

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
CENTRO DE CIÊNCIAS EXATAS  
ENGENHARIA DE COMPUTAÇÃO**

**DOCUMENTAÇÃO DO PROJETO  
“CONTROLE DE LOTAÇÃO”**

**CURITIBA  
2011**

**RODRIGO YOSHIDA TAKEDA  
TIAGO DAMBRÓS  
VINÍCIUS AUGUSTO ZANETTI**

**DOCUMENTAÇÃO DO PROJETO  
“CONTROLE DE LOTAÇÃO”**

DOCUMENTAÇÃO DO PROJETO DA  
DISCIPLINA DE MICROPROCESSADORES  
DO 6º DO CURSO DE ENGENHARIA DE  
COMPUTAÇÃO DA UNIVERSIDADE  
CATÓLICA DO PARANÁ

PROFESSOR: PROF. AFONSO MIGUEL

**CURITIBA  
2010**

## RESUMO

O controle de lotação é um dispositivo eletrônico capaz de fazer uma contagem de quantas pessoas entram em um ambiente. Além disso, ele faz um registro com uma foto na hora da entrada da pessoa. Essa foto é gravada em um cartão SD.

Essa documentação do projeto de microprocessadores vai apresentar as características e ferramentas que foram utilizadas durante o projeto controle de lotação. Isso inclui os softwares utilizados e suas respectivas bibliotecas, o processador utilizado e suas respectivas características e os circuitos eletrônicos e seus esquemáticos.

## SUMÁRIO

<b>INDICE DE FIGURAS .....</b>	<b>4</b>
<b>INTRODUÇÃO .....</b>	<b>5</b>
<b>OBJETIVOS .....</b>	<b>6</b>
<b>1 PROJETO .....</b>	<b>7</b>
<b>1.1 DESCRIÇÃO .....</b>	<b>7</b>
<b>1.2 FUNÇÕES .....</b>	<b>7</b>
<b>1.3 APLICAÇÃO .....</b>	<b>8</b>
<b>2 MODULOS UTILIZADOS .....</b>	<b>9</b>
<b>2.1 CAMERA SERIAL .....</b>	<b>9</b>
<b>2.1.1 Descrição.....</b>	<b>9</b>
<b>2.1.2 Pinagem.....</b>	<b>10</b>
<b>2.1.3 Comandos .....</b>	<b>11</b>
<b>2.2 CARTÃO SD.....</b>	<b>12</b>
<b>2.2.1 Descrição.....</b>	<b>12</b>
<b>2.2.2 Pinagem.....</b>	<b>14</b>
<b>2.3 SENSOR DE PASSAGEM .....</b>	<b>16</b>
<b>2.3.1 Descrição.....</b>	<b>16</b>
<b>2.3.2 Esquematico do sensor de passagem.....</b>	<b>17</b>
<b>2.4 PAINEL LCD .....</b>	<b>18</b>
<b>2.4.1 Descrição.....</b>	<b>18</b>
<b>2.4.2 Pinagem.....</b>	<b>19</b>
<b>3 MICROPROCESSADOR .....</b>	<b>20</b>
<b>3.1 CARACTERÍSTICAS NECESSÁRIAS .....</b>	<b>20</b>
<b>3.2 DEFINIÇÃO .....</b>	<b>21</b>
<b>3.3 PINAGEM .....</b>	<b>22</b>
<b>4 ESQUEMA GERAL .....</b>	<b>24</b>
<b>5 SOFTWARE.....</b>	<b>25</b>
<b>6 PROBLEMAS APRESENTADOS .....</b>	<b>26</b>
<b>CONCLUSÃO .....</b>	<b>28</b>
<b>REFERENCIAS.....</b>	<b>29</b>

## INDICE DE FIGURAS

Figura 1: Camera Serial .....	10
Figura 2: Cartão SD .....	12
Figura 3: Esquemático da ligação do cartão .....	13
Figura 4: Pinos do cartão SD .....	15
Figura 5: Esquemático do sensor de passagem .....	17
Figura 6: Display LCD .....	19
Figura 7: Pinagem do PIC18F4550 .....	22
Figura 8: Esquemático geral do projeto.....	24
Figura 9: Placa do projeto .....	24

## INTRODUÇÃO

O contador de lotação é um dispositivo eletrônico capaz de fazer uma contagem de quantas pessoas estão dentro de um ambiente. Ele também é capaz de saber se a pessoa entrou ou saiu do local, desde que seja por uma mesma saída.

Após um encontro com um professor, foi decidido que na hora da entrada seria tirada uma foto da pessoa com gravação do arquivo de imagem em um cartão SD.

Existem vários simuladores de presença no mercado, mas nenhum deles funciona exatamente como um contador. Normalmente, quando existe movimento por perto, ele é acionado e é ligado um rele ou interruptor que liga uma luz.

Outro equipamento eletrônico similar é uma portaria eletrônica; Ela transmite um vídeo exibindo a pessoa do outro lado do local.

O contador de lotação pode ser usado em casas de shows, casas noturnas. Com ele, é possível saber se uma casa está lotada, ou ainda se houve entrada de pessoas não autorizadas.

## OBJETIVOS

-Demonstrar o conhecimento adquirido durante a disciplina de microprocessadores do curso de Engenharia de computação da Pontifícia Universidade Católica do Paraná.

-Imprimir em um painel LCD a contagem de quantas pessoas estão dentro de um ambiente.

-Utilizar emissores e receptores de luz para fazer a contagem.

-Utilizar a porta serial para fazer comunicação com o computador

-Utilizar o serial para fazer a comunicação com o cartão sd

-Utilizar o serial para fazer a comunicação com a câmera serial.

## **1 PROJETO**

### **1.1 DESCRIÇÃO**

O projeto é composto por quatro módulos principais. O primeiro possui um par de sensores e foto receptores.

Os pares de receptores funcionam em conjunto. Quando uma pessoa ou um objeto passa na frente dos emissores de luz, a luz é retida e é enviado um sinal baixo para o microcontrolador. Dependendo da ordem desses sinais, é possível saber se a pessoa está saindo ou entrando no ambiente. A contagem dessas pessoas é enviada a um painel LCD.

A segunda parte é a comunicação com o modulo sd. O modulo sd funciona em um formato de arquivos padrão (FAT16) e serão inseridas as fotos dentro dele.

A terceira parte é câmera serial. Ela é capaz de enviar os dados de uma foto em serial. Isso facilita a gravação no cartão sd.

### **1.2 FUNÇÕES**

A função principal do projeto Controle de lotação é fazer uma contagem de quantas pessoas existem dentro de algum ambiente.

Existem algumas limitações que todas as pessoas devem entrar e sair pelo mesmo lugar. Outra função que o projeto oferece é gravação de uma foto no em um cartão de memória. Para essa função é usada uma câmera serial e um cartão de memória padrão SD. Outra função é a gravação da foto no cartão. Para ser usada para possíveis reconhecimentos e fraudes.

### 1.3 APLICAÇÃO

O projeto é útil para qualquer lugar onde precise de um controle sobre a quantidade de pessoas dentro do local. A grande vantagem é a mobilidade que o projeto oferece. Ao invés de processos burocráticos de cadastramento, é feito uma contagem exata de quantas pessoas existem no local. Uma das maiores vantagens é número exato de pessoas.

A maior parte dos processos que envolvem controle na quantidade de pessoas, as pessoas que saem não entram no processo de contagem, sendo assim, é feita uma suposição de quantas pessoas existem no local.

Outra vantagem é que é fácil descobrir que pessoas burlaram o processo de contagem. Se saírem mais pessoas do que entrarem, é evidente que existem pessoas não autorizadas no local.

A desvantagem é que o projeto não possui comunicação com outros protótipos, portanto fica limitado a uma pessoa por vez e apenas um fluxo de entrada e saída.

## 2 MODULOS UTILIZADOS

### 2.1 CAMERA SERIAL

#### 2.1.1 Descrição

A câmera serial é tipo especial de câmera capaz de transmitir os dados de forma seqüencial (serial). Ela é normalmente usada como dispositivo de segurança.

Como ele foi feita com base nesses propósitos, ela não apresenta uma qualidade de imagem espetacular e nem resoluções assombrosas. O modulo comprado foi fabricado para ser usado em um arduino, outro microcontrolador que não será usado nesse projeto. O modulo apresenta as seguintes características técnicas retirada do site adafruit, fabricante do modulo.

- Module size: 32mm x 32mm
- Image sensor: CMOS 1/4 inch
- CMOS Pixels: 30M
- Pixel size: 5.6um\*5.6um
- Output format: Standard JPEG/M-JPEG
- White balance: Automatic
- Exposure: Automatic
- Gain: Automatic
- Shutter: Electronic rolling shutter
- SNR: 45DB
- Dynamic Range: 60DB
- Max analog gain: 16DB
- Frame speed: 640\*480 30fps
- Scan mode: Progressive scan
- Viewing angle: 120 degrees
- Monitoring distance: 10 meters, maximum 15meters (adjustable)
- Image size: VGA (640\*480), QVGA (320\*240), QQVGA (160\*120)
- Baud rate: Default 38400, Maximum 115200
- Current draw: 75mA
- Operating voltage: DC +5V
- Communication: 3.3V TTL (Three wire TX, RX, GND)

### 2.1.2 Pinagem

A câmera serial possui cinco pinos de entrada nela. O primeiro delas é CVB. O CVB em conjunto com o terra são capazes de fazer uma conexão diretamente em TV. Esse pino não será utilizado no projeto.

O RX é o pino que vai receber os comandos que serão enviados de um microcontrolador. Esses comandos também funcionam de maneira serial. Os comandos variam entre tirar uma foto entre outras opções.

O TX é o pino que faz a transmissão da foto por serial. Usando ele é possível receber a imagem no formato .jpeg em serial.

O VCC é a alimentação da câmera. A câmera funciona na mesma tensão do microcontrolador (5V). Existem a possibilidade do microcontrolador funcionar a 3.3V para o uso com o cartão SD.

O GND é o comum entre todos os componentes. Ele é ligado junto com os outros componentes.

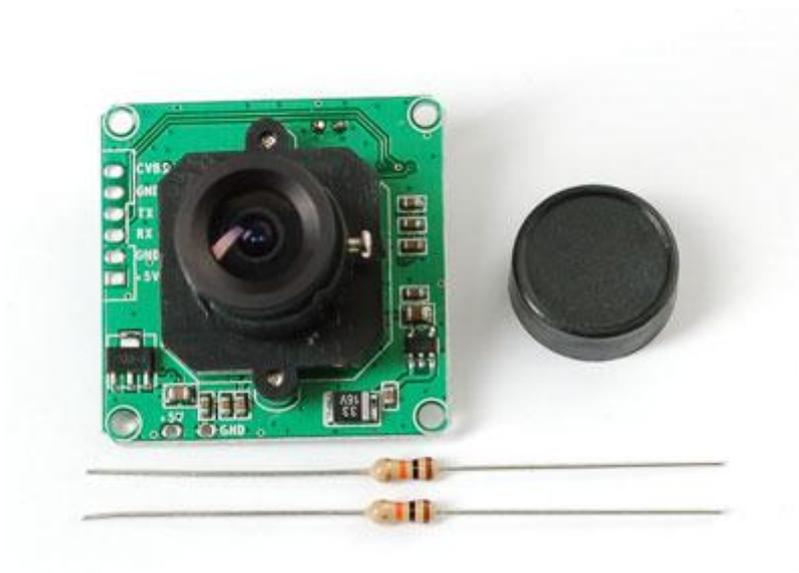


Figura 1: Camera Serial

### 2.1.3 Comandos

A câmera possui diversos recursos que não foram utilizados no projeto como reconhecimento de movimento entre outras opções.

As opções que eram necessárias para fazer a interface com a câmera eram seguintes:

- Definir o tamanho da imagem como a menor possível. (160x120)
- Iniciar a câmera com o baudrate default pelo menos (38400)
- Definir como tirar a foto usando comando serial.

Testes usando a câmera e seriais de computador foram feitos com sucesso. Os problemas encontrados na ligação com o pic pode ser visto na seção “Problemas encontrados”. Nessa seção serão colocados os problemas e as soluções para os que foram resolvidos

Foi feita uma tradução da câmera pelos nossos colegas de curso Alex Xavier e Eryck Alves. Ela, junto com o código usado no nosso projeto será postado nos anexos dessa documentação.

## 2.2 CARTÃO SD

### 2.2.1 Descrição

O cartão SD é um dispositivo de memória popular e de baixo custo muito utilizado em aparelhos portáteis. A grande vantagem é a facilidade de acesso, baixo custo e a possibilidade de ler as informações em qualquer computador que possua esse tipo de leitor.

No projeto, ele será utilizado com o intuito de gravar as fotos tiradas pela câmera serial. Como os dois funcionam de modo serial, foi feito um serial emulado.

O cartão funciona em uma tensão diferente dos demais componentes do projeto. Para uma queda de tensão são colocados resistores para o funcionamento correto do módulo. Ele funciona na tensão de 3.3V.



Figura 2: Cartão SD

O cartão SD tem outros problemas em relação à comunicação com as portas do microcontrolador. As portas do microcontrolador têm uma tensão de 4.8V, muito alta em relação aos 3.3V pedidos pelo cartão. É necessário um divisor de tensão para não prejudicar o cartão.

Outra característica é que apesar da baixa tensão, cartões de memória drenam bastante corrente, chegando ao seu pico a 100mA.

O esquemático do cartão pode ser visto a seguir:

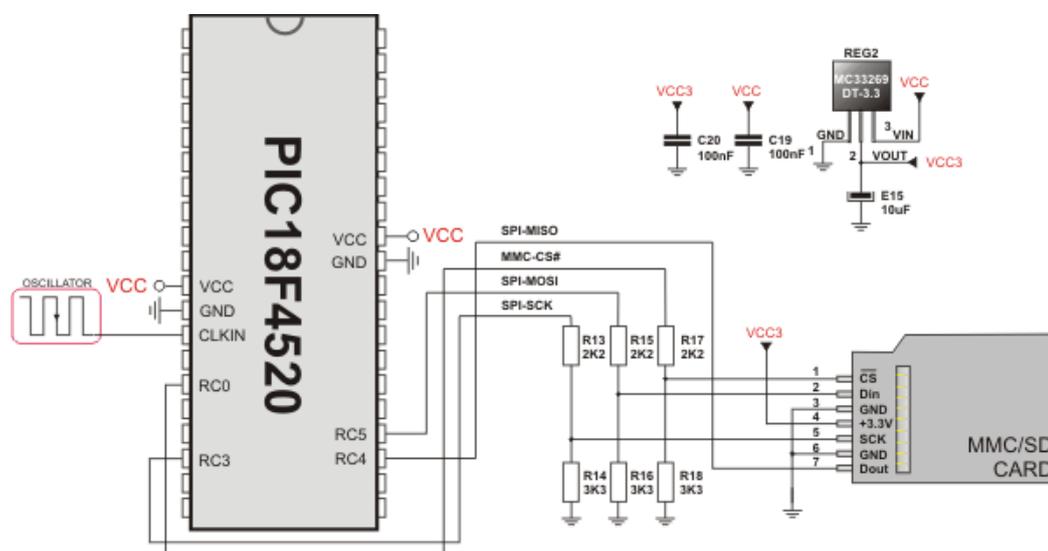


Figura 3: Esquemático da ligação do cartão

O regulador de tensão usado no projeto foi um pouco diferente. Foi usado o L78L33 para fornecer os 3.3V para o cartão SD.

Repare no divisor de tensão usando resistências de 2k2 e 3k3k aterradas.

## 2.2.2 Pinagem

A pinagem usada será para o modo SPI, mais comum entre os microcontroladores. Existe um modo mais rápido, mas tem uma biblioteca muito complicada.

Os pinos do cartão SD no modo SPI têm as seguintes características:

Pino	Nome	Entrada/Saída	Logica	Descrição
1	nCS	Entrada	Push-pull	Seletor do cartão (Negativo em alto)
2	DI	Entrada	Push-pull	Data in
3	VSS	Fonte	N	GND
4	VDD	Fonte	N	Power
5	CLK	Entrada	Push-pull	Clock
6	VSS	Fonte	Push-pull	GND
7	DO	Saída	Push-pull	Data out
8	NC nIRQ	Saída	Coletor abertor	NC interrupção Do cartão(SDIO)
9	NC	-	-	NC

Como o modulo encomendado acabou ficando retido, foi feita uma adaptação usando o esquema mostrado no tópico anterior, Como o microSD e SD normal possuem os mesmo pinos(apenas tamanhos diferentes) foi usado um adaptador com pinos soldados. Um segundo adaptador foi necessário para ler os arquivos. Na figura a seguir é possível observar quais os pinos do cartão SD.

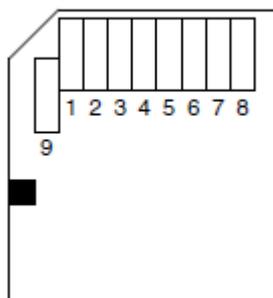


Figura 4: Pinos do cartão SD

Os pinos utilizados para o cartão foram:

Pino 1: O card selector foi colocado na porta RC0;

Pino 2: O Data in foi colocado na RC5 que é padrão serial do microcontrolador utilizado.

Pino 3: Ground.

Pino 4: Alimentação. Esse pino foi ligado junto com L78L33 para fornecer 3.3V para o circuito.

Pino 5: Ligado no RC3. É o clock do cartão SD.

Pino 6: Ground(não utilizado)

Pino 7: Data out que foi ligado ao RC4. Usado para testes do funcionamento do cartão.

Foram soldados pinos nos adaptadores seriais devido a falta de um modulo SD feito com esses propósitos.

## **2.3 SENSOR DE PASSAGEM**

### **2.3.1 Descrição**

O sensor de passagem é um dispositivo composto de um par de fotoreceptores que interceptam um emissor de luz.

Como os dois receptores se encontram em posições adjacentes, é possível saber qual é a ordem de bloqueio de sinal de cada sensor. Com essa ordem, é possível determinar se uma pessoa saiu ou entrou de um local.

Existem duas principais limitações para a lógica:

A primeira é quando uma pessoa consegue acionar somente um sensor e ir embora. Nessa situação pode ocorrer um equívoco quando uma pessoa do outro lado da pessoa que acionou o sensor fizer a passagem. A contagem nesse caso vai estar errada. A maneira encontrada para diminuir esse tipo de acontecimento, são os sensores colocados de maneira muito próxima.

Outra limitação é que apenas uma pessoa pode fazer a passagem por vez. Não é possível fazer a passagem e a contagem correta de várias pessoas fazendo passagem pelo local.

### 2.3.2 Esquemático do sensor de passagem

O sensor de passagem usa:

- 1 par de foto receptores
- 1 Emissor de infravermelho
- 2 Transistores para a amplificação de sinal
- 1 40106 porta inversora.

Figura 5: Esquemático do sensor de passagem

## **2.4 Painel LCD**

### **2.4.1 Descrição**

O painel LCD é um módulo muito utilizado em projeto microcontrolados. É um módulo, que possui clock próprio, que exibe algumas informações em um display LCD.

Um dos problemas do módulo LCD é que ele possui um clock muito menor que os microprocessadores, sendo assim são necessário atrasos para o funcionamento correto desse tipo de dispositivo.

O painel terá duas funções principais no projeto. A primeira é exibir a quantidade de pessoas que passaram pelo contador e a segunda é mostrar quando a foto está sendo armazenada no programa.

O numero de pessoas que o painel vai exibir é limitado em duas casas não existe tratamento de erro caso saia uma pessoa antes da entrada de uma pessoa.

### 2.4.2 Pinagem

Existem dois modos suportados pelo módulo LCD. O primeiro funciona em 8 bits. Nesse caso são necessárias pelo menos 10 portas para o funcionamento do display. Esse modo foi descartado para a economia de pinos para os outros módulos do projeto.

O segundo modo é modo 4 bits. Esse modo utiliza apenas 6 portas para o funcionamento correto do display. Quando é ativado esse modo, apenas as portas D4 a D7 são utilizadas. As outras portas (D0-D3) são colocadas em ground juntamente com pino R/W. Os pinos E, RS são utilizadas nas portas PORTD0 e PORTD1.

Para colocar o display em 4 bits é necessário um comando que define a função que ele possui chamado "Function Set". Nesse comando é colocado 0x20 para a inicialização com 4 bits. Na verdade esse modo manda dois comandos 4 bits para fazer a "equivalência" do modo 8 bits.



Figura 6: Display LCD

### **3 MICROPROCESSADOR**

#### **3.1 CARACTERÍSTICAS NECESSÁRIAS**

Quando a idéia do projeto estava pronta ficou uma dúvida de qual microcontrolador seria mais adequado. A grande verdade é que teria que ser rápido o suficiente para funcionar em clocks elevados devido a comunicação serial e deveria ter memória suficiente para o buffer de memória.

Acabamos optando por um PIC da família 18F por sugestão do professor e por possuir gravador para esse controlador. Além disso, o PIC conta com vários exemplos de projetos bem sucedidos com o intuito parecido com o nosso.

No começo do projeto foi utilizado um PIC16F877A, esse foi utilizado para os testes com display LCD e o circuito de contagem. Ficou evidente que ele não conseguiria fazer as outras propostas exigidas pelo professor. O projeto migrou para o PIC18F4550, por possuir mais memória e ser mais adaptável aos propósitos do projeto.

### 3.2 DEFINIÇÃO

O PIC18F4550 é um microcontrolador fabricado pela Microchip para projeto microcontrolados. Ele é um processador RISC, ou seja, possui poucas instruções para o ganho de velocidade. Além disso, possui baixo custo e pode ser encontrado com facilidade.

O PIC18F4550 possui compatibilidade com USB (não utilizada para o projeto), clock de até 48Mhz, otimizado para ser programado com C, 32Kbytes de memória flash. O fator primordial para escolha desse processador foi a memória ram, que é capaz de fazer o buffer de memória para a formatação do tipo fat16, coisa que os PIC16F não conseguem fazer de maneira satisfatória.

Ele possui um multiplicador de clock diferente dos PICs comuns. Normalmente, quando colocado um cristal, ele multiplica o clock até 96Mhz e faz a divisão para ficar em 48Mhz. O esquemático dessa multiplicação pode ser encontrado no datasheet do fabricante.

Foi utilizado na primeira tentativa um compilador diferente do fabricante microchip. O primeiro compilador utilizado foi o CCS, usado na maioria dos projetos encontrados na internet e com uma biblioteca melhor para os fins do projeto. Esse compilador não é freeware, mas é possível testar a versão demonstrativa por 30 dias. Conseguimos fazer parte do projeto utilizando esse compilador, mas obtivemos muitas dificuldades com a comunicação com o cartão SD.

O segundo compilador testado foi o C18 oferecido por 30 dias full pela própria microchip. O problema desse compilador é que a maioria dos tutoriais são feitos para os kits de desenvolvimento da empresa e não foi possível portar o código para um processador comum.

A última alternativa foi o MikroC da Mikroeletronica. Esse compilador conseguiu fazer a gravação do SD com sucesso e imediatamente portamos o código usado no CCS. Foi aprimorado algumas funções e reduzido o número de variáveis. O último problema foi a câmera serial que não conseguimos fazer a conexão com os módulos já feitos.

### 3.3 PINAGEM

O PIC18F4550 apresenta 40 pinos para disponíveis. A seguir o desenho e as explicações sobre as funções de cada um deles.

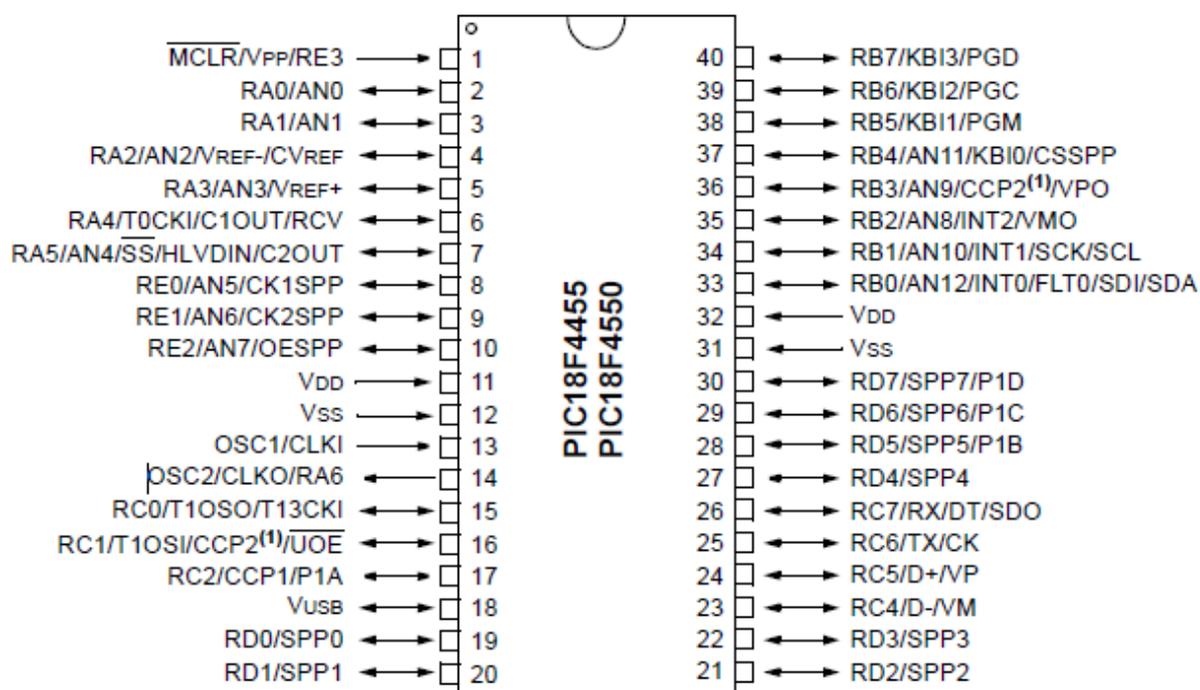


Figura 7: Pinagem do PIC18F4550

O pino MCLR é o reset do pic. Ao contrario da maioria dos processadores, esse pino deve ser ligado em zero para o reset do processador. Quando esse pino está em nível lógico um, ele funciona normalmente.

As portas A servem para os conversores analógicos digitais. Nesse projeto não houve necessidade de uso desses conversores. Vale lembrar que a porta pode ser usada como portas normais dependendo da configuração do processador.

Na porta 11 e 12 são ligados da fonte e o terra do micro respectivamente. Na porta subsequente é ligado o clock. As portas D são utilizadas exclusivamente para o painel LCD.

Uma vantagem que o pic tem que, ao contrario do 8051, ele não precisa de um reset para o funcionamento correto do processador.

Como é um prototipo, o watch dog foi desligado. O watch dog age como uma interrupção que constantemente da reset no microcontrolador. Essa função evita travamentos e outros problemas.

## 4 ESQUEMA GERAL

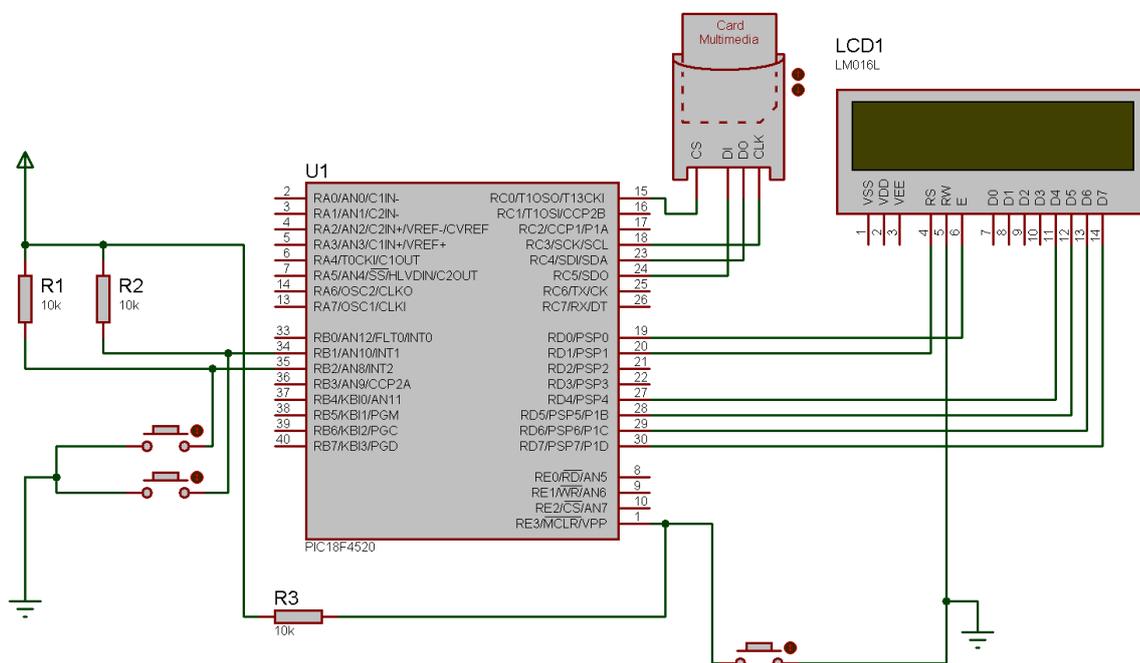


Figura 8: Esquemático geral do projeto

A camera serial não foi incluída no esquemático por ser um item que não existe em nenhum software de desenvolvimento de PCB. Os receptores estão sinalizados como push buttons para a facilitação da simulação usando o software PROTEUS.

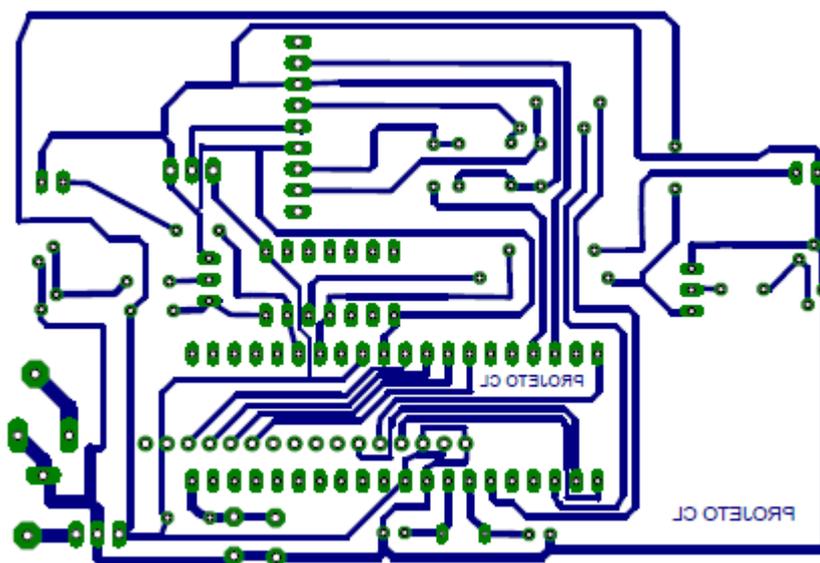


Figura 9: Placa do projeto

## 5 SOFTWARE

O papel principal do software é receber os dados da câmera e realizar a gravação das imagens em arquivos em um cartão do tipo SD. Além disto, o mesmo deve realizar a contagem das entradas e saídas do que passa pelos sensores. Conforme a ordem de detecção dos sensores, podemos saber se algo está entrando ou saindo e então incrementar ou decrementar a contagem. A foto será tirada quando um objeto entrar, ou seja, dependendo da ordem de ativação dos sensores, o software deverá acionar a câmera e receber os dados da mesma, gravando no cartão de memória.

A imagem produzida pela câmera é do formato JPEG - Joint Photographic Experts Group, um método de compressão de imagens com perdas, o que já produz uma imagem de tamanho significativamente pequeno. Seremos dependentes da velocidade de Clock que o microprocessador recebe para determinar um Baud Rate para a transmissão serial.

O funcionamento do contador tem que impedir as infinitas contagens. Como será observado todo processo de contagem possui laços que esperam o programa e só quando os sensores voltam no estado inicial acontece a contagem. Os números inteiros foram convertidos usando uma função que entra um numero inteiro e a função retorna um char. Mais detalhes podem ser vistos no anexo da documentação onde será colocado o código desses módulos.

## **6 PROBLEMAS APRESENTADOS**

O projeto teve vários problemas, alguns foram solucionados, outros atrapalharam e dificultaram o funcionamento correto do projeto

### **Atraso na entrega dos componentes**

Como o projeto era de complexidade alta, logo que foi sugerido à câmera e o cartão SD, o grupo se encarregou de comprar os módulos que facilitariam nosso trabalho. Esses módulos eram a câmera serial e módulo para usar o cartão SD. A câmera foi comprada dos EUA (pois esse tipo de câmera não pode ser encontrado no Brasil) no dia 14 de agosto de 2011. A câmera chegou ao início de novembro. Normalmente, compras feitas nos EUA levam em torno de 20 dias corridos para chegar ao destino e três meses de atraso prejudicaram o andamento do projeto.

### **Não entrega dos componentes**

Como foi dito anteriormente, no mesmo dia que foi pedido a câmera serial foi pedido um módulo SD. Esse módulo não chegou até hoje, dia 21 de novembro de 2011.

### **Painel LCD no hitech(compilador do pic 16f padrão do mplab)**

O painel LCD foi implementado em c utilizando funções mas o funcionamento ficou comprometido devido os delays. A solução foi migrar o código para o compilador CCS que atendeu bem as necessidades para esse propósito.

### **Não gravação do cartão**

Foram realizados testes e foi constatado que o serial do CCS funcionava de maneira correta. Utilizando bibliotecas prontas para a formatação fat16 e para a comunicação com o cartão SD não foi possível conseguir gravar algo usando esse compilador. Foram revisados os esquemáticos e a programação e foi verificado que a biblioteca padrão de cartão do CCS apresentava erros mostrados pelos próprios usuários. Os erros foram consertados, mas ainda sim não conseguimos gravar algo usando esse compilador. A solução foi tentar migrar de compilador. Na tentativa de fugir do CCS foi utilizado o C18. O C18 é o compilador da Microchip. Ele possui

bibliotecas poderosas como a MDFFS, para dispositivos de armazenamento. A microchip deixa disponível as bibliotecas gratuitamente. O problema é que essas bibliotecas são direcionadas para os kits de desenvolvimento oferecidos pela própria microchip. A não compilação e a alta complexidade fizeram a última migração de compiladores. A última alternativa foi o MikroC da MikroElektronika. Com esse compilador conseguimos gravar algo no cartão sd mas o prazo do projeto já estava apertado.

### **Instabilidade na gravação SD**

Foi constatada uma instabilidade na gravação do cartão SD. Muitas das vezes testadas não ocorria a gravação ao gravava lixo no lugar. Um dos motivos é um esquemático feito com os divisores de tensão que não eram muito estáveis.

### **Comunicação ruim com a câmera**

A câmera e a porta serial do computador conseguiam conversar de maneira correta. Tiravam fotos e guardavam usando o TX da câmera.

## CONCLUSÃO

O projeto controle de lotação não funcionou como o esperado. A parte de contagem e da exibição no painel LCD foram concluídas com sucesso, mas obtivemos muitos problemas relacionados à câmera e a o cartão SD. A equipe se esforçou ao máximo que pode. Foram feriados e finais de semana trabalhando para o projeto dar certo, mas aconteciam vários problemas diferentes.

O atraso da entrega dos componentes e a não entrega de componentes foram fatores que não podem ser ignorados. Outro grande problema foi à constante migração de compiladores devido aos problemas encontrados. Infelizmente não foi possível ter êxito em todas as funções oferecidas pelo projeto.

## Referencias

IBRAHIM, DOGAN, SD Card projects using PIC Microcontroller, 2010

Serial Camera, Disponivel em <<http://www.ladyada.net/products/camera/>> acesso em 25 de agosto de 2011.

Driving a serial câmera with netduino, Disponivel em<<http://fabienroyer.wordpress.com/2011/08/12/driving-an-adafruit-vc0706-ttl-serial-jpeg-camera-with-a-netduino/>> Acesso em 2 de outubro de 2011.

Microchips, Disponivel em < <http://www.microchip.com/>>

## ANEXOS

### Função principal:

```
#include "moduloSD.c"
```

```
void main() {
```

```
    // Criando a variavel de contagem;
```

```
    int contador =0;
```

```
    // Setando os pinos de entrada para os receptores
```

```
    TRISB1_bit = 1;
```

```
    TRISB2_bit = 1;
```

```
    // Iniciando o cartao...
```

```
    iniciaCartao();
```

```
    // Configurando a porta B
```

```
    Lcd_Init();
```

```
    // desligando o cursor
```

```
    Lcd_Cmd(_LCD_CURSOR_OFF);
```

```
    // Exibindo as mensagens
```

```
    Lcd_Out( 1, 1, "CL V1.0");
```

```
    Lcd_Out(2,1, "Quantidade:");
```

```
    //exibeContagem(transformaDigito((contador/10)),  
transformaDigito((contador%10)));
```

```
    /*Coloque a escreveCartao(contador) ligado a uma porta ou
```

```
    faça depois da contagem. Como ele entra em loop devido a inialização
```

não consegui testar o restante com o proteus.\*/

```
while(1){
    contador = moduloContador(contador);

}
}
```

### Modulo de contagem:

```
/*-----Inicio das configurações do LCD-----*/

// Configurando os bits do LCD
sbit LCD_RS at RD1_bit;
sbit LCD_EN at RD0_bit;
sbit LCD_D7 at RD7_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D4 at RD4_bit;

// Configurando as direções
sbit LCD_RS_Direction at TRISD1_bit;
sbit LCD_EN_Direction at TRISD0_bit;
sbit LCD_D7_Direction at TRISD7_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D4_Direction at TRISD4_bit;

/*-----Fim das configurações do LCD-----*/

// Criando a função que converte um numero de UM digito para um char equivalente
char transformaDigito(int numero){
    char u;
    switch(numero){
        case 0:u = '0';break;
```

```

    case 1:u = '1';break;
    case 2:u = '2';break;
    case 3:u = '3';break;
    case 4:u = '4';break;
    case 5:u = '5';break;
    case 6:u = '6';break;
    case 7:u = '7';break;
    case 8:u = '8';break;
    case 9:u = '9';break;
  }
  return u;
}

```

// Para exibir a contagem é só usar a função a seguir:

```

void exibeContagem(char dezena, char unidade)
{
  Lcd_Chr(2,12, dezena);
  Lcd_Chr(2,13, unidade);
}

```

/\*Essa função será usada caso seja impresso outra coisa na tela como .... gravando ou tirando foto ou qualquer coisa\*/

```

void Teladefault(){
  Lcd_Out( 1, 1, "CL V1.0");
  Lcd_Out(2,1, "Quantidade:");
  // exibeContagem(transformaDigito((contador/10)),
transformaDigito((contador%10)));
}
/*-----Fim da conversão dos numeros e das funções do painel ICD-----*/

```

```

/*-----
-----Função do modulo emissor e receptor-----

```

```

-----*/

int moduloContador(int contador){

    if(PORTB.F1 == 0 && PORTB.F2 == 1)
    {
        for(;;)
        {
            if(PORTB.F2 == 0 && PORTB.F1 == 1)
            {
                for(;;){
                    if(PORTB.F1==1 && PORTB.F2 == 1)
                    {
                        contador++;
                        exhibeContagem(transformaDigito((contador/10)),
transformaDigito((contador%10)));
                        break;
                    }
                }
                break;
            }
        }
        // Testa o bit de volta
        if(PORTB.F2 == 0 && PORTB.F1 == 1 )
        {
            for(;;)
            {
                if(PORTB.F1 == 0 && PORTB.F2 == 1)
                {
                    for(;;){
                        if(PORTB.F1 == 1 && PORTB.F2 == 1)
                        {

```

```

        contador--;
        exibeContagem(transformaDigito((contador/10)),
transformaDigito((contador%10)));
        break;
    }

}
break;
}
}
}
return contador;
}

```

### Programa feito para o compilador CCS

```

#include <ProjetoCLCSS.h>
#include <lcd.c>

char separaUnidade(int numero){
    int unidade;
    char u;
    unidade = numero%10;
    switch(unidade){
        case 0:u = '0';break;
        case 1:u = '1';break;
        case 2:u = '2';break;
        case 3:u = '3';break;
        case 4:u = '4';break;
        case 5:u = '5';break;
        case 6:u = '6';break;
        case 7:u = '7';break;
        case 8:u = '8';break;
        case 9:u = '9';break;
    }
}

```

```
    }  
    return u;  
  
}  
char separaDezena(int numero){  
    int dezena;  
    char u;  
    dezena = numero/10;  
    switch(dezena){  
        case 0:u = '0';break;  
        case 1:u = '1';break;  
        case 2:u = '2';break;  
        case 3:u = '3';break;  
        case 4:u = '4';break;  
        case 5:u = '5';break;  
        case 6:u = '6';break;  
        case 7:u = '7';break;  
        case 8:u = '8';break;  
        case 9:u = '9';break;  
    }  
    return u;  
  
}  
  
void exhibeContagem(char unidade, char dezena){  
    lcd_putc(unidade);  
    lcd_putc(dezena);  
    lcd_gotoxy(1,2);  
  
}  
  
void main()  
{  
    //char c = '1';
```

```
int contador=0,i=0;
char MSG[7]={'C','L','V','1','!','0'};

set_tris_b(1);
lcd_init();
//lcd_putc(c);
lcd_gotoxy(1,1);
for(i=0;i<6;i++)
    lcd_putc(MSG[i]);

    lcd_gotoxy(1,2);
//Programa Principal

while(true){

    // Testando para ver se está entrando alguém na porta
    // Testa o bit de ida

    // Nesse teste a porta C vai ser a modificada
    if(Input(48) == 1 && Input(49) == 0)
    {
        for(;;)
        {
            if(Input(49) == 1 && Input(48) == 0)
            {
                for(;;){
                    if(Input(48)==0 && Input(49) == 0)
                    {
                        contador++;
                        exibeContagem(separaDezena(contador),separaUnidade(contador));
                        break;
                    }
                }
            }
        }
    }
}
```

```
        break;
    }
}
}
// Testa o bit de volta
if(Input(49) == 1 && Input(48) == 0 )
{
for(;;)

{
    if(Input(48) == 1 && Input(49) == 0)
    {
        for(;;){
            if(Input(48) == 0 && Input(49) == 0)
            {
                contador--;
                exibeContagem(separaDezena(contador),separaUnidade(contador));
                break;
            }

        }
        break;
    }
}
}
}
}
```