

Experimento 6 – Projeto final

Fundamentação

Diferente dos experimentos anteriores, este experimento não possui um roteiro definido. Os alunos podem escolher qualquer projeto, individual ou em equipe. O assunto do projeto deve ser combinado com o professor. Para facilitar, a seguir temos algumas sugestões de projetos.

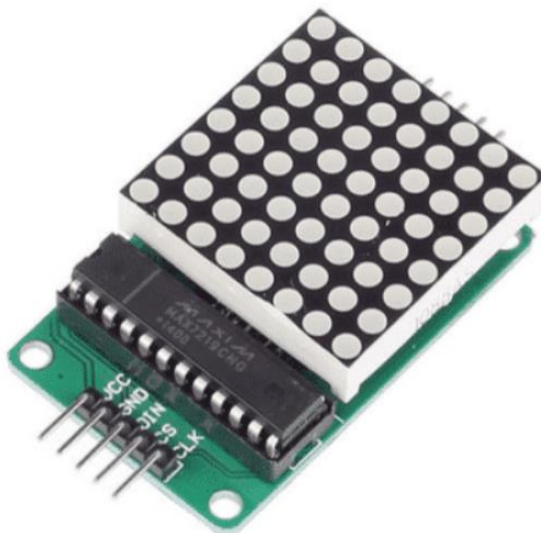
- Conexão de periféricos

Existem diversos periféricos que podem ser conectados ao microcontrolador para ampliar suas funcionalidades. A maioria destes periféricos se conecta de forma paralela, utilizando vários pinos do microcontrolador ou de forma serial, usando uma das portas de comunicação, I2C, SPI USART etc. Um possível projeto consiste em desenvolver o programa para conectar algum destes periféricos, como os exemplos a seguir.

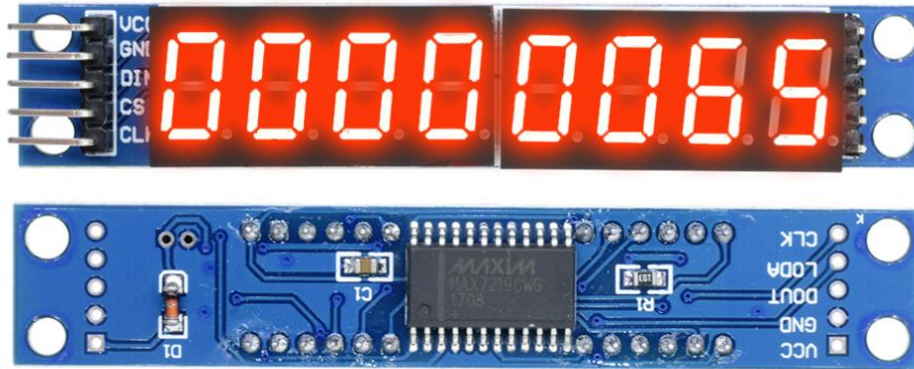
Display I2C com PCF8574:



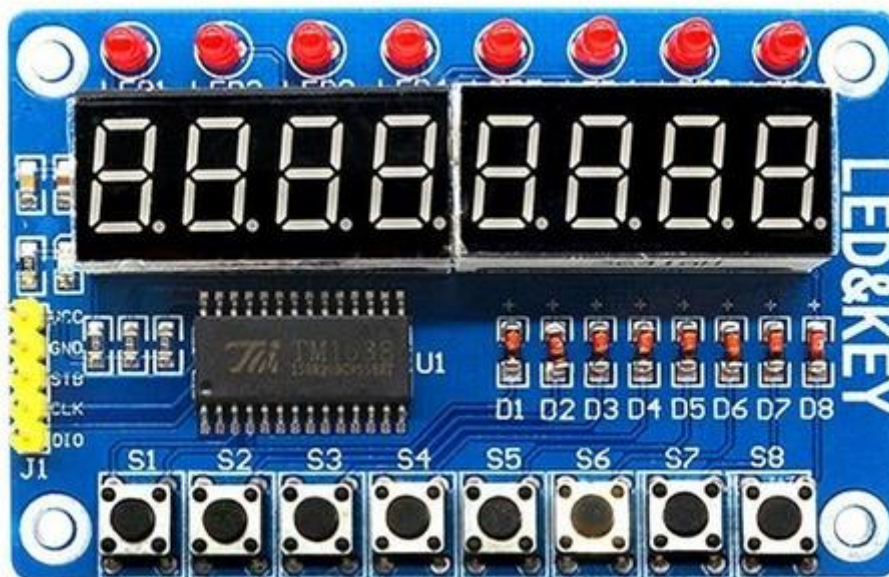
Matrix de LEDs com MAX7219:



Displays de LED com MAX7219:



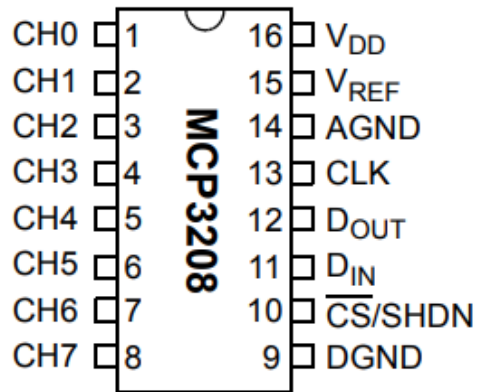
Módulo com displays botões e LEDs com Tm1638:



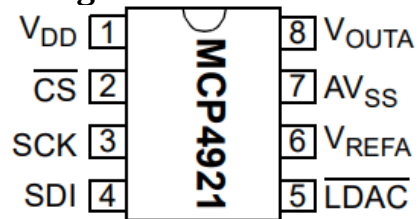
Teclado matricial:



Conversor Analógico/Digital de 12 bits e 8 entradas MCP3208:



Conversor Digital/Analógico de 12 bits MCP4921:



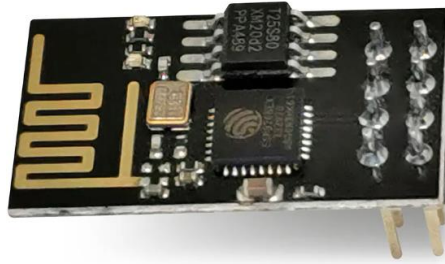
Display LCD gráfico 128x64: (Difícil)



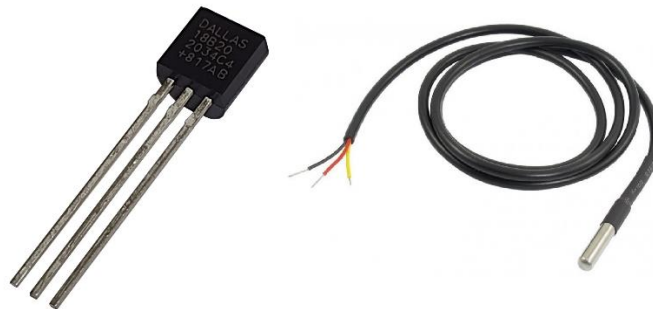
Encoder rotatório com chave:



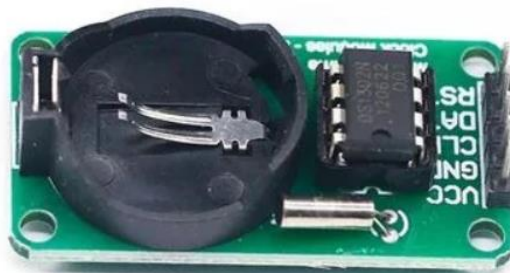
Conexão a Internet com ESP01:



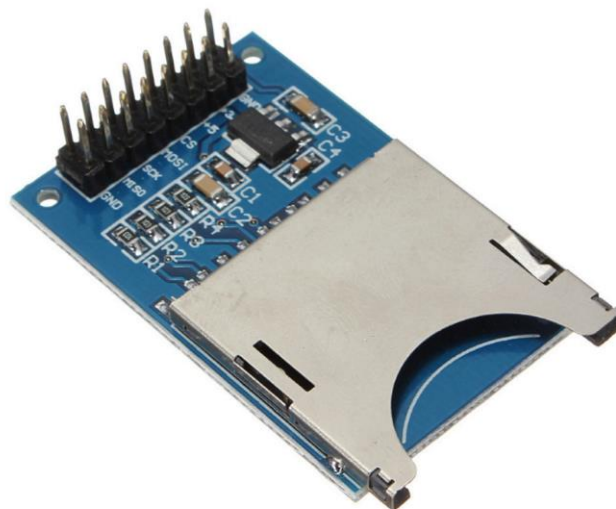
Sensor de temperatura DS18B20:



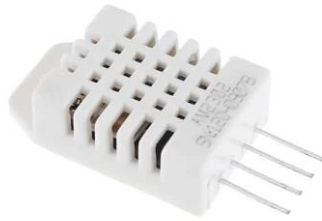
Relógio de tempo real DS1302 / DS1307:



Cartão de memória SD: (Difícil, requer ATmega328)



Sensor de Umidade e Temperatura DHT22:



OBS: é possível associar 2 ou mais periféricos para compor um projeto mais interessante, como por exemplo, um sensor e um display que apresenta o valor do sensor.

- Controlador de temperatura

Um possível experimento é um controlador de temperatura, neste experimento são utilizadas duas entradas analógicas do microcontrolador, uma conectada a um potenciômetro e outra conectada a um sensor de temperatura. A entrada do potenciômetro serve como referência de temperatura, e é comparada com o valor do sensor para determinar o comportamento do controlador. Para ajustar a temperatura a ser controlada é utilizada uma saída PWM.

Para manter a temperatura controlada é necessário utilizar um software controlador. O trecho de programa a seguir apresenta um exemplo de controlador PID simplificado.

```
// definições controlador PID
#define K_P    2.0      // Ganho proporcional
#define K_I    0.1      // Ganho integral
#define K_D    1.0      // Ganho derivativo
#define limite_I 10000.0 //limitador para o erro do integrador + ou -

// variáveis globais controlador PID
float erroAnt;        // erro anterior
float somaErro;       // somatório dos erros

void inicializaPID()
{
    // inicializa as variáveis do PID
    erroAnt = 0;
    somaErro = 0;
}

int controladorPID(/*valor desejado*/int valDes, /*valor atual do processo*/int valProc)
{
    // variáveis locais
    float erroAtual;
    float termoP;
    float termoD;
    float termoI;
    float resultado;
```

```

// calcula o erro
erroAtual = valDes - valProc;

// calcula o termo P
termoP = K_P * erroAtual;

// calcula o termo I
somaErro = somaErro + erroAtual;
if(somaErro > limite_I) somaErro = limite_I; // limita o máximo
if(somaErro < -limite_I) somaErro = -limite_I; // limita o mínimo
termoI = somaErro*K_I;

// calcula o termo D
termoD = (erroAnt-erroAtual)*K_D;
erroAnt=erroAtual;

// soma os termos P+I+D, e limita para caner em um int

resultado=termoP + termoI + termoD;
if(resultado > 32767) resultado = 32767; // limita o máximo
if(resultado < -32767) resultado = -32767; // limita o mínimo

// retorna o resultado
return(resultado);
}

```

Este controlador recebe como entradas o valor atual do processo (valProc) e o valor desejado (valDes) para o processo, e retorna um valor inteiro proporcional ao valor de atuação que se deve efetuar no processo.

As constantes K_P, K_I e K_D servem para ajustar este controlador. Um exemplo de utilização é apresentado a seguir.

```

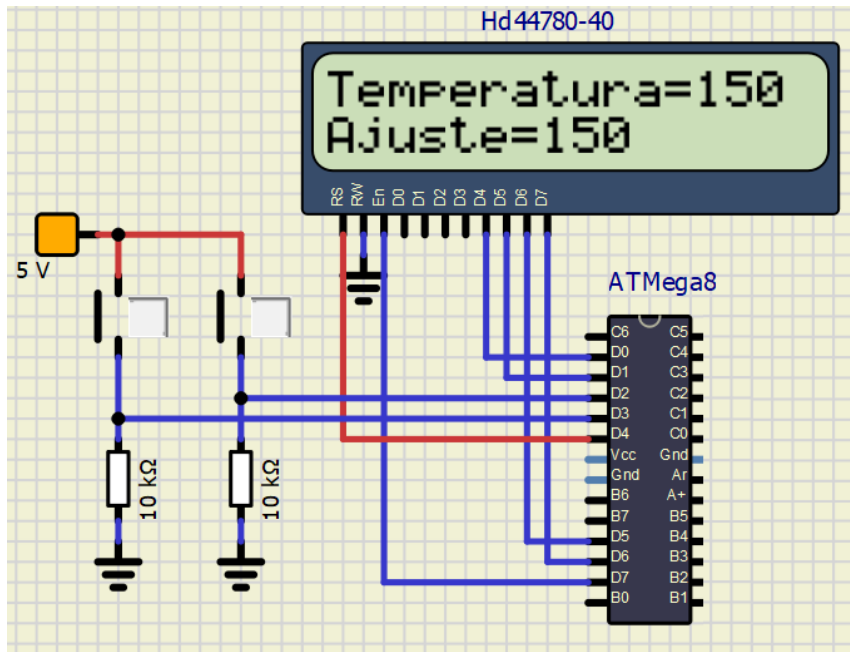
...
int main(void)
{
    ...
    int entrada;
    int desejado;
    int saida;
    inicializaPID();
    ...
    while (1)
    {
        ...
        saida = controladorPID(desejado,entrada);
        ...
        _delay_ms(100); //aguarda um tempo
    }
}

```

- Controlador de temperatura com display

Trata-se do mesmo controlador do circuito anterior, porém para este projeto o ajuste da temperatura deseja não é feito através de um potenciômetro. O ajuste é feito através de dois botões, um para aumentar e outro para diminuir.

Para este projeto é necessário algum tipo de display para apresentar o valor ajustado nos botões. A figura a seguir apresenta um exemplo de circuito de display para este projeto.



Para a utilização deste tipo de display é mais prático utilizar bibliotecas com as funções necessárias. No exemplo da figura foi utilizada a biblioteca "TextLCD.h" disponível no link: <https://github.com/RicardoKers/Text-LCD-for-AVR> A seguir temos um pequeno exemplo de utilização desta biblioteca.

```
#include <avr/io.h>
#define F_CPU 1000000UL // 1 MHz (for delay.h)
#include <util/delay.h>
#include "TextLCD.h"

int main(void)
{
    unsigned int y=0;
    LCD_init(); // initializes lcd 20x4 (TextLCD.h)

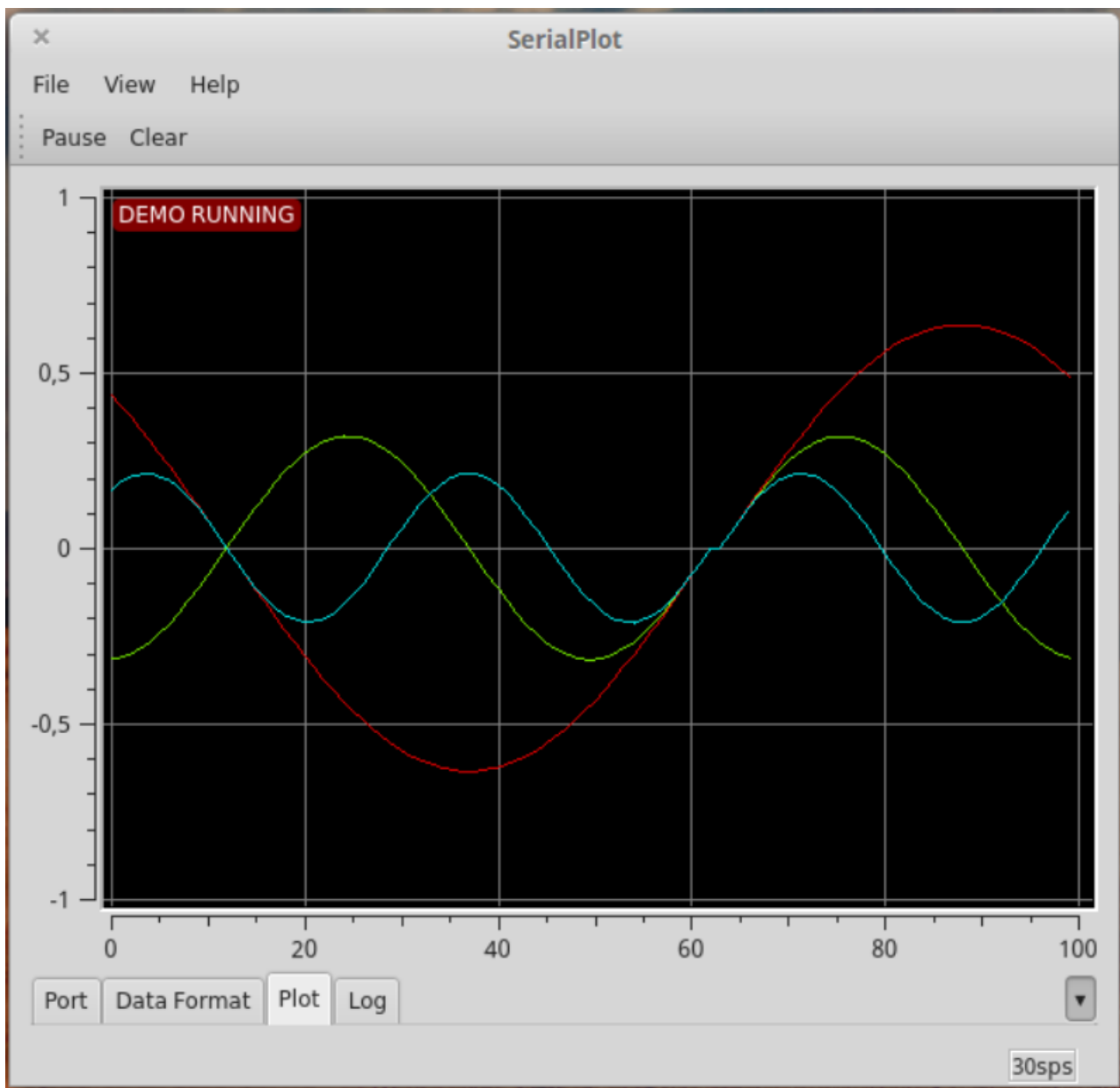
    while(1)
    {
        gotoxy(2,1);
        putstr("y = ");
        putuint(y);
        putstr("    ");
        y++;
        _delay_ms(500);
    }
}
```

- Transmissão de dados através do conversor USB-Serial

Um último exemplo de experimento consiste em ler as informações da entrada analógica, formatar dos dados de forma apropriada e transmiti-los através da porta serial para um conversor USB-serial como este na figura.



Os dados deverão ser recebidos no computador (através da USB) pelo software “serialplot” (<https://hackaday.io/project/5334-serialplot-realtime-plotting-software>). Este software permite apresentar na forma de texto ou de gráficos os dados recebidos pela USB. A seguir temos uma imagem deste software.



Relatório

Após a realização dos experimentos deve ser elaborado um relatório seguindo o modelo disponibilizado. Este relatório deve ser submetido via sistema SIGAA para avaliação, em formato PDF, até a data estipulada em aula.

O modelo do relatório pode ser encontrado em: <https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/microcontroladores-experimental/>

Serão avaliados os seguintes itens no relatório:

- Introdução
- Objetivo
- Fundamentação teórica
- Desenvolvimento
- Componentes utilizados
- Circuito eletrônico
- Código fonte do programa
- Resultados e discussões
- Conclusão