



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA - CT  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**Programador Horário Baseado no Microcontrolador ATMega328p  
ELE1717 - Grupo 01 - Problema 04 - Projeto**

Ana Beatriz Marinho Neves  
Anny Beatriz Pinheiro Fernandes  
Isaac de Lyra Junior  
Lucas Batista da Fonseca  
Wesley Brito da Silva

Natal, 27 de julho de 2021

# Resumo

O seguinte relatório visa o desenvolvimento do projeto de controle de um programador horário, utilizando do microcontrolador AVR ATMega328p, para posterior implementação em linguagem *assembly*. Além deste dispositivo, o projeto conta com a utilização de CIs auxiliares para o fornecimento de data/hora (CI PCF2123) e, também, para controle do display utilizado na interface homem-máquina proposta (CI MAX7219). Estes CIs contam com o protocolo de comunicação SPI para realizar o controle de dados com o ATMega328p. Para isso, foi apresentado o embasamento teórico a respeito dos dispositivos eletrônicos, do protocolo e de todo projeto desenvolvido em si.

**Palavras-chave:** Programador Horário, ATMega328p, PCF2123, MAX7219, comunicação SPI.

# Lista de Imagens

Figura 1 – Programador Horário Industrial . . . . .	4
Figura 2 – Interface Homem-Máquina . . . . .	5
Figura 3 – Legenda da Interface . . . . .	5
Figura 4 – Modos de Operação . . . . .	7
Figura 5 – Descrição da Chave Seletora . . . . .	8
Figura 6 – Processo de Ajuste do Relógio . . . . .	9
Figura 7 – Processo de Agendamento de Alarme . . . . .	9
Figura 8 – Pinagem ATmega328p . . . . .	10
Figura 9 – Diagrama de blocos do ATmega328p . . . . .	11
Figura 10 – Pinagem PCF2123 . . . . .	11
Figura 11 – Diagrama de Blocos de PCF2123 . . . . .	12
Figura 12 – Registradores de PCF2123 . . . . .	12
Figura 13 – Configuração dos pinos do MAX7219 . . . . .	13
Figura 14 – Diagrama do MAX7219 . . . . .	14
Figura 15 – MDE - fluxograma . . . . .	15
Figura 16 – Chave Seletora - fluxograma . . . . .	16
Figura 17 – Formato BCD para exemplo de minutos . . . . .	19
Figura 18 – Pinagem pensada para o MAX7219 . . . . .	20
Figura 19 – Demultiplexador dos leds da semana . . . . .	20
Figura 20 – Tabela 2 do datasheet do MAX7219. . . . .	21
Figura 21 – Tabela 3 do datasheet do MAX7219. . . . .	21
Figura 22 – Tabela 4 do datasheet do MAX7219. . . . .	22
Figura 23 – Tabela 8 do datasheet do MAX7219. . . . .	22
Figura 24 – Tabela 5 do <i>datasheet</i> do MAX7219. . . . .	23

# Sumário

1	INTRODUÇÃO	4
2	DESENVOLVIMENTO	6
2.1	Modos de Operação do Programador Horário	6
2.2	Chave Seletora - Funcionamento Esperado	8
2.3	Ajuste de Relógio - Funcionamento Esperado	8
2.4	Ajuste do Agendamento - Funcionamento Esperado	9
2.5	Microncontrolador AVR ATmega328p	10
2.6	CI RTC PCF2123	10
2.7	CI MAX7219	13
2.8	Protocolo de Comunicação SPI	13
3	PROJETO	15
3.1	Máquina de Estados de Alto Nível - fluxograma	15
3.2	Chave Seletora - Fluxograma	16
3.3	ATMEGA328P - Aplicação	16
3.3.1	Pinagem Utilizada	17
3.3.2	Protocolo de Comunicação SPI - Aplicação	17
3.3.3	Timer/Counter1	18
3.4	CI RTC PCF2123 - Aplicação	18
3.5	CI MAX7219 - Aplicação	19
3.5.1	Pinagem Utilizada	19
3.5.2	Configuração Necessária	20
	REFERÊNCIAS	24
	ANEXO A – RELATO SEMANAL	25
A.1	Equipe	25
A.2	Defina o problema	25
A.3	Registro de <i>brainstorming</i>	26
A.4	Pontos-chaves	26
A.5	Questões de pesquisa	26
A.6	Planejamento da pesquisa	27

# 1 INTRODUÇÃO

Por definição, o programador horário é um aparelho que permite ligar e desligar qualquer equipamento elétrico de acordo com horários pré-estabelecidos. Em sua forma mais básica, é composto de um disco de programação e os contatos de saída (Figura 1). Este tipo de equipamento são amplamente utilizados no âmbito industrial, comercial e até residencial. São exemplos de elementos controláveis, pelo programador horário, os aquecedores elétricos, os painéis comerciais, o sinal sonoro de entradas e saídas de funcionários em fábricas, irrigações, etc (COEL, 2001).

Figura 1 – Programador Horário Industrial



Fonte: COEL (2015).

Diante de sua alta taxa de utilidade em âmbito industrial e por atribuir à este o caráter de automação de processos, no problema 04 é proposto o desenvolvimento do projeto de um circuito - que utilize do microcontrolador AVR ATMEGA328P para implementar um programador horário (DIAS, 2021).

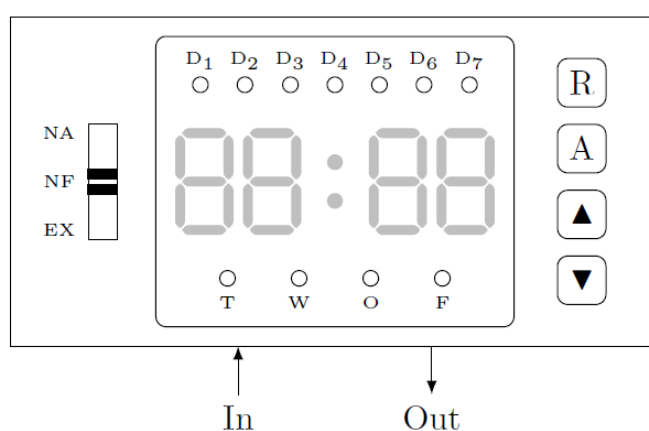
Algumas modificações foram realizadas ao projeto do equipamento original, com intuito de facilitar o entendimento e desenvolvimento do trabalho em questão. O projeto proposto irá controlar uma saída de acordo com um agendamento definido pelo usuário, além de também funcionar como um relógio normal. A interface homem-máquina (figura 2) proposta dispõe de:

1. Um *display* 7 segmentos de 4 dígitos, o qual irá apresentar os valores de hora e minutos (atuais ou a serem escolhidos em um agendamento);
2. O total de 11 LEDs que possuem a função de indicar tanto o dia atual da semana, quanto o modo de operação que se encontra o programador;
3. Quatro *pushbuttons* que, através deles, realiza-se a escolha do ajuste de relógio e de agendamento de um alarme;

4. Uma chave seletora que controla os valores atribuídos à saída de acordo com a data atual e o alarme programado.









A Figura 3 relaciona os elementos gráficos da Figura 2 com os itens explicativos citados acima. Além dessas alterações, o programador irá contar com 10 modos de operação e 3 modos de controle da saída (através da chave seletora). Com base nestas informações, o seguinte projeto desenvolvido neste relatório, visa definir todo o sistema e hardware a ser utilizado para que possa ser implementado o circuito em linguagem *assembly* a posterior.

Figura 2 – Interface Homem-Máquina



Fonte: Dias (2021).

Figura 3 – Legenda da Interface

Elemento	Descrição
	Display para exibição da hora
<div> <div>NA</div> <div>NF</div> <div>EX</div> </div> 	Chave seletora de três estados
<div> <div>D<sub>1</sub></div> <div>D<sub>2</sub></div> <div>D<sub>3</sub></div> <div>D<sub>4</sub></div> <div>D<sub>5</sub></div> <div>D<sub>6</sub></div> <div>D<sub>7</sub></div> </div> 	Painel de LEDs da sinalização do dia da semana
<div> <div>T</div> <div>W</div> <div>O</div> <div>F</div> </div> 	Painel de LEDs da sinalização dos modos de operação
	Botão para ajuste do relógio ( <i>Pushbutton</i> )
	Botão para ajuste da programação ( <i>Pushbutton</i> )
	Botão de incremento ( <i>Pushbutton</i> )
	Botão de decremento ( <i>Pushbutton</i> )

Fonte: Dias (2021).

## 2 Desenvolvimento

Diante da necessidade de controle de estados e funções do sistema do programador horário, o problema propõe a utilização do microcontrolador AVR ATMEGA328P. Dado que o projeto também necessita do fornecimento de data e hora, foi sugerido - pelo problema 04 - a utilização de um circuito integrado RTC (*Real Time Clock*) que utilize da comunicação SPI para resolver este aspecto.

Como decisão de projeto, o CI selecionado foi o PCF2123. Ainda a respeito das decisões iniciais de *hardware* a serem utilizados, foi verificada a necessidade da utilização de um CI que implementasse os valores - de data e hora - no *display* de 4 dígitos de forma correta e que utilizasse a menor quantidade de portas do ATMEGA328P, como forma de otimização de projeto. Diante disto, o CI selecionado foi o MAX7219.

Para melhor explicação do projeto que será desenvolvido, se faz necessário o entendimento a fundo dos modos de operações, da função da chave seletora, dos ajustes de relógio e de agendamento, além do conhecimento do funcionamento interno do ATMEGA328P e dos CI's que serão utilizados em complemento, juntamente com a compreensão do protocolo de comunicação SPI.

### 2.1 Modos de Operação do Programador Horário

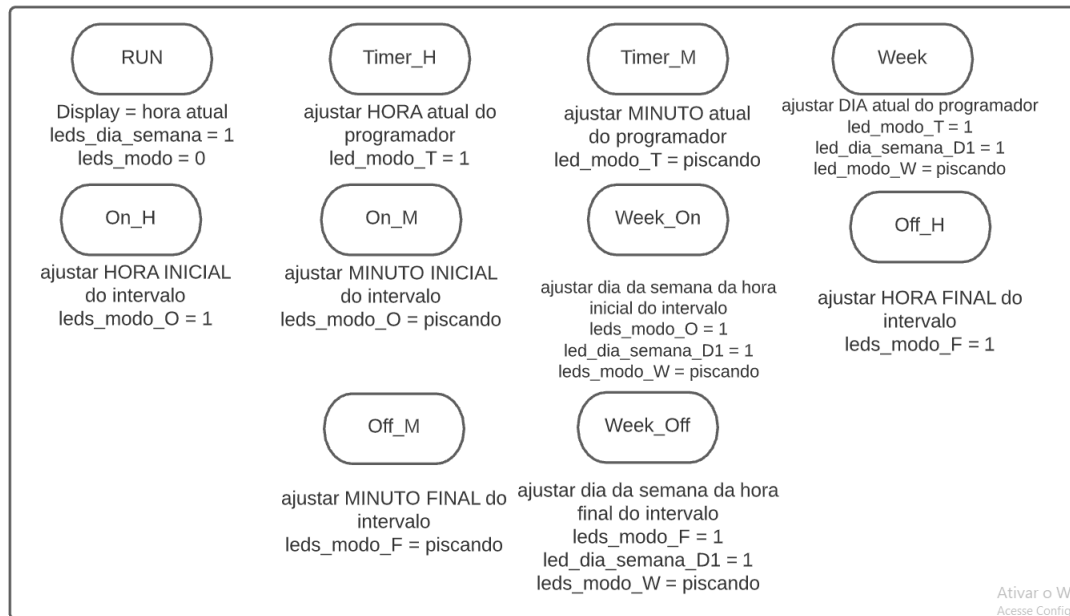
O funcionamento do programador horário proposto conta com 10 modos de operação diferentes, que em conjunto formam as etapas de ajuste de relógio e de ajuste do agendamento de um intervalo para alarme.

São eles: O modo RUN, Timer\_H, Timer\_M, Week, On\_H, On\_M, Week\_On, Off\_H, Off\_M e Week\_Off. Cada modo de operação conta com um conjunto de atribuição aos LEDS presentes na interface e, também, conta com uma sequência de instruções a serem realizadas.

A função de cada modo e o conjunto de atribuições relacionadas à eles está descrita abaixo e na Figura 4 é possível ver essa relação:

1. RUN: Estado base em que o funcionamento normal do programador é exibir a data e a hora atual. No display deve estar aparecendo a informação de HORA ATUAL, os LEDS de semana devem acender apenas aquele referente ao dia da semana atual e os LEDS de modo de operação (T, W, O e F) devem estar todos APAGADOS;
2. Timer\_H: Este modo permite que o usuário realize o ajuste da HORA atual do relógio. Neste modo, o LED T deve estar aceso e todos os outros apagados. O

Figura 4 – Modos de Operação



Fonte: Desenvolvido pelos Autores.

- display deve apresentar o valor atual e variar conforme indicado pelo usuário e por fim apresentar o valor de HORA escolhido;
3. **Timer\_M**: Com funcionamento semelhante ao **Timer\_H**, este modo permite o ajuste do valor de MINUTO ATUAL e apresenta o LED T piscando em uma frequência de 2Hz;
  4. **Week**: Último modo de operação para o ajuste do relógio, permite a escolha do DIA atual da semana. Esta escolha se dá por meio da variação entre os LEDS de dia (D1, D2, D3, D4, D5, D6, D7). O LED T permanece aceso neste modo, o LED W passa a piscar na frequência de 2Hz e os LEDS de dia iniciam com o D1 aceso e variam de acordo com a escolha do usuário;
  5. Modos ON (H, M e Week) permitem, respectivamente, o ajuste de hora, minuto e dia da semana em relação à data e hora de INÍCIO DO INTERVALO do agendamento:
    - a) **On\_H**: LED O aceso;
    - b) **On\_M**: LED O piscando em 2Hz;
    - c) **Week\_On**: LED O aceso, LED D1 inicia aceso e LED W piscando em 2Hz.
  6. Modos Off (H, M, e Week) permitem, respectivamente, o ajuste de hora, minuto e dia da semana em relação à data e hora de FIM DO INTERVALO do agendamento:
    - a) **Off\_H**: LED F aceso;
    - b) **Off\_M**: LED F piscando em 2Hz;
    - c) **Week\_Off**: LED F aceso, LED D1 inicia aceso e LED W piscando em 2Hz.

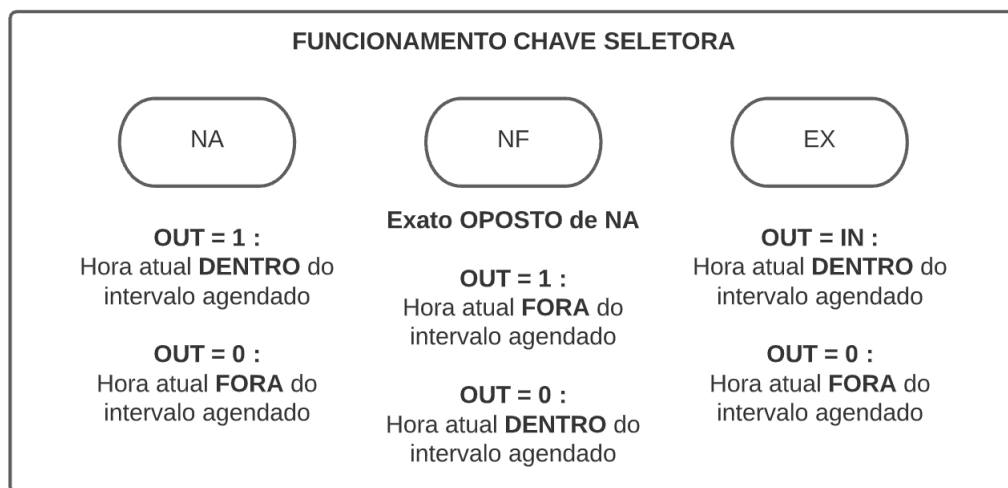


## 2.2 Chave Seletora - Funcionamento Esperado

O programado horário conta, também, com três possíveis seleções de controle do valor a ser inserido na porta de saída deste. Essa seleção é controlada através de uma chave seletora com 3 estados. São estes o NA, NF e EX (Figura 5):

1. NA: Estado em que a saída é 1 caso a hora atual esteja dentro do intervalo agendado ou 0, caso contrário;
2. NF: realiza o exato oposto de NA. A saída é 0 caso esteja dentro do intervalo agendado ou 1, caso contrário;
3. EX: Nesse estado a saída (OUT) passa a receber o valor da entrada (IN) caso a hora atual esteja dentro do intervalo ou recebe 0 no caso contrário.

Figura 5 – Descrição da Chave Seletora



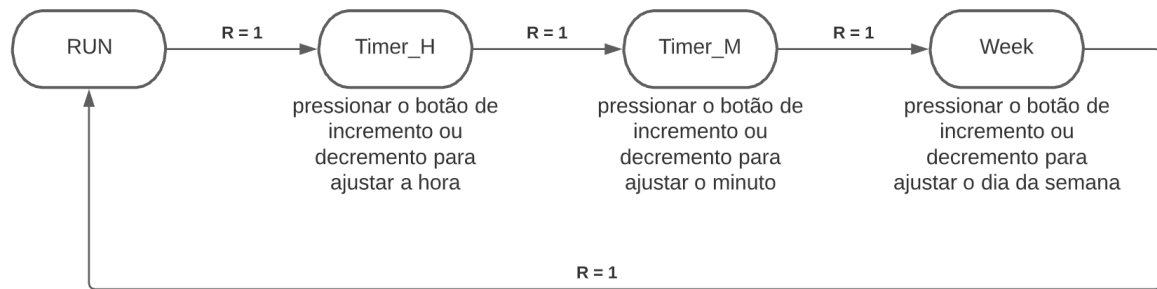
Fonte: Desenvolvido pelos Autores.

## 2.3 Ajuste de Relógio - Funcionamento Esperado

Para melhor descrever o funcionamento do estado de ajuste de relógio, do programador horário, foi desenvolvido um fluxograma (Figura 6) utilizando 4 dos modos de operações disponíveis.

O sistema inicia em RUN, caso seja pressionado o botão R, este segue para o Timer\_H e assim em diante. É válido ressaltar que após iniciado o ajuste do relógio, o programador só voltará para RUN após passar todos os modos necessários para encerrar o ajuste requerido. Cada ajuste (valor de hora, minuto e dia) é realizado a partir do pressionamento dos botões de incremento ou decremento, como for desejado.

Figura 6 – Processo de Ajuste do Relógio



Fonte: Desenvolvido pelos Autores.

## 2.4 Ajuste do Agendamento - Funcionamento Esperado

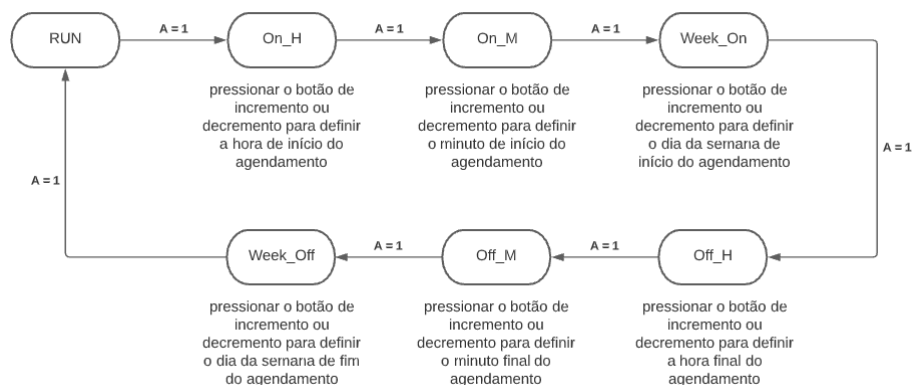
Para melhor descrever o funcionamento do estado de ajuste de agendamento do alarme, do programador horário, foi desenvolvido um fluxograma (Figura 7) utilizando 7 dos modos de operações disponíveis.

O sistema inicia em **RUN**, caso seja pressionado o botão **A**, este segue para o **On\_H** e assim em diante. É válido ressaltar que após iniciado o ajuste de agendamento, o programador só voltará para **RUN** após passar todos os modos necessários para encerrar o ajuste requerido, assim como no ajuste de relógio.

Cada ajuste (valor de hora, minuto e dia) é realizado a partir do pressionamento dos botões de incremento ou decremento, como for desejado. E, é válido lembrar que cada modo de operação possui um conjunto de atribuições de LED e display já comentados anteriormente.

O agendamento é dado através de um intervalo, este se inicia realizando o ajuste da hora de **ÍNICIO** do intervalo, seguido pelo ajuste do minuto, dia de início, hora final, minuto final e se encerra ajustando o dia da semana em que se dá o **FIM** do intervalo, como descrito na Figura 7.

Figura 7 – Processo de Agendamento de Alarme

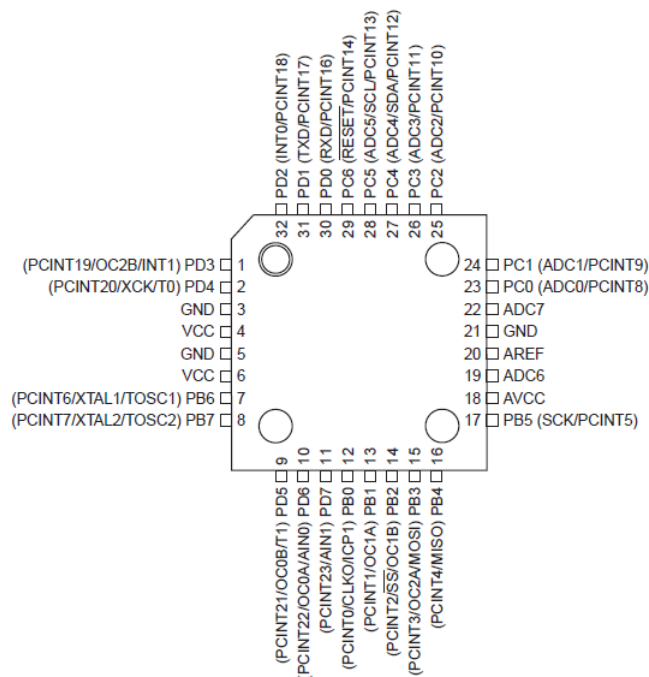


Fonte: Desenvolvido pelos Autores.

## 2.5 Microncontrolador AVR ATMega328p

O ATMEGA328P trata-se de um CMOS (*complementary metal oxide semiconductor*) de baixa potência, microcontrolador de arquitetura RISC AVR avançada de 8 bits, o qual possui alto desempenho e baixo consumo (ATMEL, 2015). O dispositivo conta com 32 portas que se dividem entre entradas/saídas programáveis digitais e analógicas, aterramento e alimentação (Figura 8).

Figura 8 – Pinagem ATmega328p



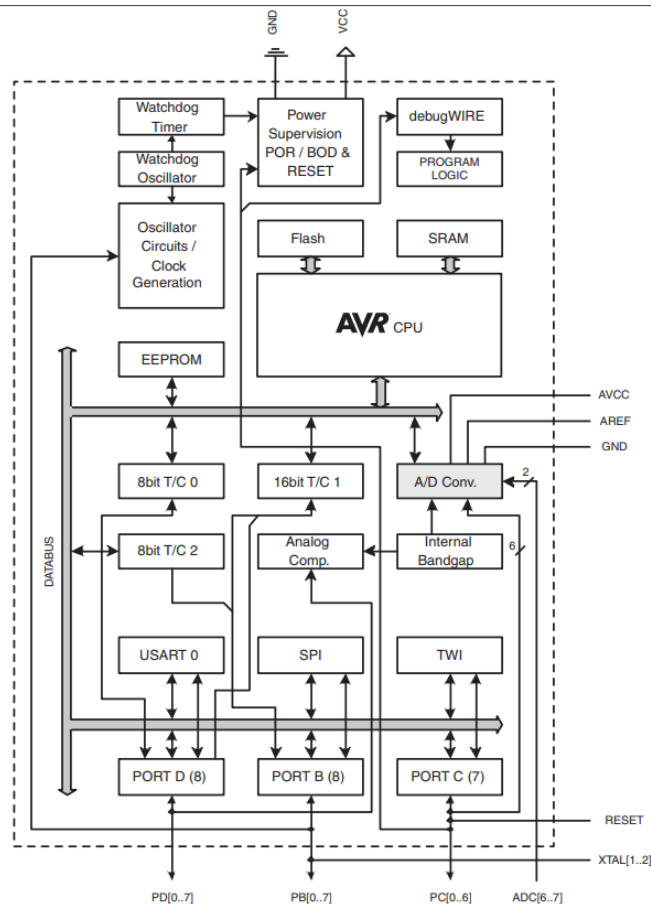
Fonte: ATMEL (2015).

Para melhor entendimento do funcionamento interno do ATmega328p, a Figura 9 é o diagrama de blocos internos do dispositivo. Nele é possível observar o bloco de comunicação do protocolo SPI, o qual iremos usar para realizar a comunicação entre os CI periféricos - de controle de hora e display - e o microcontrolador em questão. É com este dispositivo que será implementado o controle do programador horário, através de linhas instruções, manipulação de registradores e controle de dados através da pinagem disponível.

## 2.6 CI RTC PCF2123

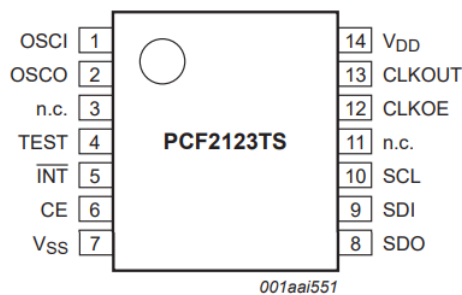
O CI PCF2123 é um CMOS RTC, este fornece data e hora de forma otimizada para aplicações baixo consumo. Os dados deste são transferidos através de uma interface periférica serial (SPI-bus) com uma taxa máxima de 6,25Mbits/s. Este dispositivo também dispõe de uma função de alarme e de temporizador (NXP, 2013). A pinagem deste CI pode ser visualizada na Figura 10.

Figura 9 – Diagrama de blocos do ATmega328p



Fonte: ATMEL (2015).

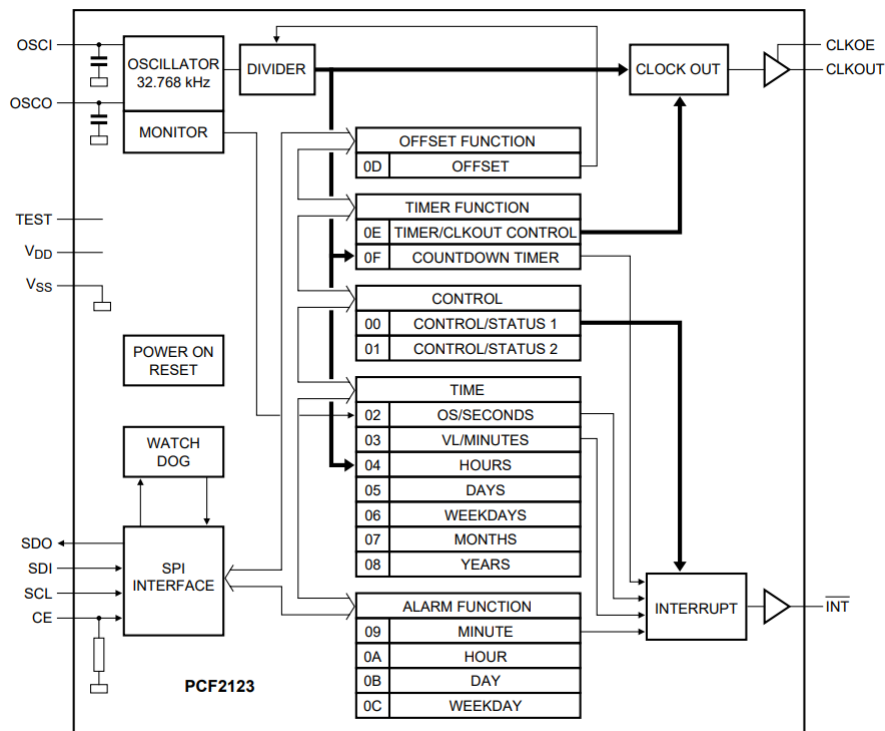
Figura 10 – Pinagem PCF2123



Fonte: NXP (2013).

Este CI possui 16 registradores de 8 bits, um contador de endereço autoincrementável, um oscilador de 32,768 kHz, um divisor de frequência que fornece o relógio fonte para o RTC, uma saída de relógio programável e um barramento SPI. O diagrama de blocos e a relação de registradores do PCF2123 estão dispostos, respectivamente, nas Figuras 11 e 12.

Figura 11 – Diagrama de Blocos de PCF2123



Fonte: NXP (2013).

Figura 12 – Registradores de PCF2123

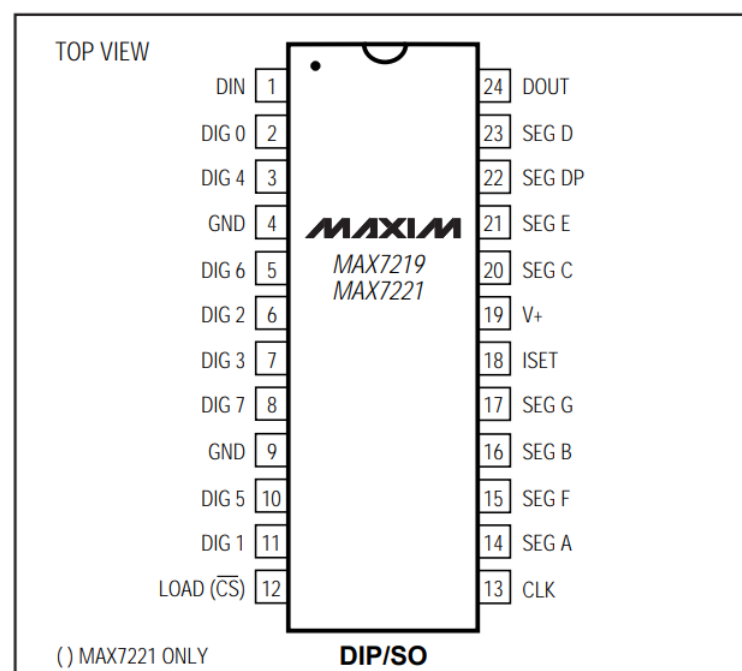
Address	Register name	Bit							
		7	6	5	4	3	2	1	0
Control and status registers									
00h	Control_1	EXT_TEST	N	STOP	SR	N	12_24	CIE	N
01h	Control_2	MI	SI	MSF	TI_TP	AF	TF	AIE	TIE
Time and date registers									
02h	Seconds	OS	SECONDS (0 to 59)						
03h	Minutes	--	MINUTES (0 to 59)						
04h	Hours	-	-	AMPM	HOURS (1 to 12) in 12 h mode				
				HOURS (0 to 23) in 24 h mode					
05h	Days	-	-	DAYS (1 to 31)					
06h	Weekdays	-	-	-	-	-	WEEKDAYS (0 to 6)		
07h	Months	-	-	-	MONTHS (1 to 12)				
08h	Years	YEARS (0 to 99)							
Alarm registers									
09h	Minute_alarm	AE_M	MINUTE_ALARM (0 to 59)						
0Ah	Hour_alarm	AE_H	-	AMPM	HOUR_ALARM (1 to 12) in 12 h mode				
				HOUR_ALARM (0 to 23) in 24 h mode					
0Bh	Day_alarm	AE_D	-	DAY_ALARM (1 to 31)					
0Ch	Weekday_alarm	AE_W	-	-	-	-	WEEKDAY_ALARM (0 to 6)		
Offset register									
0Dh	Offset_register	MODE	OFFSET[6:0]						
Timer registers									
0Eh	Timer_clkout	-	COF[2:0]			TE	-	CTD[1:0]	
0Fh	Countdown_timer	COUNTDOWN_TIMER[7:0]							

Fonte: NXP (2013).

## 2.7 CI MAX7219

Este CI é um driver de interface de microprocessadores para displays de 7 segmentos de 8 dígitos ou similares. Este foi utilizado para fazer a multiplexação dos 4 dígitos do display, onde a configuração de pinos está exposta na Figura 13. Este CI utiliza o protocolo de comunicação SPI e entre suas características possui a capacidade de decodificar o dado inserido de maneira serial através da entrada DIN em uma frequência de transmissão de até 10MHz a depender do valor que está sendo alimentado, como também a multiplexação de até 8 dígitos em uma frequência de 800Hz.

Figura 13 – Configuração dos pinos do MAX7219



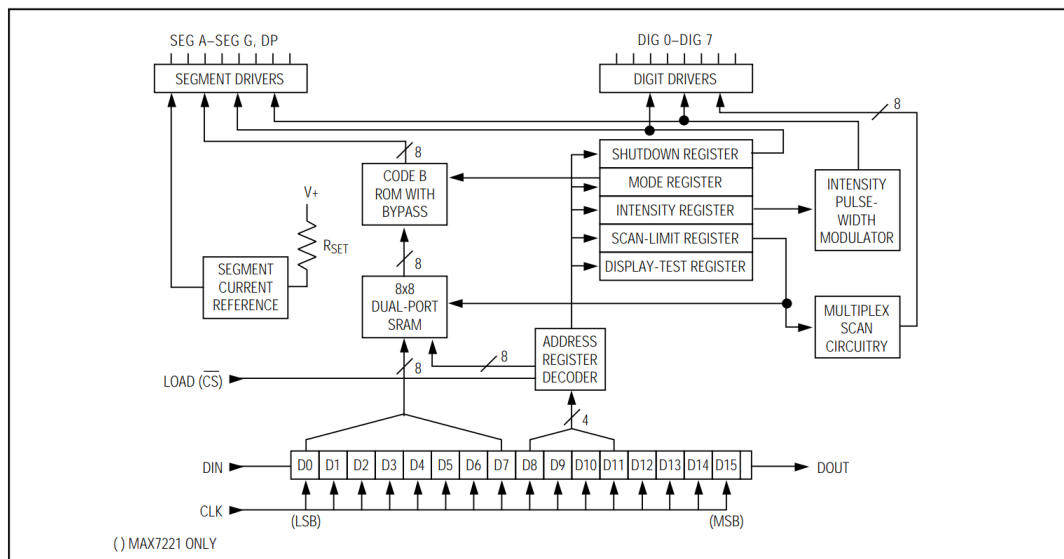
Fonte: MAXIM (1997)

Uma das configurações permitem também que dígitos individuais sejam atualizados e possui 3 convenientes entradas para a comunicação serial, como pode ser vista na Figura 14. Nela ainda pode ser analisado que os 8 primeiros bits serão armazenados para, posteriormente, serem enviados ao enables dos displays que vão ser ligados, também pode ser visto que os bits de D8 a D11 serão responsáveis por selecionar os modos de configuração que, mediante a eles, pode ser enviado o dígito desejado ao display que está habilitado no momento.

## 2.8 Protocolo de Comunicação SPI

A comunicação *Serial Peripheral Interface* (SPI) é um protocolo síncrono de comunicação serial usado entre circuitos integrados ou até mesmo entre microcontroladores capaz de operar no modo *Full Duplex*, ou seja, pode tanto receber quanto enviar dados.

Figura 14 – Diagrama do MAX7219



Fonte: MAXIM (1997).

Essa operação de envio e recebimento dos dados são classificados como *Master* ou *Slave*. Para que realizar a comunicação, os dispositivos terão que compartilhar 4 pinos sendo eles: *Master Out - Slave In* (MOSI), *Master In - Slave Out* (MISO), *Clock* (SCK) e *Slave Select* (SS).

No MOSI o barramento é usado para enviar os dados do *Master* para os periféricos (*Slave*), o MISO funciona de maneira oposta, pois os dados que estão no *Slave* serão enviados para o *Master*. O SCK é o sinal de *clock* gerado pelo *Master* e o SS indica qual foi o *Slave*, selecionado pois o circuito pode apresentar mais de um *Slave*. O sinal SS possui lógica barrada, então ativarão em nível lógico baixo. Além dessas 4 pinagens para o uso do protocolo é necessário saber que o dispositivo *Master* sempre será o responsável por iniciar a transferência de dados.

O projeto desenvolvido para a implementação de um programador de horário neste relatório contará com a comunicação entre 3 dispositivos utilizando o protocolo *SPI* sendo eles: o microcontrolador ATMEGA328P (como *Master*), o CI PCF2123 para ajuste de horário do nosso relógio (como *Slave*) e o CI MAX7219 (como *Slave*).

## 3 PROJETO

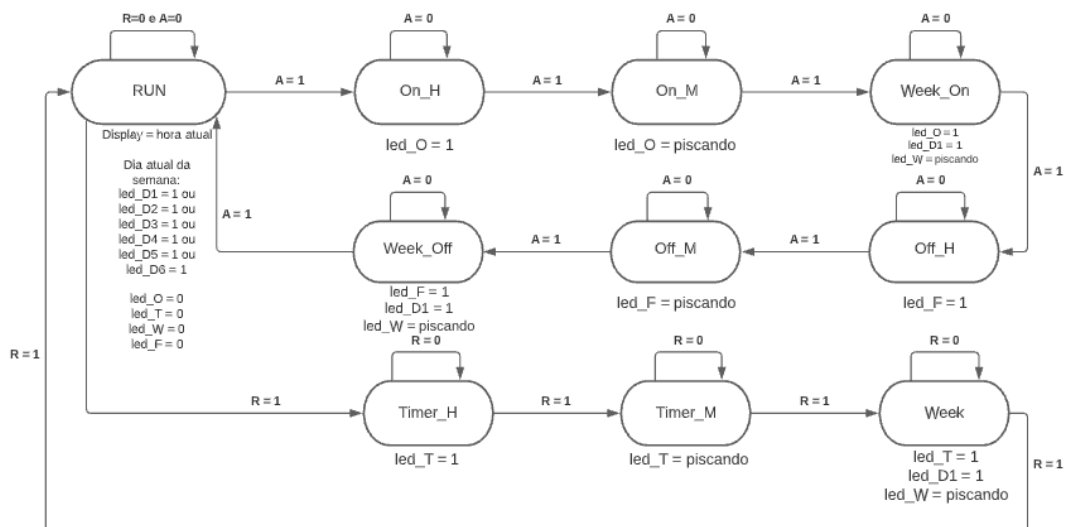
Em ciência dos modos de operação, dos estados de seleção, dos dispositivos eletrônicos a serem utilizados e do protocolo de comunicação a ser aplicado à estes. O projeto se subdivide na criação de uma máquina de estados de alto nível (fluxograma) que retrate o funcionamento geral do programador, incluindo os ajustes de relógio e de agendamento de intervalo, a criação de um fluxograma para a descrição do funcionamento da chave seletora de controle da saída OUT e as aplicações atribuídas ao ATmega328p, ao CI RTC PCF2123 e ao CI MAX7219. Com isso, todas as definições de projeto são esclarecidas e o problema é passível de implementação.

### 3.1 Máquina de Estados de Alto Nível - fluxograma

Para melhor entendimento do sistema do programador horário, foi desenvolvido o fluxograma da Figura 15. É a partir dele que será possível identificar o fluxo de instruções necessárias para a implementação em *assembly*.

Na MDE temos a representação de cada modo de operação e suas respectivas atribuições de LED e display. O controle de fluxo de cada estado se dá por meio das variáveis A e R que representam, quando em 1 ou em 0, o pressionar ou não dos botões de relógio e de agendamento inseridos na interface. Repare que o estado natural do programador é no modo RUN que apresenta a data e hora atuais, com todos os LED de modo de operação apagados.

Figura 15 – MDE - fluxograma



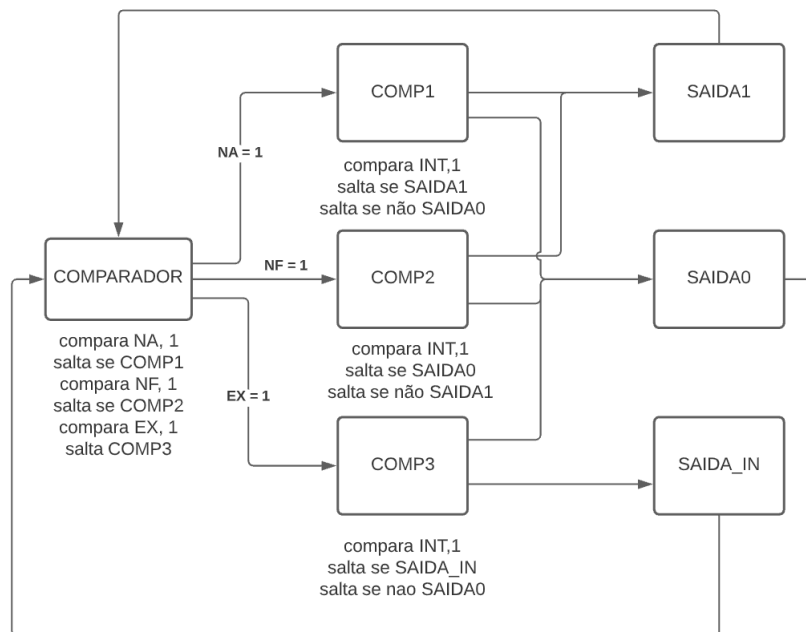
Fonte: Desenvolvido pelos Autores.



## 3.2 Chave Seletora - Fluxograma

Aqui é apresentado a lógica de como implementar as o tratamento dos dados das chaves seletoras. A Figura 16 mostra que no primeiro bloco há 3 comparadores e que será verificado se cada chave está em seu estado alto, caso não estejam, haverá um pulso verificando as próximas. Após essa conjuntura, se o NA for 1 irá para o COMP1 e nele é verificado se a entrada IN é 1: caso seja, a saída OUT vai ser 1, caso não seja, a saída será 0. Se NF for 1, no COMP2 haverá uma verificação da entrada 1: caso seja 1 a saída será 0, senão vai ser 1. Se EX for 1, o COMP3 vai comparar o valor da entrada com 1: caso seja verdadeiro, a saída OUT terá o valor da entrada IN, caso contrário será 0. Por fim, após a inserção do valor a saída, deverá haver uma nova verificação da entrada, voltando ao ciclo da explicação.

Figura 16 – Chave Seletora - fluxograma



Fonte: Desenvolvido pelos Autores.

## 3.3 ATMEGA328P - Aplicação

O ATMEGA328p irá utilizar o clock interno de 8MHz para executar as instruções e terá ao todo 11 pinos configurados como saídas e 8 como entradas, por fim, iremos utilizar de periférico apenas o *Timer/Counter1* de 16 *bits* para a frequência de 2Hz exigida na problemática. Por estamos utilizando o protocolo de comunicação SPI com dois CI's que serão configurados como escravos, o ATMEGA328P será o mestre da comunicação.

### 3.3.1 Pinagem Utilizada

A configuração pensada pela equipe de projeto para a pinagem está exposta na Tabela 1. Perceba que os pinos PB3 e PB5 são compartilhados por dois CI's, essas entradas serão responsáveis por enviar os dados e a frequência de transmissão de dados, respectivamente, para os dois CI's escravos da comunicação SPI.

Tabela 1 – Configuração de pinagem

Pino	Entrada/Saída	Descrição
PB1	Saída	Slave select para o MAX7219
PB2	Saída	Slave select para o PCF2123
PB3	Saída	Configurada para ser o MOSI que envia o dado para os escravos
PB4	Saída	Configurada para ser o MISO que recebe o dado dos escravos
PB5	Saída	Configurada para fornecer a frequência de transferência dos dados para os escravos
PB6	Entrada	Input da problemática
PB7	Saída	Output da problemática a ser definido pelo microcontrolador com base no modo escolhido
PC0	Entrada	Botão down
PC1	Entrada	Botão up
PC2	Entrada	Botão R
PC3	Entrada	Botão A
PC4	Saída	Definição do input do demultiplexador
PC5	Saída	LED F
PD2	Entrada	NA
PD3	Entrada	NF
PD4	Entrada	EX
PD5	Saída	LED T
PD6	Saída	LED W
PD7	Saída	LED O

Fonte: Desenvolvido pelos Autores.

Todas as entradas e saídas utilizadas serão configuradas para leitura/escrita digital, pois não há nenhum dado analógico para ser convertido no microcontrolador. Os pinos que não apareceram na Tabela 1 não foram utilizados, ficando livre para a equipe de implementação a utilização dos mesmos para propósitos gerais.

### 3.3.2 Protocolo de Comunicação SPI - Aplicação

Utilizando o microcontrolador ATMEGA328P que possui encapsulamento de 32 porta, selecionamos os pinos que trabalharemos para efetuar a comunicação entre o microcontrolador ATMEGA328P, o CI PCF2123 e o CI MAX7219 sendo eles: PB1(*Slave Select* do CI MAX7219), PB2(*Slave Select* do CI PCF2123), PB3(MOSI), PB4(MISO) e PB5(SCK). Como dito anteriormente, precisaremos utilizar 3 portas (MOSI, MISO e SCK) mais  $n$  portas referentes a quantidade de dispositivos *Slave*, ou seja, no total usaremos 5 pinos.

Para iniciar o projeto, foi necessário verificar algumas informações. A primeira era conferir no *datasheet* dos CI's qual são as velocidades SPI máximas que poderíamos utilizar. Constatando que o dispositivo responsável pela atualização do horário poderia operar até em 8MHz e o outro dispositivo poderia chegar até em 10MHz vimos que poderíamos

utilizar o *clock* interno do microcontrolador e setamos um *Prescale* de 8. A segunda consideração foi decidir que o deslocamento dos *bits* iniciará pelo *bit* mais significativo (MSB). E a última consideração era sobre o modo que iríamos operar os dados, ou seja, configurar o *Clock Phase* (CPHA) e o *Clock Polarity* (CPOL). O modo escolhido possuía configuração do CPHA = 0 e o CPOL = 0.

A partir das informações passadas no parágrafo anterior, é possível configurar os 4 registradores necessários para o uso do protocolo SPI sendo eles: AVR *Status Register* (SREG), SPI *Control Register* (SPCR), SPI *Status Register* (SPSR) e SPI *Data Register* (SPDR). Os registradores serão configurados de acordo com as informações passadas no parágrafo anterior.

### 3.3.3 Timer/Counter1

Os botões de incremento e decremento assim como a frequência dos leds piscando é de 2Hz ou 0,5 segundos. Assim, se faz necessário o uso de um temporizador dedicado para essa contagem. O ATMEGA328P possui três temporizadores, sendo dois de 8 *bits* e um de 16 *bits*. Para esse projeto, utilizando o *clock* interno de 8MHz com o *Prescale* de 256, usaremos o *Timer/Counter1*, pois precisaremos de um temporizador de 16 *bits* para contar o número de ciclos até chegar em 0,5 segundos.

Para configurar como o *Timer/Counter1* funcionará iremos utilizar 4 registradores específicos para essa operação sendo eles: *Timer Counter1 Control Register A* (TCCR1A), *Timer Counter1 Control Register B* (TCCR1B) e os registradores de armazenamento de contagem TCNT1H e TCNT1L. Por fim, o modo utilizando é o CTC, gerando interrupção por comparação.

## 3.4 CI RTC PCF2123 - Aplicação

O dispositivo responsável pela atualização do horário, o *Real Time Clock* (RTC), definido para esse projeto é o PCF2123 e ele possui configurações necessárias para que seja possível a contagem do tempo.

Os registradores que utilizaremos são o de controle, apenas o registrador 00h, e os de contadores para a função do relógio (02h até 08h). Para o registrador 00h, deve ser configurado no modo normal, em formato de 24 horas e caso não seja resetado deve-se colocar os pinos 3, 4 e 6 em baixo lógico.

No registrador 02h definirá a saída dos 'Segundos' utilizando apenas 7 dos 8 *bits*, o 03h é a dos 'Minutos' também utilizando 7 *bits*, no 04h são às 'Horas' (setado para PM, ou seja, 24 horas) usando 6 *bits*, no registrador 05h são os 'Dias' também usando 6 *bits*, o 06h é o dos 'Dias da Semana' precisando apenas de 3 *bits*, o 07h é do 'Mês' usando 5

*bits* e por último é o registrador do 'Ano' que utilizamos todos os 8 *bits*. Os registradores citados anteriormente são todos codificados no formato BCD, como mostra a Figura 17.

Figura 17 – Formato BCD para exemplo de minutos

Minutes value (decimal)	Upper-digit				Digit			
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	1
02	0	0	0	0	0	0	1	0
:	:	:	:	:	:	:	:	:
09	0	0	0	0	1	0	0	1
10	0	0	0	1	0	0	0	0
:	:	:	:	:	:	:	:	:
58	0	1	0	1	1	0	0	0
59	0	1	0	1	1	0	0	1

Fonte: NXP (2013)

## 3.5 CI MAX7219 - Aplicação

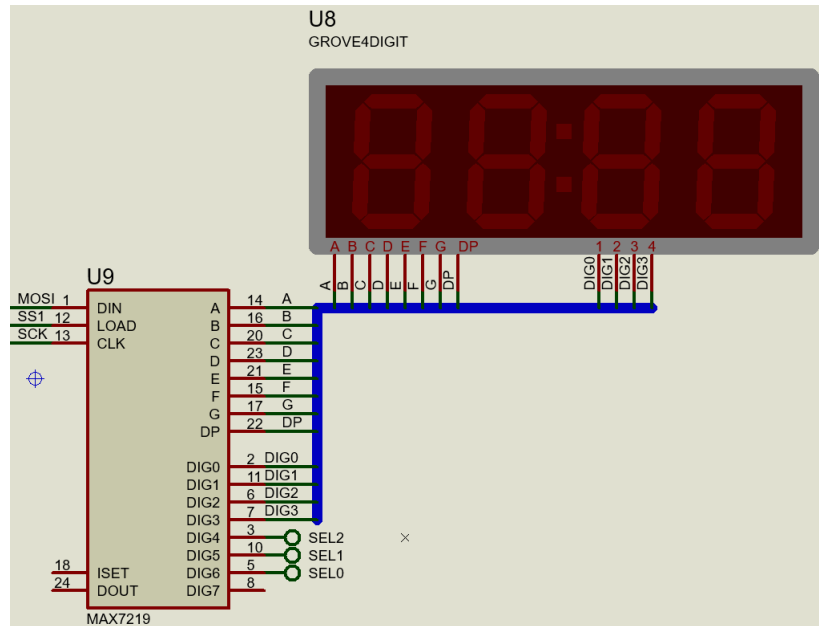
Este CI foi selecionado para solucionar a problemática de poucos pinos disponíveis para o *display* de 4 dígitos, pois com apenas 3 pinos é possível enviar todos os dados necessários para exibir os caracteres no *display*.

### 3.5.1 Pinagem Utilizada

A pinagem pensada pela equipe projetista está exposta na Figura 18, onde o CI recebe o *clock* de transferência de dados através da entrada CLK, o dado a ser decodificado através da entrada DIN e o *bit* que indica quando o dispositivo deverá ativar a leitura. Como o CI utiliza o protocolo de comunicação SPI e ele servirá apenas para decodificar os dados enviados pelo microcontrolador que será o mestre da comunicação, este será configurado como um dos escravos do mesmo, com isso dito, a entrada DIN será a leitura da saída MOSI do microcontrolador.

As saídas do MAX7219 servirão para indicar ao *display* quais segmentos necessitam acender por caractere, sendo de A~G os 7 segmentos, DP os dois pontos que separam as horas dos minutos e DIG0~DIG7 as saídas responsáveis por fazer a multiplexação dos 8 dígitos, repare que como utilizamos apenas 4 dígitos, reaproveitamos 3 das outras 4 saídas restantes de dígitos como chave seletora do MUX responsável por ligar os leds da semana, como a frequência de multiplexação do CI é de 800Hz aparentará que essas saídas são constantes e sem oscilações. Na Figura 19 é possível visualizar a chave seletora que será definida pelo MAX7219 sendo utilizada em um demultiplexador, repare que na imagem

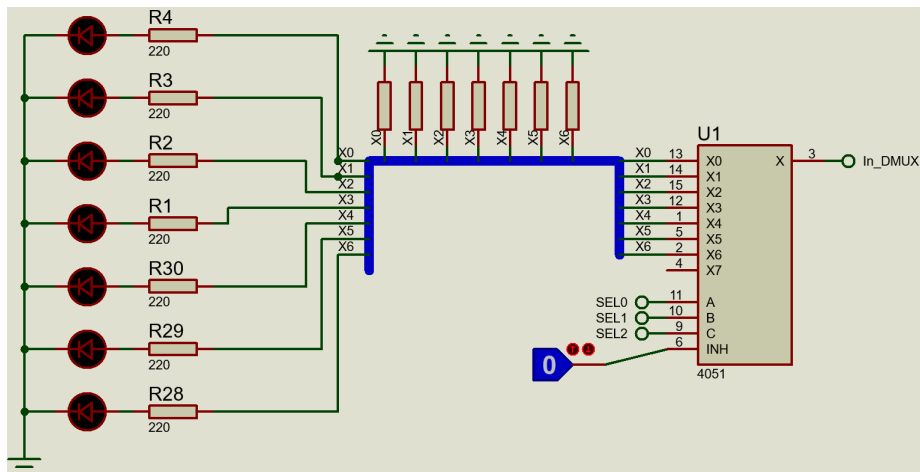
Figura 18 – Pinagem pensada para o MAX7219



Fonte: Desenvolvido pelos Autores.

utilizamos o CI 4051 em sua configuração de demultiplexador, porém fica livre para que a equipe de implementação utilize qualquer CI com a mesma função.

Figura 19 – Demultiplexador dos leds da semana



Fonte: Desenvolvido pelos Autores.

### 3.5.2 Configuração Necessária

Para as configurações, devesse, inicialmente, ajustar o mapeamento do endereço de registro. Ele consiste nos pinos D8 ao D11 e é lá que vai ser inserido quais dígitos vão ser modificados (dígito 0 ao 7), a decodificação do modo, a intensidade, limite da leitura, desligamentos e até mesmo um teste. Neste momento, serão introduzidos os valores "1100"(xC) para D8 a D11 (Figura 20), setando o modo "shutdown" e para o D0 em 1 será selecionado o "normal mode"(pode ser visto da Figura 21).

Figura 20 – Tabela 2 do datasheet do MAX7219.

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	X0
Digit 0	X	0	0	0	1	X1
Digit 1	X	0	0	1	0	X2
Digit 2	X	0	0	1	1	X3
Digit 3	X	0	1	0	0	X4
Digit 4	X	0	1	0	1	X5
Digit 5	X	0	1	1	0	X6
Digit 6	X	0	1	1	1	X7
Digit 7	X	1	0	0	0	X8
Decode Mode	X	1	0	0	1	X9
Intensity	X	1	0	1	0	XA
Scan Limit	X	1	0	1	1	XB
Shutdown	X	1	1	0	0	XC
Display Test	X	1	1	1	1	XF

Fonte: MAXIM (1997)

Figura 21 – Tabela 3 do datasheet do MAX7219.

MODE	ADDRESS CODE (HEX)	REGISTER DATA							
		D7	D6	D5	D4	D3	D2	D1	D0
Shutdown Mode	XC	X	X	X	X	X	X	X	0
Normal Operation	XC	X	X	X	X	X	X	X	1

Fonte: MAXIM (1997)

Após isso, é adicionado no D8-D11 o valor "1001"(x9 em hexadecimal), referente a Figura 20, selecionando a opção "decode mode" para definir quais dígitos serão utilizados para os enables do *display*, que neste caso são os dígitos 0 ao 3, então é inserido no D7-D0 o valor "00001111"(0F em hexadecimal). As opções dessa configuração podem ser vista na Figura 22.

Figura 22 – Tabela 4 do datasheet do MAX7219.

DECODE MODE	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7–0	0	0	0	0	0	0	0	0	00
Code B decode for digit 0 No decode for digits 7–1	0	0	0	0	0	0	0	1	01
Code B decode for digits 3–0 No decode for digits 7–4	0	0	0	0	1	1	1	1	0F
Code B decode for digits 7–0	1	1	1	1	1	1	1	1	FF

Fonte: MAXIM (1997)

Depois, na tabela da Figura 20, são inseridos os valores em "1011"(xB), objetivando setar as quantidades de saídas para o uso no display e para o MUX dos leds. Desta forma, tem-se a Figura 23, e mediante as configurações disposta por ela, seleciona-se o X6 (inserindo "110"no D2-D0).

Figura 23 – Tabela 8 do datasheet do MAX7219.

SCAN LIMIT	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
Display digit 0 only*	X	X	X	X	X	0	0	0	X0
Display digits 0 & 1*	X	X	X	X	X	0	0	1	X1
Display digits 0 1 2*	X	X	X	X	X	0	1	0	X2
Display digits 0 1 2 3	X	X	X	X	X	0	1	1	X3
Display digits 0 1 2 3 4	X	X	X	X	X	1	0	0	X4
Display digits 0 1 2 3 4 5	X	X	X	X	X	1	0	1	X5
Display digits 0 1 2 3 4 5 6	X	X	X	X	X	1	1	0	X6
Display digits 0 1 2 3 4 5 6 7	X	X	X	X	X	1	1	1	X7

Fonte: MAXIM (1997)

Por fim, para selecionar os displays e inserir os valores corretamente em cada, devesse selecionar, através do *Register Adresse Map* (Figura 20), os valores "0001"em D11-D8 (x1) e para o "Code B"(Figura 24) se define os valores por meio das entradas D3-D0. Para os *enables* do segundo, terceiro e quarto *display* também funcionará assim.

Figura 24 – Tabela 5 do *datasheet* do MAX7219.

7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6-D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

Fonte: MAXIM (1997)



## Referências

ATMEL CORPORATION. *ATMega328p*: Atmel 8-bit microcontroller with 4/8/16/32kbytes in-system programmable flash. Califórnia, USA, 2015. 660 p.

COEL. *Programadores Horários*. 2015. Acessado em: 25 jul. 2021. Disponível em: <[https://cdn.media.coel.com.br/uploads/2016/06/Manual-de-Instrucoes-RTST20\\_r8.pdf](https://cdn.media.coel.com.br/uploads/2016/06/Manual-de-Instrucoes-RTST20_r8.pdf)>.

DIAS, S. M. *Problema 04: Projeto*. [S.l.]: Departamento de Engenharia Elétrica, 2021.

MAXIM. *Serially Interfaced, 8-Digit LED Display Drivers*. 1997. Acessado em: 25 jul. 2021. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/73745/MAXIM/MAX7219.html>>.

NXP. *PCF2123 - SPI Real time clock/calendar*. 2013. Acessado em: 25 jul. 2021. Disponível em: <<https://www.nxp.com/docs/en/data-sheet/PCF2123.pdf>>.

VAHID, F. *Sistemas Digitais: Projeto, Otimização e HDLs*. [S.l.]: Artmed Bookman, 2008.

# ANEXO A – Relato semanal

**Líder:** Lucas Batista da Fonseca

## A.1 Equipe

Tabela 2 – Identificação da equipe

<b>Função no grupo</b>	<b>Nome completo do aluno</b>
Redator	Anny Beatriz Pinheiro Fernandes
Debatedor	Isaac de Lyra Junior
Videomaker	Wesley Brito da Silva
Auxiliar	Ana Beatriz Marinho Neves

Fonte: Produzido pelos autores.

## A.2 Defina o problema

O problema consiste em projetar um programador horário baseado em  $\mu$ C AVR (ATMega328P) que realiza o controle da saída OUT de acordo com o horário agendado. Além de ter três modos de operação definidos pela chave seletora de 3 posições, esses modos são: NA; NF; EX. Esse programador possui um display de 7 segmentos de 4 dígitos, 7 leds de indicação dos dias da semana, outros 4 leds que informam o modo de operação, e para configurar o funcionamento são utilizados 4 botões, desses dois são setas que incrementa ou decrementa, e os outros dois direcionam para a calibrar a hora ou para o agendamento.

Essa máquina possui 10 modos de operação, sendo elas: RUN; On\_H; On\_M; Week\_On; Week\_Off; Off\_M; Off\_H; Timer\_H; Timer\_M; Week. Em cada um desses modos de operação, os leds e o display estão configurados em um determinado modo, vale salientar, que em alguns leds que indicam o modo de operação podem estar piscando em algum determinado modo, como exemplo, no modo Time\_M o led T deve piscar a uma taxa de 2Hz.

O circuito utiliza um CI RTC que fornece o dia e horário que serão tratados pelo ATMega328P. Essa comunicação é feita através da comunicação serial SPI, em que o  $\mu$ C é o mestre e o RTC é o escravo.

### A.3 Registro de *brainstorming*

A primeira reunião para fazer o brainstorm foi discutido como seria feita a MDE do projeto, além de como iríamos utilizar tantos leds sabendo que as portas do  $\mu C$  não seriam o suficiente, no fim dessa primeira reunião entramos em um consenso de definir o CI RTC depois de sabermos exatamente qual seria a nossa necessidade.

Na reunião seguinte decidimos usar outro protocolo para controlar o display, pois assim iria consumir menos portas do  $\mu C$ , e que iríamos usar um shifter register para acionar os leds da semana. Em pesquisas feitas foi encontrado um CI SAA1064 que controla display de 7 segmentos utilizando protocolo I<sup>2</sup>C, assim ficou para a reunião seguinte o estudo sobre esse protocolo.

No último brainstorming percebemos que utilizar dois protocolos diferentes seria contraproducente, dessa forma decidimos utilizar o SPI tanto para o CI RTC quanto para outro CI controlador de display de 7 segmentos. Na pesquisa foi encontrado o CI MAX7219 que utiliza protocolo SPI para controlar um display de 7 segmentos, assim sobrando portas para acionar os leds e os botões. Diretamente ao ATmega deveria estar ligados os leds dos modos de operação e os leds da semana são acesos de acordo com um demux ligado a portas que sobrariam do CI MAX7219.

### A.4 Pontos-chaves

- Entender como funciona o protocolo SPI.
- Entender como funciona o CI RTC, como configurar os registradores internos.
- Compreender o funcionamento do programador horário.
- Utilizar interrupção para mudança de operação sem parar de realizar a contagem.
- Economizar quantidade de portas lógicas utilizadas do ATmega328P
  - Como controlar o display de 7 segmentos de 4 dígitos.
  - Como ligar os 11 leds da máquina e como realizar o piscar.

### A.5 Questões de pesquisa

Protocolo SPI; CI RTC; ATmega328P; Interrupção; CI MAX7219; Demultiplexador.

## A.6 Planejamento da pesquisa

O plano da pesquisa inicialmente foi entender como é o funcionamento da máquina, por esse motivo o dia seguinte a aula foi definido como dia de estudar o funcionamento da máquina. Na reunião seguinte o plano era pesquisar o protocolo SPI e qual o CI que nós iríamos usar. Na reunião seguinte definir como iremos poupar as portas do ATmega328P. Por fim definir o modo de operação de todos os periféricos utilizados dentro do ATmega e dos CI externos que são utilizados. No último dia realizar a escrita do projeto.