



Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

SISTEMA DE MONITORAMENTO DE NÍVEL DE  
ÁGUA EM RESERVATÓRIOS E DE  
TEMPERATURA DE BOMBAS D'ÁGUA

Wagner Irlanda de Souza Campos

Campina Grande - PB  
Fevereiro de 2009

Wagner Irlanda de Souza Campos

Trabalho de Conclusão de Curso

SISTEMA DE MONITORAMENTO DE NÍVEL  
DE ÁGUA EM RESERVATÓRIOS E DE  
TEMPERATURA DE BOMBAS D'ÁGUA (SMNT).

Relatório apresentado à Coordenação de  
Graduação em Engenharia Elétrica da UFCG,  
como parte dos requisitos para obtenção do título  
de Engenheiro Eletricista.

Aprovado em \_\_\_\_\_ de \_\_\_\_\_ de 2009

Banca Examinadora

---

Orientador: Prof. Edmar Candeia Gurjão

---

Professor convidado: Eurico Bezerra de Souza Filho

## **DEDICATÓRIA**

Dedico este trabalho aos meus colegas de profissão e a todos que me apoiaram e me orientaram na realização do meu estágio e na realização deste relatório.

## **AGRADECIMENTOS**

Agradeço em primeiro lugar a Deus que é a grande referência de minha vida, aos meus pais Jose de Sousa Campos e Ilda Maria de Souza Campos, a minha esposa Joana Darc Santa Baracho e minha Irma Leise Flaviana de Souza Campos por terem me dado o apoio que sempre precisei, aconselhando, incentivando e acreditando no meu potencial.

# SUMARIO

## 1- INTRODUÇÃO

## 2 - Desenvolvimento do sistema.

### 2.1 - Microcontrolador (PIC 16F877)

- 2.1.1 - Introdução;
- 2.1.2 - Arquitetura do PIC;
- 2.1.3 - Circuito;
- 2.1.4 - Programa;

### 2.2 Sinalização

#### 2.2.1 - Led Indicador de Power-on / Led indicador bomba acionada;

- 2.2.1.1 - Introdução;
- 2.2.1.2 - Circuito;

#### 2.2.2 - Buzzer (sinalizado sonoro)

- 2.2.2.1 - Introdução;
- 2.2.2.2 - Circuito;

#### 2.2.3 - Display de LCD

- 2.2.3.1 - Introdução;
- 2.2.3.2 - Circuito;

### 2.3 Sensores

#### 2.3.1 Sensor - Temperatura;

- 2.3.1.1 - Introdução;
- 2.3.1.2 - Algumas aplicações para o LM35;
- 2.3.1.3 - Diagrama de Conexões da LM35;
- 2.3.1.4 - Circuito;

## 2.3.2 Sensor – Nível

2.3.2.1 - Introdução;

2.3.2.2 - Circuito (Montagem Elétrica);

2.3.2.3 - Montagem Mecânica;

## 2.4 - Atuador (relé)

2.4.1 - Introdução;

2.4.2 - Circuito;

## 3 - Conclusões.

## 4 - Referência Bibliografia;

## 5 - Anexo: Programa do PIC 16F877;

# 1. INTRODUÇÃO

Um sistema de abastecimento de água de um prédio pode ser feito por um reservatório que armazena a água e que é alimentado por um motor. Esse tipo de sistema precisa de controles, dentre os quais pode-se citar o nível da água e a temperatura do motor. Baseando-se em um problema prático, o projeto foi desenvolvido com a finalidade de atender às exigências feitas por um cliente de uma empresa, que passou por problemas no sistema de abastecimento de um hotel e na manutenção do mesmo.

O sistema desenvolvido é composto por sensores e atuadores. O sistema vai contar com dois sensores, um de temperatura e um de nível. O sensor de nível tem com objetivo informar ao cliente a situação em que se encontram os reservatórios (Caixa D'água e Cisterna). Como as bombas D'água são equipamentos caros, o sensor de temperatura irá informar constantemente as temperaturas, com o objetivo de evitar o super-aquecimento.

Para atuar na bomba em casos exceções, foi instalado um relé. O sistema também irá contar com display de LCD, a fim de visualizar as informações.

O sistema para o qual o software foi desenvolvido é composto por: reservatórios, motor e sensores como já foi dito. Então software é capaz de:

- Gerenciar o sistema de abastecimento de água;
- Monitorar o comportamento do sistema com base nas entradas de sensores;
- Atuar no sistema em casos de exceções;
- Prover a segurança e bom funcionamento ao sistema;
- Auxiliar na identificação dos defeitos.

O cenário atual do sistema apresenta os seguintes pontos:

- Problemas gerados por causa do racionamento de água pela concessionária;
- Consumidores finais sem abastecimento em horários de maior demanda;
- Dificuldades em determinar os horários em que o reservatório e a cisterna deverão ser abastecidos;
- Dificuldades de identificação dos pontos de falhas no sistema.

Dentre os critérios de aceitação, podemos destacar comparativos entre a eficiência do sistema inicial e o final. Como, por exemplo, valores de tempo de detecção de falhas antes e depois da implantação do sistema. E um comparativo entre os níveis de água nos reservatórios ao longo do dia, permitindo observar a sua regularidade.

Nos capítulos a seguir será descrito o sistema desenvolvido e suas aplicações.

## 2. DESENVOLVIMENTO DO SISTEMA

As características técnicas do sistema são:

2.1 – “**Cérebro**”: Microcontrolador PIC 16F877 Microchip;

2.2 - **Sinalização**: Led Indicador de *Power-on*, Led indicador bomba acionada, *buzzer* (sinal sonoro) e *Display* LCD;

2.3 - **Sensores**: Temperatura e Nível

2.4 - **Atuadores**: Relé;

2.5 - **Alimentação**: Fonte regulada de 5Vdc com circuito integrado dedicado;



## 2.2 – “CÉREBRO” - MICROCONTROLADOR (PIC 16F877)

Os microcontroladores são chips inteligentes, que tem um processador, pinos de entradas/saídas e memória. Através da programação dos microcontroladores podemos controlar suas saídas, tendo como referencia as entradas ou um programa interno.

O que diferencia os diversos tipos de microcontroladores, são as quantidades de memória interna (programa e dados), velocidade de processamento, quantidade de pinos de entrada/saída (I/O), alimentação, periféricos, arquitetura e set de instruções.

### 2.1.2 - Arquitetura do PIC

A memória é a parte do microcontrolador cuja função é guardar dados de maneira mais fácil para eventuais operações aritmética e essas operações matemáticas são executados pela unidade central de processamento. Para saída e entrada dos dados o microcontrolador dispõem de porta (conjunto de pino do PIC) que funciona como um local de memória e quando qualquer coisa que se queira ler ou escrever é possível identificar facilmente através dos seus pinos.

Para que possamos usar alguns dados em um determinado tempo, o PIC possui um bloco de temporização, e ele é responsável por informações acerca da hora, duração, protocolo, etc. A unidade básica do temporizador é um contador que é na realidade é um registrador cujo conteúdo aumenta de uma unidade num intervalo de tempo fixo.

Como os sinais analogicos são substancialmente diferentes daqueles que o microcontrolador pode entender (zero e um), eles devem ser convertidos

num formato que possa ser compreendido pelo microcontrolador. Esta tarefa é executada por intermédio de um bloco destinado à conversão analógica-digital ou com um conversor A/D. Este bloco vai ser responsável pela conversão de uma informação de valor analógico para um número binário e pelo seu trajeto através do bloco do CPU, de modo a que este o possa processar de imediato

**2.1.3 - Circuito** - Como foi utilizado em nosso projeto o PIC 16F877, estão algumas de suas principais especificações:

- Arquitetura Harvard;
- Código de instruções reduzido (35 instruções);
- Proteção contra cópia de código;
- Memória FLASH de programa com 8 kbytes;
- 256 bytes de EEPROM;
- 368 bytes de RAM;
- Conversor Analógico/digital (A/D) em oito canais de 10 bits;
- Pinagem compatível com outros modelos de Pic (40 pinos);
- 33 pinos de I/O (dreno de 25mA);
- USART (para comunicação RS-232);
- Comunicação SPI ou  $I^2C$ ;
- USART com detecção de nove bits de endereço.
- Dois timers de 8 bits e um de 16 bits;
- Dois canais CCP ( Capture, Compare and PWM);
- Watch Dog Timer interno;
- Alta frequência de operação (até 20MHz);
- Controles “Power-on Reset” e “Power-up Timer”
- Ampla faixa de tensão alimentação: 2V a 5,5V.
- Encapsulamento DIP com 40 pinos;
- Entre outras.

Na figura 9 e 10 apresentamos o circuito elétrico do SMNT. Note que o microcontrolador precisa de poucos componentes externos para o seu funcionamento, pois ele possui praticamente todos os periféricos em seu interior. O circuito fica, portanto, “enxuto”.

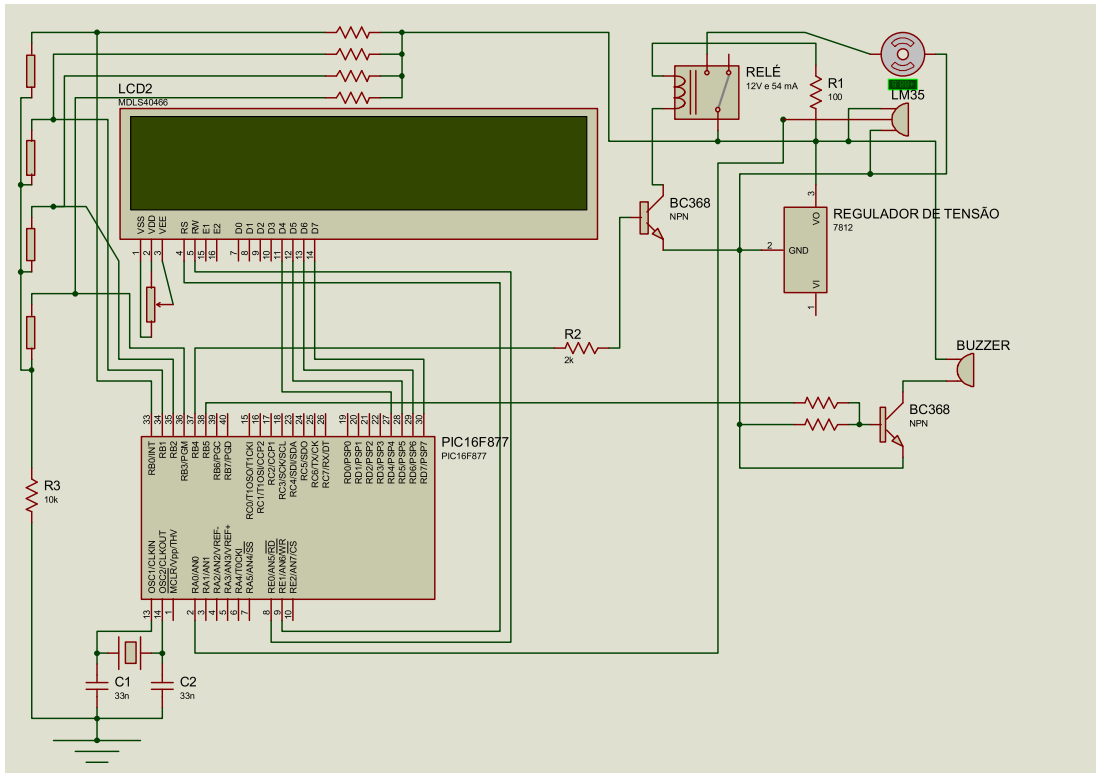


Figura 9 – Diagrama elétrico

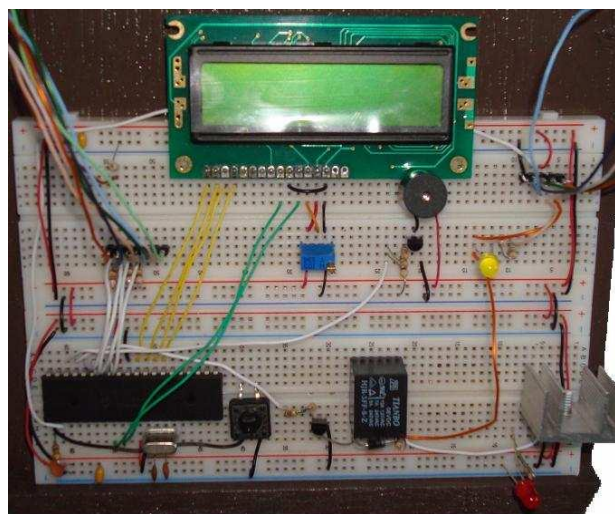


Figura 10 – Montagem do circuito

Para a geração de *clock* do PIC foi utilizado um cristal de 4 MHz em conjunto com dois capacitores de 33 nF como mostra a figura 11 e o oscilador interno cuida do resto

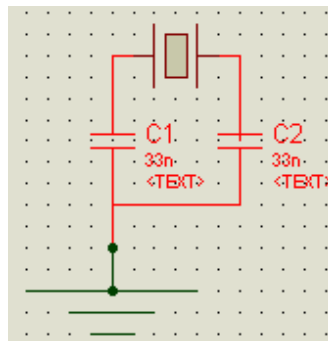


Figura 11 – Circuito oscilado do PIC

O *Reset*, leva o microcontrolador a reiniciar seus registradores para valores iniciais pré-definidos na fabricação. Um dos mais importantes efeitos de um reset é o que zera o contador de programa (PC), fazendo com que o programa comece a ser executado a partir da primeira instrução do software programado. A condição de reset normal do PIC ocorre quando colocamos o pino MCLR em nível baixo. Normalmente, isto é feito através de uma chave push-botton, como na figura - 12.

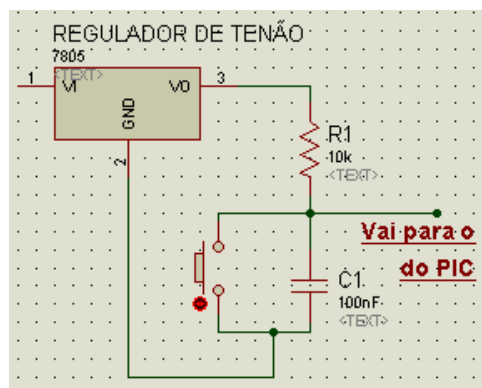


Figura - 12

**2.1.4 - Programa** - O funcionamento do circuito depende do programa aplicado no PIC, e este é conectado a todos os “circuitos auxiliares” (Sensor de nível, sensor de temperatura, oscilador, atuado e sinalizador). O programa foi desenvolvido na linguagem C e segue em anexo. O programa do PIC foi elaborado de modo a informar ao LCD os níveis da caixa d’água através dos valores lidos nos pinos B0, B1, B2 e B3 do microcontrolador. Sendo que o pino B0 representa a caixa vazia e o pino B3 a caixa cheia. A bomba só será desligada quando a caixa estiver cheia ou a temperatura do motor subir além do normal. As chaves magnéticas estão igualmente espaçadas e na medida em que vão sendo acionadas, o PIC captura esta informação, analisa e informa no LCD o valor correto do nível do reservatório. De acordo com a programação, a bomba d’água é acionada ou não por meio de um relé. Este relé é controlado pelo PIC através do pino B4. Quando a temperatura da bomba subir acima de normal, o *buzzer* (Sinal Sonoro) é acionado e a bomba d’água é desligada como já foi dito. Do mesmo modo acontece com o sensor de temperatura, pois através da programação o valor analógico é lido pelo pino A0 do PIC, convertido de analógico para um número binário e o resultado desta operação é informado no cristal de LCD.

## **2.2 - SINALIZAÇÃO**

### **2.2.1 - Led Indicador de Power-on / Led indicador bomba acionada**

**2.2.1.1 - Introdução** - O LED é um componente eletrônico semicondutor, ou seja, um diodo emissor de luz ( L.E.D = Light emitter diode ), mesma tecnologia utilizada nos chips dos computadores, que tem a propriedade de transformar energia elétrica em luz. Tal transformação é diferente da encontrada nas lâmpadas convencionais que utilizam filamentos metálicos, radiação ultravioleta e descarga de gases, dentre outras. Nos LEDs, a transformação de energia elétrica em luz é feita na matéria, sendo, por isso, chamada de Estado sólido.

O LED é um componente do tipo bipolar, ou seja, tem um terminal chamado anodo e outro, chamado catodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz.

**2.2.1.2 - Circuito** - Como a corrente que passa pelo LED é em torno de 15mA, então para o cálculo do resistor em série com o LED, temos:

$$R = \frac{V_{cc} - V_{LED}}{15 \times 10^{-3}} = \frac{V_{cc} - 1,4}{15 \times 10^{-3}}$$

## **2.2.2 – Buzzer (sinalizado sonoro)**

**2.2.2.1 - Introdução** - A função do *buzzer* alarma quando a temperatura da bomba elevar acima do normal e assim alertado que houve falha no funcionamento bomba d'água. Mas para acionar o *buzzer* de 5V e 50mA foi usado um transistor, pois o PIC tem um dreno de corrente de apenas 25mA em seus pinos de I/O.

**2.2.2.2 - Circuito** - Para o projeto de circuito de acionamento, alguns parâmetros dos transistores são necessários, como  $h_{FEsat}$ ,  $V_{CEsat}$  e  $V_{BEsat}$ , no entanto nem todos os manuais fornecem esses parâmetros e quando o fazem, eles valem para condições predeterminadas de  $I_c$ ,  $I_b$  e  $V_{CE}$ .

Assim, caso esses valores não sejam fornecidos pelos manuais ou as condições de operação não sejam compatíveis com as previstas nos projetos, podemos usar alguns critérios técnicos que garantam a saturação do transistor, conforme a tabela -3.

**Tabela - 3**

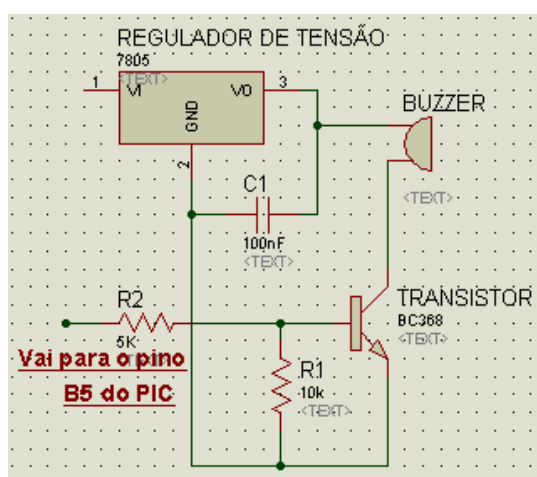
Especificações	Transistor Simples	Transistor Darlington
$h_{FEsat}$	$h_{FEmin} / 2$	$h_{FEmin} / 2$
$V_{BEsat}$	0,8V	1,4V
$V_{CEsat}$	0,2V	1,0V
$I_{Csat} = I_L$	Corrente de operação da carga	

### Resistor de Base – Rb

Qualquer que seja o tipo de circuito de acionamento, o resistor de polarização da base Rb deve garantir a saturação do transistor quando a tensão de entrada “Vi” estiver em nível alto.

Assim, pela malha de entrada do circuito dos da figura - 13 temos:

$$V_i = R_b \times I_{Bsat} + V_{BEsat} \Rightarrow R_B = \frac{V_i - V_{BEsat}}{I_{Bsat}} \text{ em que } I_{Bsat} = \frac{I_{Csat}}{h_{FEsat}}$$



**Figura – 13**

### 2.2.3 – DISPLAY DE LCD

**2.2.3.1 - Introdução** - A idéia é realizar a conexão de um LCD paralelo com microcontrolador PIC, utilizando apenas 4 vias de dados. Devido à sua simplicidade, estes LCDs não possuem recursos como gerenciamento automático de pixels, não são coloridos (*full color*), não possuem iluminação ativa entre outras limitações, mas ainda são largamente utilizados na indústria. Basta ver que muitas registradoras, equipamentos *hand-held*, sistemas de informação de computadores servidores entre outros, ainda utilizam largamente este dispositivo.

Existem, comercialmente, diversos tipos e tamanhos de LCD alfanuméricos que podem ser incorporados a um microcontrolador PIC. Os mais comuns são os gerenciados por um processador Hitachi HD44780.

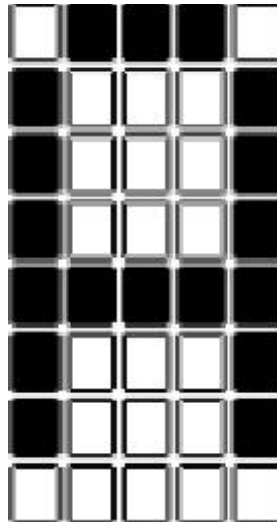


Figura - 14

Alguns módulos possuem iluminação em "*back-light*" e estão disponíveis nas cores (monocromáticas) azul, âmbar, verde entre outras. Mas, no fundo, todos eles são compatíveis e podem ser controlados pelo PIC exatamente da mesma maneira. Isto é importante para que, no futuro, uma rotina desenvolvida para um LCD 16X02 possa ser utilizada para controlar um LCD 20X04 sem a necessidade de alteração de código.



Basicamente, cada "célula" (caractere) do LCD é composto de 8 pixels na horizontal e de 5 pixels na vertical, ou seja, cada "célula" é representada da seguinte forma:



**Figura - 15**

Estes caracteres são conhecidos como 5X7, uma vez que a linha inferior é normalmente reservada para o cursor.

**2.2.3.2 - Circuito** - Basicamente os LCDs alfa numéricos seguem um padrão de especificação de interface, onde estão presentes 14 pinos de acesso (para os LCDs sem iluminação em "*back-light*") ou 16 pinos (para os que possuem iluminação em "*back-light*"). Na verdade, estes pinos são "*pads*" soldáveis para a inserção de conectores IDC e são classificados em 8 pinos para controle de dados, três linhas de controle e três linhas de alimentação. Iremos detalhar cada um a seguir. A função de cada pino é resumida de acordo com a tabela 1.

Tabela - 1

Pino	Nome	Função
1	Vss	Terra
2	Vdd	Positivo (normalmente 5V)
3	Vo	Contraste do LCD. Às vezes também é chamado de Vee
4	RS	Register Select
5	R/W	Read/Write
6	E	Enable
7	D0	Bit 0 do dado a ser escrito no LCD (ou lido dele).
8	D1	Bit 1 do dado a ser escrito no LCD (ou lido dele).
9	D2	Bit 2 do dado a ser escrito no LCD (ou lido dele).
10	D3	Bit 3 do dado a ser escrito no LCD (ou lido dele).
11	D4	Bit 4 do dado a ser escrito no LCD (ou lido dele).
12	D5	Bit 5 do dado a ser escrito no LCD (ou lido dele).
13	D6	Bit 6 do dado a ser escrito no LCD (ou lido dele).
14	D7	Bit 7 do dado a ser escrito no LCD (ou lido dele).
15	A	Anodo do back-light (se existir back-light).
16	K	Catodo do back-light (se existir back-light).

Apesar dos LCDs serem projetados para trabalharem com 5V, consumindo apenas alguns miliampéres, tensões entre 4,5V e 6V funcionam perfeitamente. Alguns módulos também funcionam com 3V. Por isso, o mais indicado é sempre ler o *data sheet* do LCD que você possui para saber melhor seus recursos e limitações.

Como já dito antes, o pino 1 (Vss) é o terra do módulo LCD e pode ser ligado juntamente com o pino Vss do PIC.

O Pino 2 (Vdd) é o positivo e deve ser ligado juntamente com o Vdd do PIC .

O pino 3 (Vo) é onde controlamos o contraste do LCD, ou seja, onde controlamos se o que irá aparecer no LCD será "escuro" ou "claro". Para isso devemos ligá-lo a um potenciômetro ou trim-pot de  $10K\Omega$  conectado ao Vss e Vdd (atuando como um divisor de tensão) de acordo com o esquemático a seguir:

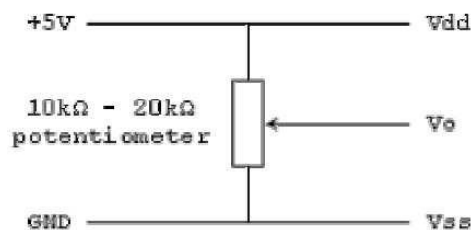


Figura - 16

O pino 4 (RS) é o Seletor de Registros. Resumindo, quando este pino está em nível lógico baixo (0), os dados enviados para o LCD são tratados como comandos e os dados lidos do LCD indicam o seu estado atual (status). Quando este pino está em nível lógico alto (1), os dados são tratados como caracteres, tanto para leitura como para escrita.

Para indicar se os dados serão lidos ou escritos no LCD, existe o pino 5 (R/W) que controla exatamente se a operação em andamento será de leitura (quando estiverem em nível lógico alto - 1) ou gravação (quando estiver em nível lógico baixo - 0).

O pino 6 (E) é a linha de habilitação para os comandos do LCD. É utilizado para iniciar a transferência de comandos ou caracteres entre o módulo e as linhas de dados. Quando estiver escrevendo para o display, os dados serão transmitidos apenas a partir de uma transição de high para low (H -> L) deste

sinal. No entanto, para ler informações do display, as informações estarão disponíveis imediatamente após uma transição L -> H e permanecerá lá até que o sinal volte para o nível lógico baixo (0) novamente.

Os pinos 7 a 14 fazem parte dos barramentos de dados. Ele trabalha com os oito sinais em paralelo ou ainda pode trabalhar com um barramento de 4 vias (normalmente D4 a D7), mas os dados devem ser transmitidos em dois pacotes. Cada pacote de quatro bits é conhecido como "*nibble*". Este é um excelente recurso para minimizar o uso de pinos de I/O do microcontrolador, mas ocupa um pouco mais de memória. A decisão de utilizar 8 vias ou 4 vias é exclusiva do projetista.

Neste modo, apenas as vias de dados D4 a D7 são utilizadas, e as vias D0 a D3 devem ser deixadas flutuando (sem conexão alguma) ou conectadas ao positivo da alimentação, via resistor limitador que tem seu valor entre 4K7 e 47K. Não é aconselhável aterrá-los, a não ser que o pino R/W também esteja aterrado, prevenindo que os pinos sejam configurados como saída.

## **2.3 - Sensores**

### **2.3.1 Sensor – Temperatura**

**2.3.1.1 - Introdução** - Existem no mercado diversos tipos de sensores de temperatura, que vão desde os NTC's, PTC's e diodos até os mais variados tipos de termopares, dentre outros. Porém, estima-se que talvez nenhum dos citados anteriormente seja de tão simples manuseio e exija tão poucos aparatos eletrônicos para que funcione quanto o modelo LM35, pois o circuito usual é bastante simples, necessitando apenas do sensor propriamente dito.

O sensor LM35 é um sensor de precisão, fabricado pela National Semiconductor ([www.national.com](http://www.national.com)), que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontrar no momento em que for alimentado por uma tensão de 4 -20Vdc e GND, tendo em sua saída um sinal de 10mV para cada Grau Celsius de temperatura, sendo assim, apresenta uma boa vantagem com relação aos demais sensores de temperatura calibrados em "KELVIN", não necessitando nenhuma subtração de variáveis para que se obtenha uma escala de temperatura em Graus Celsius.

Este sensor poderá ser alimentado com alimentação simples ou simétrica, dependendo do que se desejar como sinal de saída, mas independentemente disso, a saída continuará sendo de 10mV/°C. Ele drena apenas 60µA para estas alimentações, sendo assim seu auto-aquecimento é de aproximadamente 0.1°C ao ar livre.

O sensor LM35 é apresentado com vários tipos de encapsulamentos, sendo o mais comum o TO-92, que mais se parece com um transistor, e oferece ótima relação custo benefício, por ser o mais barato dos modelos e propiciar a mesma precisão dos demais. A grande diversidade de encapsulamentos se dá devido à alta gama de aplicações deste integrado.

O sensor LM35 pode ser facilmente utilizado, da mesma maneira que qualquer outro sensor de temperatura, colando-o sobre a superfície que se deseja medir a temperatura e sua temperatura estará em torno de 0.01°C abaixo da temperatura da superfície que se encontra colado, pressupondo que a temperatura da superfície seja a mesma que a temperatura do ar que se encontra ao redor desde ambiente.

Esta regra se aplica especialmente para o encapsulamento do tipo TO-92 de encapsulamento plástico, onde as ligações de cobre são o trajeto térmico principal para carregar o calor através do dispositivo, fazendo com que a temperatura fique mais próxima da temperatura do ar do que da superfície em que se encontra colado. Para amenizar este problema, tenha certeza de que a fiação que ligará o LM35 esteja presa juntamente a superfície de interesse, para que ambas as partes estejam praticamente sempre na mesma temperatura. A maneira mais fácil de fazer isto é fixar os fios e o próprio LM35 com um leve revestimento de cola epóxi à superfície de interesse, assim, o LM35 e seus condutores não estarão em contato com o ar, logo, a temperatura do ar não afetará na medição do integrado.

#### **2.3.1.2 - Algumas aplicações para o LM35:**

- Termômetros para câmeras frias, chocadeiras etc;
- Controles de temperatura de máquinas;
- Aquisição de dados para pesquisas;
- Proteção para dispositivos industriais (motores, inversores, fontes);

#### **2.3.1.3 - Diagrama de Conexões (LM35) como mostra a figura - 17.**

- Modelo TO-46;
- Modelo TO-92;
- Modelo SO-8;
- Modelo TO-220;

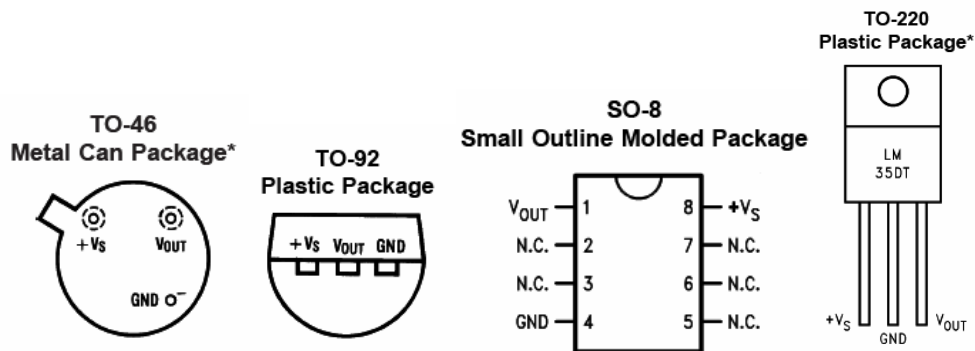


Figura – 17 (<http://www.national.com>)

Ascensão de Temperatura devido ao Auto-aquecimento (Resistência Térmica) é mostrado na tabela - 2.

Tabela - 2

	TO-92 sem dissipador	TO-92 c/ pequeno dissipador
Ar livre	180 °C/W	140 °C/W
Ar em movimento	90 °C/W	70 °C/W
Óleo parado	90°C/W	70 °C/W
Óleo em movimento	45 °C/W	40 °C/W

**2.3.1.4 - Circuito** - Na figura - 18 temos o circuito elétrico do sensor de temperatura. A sua alimentação é estabilizada através de regulador de tensão e o LM35 fica conectado na carcaça do morto, ou melhor, bomba da d'água.





Na verdade o sensor de nível é um sensor magnético. Pois os componentes usados são: Chaves magnéticas e imã. Uma chave magnética (também conhecida como “*reed switch*”) nada mais é do que uma chave digital, só que acionada por um imã. Tratam-se de dois contatos de metal que no estado normal mantêm-se em aberto. Porém, na presença de um campo magnético, os contatos se fecham, podendo conduzir a corrente elétrica. Os contatos estão dentro de uma cápsula de vidro, mantendo-se isolados da corrosão atmosférica.

A chave magnética é um componente relativamente frágil, portanto são necessários alguns cuidados durante a sua utilização. Caso seja necessário dobrar seus terminais, para montá-la numa placa de circuito impresso, faça-o a certa distância da sua cápsula. Se todo o terminal do componente for dobrado, ele poderá ser danificado.

**2.3.2.2 - Circuito (Montagem Elétrica)** – Na figura - 6 temos o circuito elétrico do “sensor de nível”. Ele é composto por quatro chaves magnéticas e vão sendo atuada na medida em que o imã se aproxima. Enquanto a chave magnética estiver aberta, haverá uma tensão de 5 V na saída. Se aproximarmos um imã da chave, ela será fechada, e então a tensão de saída do circuito cairá para 0 V. Porém, assim como as chaves comuns, podem ocorrer algum mal-contato durante sua utilização. Para solucionar esse problema, podemos também utilizar um capacitor para estabilizar a tensão.

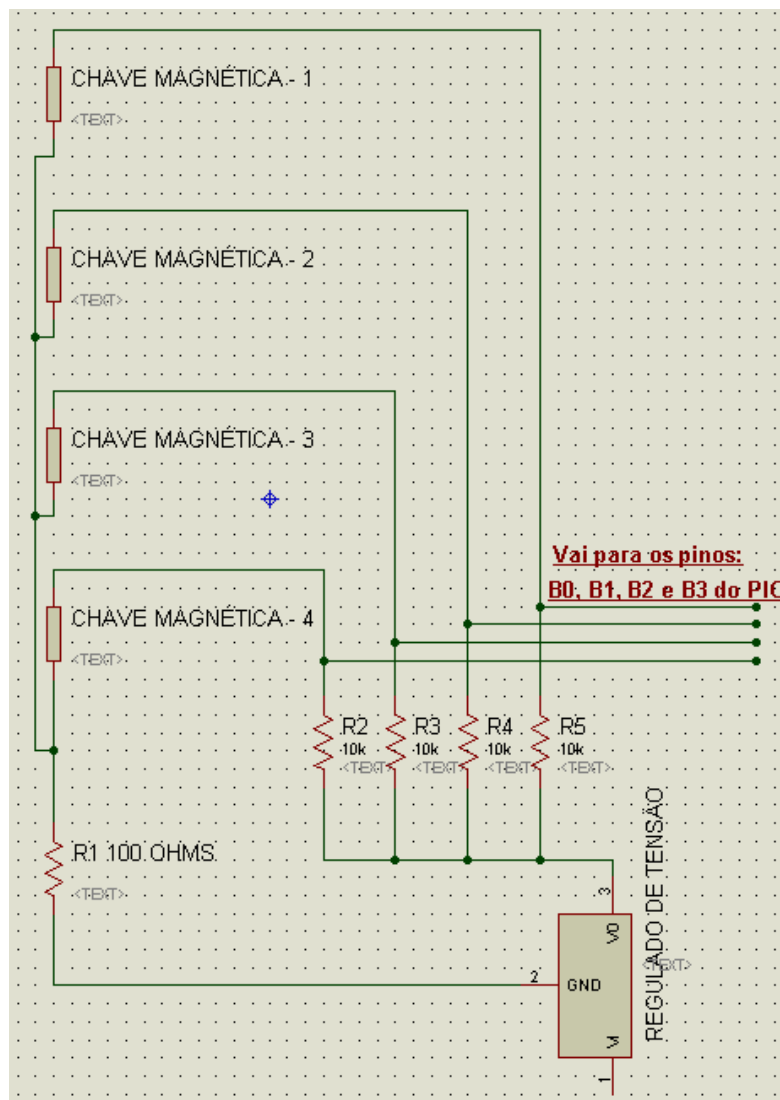


Figura - 19

Os resistores R2, R3, R4 e R5 que mantém nível lógico alto ( cinco volts) nos pinos de entrada B0, B1, B2 e B3 do PIC. Quando uma das chaves é atuada com a proximidade do imã, o nível informado no pino de entrada será baixo (GND) como já foi mencionado.

**2.3.2.3 - Montagem Mecânica** – O elemento do sensor foi construído com o uso de canos e outras peças de PVC, utilizadas no encanamento d'água. Além desta peças, utilizamos também um pedaço de isopor e um ímã retirado de um alto-falante fora de uso. Ao observar a figura -7 abaixo percebe-se um pequeno “T” montado na base do sensor serve para impedir que o ímã escape, quando o nível da água se aproximar do nível mais baixo da caixa d'água.

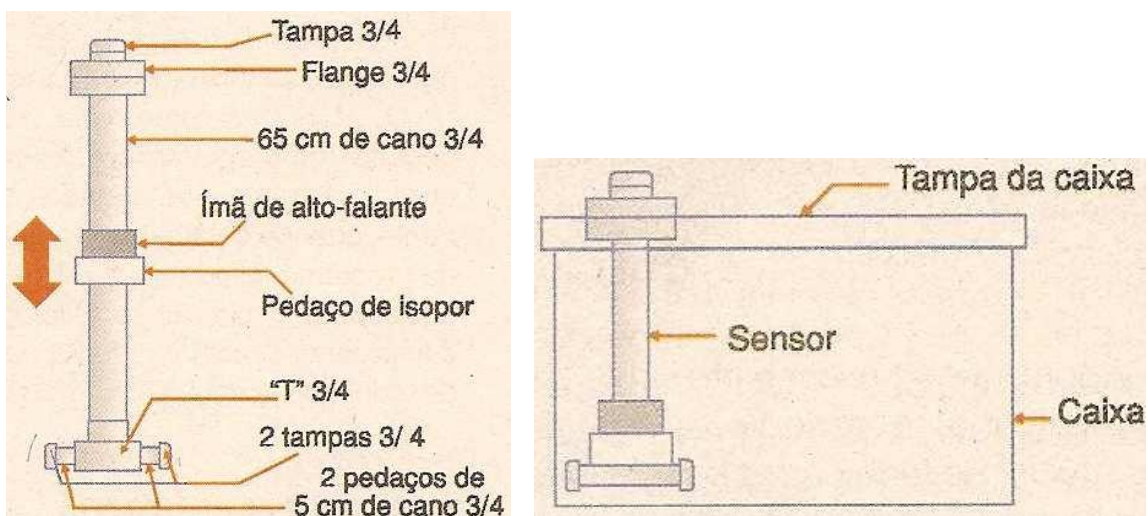


Figura – 20

Para construir o “conjunto bóia” (ímã + isopor), deverá fazer o furo no isopor, com o mesmo diâmetro interno do diâmetro do ímã. Para colar o ímã no isopor foi utilizada a cola do tipo epóxi e foi observada a flutuação do conjunto bóia. Caso o mesmo afunde, é sinal que o pedaço de isopor está muito pequeno ou ainda não tenha a densidade adequada. Quanto maior a densidade (dureza) do mesmo, melhor. A primeira chave magnética deverá ser montada na extremidade inferior do suporte (madeira ou fio rígido) e a ultima na parte superior. As demais devem ser distribuídas ao longo do suporte de maneira uniforme (distancia iguais). As chaves magnéticas foram fixada com fita.

## 2.4 - ATUADOR (RELÉ)

**2.4.1 - Introdução** - A função principal do relé é de ligar e desligar o a bomba d'água quando a caixa d'água estiver vazia e cheia respectivamente e quando a temperatura da bomba for elevada, assim evitando o risco de ser danificada a sua estrutura. Mas para acionar o rele de 6V e 60mA foi usado um transistor, pois o PIC tem um dreno de corrente de apenas 25mA em seus pinos de I/O.

**2.4.2 - Circuito** - Como já foi mencionado, para o projeto de circuito de acionamento, alguns parâmetros dos transistores são necessários, como  $h_{FEsat}$ ,  $V_{CEsat}$  e  $V_{BEsat}$ , no entanto nem todos os manuais fornecem esses parâmetros e quando o fazem, eles valem para condições predeterminadas de  $I_c$ ,  $I_b$  e  $V_{CE}$ .

- **Resistor de Base – Rb**

Qualquer que seja o tipo de circuito de acionamento, o resistor de polarização da base Rb deve garantir a saturação do transistor quando a tensão de entrada Vi estiver em nível alto.

Assim, pela malha de entrada do circuito dos da figura 4, temos:

$$V_i = R_b \times I_{Bsat} + V_{BEsat} \Rightarrow R_B = \frac{V_i - V_{BEsat}}{I_{Bsat}} \text{ em que } I_{Bsat} = \frac{I_{Csat}}{h_{FEsat}}$$

- **Resistor do coletor – Rc**

O resistor de coletor Rc foi obtido pela malha de saída do circuito da figura 4.

$$R_c = \frac{V_{cc} - V_L - V_{CEsat}}{I_{Csat}}$$

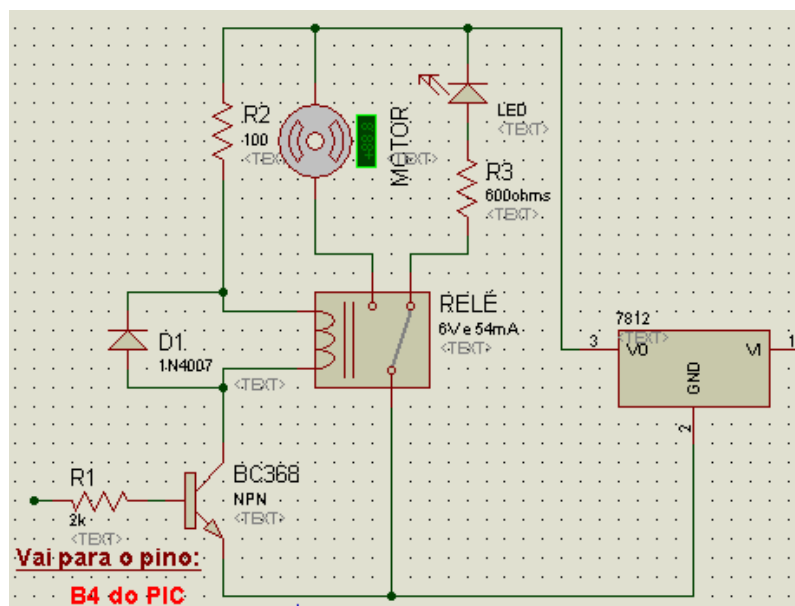


Figura - 21

## 2.5 – ALIMENTAÇÃO: FONTE REGULADA.

Introdução - O LM78XX é um regulador de tensão linear fornecido por vários fabricantes como a Fairchild ou ST Microelectronics. Ele pode vir em vários encapsulamentos. Para corrente de saída até 1A existem dois encapsulamentos: TO-220 (vertical) e D-PAK (horizontal).

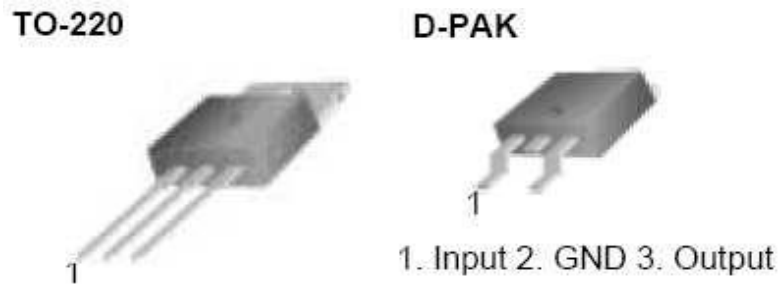


Figura – 22 ([www.fairchildsemi.com](http://www.fairchildsemi.com))

Circuito - Com um dissipador de calor apropriado, os LM78xx podem fornecer até mais que 1A de corrente. Ele também possui proteção contra sobre-temperatura e curto-circuito.

Logo abaixo está a tabela - 3 com os vários tipos de LM78XX, qual a sua tensão de saída e faixa de entrada. Para alimentação do circuito foi usado o LM7805, pois a faixa de alimentação do PIC é 2,0-5,5V e a do display de LCD 4,0-5,0V.

Tabela - 3

Componente	Saída (V)	Faixa Entrada (V)
LM7805	5	7–25
LM7806	6	8–25
LM7808	8	10.5–25
LM7809	9	11.5–25
LM7810	10	12.5–25
LM7812	12	14.5–30
LM7815	15	17.5–30
LM7818	18	21–33
LM7824	24	27–38

A conexão típica é muito simples:

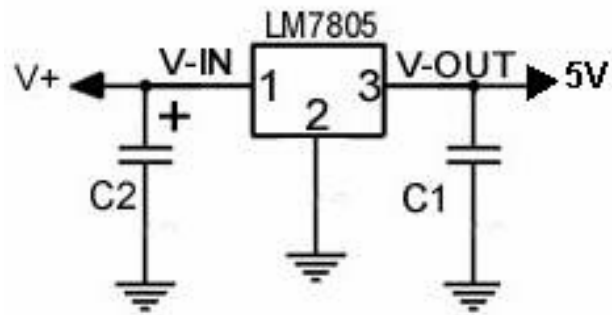


Figura - 23

## **CONCLUSÃO**

A realização do TCC possibilitou o contato direto com os componentes eletrônicos que existe no comércio e possibilitando um maior domínio na sua utilização. Além de se familiarizar com os dispositivos eletrônicos, também proporcionou uma maior revisão conceitos adquiridos ao longo do curso de engenharia elétrica.



## Referência Bibliografia:

THOMAZINI, Daniel. ; ALBUQUERQUE, Pedro. *Sensores Industriais. São Paulo. 3. Ed, Editora Érica, 2007.*

NILSSON, James. *Circuitos Elétricos. 5. Ed, Editora Livros Técnicos e Científicos, 1996.*

PEREIRA, Fabio. *Microcontroladores pic: Programação em C. 4. Ed. Editora Érica, 2005.*

FITTIPALDI, Hélio. *Monitoramento do nível de água. Revista Mecatrônica fácil. São Paulo. Editora Saber LTDA. Vol. 20, 2004.*

FITTIPALDI, Hélio. *Display LCD. Revista Mecatrônica fácil. São Paulo. Editora Saber LTDA. Vol. 12, 2003.*

Transmissor de dados por radio freqüência; Site URL:  
[http://www.wenshing.com.tw/English/prouducts\\_info.asp?bookbm=167](http://www.wenshing.com.tw/English/prouducts_info.asp?bookbm=167). Acessado: 17 de dezembro de 2008.

Transmissor de dados por radio freqüência; Site URL:  
[http://www.wenshing.com.tw/English/prouducts\\_info.asp?bookbm=457](http://www.wenshing.com.tw/English/prouducts_info.asp?bookbm=457). Acessado: 17 de dezembro de 2008.

Microcontrolador 16f877; Site URL:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41291E.pdf>. Acessado: 17 de dezembro de 2008.

Sensor de Temperatura LM35; Site URL:  
<http://cache.national.com/ds/LM/LM35.pdf>. Acessado: 17 de dezembro de 2008.

## Anexo: Programa do PIC 16F877

### Configuração do PIC

```
#include <16F877.h>
//Biblioteca do pic 16f877

#device adc=10
//Configura o compilador para a conversão A/D de dez bits.

#use delay(clock=4000000)
// Clck externo de 4Mhz.

//NOWDT=>Watch dog timer desabilitado.
//XT=>Oscilador cristal<=4mhz.
//NOPUT=>Power up timer desabilitado.
//NOPROTECT=>Sem proteção para leitura da eprom.
//BROWNOUT=>Resetar quando detectar brownout.
//NOLVP=>Programa baixa voltagem desabilitado.
#fuses NOWDT,XT, NOPUT, NOPROTECT, BROWNOUT, NOLVP, NOCPD,
NOWRT, NODEBUG

#include <LCD_16x2_lib.c>
//Biblioteca de manipulação do LCD

#int_TIMER0
//Biblioteca de manipulação do timer zero

long int temp;
//Define um número com 16 bits faixa de 0 a 32767

int valor;
//Define um número com 1 bit faixa de 0 a 1

TIMER0_isr()
{
temp=0;
}
//Função timer zero

void Sistema_M (int var);
//Função de monitoramento de nível de água e temperatura
```

## Função Principal

```
void main(){
//Programa Principal.

    setup_adc_ports(RA0_analog);
//Configura a entrada analogica, que neste caso é RA0.

    setup_adc(ADC_CLOCK_INTERNAL);
//configura o conversor AD interno.

    setup_psp(PSP_DISABLED);
//Porta paralela escrava desativada.

    setup_spi(FALSE);
//Desabilitado o módulo SSP ou MSSP para a comunicação SPI.

    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_128);
//timer 0 abilitado com prescaler dividido por 128.

    setup_timer_1(T1_DISABLED);
//timer 1 desabilitado.

    setup_timer_2(T2_DISABLED,0,1);
// timer 2 desabilitado.

    set_adc_channel(0);
//Seleciona um canal de entrada para módulo AD interno.

    enable_interrupts(INT_TIMER0);
//Estouro de contagem do timer zero.

    enable_interrupts(GLOBAL);
//Habilitação global de interrupções.

    ini_lcd_16x2();
//Função que habilita o LCD.
```

```

while(1) {

    if (input(PIN_B0)==0 && input(PIN_B1)==1 && input(PIN_B2)==1) {
        Sistema_M(0); //A caixa vazia, bomba D'água acionada!

    }
    if (input(PIN_B0)==0 && input(PIN_B1)==0 && input(PIN_B2)==1){
        Sistema_M(1); //A caixa 25% do volume, bomba D'água acionada!

    }
    if (input(PIN_B0)==1 && input(PIN_B1)==0 && input(PIN_B2)==1){
        Sistema_M(2); //A caixa com 50% do volume, bomba D'água acionada!

    }
    if (input(PIN_B0)==1 && input(PIN_B1)==0 && input(PIN_B2)==0){
        Sistema_M(3); //A caixa 75% do volume, bomba D'água acionada!

    }
    if (input(PIN_B0)==1 && input(PIN_B1)==1 && input(PIN_B2)==0){
        Sistema_M(4); //A caixa com quase 100% do volume, bomba D'água
acionada!

    }
    if (input(PIN_B3)==0){
        Sistema_M(5); //A caixa cheia, bomba D'água desligada!

    }

    delay_ms(1000);

}

```

```

void Sistema_M (int var) {

    switch(var) {

        case 0:
            output_bit(pin_B4,1);
            //Aciona a bomba D'água e caixa vazia.

            temp = read_adc();
            //efetua a conversão AD.

            temp=((temp+1)*500)/1024;
            //Valor do escalonamento.
            // \f =>apaga o display, \n =>próxima linha e lu=>decimal
            printf(exibe_lcd,"\f Caixa Vazia\n Temp_M=%lu C ",temp);
            if(temp>35){
                //Para temperaturas acima da temperatura do
                funcionamento normal do morto (Bomba D'água).
                while(1){
                    output_bit(pin_B4,0);
                    output_high(PIN_B5);
                    delay_ms(1);
                    output_low(PIN_B5);
                    delay_ms(1);
                }
            }
            break;

        case 1:
            output_bit(pin_B4,1);
            //Aciona a bomba D'água e volume 25%
            temp = read_adc();
            temp=((temp+1)*500)/1024;
            printf(exibe_lcd,"\f Volume 25%%\n Temp_M=%lu C
            ",temp);
            if(temp>35){
                while(1){
                    output_bit(pin_B4,0);
                    output_high(PIN_B5);
                    delay_ms(1);
                    output_low(PIN_B5);
                    delay_ms(1);
                }
            }
            break;
    }
}

```

```

case 2:
    output_bit(pin_B4,1);
    //Aciona a bomba D'água e volume 50%
    temp = read_adc();
    temp=((temp+1)*500)/1024;
    printf(exibe_lcd,"\f Volume 50%%\n Temp_M=%lu C
",temp);

    if(temp>35){
        while(1){
            output_bit(pin_B4,0);
            output_high(PIN_B5);
            delay_ms(1);
            output_low(PIN_B5);
            delay_ms(1);
        }
    }
    break;

case 3:
    output_bit(pin_B4,1);
    //Aciona a bomba D'água e volume 75%
    temp = read_adc();
    temp=((temp+1)*500)/1024;
    printf(exibe_lcd,"\f Volume 75%%\n Temp_M=%lu C
",temp);

    if(temp>35){
        while(1){
            output_bit(pin_B4,0);
            output_high(PIN_B5);
            delay_ms(1);
            output_low(PIN_B5);
            delay_ms(1);
        }
    }
    break;

```

```

case 4:
    output_bit(pin_B4,1);
    //Aciona a bomba D'água e volume quase 100%
    temp = read_adc();
    temp=((temp+1)*500)/1024;
    printf(exibe_lcd,"\f Volume 100%%\n Temp_M=%lu C
",temp);

    if(temp>35){
        while(1){
            output_bit(pin_B4,0);
            output_high(PIN_B5);
            delay_ms(1);
            output_low(PIN_B5);
            delay_ms(1);
        }
    }
    break;

case 5:
    output_bit(pin_B4,0);
    //A caixa cheia, bomba bomba desligada!
    temp = read_adc();
    temp=((temp+1)*500)/1024;
    printf(exibe_lcd,"\f Volume 100%%\n Temp_M=%lu C
",temp);

    if(temp>35){
        while(1){
            output_bit(pin_B4,0);
            output_high(PIN_B5);
            delay_ms(1);
            output_low(PIN_B5);
            delay_ms(1);
        }
    }
    break;
}
}
}

```