



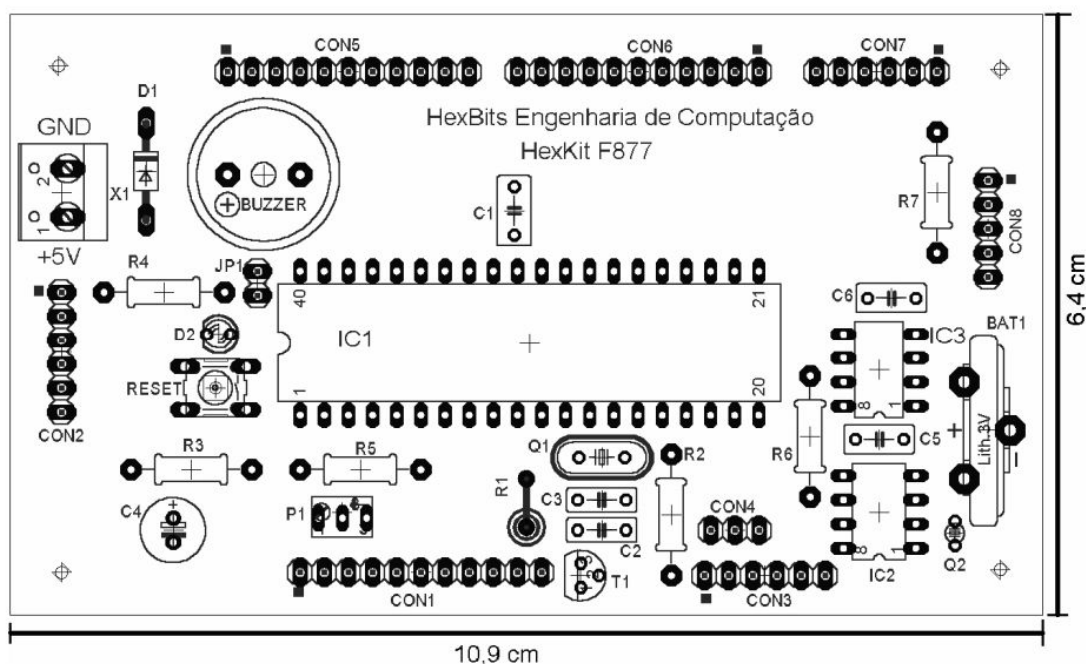
Centro de Informática
U F P A

PROJETO INTEGRADO : PLUVIÔMETRO

Thales Monteiro Soares

João Pessoa – PB

2019



O chip de memória eeprom 24LC256 é uma memória de 256 kbit com organização de 32k x 8, com tensão de operação de 2.5V até 5.5V (para mais informação consultar referência).

O RTC é um relógio de tempo real, cujo fornecimento dos segundos, minutos, horas, dias, mês e ano é feita em BCD (consultar referência).

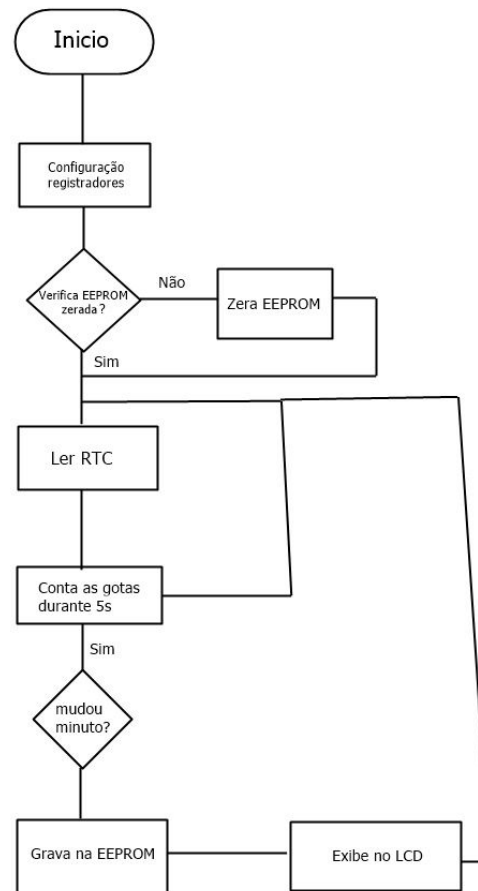
ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

Exemplo: se for feita a leitura do RTC no endereço 01h e o valor atual for 19 ele retornará 0001 1001, 1 e 9 representados em BCD.

3. O CÓDIGO

O fluxograma simplificado do código é apresentado a seguir:



- **Configuração dos registradores:**

A porta RA0 do PORTA é utilizado como entrada do sinal do pluviômetro e essa porta foi definida como entrada do sinal do comparador. O vrcon foi iniciado com o valor 9, que corresponde ao nível de tensão 1.8.

Aqui também é definido a velocidade de comunicação do I2C através do registrador SSPADD onde o valor 49 define a velocidade 100khz. O I2C é iniciado com o pic sendo master no SSPCON.

- **Verifica EEPROM zerada**

É utilizada a posição de memória 0x7FFE como indicador, se a posição estiver com o valor 0 é pq já foi zerada, caso contrário é zerado essa posição e é iniciado o processo de zerar a eeprom.

O processo de zerar a eeprom é bastante simples, é iniciado o Endereço low e Endereço high em 0 e é escrito via i2c o valor 0, a posição é incrementada em

uma unidade após cada interação. Quando o endereço chega no valor de 3FF é encerrada o processo, isso é feito para evitar que zeremos toda a eeprom sendo que não utilizaremos toda essas posições.

- **Ler RTC**

Para ler o RTC é utilizado os seguintes endereços:

- 0x0 : Segundos
- 0x1 : Minutos
- 0x2 : Horas
- 0x3 : Dia da semana
- 0x4 : Data
- 0x5 : Mês
- 0x6 : Ano

Nesta etapa é lida o valor do minuto, hora, data, mês e é salva em duas variáveis `minuto` e `minuto_anterior`, por exemplo, isso é feito para permitir o conhecimento da mudança de minuto para realizar a gravação do valor medido na posição de memória específica.

Como o valor dado pelo RTC é dada em BCD, precisamos separar esses valores, para isso é utilizada a função `SEPARA_RTC`.

A `separa_rtc` pega o valor lido do rtc e separa os dois nibbles, seleciona o nibble que representa a dezena, multiplica por 10 e soma com o nibble que representa a unidade.

Ex: Foi lido o valor 15 no RTC a função separa colocando 00000001 na dezena e 00000101 na unidade.

$00000001 * 10$ e é somada com 00000101 isso nos dá o valor 15 em binário.

Neste trecho foi utilizado a função de multiplicação retirada do livro Conectando o Pic PIC16F877A.

É necessário que o valor lido do RTC esteja em binário pois ele é utilizado no cálculo do endereçamento da eeprom.

- **Conta Gota**

A rotina conta gota utiliza o timer 1 para realizar a contagem de 5s, ou seja, passamos 5s contando o número de gotas.

Para contar 5s no timer1 foi esperado o timer estourar 10 vezes, sendo utilizado o prescaler de 1:8, contando 0,5s e para isso foi preciso iniciar o timer em 3036.

- **Mudou minuto?**

Esta rotina verifica se o minuto lido é diferente do minuto anterior, se for é iniciado a gravação do dado coletado no último minuto na posição de memória referente ao último minuto.

- **Gravação na EEPROM**

A eeprom foi separada em blocos de memória referentes ao minuto, hora, dia e mês, sendo separada dois bytes para cada valor medido.

120 bytes foram reservados para guardar informações de chuva referente aos minutos de 0 a 59.

48 bytes reservados para cada hora do dia e 732 bytes referentes a cada dia do ano.

Foi feita uma expressão matemática para calcular o endereço com base no valor lido. Quando é pedido para ler os minutos é retornado um valor entre 0 e 59, como cada minuto precisa ter dois bytes a equação que gera o endereçamento dos minutos é : **MINUTO * 2**.

Ex: foi lido o minuto 40, o endereço onde ficará guardado o número de gotas neste minuto é a posição de endereço 80 em decimal. Quando é pedido para ler a eeprom na posição 80 em decimal ou 50 em hexa é retornado os dois bytes.

[illegible]

O RTC retorna valores referentes a hora entre 0 e 23, e a expressão matemática que calcula o endereço é a seguinte:

$$. \text{HORA} * 2 + 120$$

Exemplo: Se for lido Hora 1, $1 * 2 + 120$, seria na posição 122, ou 7A em hexadecimal

I2C Memory Internal Memory - U2

0000	06 22 07 42	06 A9 07 54	07 5A 07 C1	07 29 07 6F	05 DA 06 41	07 40 06 F2	07 59 07 C1	07 58 07 72
0020	07 D9 07 2C	06 8A 06 F1	06 59 07 47	07 0A 07 71	08 18 07 22	07 89 06 F1	07 33 06 A2	07 09 07 04
0040	06 BA 07 21	07 89 08 1F	07 3A 07 A1	06 F0 06 52	06 B9 04 F2	00 00 00 00	00 00 00 00	00 00 00 00
0060	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 47 00	00 00 59 AE	00 00 A4 73
0080	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 E4 86	25 68 02 D6	3C E4 2B 7A	27 E4 A1 F0
00A0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00E0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0100	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0120	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0140	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

Para a definição do dia entre os 720 bytes foi utilizada a seguinte expressão.

$$\text{DIA} * 2 + 166 + 62(\text{MES} - 1)$$

Já é gravado na posição de memória referente ao dia do mês corrente.

Exemplo: Se for lido Dia 3 e Mês 4 o valor do endereço é 420 em decimal ou 1A4.

I2C Memory Internal Memory - U2

0000	06 22 07 42	06 A9 07 54	07 5A 07 C1	07 29 07 6F	05 DA 06 41	07 40 06 F2	07 59 07 C1	07 58 07 72
0020	07 D9 07 2C	06 8A 06 F1	06 59 07 47	07 0A 07 71	08 18 07 22	07 89 06 F1	07 33 06 A2	07 09 07 04
0040	06 BA 07 21	07 89 08 1F	07 3A 07 A1	06 F0 06 52	06 B9 04 F2	00 00 00 00	00 00 00 00	00 00 00 00
0060	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0080	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00A0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00E0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0100	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0120	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0140	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0160	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0180	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
01A0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
01C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
01E0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0200	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0220	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0240	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

Relembrando que todos os valores de minuto, hora, dia e mês são em decimal, após serem passados pela função `separa_rtc`.

- **Exibe LCD**

Foi utilizado um display 16x2, onde a comunicação foi feita em 8 vias, utilizando todo o PORTD como via de dados e RS em RB1 e ENABLE em RB0.

Para exibir o número lido na eeprom em hexadecimal no LCD foi preciso fazer um processamento com o dado.

Vamos supor que tenhamos lido o valor 007F da eeprom, para exibir este valor cada dígito do dado é separado e verificado se o dígito separado é menor que 10 ou não. Caso o número seja menor que 10 (esteja entre 0 e 9) é somado o valor 0x30 ao dígito que cai exatamente na sua representação na tabela ASCII. Caso o número seja maior que 10 (entre 10 e 15) é preciso somar com 0x37 que cai na tabela ascii nas letras A,B,C,D,E,F.

4. PROTOCOLO I2C

O pic 16F877A já possui o protocolo I2C implementado, sendo necessário apenas a manipulação de registradores.

A rotina que realiza a leitura da eeprom externa funciona da seguinte forma: start bit, envia endereço da eeprom (1010000) e comando de escrita , envia endereço high, envia endereço low, envia restart bit, envia comando de leitura, inicia leitura do byte1 , inicia leitura do byte2, stop bit.

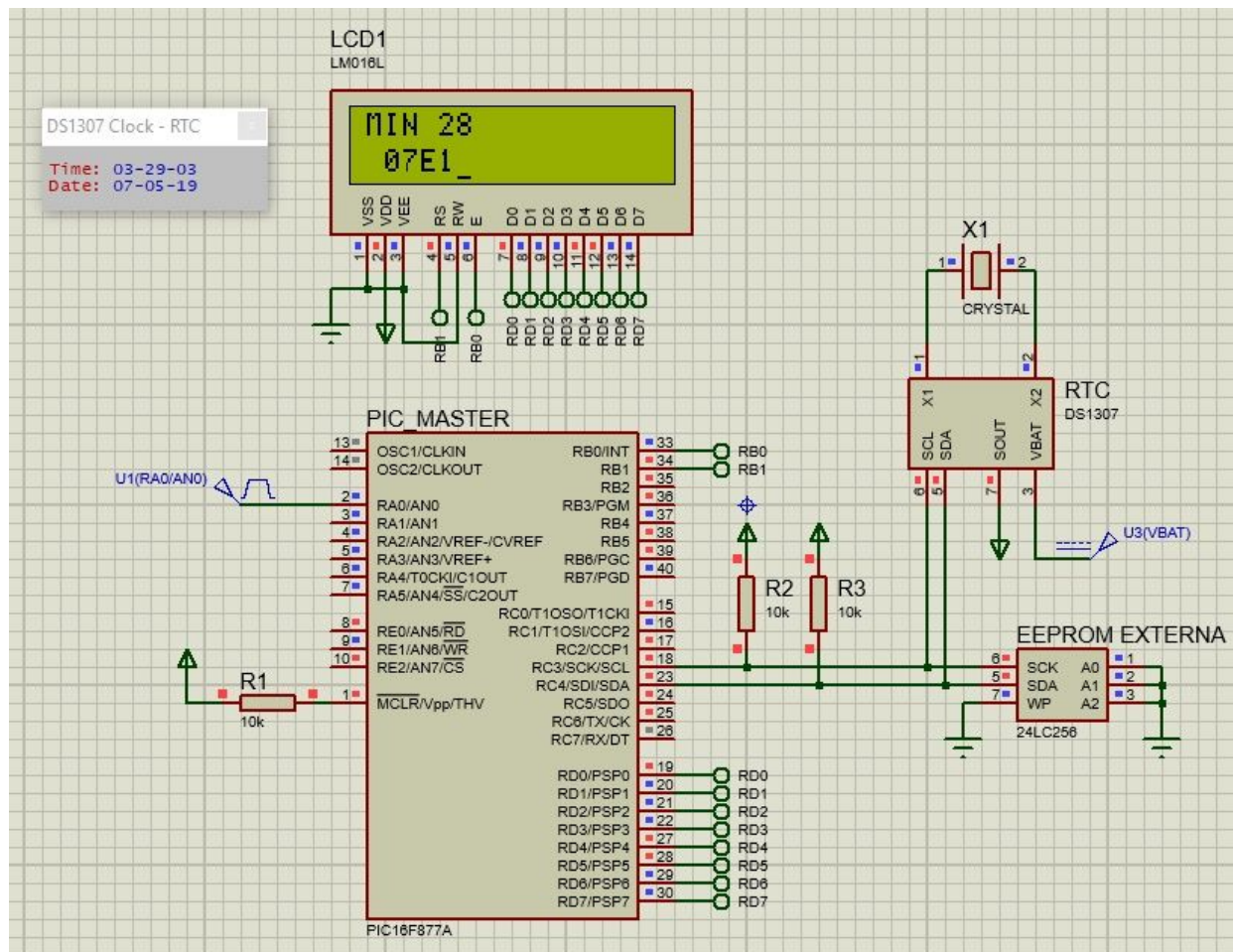
Para a escrita na eeprom é feito o start bit, endereço da eeprom com comando de escrita, envio do endereço high, envio do endereço low, envia parte alta do dado a ser gravado, envia parte baixa do dado, stop bit.

As rotinas que fazem a leitura e gravação do rtc são similares, exceto que a de leitura só ler um byte.

5. RESULTADOS

Foram executadas algumas simulações no software Proteus 8.5, onde apresentou correto funcionamento. Foi posto na simulação o RTC 1307, a EEPROM 24LC256, o barramento I2C com os resistores de pull up, o oscilador externo do rtc. O oscilador externo do PIC foi configurado para 20 Mhz, sem precisar instanciar um oscilador externo.

No LCD é exibido um pequeno menu contendo o nome MIN e o minuto anterior, logo embaixo aparece o número de gotas em hexadecimal medidos naquele minuto. A exibição é sempre feita mostrando a quantidade de gotas contadas no minuto anterior ao minuto lido no RTC.



A cada minuto o display LCD mostra o minuto anterior e a quantidade de gotas medidas.

Para simulação foi utilizado um gerador de sinal para simular o pluviômetro físico, este sinal possui 10ms de largura de pulso, 2us de tempo de descida e subida e uma frequência de 4 Hz.

Os testes na placa não apresentaram total sucesso visto que o RTC apresentou problema, pois sempre retornava o valor 0 a qualquer valor solicitado. Como a aplicação depende do valor lido do RTC para sincronização com a eeprom a aplicação ficava esperando o valor do minuto mudar, o que nunca acontecia.

6 . REFERÊNCIAS

MICROCHIP. Datasheet: PIC16F87XA, 28/40/44-Pin Enhanced Flash Microcontrollers. Disponível em:

<http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>

MICROCHIP. Datasheet: 256K I2C CMOS Serial EEPROM

24AA256/24LC256/24FC256. Disponível em:

<http://ww1.microchip.com/downloads/en/DeviceDoc/24AA256-24LC256-24FC256-Datasheet-20001203V.pdf>

MAXIM INTEGRATED. Datasheet: DS1307 64 x 8, Serial, I2 C Real-Time Clock .

Disponível em: <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>