



PUCPR

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

**PROJETO INTEGRADO PiSP
DOCUMENTAÇÃO FINAL**

CURITIBA, 2011

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

LUCAS GUIMARÃES GONÇALVES

**PROJETO INTEGRADO PiSP
DOCUMENTAÇÃO FINAL**

Documentação do projeto integrado realizada pelo aluno Lucas Guimarães Gonçalves para a disciplina de Resolução de Problemas em Engenharia (RPE) do curso de Engenharia de Computação.

CURITIBA, 2011

Índice

0 – RESUMO.....	1
1 – INTRODUÇÃO.....	1
2 – OBJETIVO.....	1
2.1 – DO PROJETO.....	1
2.2 – ESPECÍFICO PARA A REALIZAÇÃO DO PROJETO.....	1
3 – MATERIAIS UTILIZADOS.....	2
3.1 – DESCRIÇÃO.....	2
3.2 – IMAGENS DOS COMPONENTES ELETRÔNICOS.....	2
4 – ETAPAS DETALHADAS DE REALIZAÇÃO DO PROJETO.....	4
5 – CÓDIGOS E DESENHOS ESQUEMÁTICOS.....	4
5.1 – MÓDULO PiSPic.....	4
5.2 – PCI MÓDULO PiSPic.....	5
5.3 – CÓDIGO FONTE DO PIC.....	5
6 – CONCLUSÃO.....	8

0 – RESUMO

Essa documentação descreve as informações, códigos, desenhos esquemáticos relacionados ao projeto integrado **PiSP (Piso Sensível à Pressão)**.

A ideia do projeto foi dada pelo aluno Rodrigo Ramirez Echegorri e realizada também pelo aluno Lucas Guimarães Gonçalves para a disciplina de Resolução de Problemas em Engenharia (RPE), sob orientação do professor Afonso Ferreira Miguel.

1 – INTRODUÇÃO

Os sistemas de piso sensível a pressão são utilizados principalmente no trânsito, com o objetivo de calcular a velocidade dos veículos. Existem também alguns utilizados para abrir uma porta quando algo se aproxima dela, mas um dos problemas é o da porta e do piso sensível serem vendidos juntos como uma peça individual.

O projeto PiSP também é um piso sensível a pressão, mas ele possui uma interface para o usuário ao qual pode ser conectado um dispositivo que será ativado quando o piso for pressionado. Além disso ele possui uma ativação magnética com resultados muito rápidos.

2 – OBJETIVO

Nesse item será focado o objetivo do projeto com o usuário e o objetivo para realização do mesmo.

2.1 – DO PROJETO

Atualmente, em ambientes urbanos, a necessidade da criação de dispositivos de presença se torna cada vez maior, sejam eles para controlar a iluminação de uma residência até para serem utilizados como dispositivos de segurança.

Logo, o PiSP entra como uma alternativa, tornando possível a implementação de dispositivos eletrônicos externos quaisquer (portas, alarmes, trancas).

2.2 – ESPECÍFICO PARA A REALIZAÇÃO DO PROJETO

- Relatar o projeto e criar um cronograma para o mesmo.
- Montagem do piso protótipo.
- Programação dos componentes digitais.
- Montagem das placa eletrônica.
- Implementação da placa no piso.
- Implementação da interface de alimentação e ativação.

3 – MATERIAIS UTILIZADOS

Nesse item serão mostrados os materiais utilizados para a construção do projeto.

3.1 – DESCRIÇÃO

<Quantidade> <Nome>: <Descrição>

- 1 PIC12F675 (SIP 8 pinos): Microcontrolador utilizado para fazer a conversão A/D do sinal do sensor de hall e ativar o dispositivo externo quando o piso for pressionado. Demonstrado na figura 1.
- 1 Ímã (12x6x3 mm): Objeto magnético feito de elemento Neodímio. Demonstrado na figura 2.
- 1 A1321LUA-T: Sensor de Hall linear. Demonstrado na figura 3.
- 1 IRF640: Transistor de efeito de campo (FET). Demonstrado na figura 4.
- 3 Resistores: Para uso específico, 10K Ω , 2K9 Ω e 100K Ω .
- 1 Caixa de madeira (10x10 cm): Caixa utilizada para fazer o piso protótipo.
- 4 Molas: Utilizadas para variar o piso.
- Fios: Utilizados para fazer conexões externas.
- 4 Parafusos: Utilizados para prender a placa no piso.
- 1 Placa de Fenolite Cobreada: Placa para montagem do circuito.
- Cola bastão: Utilizada para prender as molas no piso.
- 1 Fechadura elétrica.

3.2 – IMAGENS DOS COMPONENTES ELETRÔNICOS

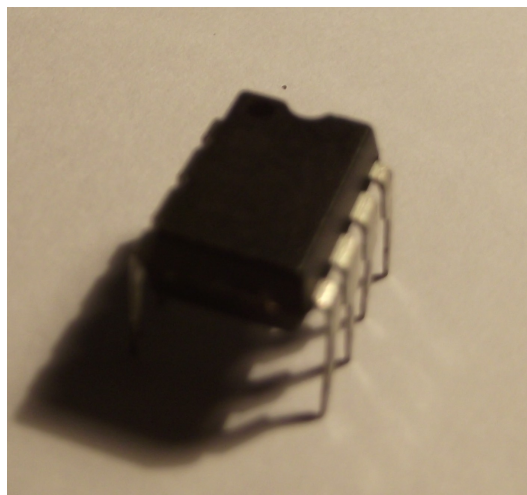


Figura 1: Microcontrolador PIC12F675

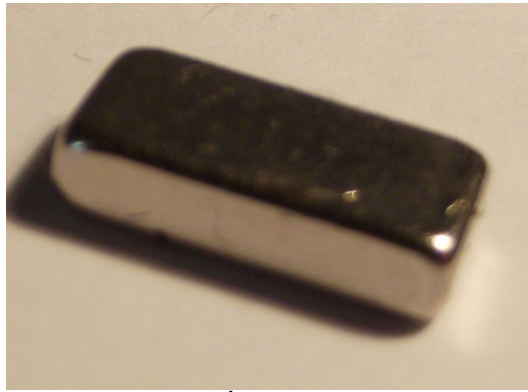


Figura 2: Ímã de Neodímio

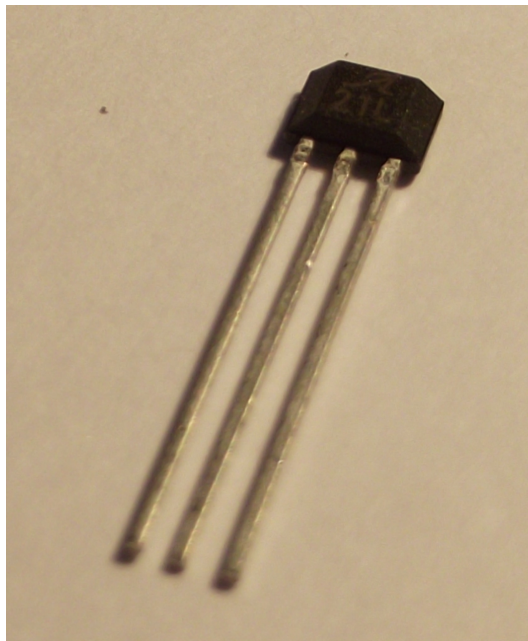


Figura 3: Sensor de Hall A1321LUA-T

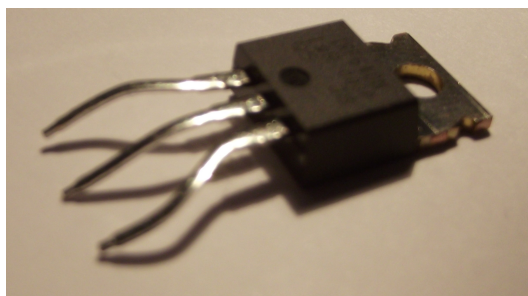


Figura 4: Transistor FET

4 – ETAPAS DETALHADAS DE REALIZAÇÃO DO PROJETO

O plano de trabalho foi utilizado como um guia para a montagem de um cronograma, que foi cumprido pelos integrantes da equipe para a construção do projeto.

Para a montagem do piso protótipo, foi necessário uma caixa de madeira 10x10 cm, 4 molas, cola bastão e ferramentas para corte de madeira, que foram fornecidas pelo aluno Lucas. O caixa foi adquirida pelo aluno Rodrigo, que ficou encarregado de cortar a caixa e colar as molas, fazendo uma espécie de tampa que variava com a pressão sobre ela.

Para programar o PIC, foi necessário um compilador Assembly e uma ferramenta de programação PICStart Plus. No PIC, foram utilizados um pino de entrada analógica para receber a tensão de saída do Sensor de Hall e um outro pino de saída digital para ativar o amplificador. O encarregado de realizar essa tarefa foi Lucas, que também forneceu as ferramentas necessárias.

A etapa de desenho da PCI e montagem da placa foi concluída com as ferramentas fornecidas pela PUCPR. Para o desenho da PCI (placa de circuito impresso) foi utilizada a ferramenta EAGLE. Essa etapa e as etapas de implementação foram realizadas nos laboratórios da faculdade com as ferramentas disponíveis. O encarregado dessas atividades foi Lucas.

5 – CÓDIGOS E DESENHOS ESQUEMÁTICOS

Nesse item serão demonstrados o esquemático do módulo eletrônico do projeto (figura 5), a PCI (figura 6) e o código fonte na linguagem assembly que é compilada e gravada no PIC.

5.1 – MÓDULO PiSPic

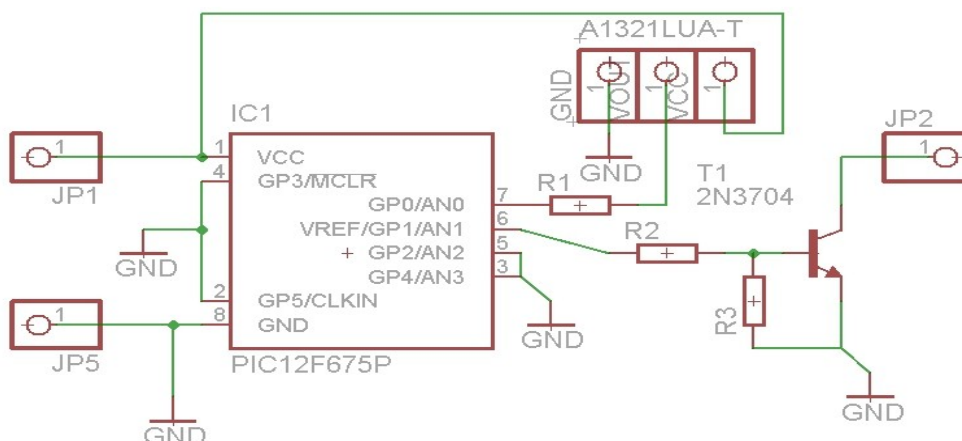


Figura 5: PiSPic

A figura 5 representa o modelo esquemático feito na ferramenta EAGLE.

5.2 – PCI MÓDULO PiSPic

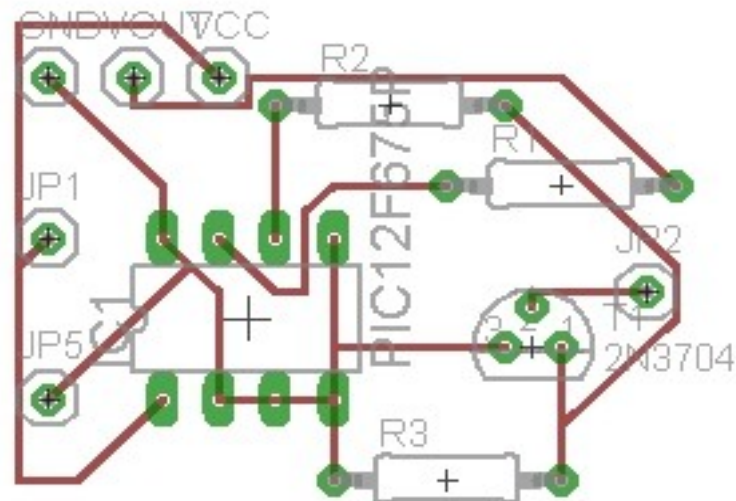


Figura 6: PCI PiSPic

A figura 6 representa o desenho do esquemático para impressão na placa.

5.3 – CÓDIGO FONTE DO PIC

O código foi escrito em Assembly por Lucas Guimarães Gonçalves, os comentários estão em cor verde.

```
list p=12f675 ; Processador utilizado.
#include <p12f675.inc> ; Biblioteca do PIC 12f675.

; Configuração, _INTRC_OSC_NOCLKOUT: Ativa clock interno (4MHz).
_CONFIG _INTRC_OSC_NOCLKOUT & _WDT_OFF & _PWRTE_ON &
_MCLR_OFF & _CP_OFF & _CPD_OFF

; Variáveis (0x20 = Início da RAM de acesso livre)

; Variáveis contadores para o timer programado.
contador1 equ 0x20
contador2 equ 0x22
contador3 equ 0x24

; Variável para calcular a média.
media equ 0x26
led equ 0x28 ; bit de controle

; -----
org 0x000 ; RESET.
goto inicio
```



```

;-----
org 0x004 ; INTERRUPÇÃO.
retfie ; Retorna.

;-----
inicio:
; RP0 é o 5º bit do registro STATUS que controla os bancos de registros.
bsf STATUS, RP0 ; Muda para o banco 1 (RP0 = 1).

movlw B'00000001' ; Move 00000001 para o registrador W.

movwf TRISIO ; Bota o pino GP0 (pino 7 do diagrama) como entrada e o resto
como saída (bit's 7 e 6 não implementados).
movwf ANSEL ; Seta entrada analógica AN0 e o resto como entrada/saída digital.

bcf STATUS, RP0 ; Muda para banco 0.

movwf ADCON0 ; Ativa conversor A/D (bit menos significativo do registro
ADCON0).

bcf led, 0
bcf GPIO, 1

codigo:
movlw 0x6
movwf media

amedia:
bcf ADCON0, 1 ; Prepara para pegar a amostragem (A/D).
bsf ADCON0, 1 ; Pega amostragem (A/D).

call esperar100ms ; Espera 100 ms

movlw 0x20
bsf STATUS, C
subwf ADRESH, w

btfss led, 0
goto mediab

goto mediac

mediab:
btfss STATUS, C ; btfss STATUS, C: Se C=0, a próxima instrução é executada.
call d1media

goto codigo

mediac:
btfsc STATUS, C

```

```
call d2media
```

```
goto codigo
```

```
d1media:
```

```
decfsz media, f
```

```
goto amedia
```

```
bsf GPIO, 1 ; Ativa tranca.
```

```
bsf led, 0
```

```
goto codigo
```

```
d2media:
```

```
decfsz media, f
```

```
goto amedia
```

```
bcf GPIO, 1; Apaga led.
```

```
bcf led, 0
```

```
goto codigo
```

```
; 1 ciclo = 2,5 us
```

```
; total = 5 + 35 (=40 ciclos) = call+return-1+movlw+movwf + decfsz+goto (instruções)  
= 12,5us + 87,5us (tempo)
```

```
esperar100us: ; 2 ciclos
```

```
movlw 0xB ; 1 ciclo
```

```
movwf contador1 ; 1 ciclo
```

```
decfsz contador1, f ; 2 ciclos
```

```
goto $-1 ; 1 ciclo
```

```
return ; 2 ciclos
```

```
; total = 5 + 3995 (=4000 ciclos) = 12,5us + 9987,5us
```

```
esperar10ms: ; 2 ciclos
```

```
movlw 0x5D ; 1 ciclo
```

```
movwf contador2 ; 1 ciclo
```

```
call esperar100us ; 40 ciclos
```

```
decfsz contador2, f ; 2 ciclos
```

```
goto $-2 ; 1 ciclo
```

```
return ; 2 ciclos
```

```
; total = 5 + 39995 (=40000 ciclos) = 12,5us + 99987,5us
```

```
esperar100ms: ; 2 ciclos
```

```
movlw 0xA ; 1 ciclo
```

```
movwf contador3 ; 1 ciclo
```

```
call esperar10ms ; 4000 ciclos  
decfsz contador3, f ; 2 ciclos  
goto $-2 ; 1 ciclo
```

```
return ; 2 ciclos
```

```
END ; Identificação para fim do programa.
```

6 – CONCLUSÃO

O projeto integrado foi um grande projeto de aprendizado, desde o entendimento de como funciona um sensor de hall até a programação de um microcontrolador, incluindo a montagem das placas. Além disso, a equipe ganha como experiência também um avanço na organização de ideias e tomada de decisões frente a problemas encontrados.

A equipe agradece ao professor Afonso Ferreira Miguel, ao professor Gil Marcos Jess e à PUCPR, que nos disponibilizou os laboratórios e as ferramentas.