=0$	5			V
I_{CES}	Collector Cut-off Current	$V_{CE}=45\text{V}, I_B=0$		2	100	nA
	: BC337	$V_{CE}=25\text{V}, I_B=0$		2	100	nA
	: BC338					
h_{FE1}	DC Current Gain	$V_{CE}=1\text{V}, I_C=100\text{mA}$	100		630	
h_{FE2}		$V_{CE}=1\text{V}, I_C=300\text{mA}$	60			
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C=500\text{mA}, I_B=50\text{mA}$			0.7	V
$V_{BE(on)}$	Base Emitter On Voltage	$V_{CE}=1\text{V}, I_C=300\text{mA}$			1.2	V
f_T	Current Gain Bandwidth Product	$V_{CE}=5\text{V}, I_C=10\text{mA}, f=50\text{MHz}$		100		MHz
C_{ob}	Output Capacitance	$V_{CE}=10\text{V}, I_E=0, f=1\text{MHz}$		12		pF
h_{FE} Classification						
Classification	15	25	40			
h_{FE1}	100 ~ 250	160 ~ 400	250 ~ 630			
h_{FE2}	60-	100-	170-			

BC337/338

ANEXO B - Datasheet diodo 1N4007

MOTOROLA
 SEMICONDUCTOR TECHNICAL DATA

 Order this document
 by 1N4001/D

Axial Lead Standard Recovery Rectifiers

This data sheet provides information on subminiature size, axial lead mounted rectifiers for general-purpose low-power applications.

Mechanical Characteristics

- Case: Epoxy, Molded
- Weight: 0.4 gram (approximately)
- Finish: All External Surfaces Corrosion Resistant and Terminal Leads are Readily Solderable
- Lead and Mounting Surface Temperature for Soldering Purposes: 220°C Max. for 10 Seconds, 1/16" from case
- Shipped in plastic bags, 1000 per bag.
- Available Tape and Reeled, 5000 per reel, by adding a "RL" suffix to the part number
- Polarity: Cathode Indicated by Polarity Band
- Marking: 1N4001, 1N4002, 1N4003, 1N4004, 1N4005, 1N4006, 1N4007

**1N4001
thru
1N4007**

 1N4004 and 1N4007 are
 Motorola Preferred Devices

**LEAD MOUNTED
RECTIFIERS
50-1000 VOLTS
DIFFUSED JUNCTION**

 CASE 68-03
 DO-41

MAXIMUM RATINGS

Rating	Symbol	1N4001	1N4002	1N4003	1N4004	1N4006	1N4008	1N4007	Unit
*Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage	V_{RRM} V_{RWM} V_R	50	100	200	400	600	800	1000	Volts
*Non-Repetitive Peak Reverse Voltage (halfwave, single phase, 60 Hz)	V_{RSM}	60	120	240	480	720	1000	1200	Volts
*RMS Reverse Voltage	$V_{R(RMS)}$	35	70	140	280	420	560	700	Volts
*Average Rectified Forward Current (single phase, resistive load, 60 Hz, see Figure 8, $T_A = 75^\circ\text{C}$)	I_O	1.0							Amp
*Non-Repetitive Peak Surge Current (surge applied at rated load conditions, see Figure 2)	I_{FSM}	30 (for 1 cycle)							Amp
Operating and Storage Junction Temperature Range	T_J T_{stg}	-65 to +175							$^\circ\text{C}$

ELECTRICAL CHARACTERISTICS*

Rating	Symbol	Typ	Max	Unit
Maximum Instantaneous Forward Voltage Drop ($I_F = 1.0$ Amp, $T_J = 25^\circ\text{C}$) Figure 1	V_F	0.93	1.1	Volts
Maximum Full-Cycle Average Forward Voltage Drop ($I_O = 1.0$ Amp, $T_L = 75^\circ\text{C}$, 1 inch leads)	$V_F(AV)$	—	0.8	Volts
Maximum Reverse Current (rated dc voltage) ($T_J = 25^\circ\text{C}$) ($T_J = 100^\circ\text{C}$)	I_R	0.05 1.0	10 50	μA
Maximum Full-Cycle Average Reverse Current ($I_O = 1.0$ Amp, $T_L = 75^\circ\text{C}$, 1 inch leads)	$I_{R(AV)}$	—	30	μA

*Indicates JEDEC Registered Data

Preferred devices are Motorola recommended choices for future use and best overall value.

Rev 5

© Motorola, Inc. 1996



ANEXO C - Datasheet relé Metaltex

METALTEX

Relé miniatura / Miniature relay

A

- 1 contato reversível para 12 A
- Montagem direta em circuito impresso
- Baixo custo
- Produção automatizada
- Disponível na versão selada

- 12 A contact
- Direct PC mounting
- Low cost
- Automated assembly
- Sealed version available

Chave de código / How to order

A

1

C2

↳ Tensão nominal da bobina / Nominal voltage

C0V - 5 VCC / VDC	C2 - 12 VCC / VDC
C1 - 6 VCC / VDC	C3 - 24 VCC / VDC
C0V - 9 VCC / VDC	C4 - 48 VCC / VDC

↳ Modelo / Type

A - Não selado - padrão / Non sealed - Standard
AL - Selado - baixo consumo / Sealed - Low consumption

Especificações de bobina / Coil specifications

Modelo / Type	Tensão Nominal / Nominal Voltage VCC / VDC	Mín. Tensão Contínua / Min. Allowable Voltage VCC / VDC	Tensão de Operação / Pick-up Voltage VCC / VDC	Tensão de Desoperação / Drop-out Voltage VCC / VDC	Corrente Nominal / Nominal current mA	Resistência (±10%) / Resistance (±10%) Ω*
A	C0V	5	≤ 4,0	≥ 0,5	96	52
	C1	6	≤ 4,0	≥ 0,6	80	75
	C0V	9	≤ 7,2	≥ 0,9	67	135
	C2	12	≤ 9,6	≥ 1,2	40	300
	C3	24	≤ 19,2	≥ 2,4	20	1200
C4	48	≤ 38,4	≥ 4,8	10	4800	
AL	C2	12	≤ 9,6	≥ 1,2	30	400

* ±10% sobre 100Ω de 1200Ω

Especificações de contato / Contact specifications

Capacidade do contato / Rated current (Carga relativa / Relative load)	110 VCA / VAC 12 A 250 VCA / VAC 10 A 24 VCC / VDC 12 A
Corrente de comutação máx. / Maximum switching current	12 A
Tensão de comutação máx. / Maximum switching voltage	250 VCA / 110 VCC / 250 VAC / 110 VDC
Corrente de condução máx. / Maximum allowable current	12 A
Resistência de contato inicial máx. / Maximum initial contact resistance	50 mΩ
Vida mecânica / Mechanical life	10 ⁷ operações máx. / operations min. (300 operações/minuto) (operations/minute)
Vida elétrica / Electrical life	10 ⁷ operações máx. / operations min. (20 operações/minuto) (operations/minute)
Tempo de operação / Operate time	10 ms máx.
Tempo de desoperação / Release time	10 ms máx.
Material dos contactos / Contact material	AgNi 90/10

Características gerais / Characteristics

Rigidez dielétrica entre bobina e contato / Breakdown voltage between contact and coil	2000 VCA / VAC (1ml/mín) (1ml/min)
Rigidez dielétrica entre contatos abertos / Breakdown voltage between open contacts	1000 VCA / VAC (1ml/mín) (1ml/min)
Resistência de isolamento / Insulation resistance	100 MΩ/mín. (500 VCC/VDC)
Temperatura de operação / Operating ambient temperature	-30 a 80°C
Resistência à vibração / Vibration resistance	10 a 55 Hz dupla amplitude 1,5 mm 10 to 55 Hz d.a. 1,5 mm
Resistência a impacto / Shock resistance	10 G

Dimensões e diagramas / Dimensions and layouts

Tolerâncias não indicadas / Not indicated tolerance: ±0,1 mm

Diagrama PCI / Printed circuit layout

Vista de baixo / Bottom view

Diagrama Elétrico / Schematic

Vista de baixo / Bottom view

Todas as dimensões em milímetros / All dimensions in millimeters