



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica
Laboratório de Arquitetura de Sistemas Digitais

Documentação do Hardware: Gerenciador de Rendimento de Estudo

Sara Andrade Dias - 116110719
Saulo Afonso Sobreira Lima - 115210618

Campina Grande, PB
Dezembro de 2020

Sumário

1	Atmega 328P	2
2	LCD Pcd8544	4
3	EEPROM	4
4	LedBar	5
5	ResistorDrip	7
6	Demultiplexador (DEMUX)	8
7	Push	9
8	Montagem dos Componentes	9

1 Atmega 328P

Um microcontrolador é um circuito integrado compacto projetado para governar uma operação específica em um sistema embarcado. Um microcontrolador típico inclui um processador, memória e periféricos de entrada/saída (IN/OUT) em um único chip. Para a aplicação sugerida, foi utilizado o microcontrolador atmega 328P, que, além da grande diversidade de aplicações e facilidade de programação, apresenta as seguintes características:

- Baseado em um processador AVR RISC (Reduced Instruction Set Computing).
- Arquitetura Harvard (memórias de programa e dados independentes).
- 32kB de memória flash para armazenamento de programas.
- 2kB de memória RAM estática para armazenamento de dados.
- 1kB EEPROM para armazenamento não-volátil.
- 23 linhas de entrada/saída de propósito geral (GPIO).
- 32 registradores de propósito geral.
- 3 temporizadores/contadores.
- USART (Universal Synchronous/Asynchronous Receiver Transmitter).
- Porta serial TWI (2-wire Serial Interface).
- Porta serial SPI (Serial Peripheral Interface).
- 6 canais de 10 bits para conversão A/D.

Figura 1: ATmega328P: Arquitetura

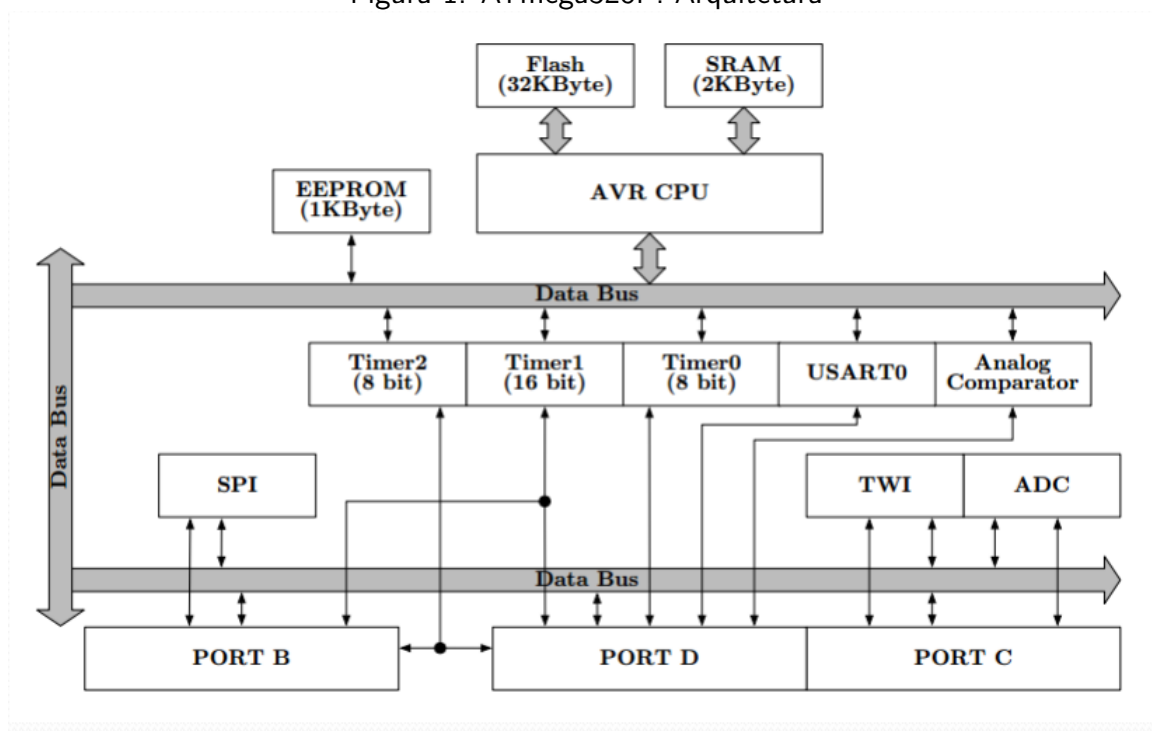


Figura 2: CPU AVR

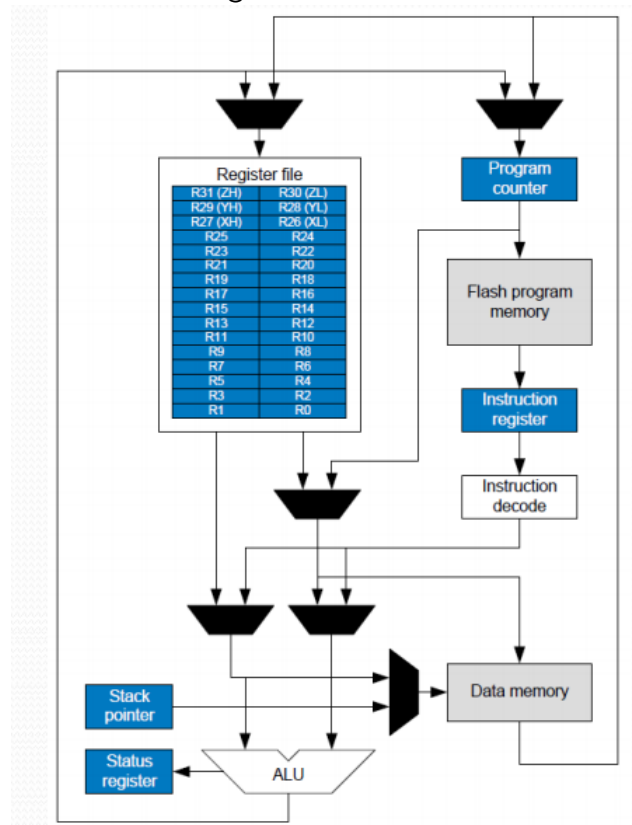
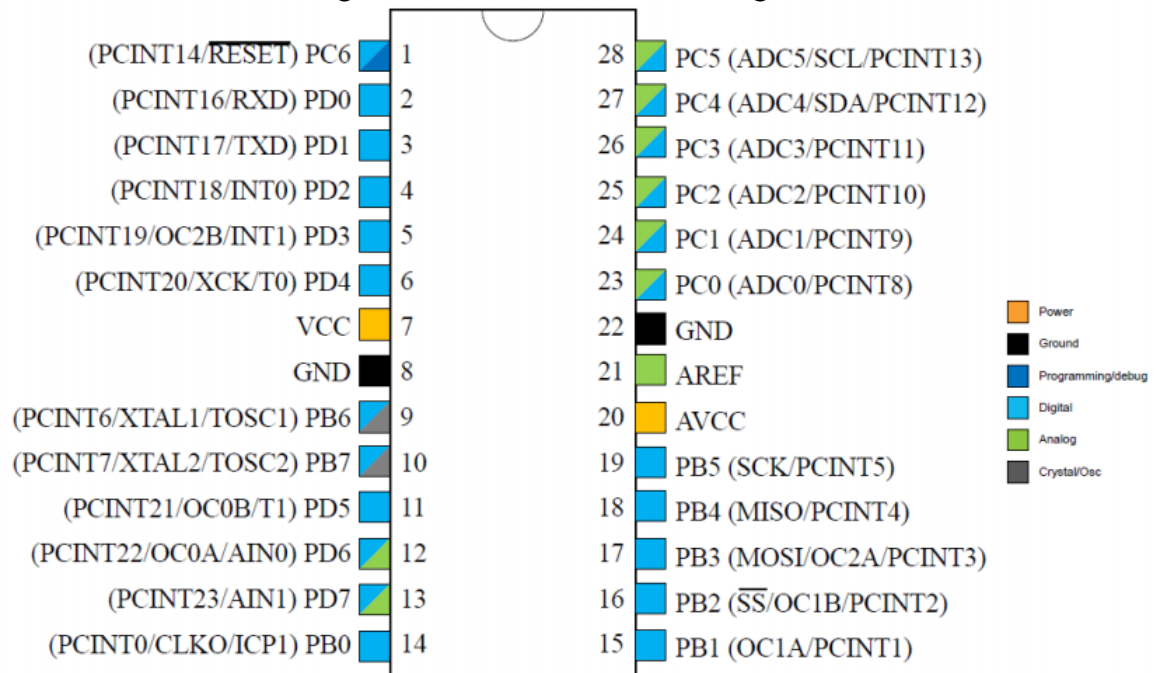


Figura 3: Microcontrolador ATmega328P

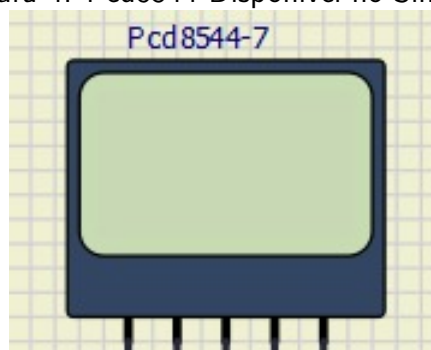


2 LCD Pcd8544

O Pcd8544 é um CMOS LCD *driver* de baixa potência, projetado para conduzir uma exibição gráfica de 48 linhas e 84 colunas. Todas as funções necessárias para o display são fornecidas em um único chip, incluindo geração on-chip de fornecimento de LCD e tensões de polarização, resultando em um mínimo de componentes externos e baixo consumo de energia. O PCD8544 faz interface com microcontroladores por meio de um interface de barramento serial.

No projeto foi utilizado o LCD Pcd8544 disponível na plataforma do Simulide com a função de exibir orientações e informações ao usuários do dispositivo.

Figura 4: Pcd8544 Disponível no Simulide



3 EEPROM

A memória EEPROM é utilizada com dados que podem ser alterados e devem estar disponíveis para o programa, ou seja, não podem ser perdidos. A EEPROM é como um pendrive do microcontrolador. Pode ser utilizada, por exemplo, para a armazenagem de senhas, variáveis lidas pelo conversor AD, valor de tempo configurado para alarmes ou quaisquer valores que devam ser salvos. Em resumo, a EEPROM armazena dados para futura leitura pelo programa.

A EEPROM é uma memória de escrita lenta, o que deve ser levado em conta quando for utilizada. O tempo de escrita de um dado é de aproximadamente 3,3 ms, o que pode ser longo demais para certas aplicações. Caso esse tempo seja crítico, pode-se empregar a interrupção da EEPROM para avisar quando uma escrita foi completada, liberando o programa desse monitoramento.

O ATmega328 possui 1 kbyte de memória EEPROM, a qual é organizada em um espaço de memória separado, em que bytes individuais podem ser lidos e escritos. A EEPROM suporta pelo menos 100 mil ciclos de escrita e apagamento.

A comunicação entre a EEPROM e a CPU é feita pelos registradores de endereço, de dados e de controle da EEPROM. De acordo com os registradores e bits de trabalho da EEPROM, os seguintes passos são necessários para efetuar-se uma escrita (a ordem dos passos 2 e 3 não é importante). Esses passos são necessários para se evitar um eventual escrita acidental.

1. Esperar até que o bit EEPE do registrador EECR se torne zero.
2. Escrever o novo endereço da EEPROM no registrador EEAR (opcional).
3. Escrever o dado a ser gravado no registrador EEDR (opcional).
4. Escrever 1 lógico no bit EEMPE enquanto escreve 0 no bit EEPE do registrador EECR.

5. Dentro de quatro ciclos de clock após ativar EEMPE, escrever 1 lógico no bit EEPE.

Código em C:

```
void EEPROM_escrita(unsigned int uiEndereco, unsigned char ucDado)
{
    while(EECR & (1<<EEPE)); //espera completar um escrita prévia
    EEAR = uiEndereco;        //carrega o endereço para a escrita
    EEDR = ucDado;            //carrega o dado a ser escrito
    EECR |= (1<<EEMPE);       //escreve um lógico em EEMPE
    EECR |= (1<<EEPE);        //inicia a escrita ativando EEPE
}

unsigned char EEPROM_leitura(unsigned int uiEndereco)
{
    while(EECR & (1<<EEPE)); //espera completar um escrita prévia
    EEAR = uiEndereco;        //escreve o endereço de leitura
    EECR |= (1<<EERE);        //inicia a leitura ativando EERE
    return EEDR;              //retorna o valor lido do registrador de
                              //dados
}
```

Para a programação em C da EEPROM no AVR Studio, basta incluir no arquivo fonte a biblioteca `<avr/eeprom.h>` do AVR-GCC, a qual possui várias funções para o trabalho com a EEPROM. Uma importante questão na gravação de dados, é que muitas vezes eles devem ser definidos previamente à gravação da memória de programa (inicialização da EEPROM). Na aplicação, a memória EEPROM foi utilizada para armazenar dados de tempo das tarefas realizadas pelo usuário.

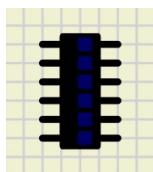
4 LedBar

O LED (Light Emitting Diode) é um tipo especial de diodo que tem a capacidade de emitir luz quando é polarizado diretamente, ou seja quando é percorrido por uma corrente. Este dispositivo tem o mesmo comportamento do diodo simples, permitindo a passagem de corrente no sentido direto ânodo-cátodo (neste caso o LED acende) e bloqueando a passagem de corrente no sentido inverso catodo-anodo (neste caso o LED não acende).

Figura 5: LEDS



Figura 6: LEDbar do Simulide



Na Tabela abaixo, observa-se as propriedades de alguns LEDs em função do tipo e da cor da luz emitida.

Tipo	Cor	I_F max	V_F typ	V_F max	V_R MAM	Intensidade luminosa	Ângulo de visu- alização	Comprimento de onda
Padrão	Vermelho	30mA	1,7V	2,1V	5V	5mcd @ 10mA	60°	660nm
Padrão	Brilhante vermelho	30mA	2,0V	2,5V	5V	80mcd @ 10mA	60°	625nm
Padrão	Amarelo	30mA	2,1V	2,5V	5V	32mcd @ 10mA	60°	590nm
Padrão	Verde	25mA	2,2V	2,5V	5V	32mcd @ 10mA	60°	565nm
Alta inten- sidade	Azul	30mA	4,5V	5,5V	5V	60mcd @ 20mA	50°	430nm
Super brilho	Vermelho	30mA	1,85V	2,5V	5V	500mcd @ 20mA	60°	660nm
Baixa cor- rente	Vermelho	30mA	1,7V	2,0V	5V	5mcd @ 2mA	60°	625nm

Figura 7: Descrição dos LEDS no Simulide

LedBar-19

itemtype

LedBar

id

LedBar-6

Show id

false

Color

yellow

Size

6

Threshold

2,4

MaxCurrent

0,03

Resistance

0,6

Grounded

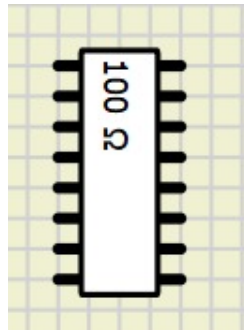
false

Foram utilizados 6 LEDS em formato de barra no desenvolvimento do projeto. Cada LED da barra simboliza 1 hora e a intensidade do brilho é proporcional aos minutos. Variando proporcionalmente da intensidade 0% para 0 minutos até 100% a partir de 49 minutos.

5 ResistorDrip

Um resistor é um componente elétrico passivo de dois terminais que implementa a resistência elétrica como um elemento de circuito. Em circuitos eletrônicos, resistores são usados para reduzir o fluxo de corrente, ajustar níveis de sinal, dividir tensões, polarizar elementos ativos e encerrar linhas de transmissão, entre outros usos.

Figura 8: Resistor de 100 ohm usado no Simulide



Na aplicação, foi utilizado um resistor de 100 Ω para controlar a corrente que chega na barra de LEDS. Por se tratar de um simulador, aplicou-se um resistor virtual e não existia risco do aparelho ser danificado. Entretanto, na realização do projeto físico, usaria-se o seguinte resistor.

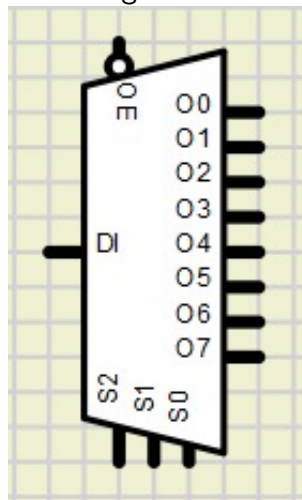
Figura 9: Resistor de 100 ohm



6 Demultiplexador (DEMUX)

O demultiplexador é um circuito lógico combinacional projetado para alternar uma linha de entrada comum para uma das várias linhas de saída separadas. Ele pega uma única linha de dados de entrada e a troca para qualquer uma de várias linhas de saída individuais, uma de cada vez. O demultiplexador converte um sinal de dados seriais na entrada em dados paralelos em suas linhas de saída, conforme mostrado abaixo.

Figura 10: Simbologia do demux no Simulide



No projeto, o demux possui uma entrada PWM e 6 saídas (1 saída para cada LED). O direcionamento da saída é feito por meio do chaveamento dos valores das chaves S0, S1 e S2, como representado na tabela a seguir.

S0	S1	S2	Led
0	0	0	LED 0
0	0	1	LED 1
0	1	0	LED 2
0	1	1	LED 3
1	0	0	LED 4
1	0	1	LED 5

7 Push

Um *pushbutton* (botão de pressão) é uma chave que contém um botão que ao ser pressionado abre ou fecha os contatos do dispositivo, abrindo ou fechando o circuito onde ele está conectado.

É comum que um *pushbutton* possua ação de contato momentânea, o que significa que a conexão é aberta ou fechada apenas momentaneamente, enquanto o botão estiver sendo pressionado. No projeto, os botões são os responsáveis pelas interrupções externas. Ou seja, quando o usuário pressiona algum botão uma interrupção é acionada e uma ação é executada logo em seguida.

O *pushbutton* usado é do tipo NOPB (*Normally Open Pushbutton*) ou normalmente aberto. É um interruptor que fecha o circuito e deixa a corrente passar quando pressionado e sua simbologia é apresentada da seguinte forma.

Figura 11: *Pushbutton* NOPB

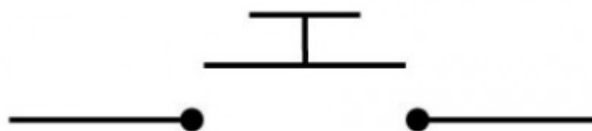
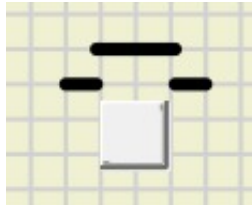


Figura 12: *Pushbutton* no Simulide



8 Montagem dos Componentes

Por meio de todos os recursos apresentados nas seções anteriores criou-se o dispositivo Gerenciador de Rendimento de Estudo como projeto para finalização da disciplina do Laboratório de Arquitetura de Sistemas Digitais. Os detalhes e instruções do funcionamento estão dispostos no documento Manual de Uso.

Figura 13: Diagrama do Gerenciador de Rendimento de Estudo

