

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANA
CCET – CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
ENGENHARIA DE COMPUTAÇÃO – 6.º PERÍODO**

Smart Cooler

Angelo André Lima, Daniel Vosnika, Eduardo Francisco Colares,
Eduardo Froes da Motta Budant

Resumo

O projeto foi desenvolvido por alunos do 6º período do curso de Engenharia de Computação da Pontifícia Universidade Católica do Paraná, e propõe criar um cooler inteligente. Conforme a temperatura ambiente, ele aciona o cooler com maior ou menor intensidade. O microcontrolador lê o valor vindo de um sensor de temperatura analógico, e de acordo com a temperatura, calcula a intensidade com que o motor do cooler deve ser acionado, em seguida envia o valor da temperatura e intensidade do cooler para o PC.

Abstract

The project was developed by Computer Engineer's students from the sixth period of the Pontifícia Universidade Católica do Paraná, and the main idea was to project and build a smart cooler. The cooler is activated with different intensity after the temperature is readen from the analogic temperature sensor, than it process the information and calculates how fast the cooler should spin, after that sends the temperature and intensity from cooler to computer..

Sensor de temperatura

O sensor utilizado foi o LM35. Escolhemos ele por já termos trabalhado com ele em uma experiência na matéria de Microprocessadores II. Também por ser um sensor de temperatura linear, o que nos facilitaria na hora que ler a informação e trabalhar com o valor convertido pelo conversor AD.

O sensor LM35 é um sensor de precisão, fabricado pela National Semiconductor, 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.

Abaixo segue a tabela com os valores da conversão analógica/digital do sensor LM35, tensão gerada pelo sensor, e a porcentagem da velocidade do cooler a ser ligado. Tudo de acordo com a temperatura.

°C	Valor A/D	V (Tensão) LM35	PWM (%)
≤ 22	46	0,22	30
≤ 35	71	0,35	45
≤ 60	122	0,60	70
> 60	x	x	99

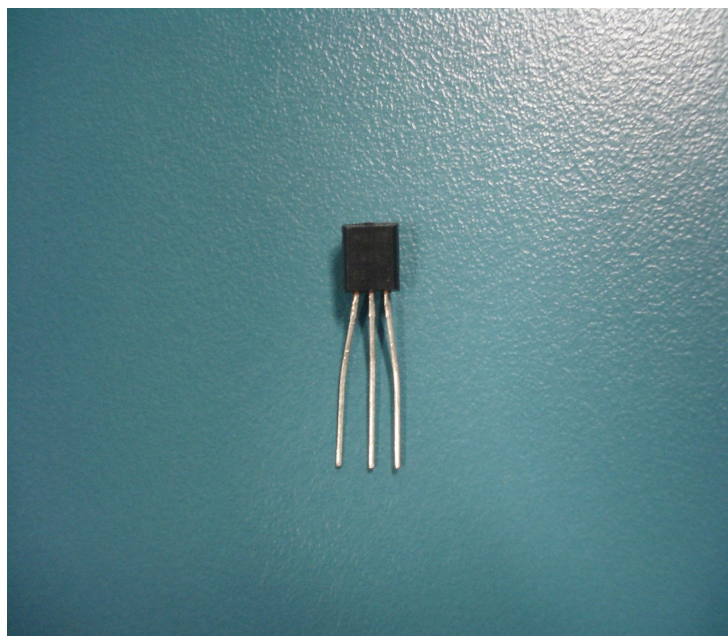


Figura 1- Sensor de temperatura

Cooler

O aumento excessivo da temperatura de muitos equipamentos podem fazer os mesmos queimarem. Em processadores, por exemplo, um aumento excessivo de temperatura pode através dos diferentes índices de expansão dos metais, causar microrrupturas na superfície do chip, ou em casos extremos uma fundição do chip causado por um aumento exponencial da resistência elétrica.

Então, podemos concluir que o calor em excesso pode derreter os minúsculos circuitos do processador caso não exista um *cooler* instalado na maioria absoluta dos casos, salvo chips que consomem pouca eletricidade, e que por isso não demandam por uma solução de refrigeração muito eficiente, como chipsets, processadores de baixíssimo consumo, controladoras, etc.

Usamos um Cooler Intel, que trabalha com uma tensão de 5-12V. E controlamos sua velocidade através do modulo PWM do microcontrolador PIC.

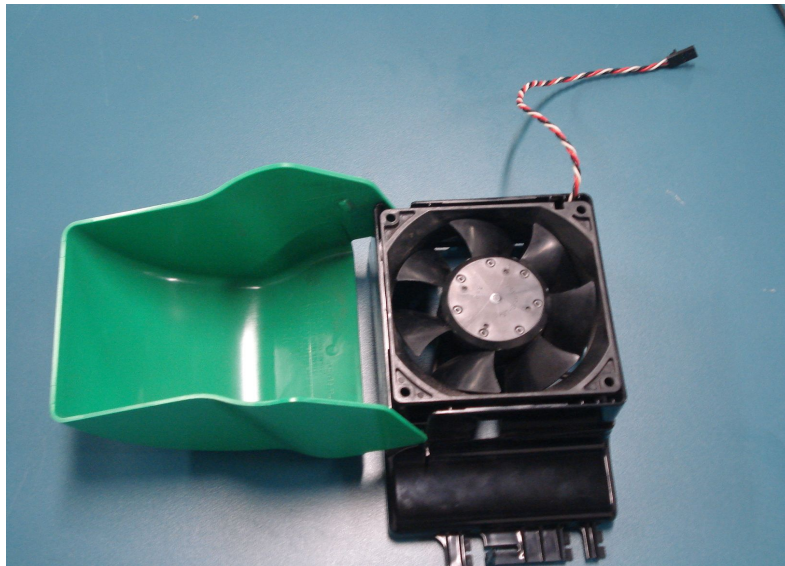


Figura 2 – Cooler

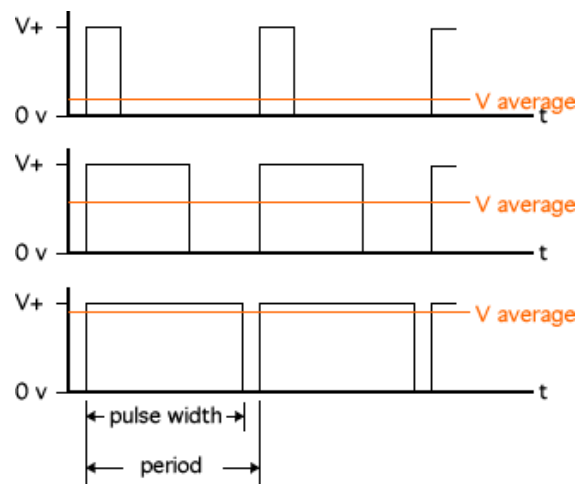


Figura 3 – Modulo PWM

Microcontrolador

Escolhemos o microcontrolador PIC 16F877 por já termos trabalhado com ele durante as aulas, e por ter interesse em fazer sua programação em linguagem C. Usamos o compilador CCS C.

PIC 16F877A

-RISC Architecture

Only 35 single word instructions
8K x 14 words of FLASH (100,000 erase/write)
368 x 8 bytes of Data Memory (RAM)
256 x 8 bytes of EEPROM (1,000,000 erase/write Cycle)

-Peripheral Features

Two 8-bit and One 16-bit Timer/Counters
Two PWM Channels
8-channel, 10-bit ADC
UARTSPI, I2C

-Operating Voltages

2.0V to 5.5V

-Speed Grades

20 MHz clock input

-Register

PORTx: Data register using output data to port and read port value
TRISx: Setting data direction. 1 = Input, 0 = Output

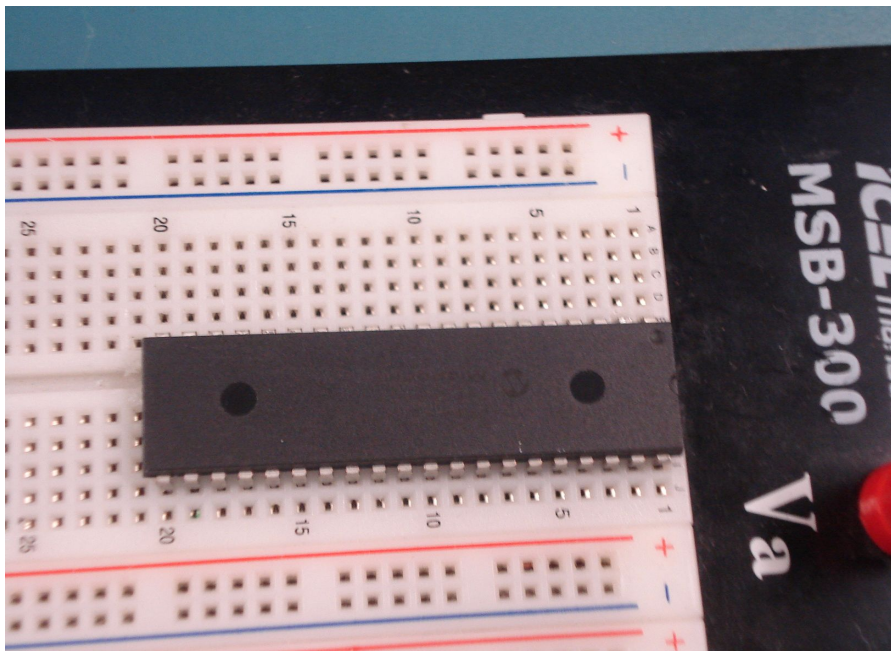


Figura 4 – PIC 16F877

Algoritmo do Projeto

O projeto Smart Cooler contou com dois softwares. Um embarcado feito em linguagem C para PIC, e outro em C# usado no computador para visualizar a temperatura e rotação do Cooler. Foi escolhido a linguagem C pela sua facilidade e por querer diferenciar do assembly, que foi usado durante a disciplina de microprocessadores. E resolvemos também usar o C# .NET por já possuir classes prontas para comunicação Serial.

Software embarcado:

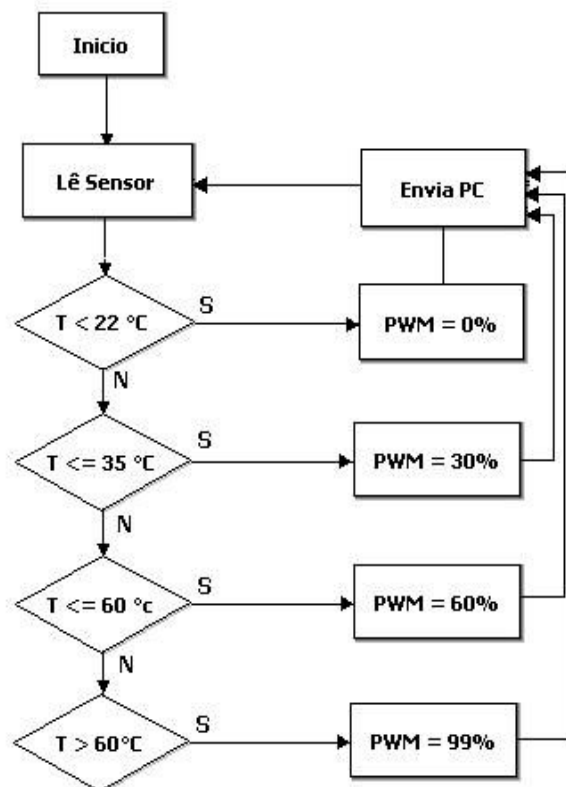


Figura 5 – Diagrama de fluxo software embarcado

Software PC:

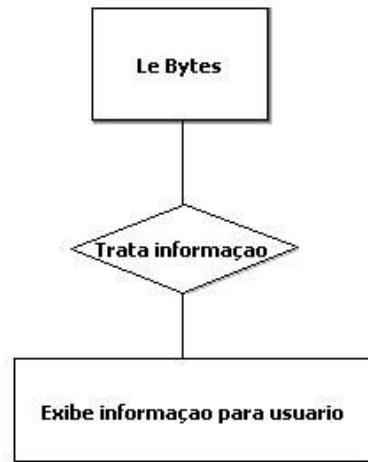


Figura 6 – Diagrama de fluxo software PC

Esquemático e Circuitos

Abaixo seguem o esquema de funcionamento do sistema de resfriamento inteligente. Que consiste na leitura do sensor, calculo da velocidade do cooler e da temperatura, e o envio das informações via comunicação serial para o PC fazer a leitura através do software.

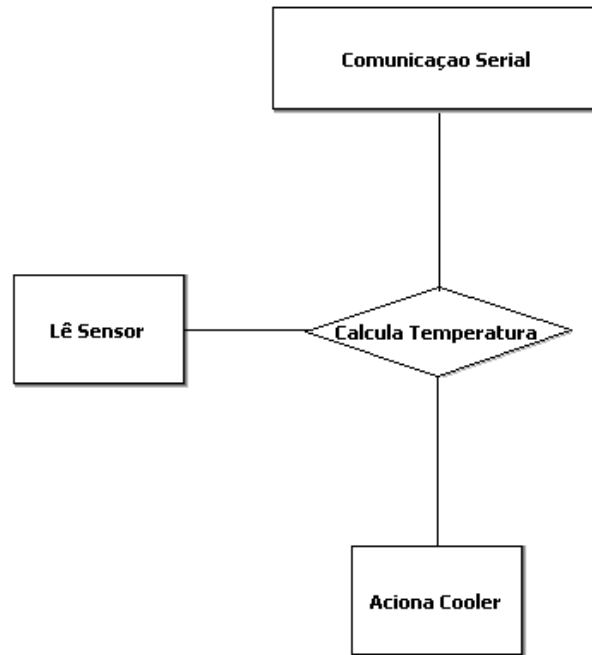


Figura 7 – Esquemático de Funcionamento

Circuito Comunicação Serial

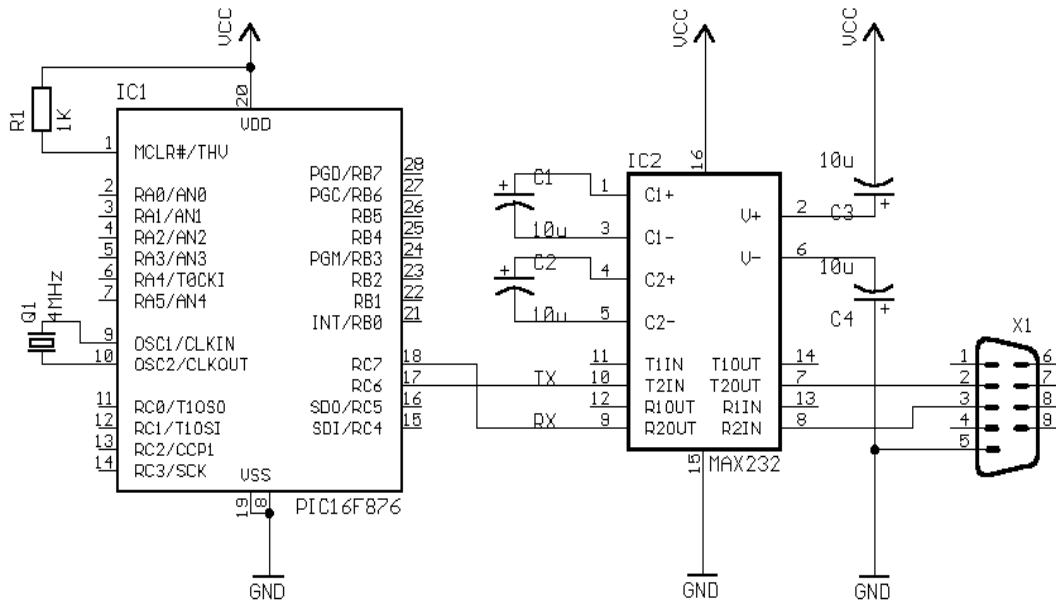


Figura 8 – Circuito Comunicação Serial

Circuito Projeto

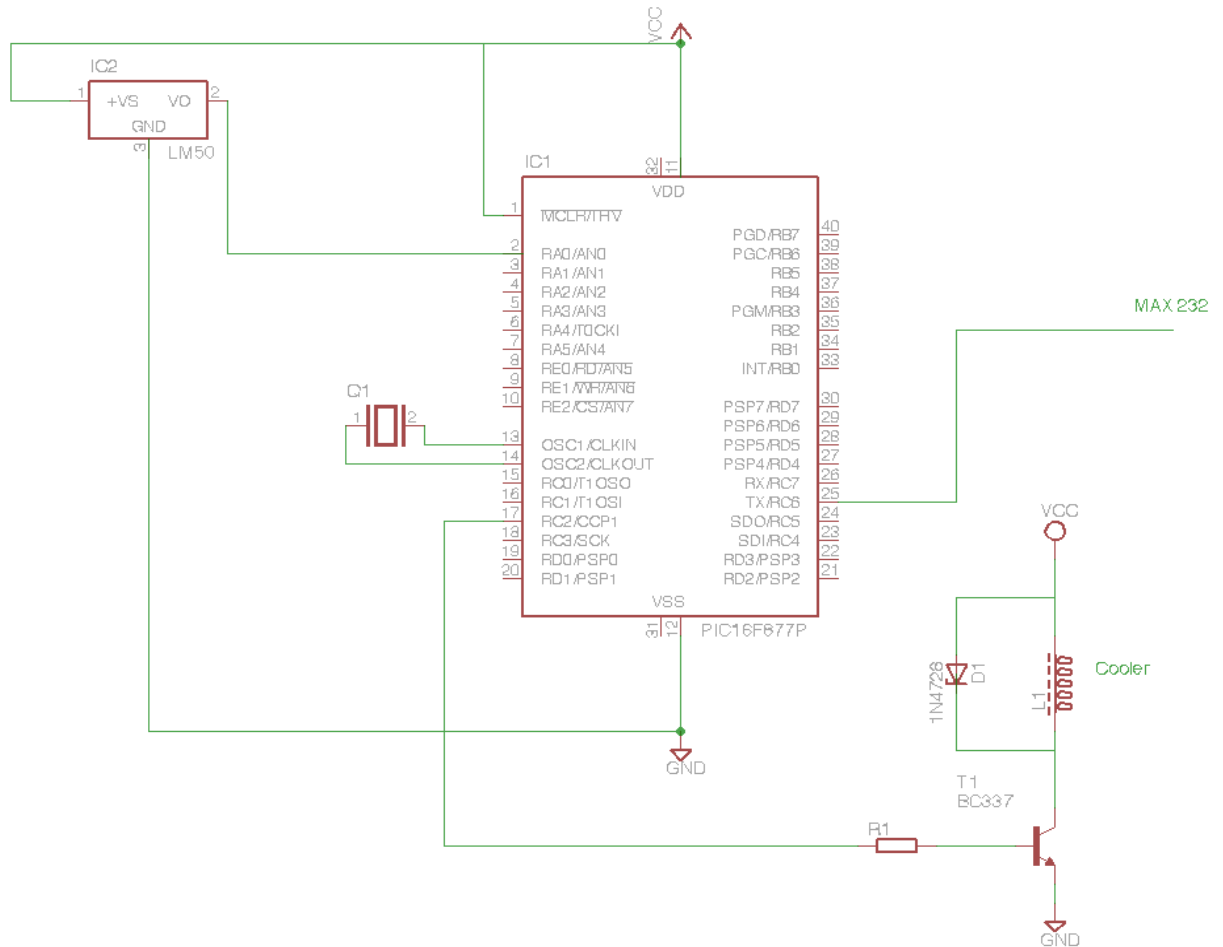


Figura 9 – Circuito Projeto

Custo do Projeto

No projeto pudemos usar alguns componentes que já tínhamos comprado para usar na matéria de microprocessadores como o microcontrolador PIC 16F877A e o modulo MAX 232 que havíamos usados em um projeto anterior.

Nos restou comprar o acrílico, sensor de temperatura e o cooler. Usamos acrílico para montar a maquete. O sensor para detectar a temperatura, e o cooler para resfriar o sensor.

Dispositivo	Quantidade	Preço	Custo Total
PIC 16F877A	1	R\$ 12,00	R\$ 12,00
LM35	2	R\$ 2,40	R\$ 4,80
MAX 232 Modulo	1	R\$ 10,00	R\$ 10,00
Cooler	1	R\$ 20,00	R\$ 20,00
Acrilico	1	R\$ 15,00	R\$ 15,00
		TOTAL	R\$ 61,80

Conclusões

O trabalho teve como objetivo criar um sistema inteligente de resfriamento, que faz a leitura da temperatura ambiente através de um sensor analógico de temperatura, e de acordo com a temperatura regula um cooler de resfriamento. O trabalho foi uma junção de muitas matérias que vimos durante o curso e pudemos por em pratica junto. Como por exemplo a comunicação de um sistema embarcado com um software alto nível em C#.

Usamos a programação alto novel para o software do PC que recebe os dados via comunicação serial. Programação baixo nível para o PIC que recebe os dados dos sensores e aciona o cooler via PWM. O sistema de resfriamento que usa transistor e um diodo para fazer o chamamento do cooler.

Como sempre os projetos integrados tem servido para pormos em prática todo o conhecimento adquirido até aqui no curso. Nos deparamos com situações em que tivemos que pesquisar componentes e decidir qual seria melhor usar. Isso engrandeceu muito nosso conhecimento.

Referências

PIC 16F87XA, Datasheet “Microcontrolador” Disponível em
<ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>

LM35, Datasheet “LM35” Disponível em <www.national.com/ds/LM/LM35.pdf>

CCS, apostila

Cooler, datasheet