

DESENVOLVIMENTO DE UM RELÓGIO DIGITAL COM SINCRONISMO DE HORÁRIO VIA SERVIDOR NTP

ELIASIBE ALVES DE SOUZA¹, PABLO RODRIGO DE SOUZA².

¹Graduando em Tecnologia de Engenharia Elétrica, Bolsista PIBIFSP, IFSP, Câmpus Piracicaba
E-mail: Eliasibe.alves@aluno.ifsp.edu.br

² Professor Dr. Pablo Rodrigo de Souza, Instituto Federal de Educação Ciência e Tecnologia de São Paulo – Câmpus Piracicaba.
E-mail: pablo@ifsp.edu.br
Área de conhecimento (Tabela CNPq): 3.04.03.03-0 Circuitos Eletrônicos

Apresentado no
10º Congresso de Inovação, Ciência e Tecnologia do IFSP
27 e 28 de novembro de 2019- Sorocaba-SP, Brasil

RESUMO: Diversos tipos de equipamentos utilizados no cotidiano possuem um relógio digital para fornecer o horário para os usuários. É bastante comum que o horário indicado apresente erro após um certo tempo de funcionamento. Esse erro está associado a técnica utilizada para obtenção da base de tempo do relógio e tolerâncias dos componentes que formam o sistema, por exemplo, o cristal de quartzo que normalmente é utilizado no circuito oscilador do microcontrolador. Este trabalho apresenta um relógio digital capaz de se conectar na internet para sincronizar seu horário com a Hora Legal Brasileira fornecida pelo servidor NTP.br, de forma a minimizar o erro na representação do seu horário. Resultados experimentais mostram que sincronizando o horário do relógio com o servidor a cada vinte e quatro horas, o erro máximo observado foi de aproximadamente três segundos.

PALAVRAS-CHAVE: Microcontrolador; Relógio Digital; Servidor NTP; Internet.

DEVELOPING A DIGITAL CLOCK WITH TIME SYNCHRONIZATION BY THE NTP SERVER

ABSTRACT: Several types of equipment used in everyday life have a digital clock to provide the time for users. It is quite common to observe a time error after a certain operating time. This error is associated with the technique used to obtain the clock time base and tolerances of the components that make up the system, for example, the quartz crystal that is normally used in the oscillator circuit of the microcontroller. This work presents a digital clock capable of connecting to the internet to synchronize its time with the Brazilian Legal Time provided by the NTP.br server, in order to minimize the error in the representation of its time. Experimental results show that by synchronizing the clock with the server every twenty-four hour, the maximum error observed was approximately three seconds.

KEYWORDS: Microcontroller; Digital Watch; NTP; Timer; Engineering.

INTRODUÇÃO

O tempo é uma das grandezas físicas elementares. Medir o tempo com exatidão não é tarefa simples. Desde os primórdios, o homem desenvolveu diversas técnicas para medir o tempo, inicialmente baseadas na observação da movimentação dos astros e recentemente, em propriedades físicas de elementos da natureza. O dispositivo utilizado para medir o tempo é o relógio e a sua unidade de medida é o segundo. O segundo é definido como a duração de 9.192.631.770 períodos da radiação correspondente à transição entre dois níveis hiperfinos do estado fundamental do átomo de Césio 133 (Inmetro, 2012). Os relógios atômicos utilizam essas oscilações como base de tempo.

Uma das formas de obter a base de tempo de um relógio digital baseado em microcontrolador é utilizando rotinas de atraso (Godse e Godse, 2010). Esta técnica baseia-se no fato de que as instruções

do firmware são executadas em intervalos constantes de acordo com a frequência do sinal de clock. O problema é que se a frequência do sinal de clock do microcontrolador variar, devido as imperfeições do circuito oscilador tais como as tolerâncias do cristal de quartzo (SJK, 2018), ocorrerá variações na base de tempo e a representação do tempo do relógio estará errada. Equipamentos utilizados no cotidiano, tais como, refrigeradores, fornos micro-ondas, automóveis, e outros, que possuem um relógio para informar o horário ao usuário, normalmente apresentam esse problema, sendo necessário o ajuste do horário com certa frequência. Para minimizar o erro na representação do tempo, este trabalho apresenta o desenvolvimento de um relógio digital baseado em microcontrolador capaz de se conectar na internet para sincronizar o seu horário com a Hora Legal Brasileira fornecida pelo servidor NTP.br. Este dispositivo poderá ser utilizado como um módulo integrado em equipamentos que utilizam relógios, minimizando o erro na representação do horário.

MATERIAIS E MÉTODOS

Há diversas plataformas de desenvolvimento de sistemas microcontrolados disponíveis no mercado. O módulo NodeMCU ESP-12E, que possui o microcontrolador ESP8266, é uma opção extremamente interessante devido a seu custo bastante reduzido e aos seus recursos, suficientes para o desenvolvimento de diversos tipos de aplicações, inclusive na área de Internet das Coisas (Oliveira, 2017). A Figura 1 apresenta o diagrama esquemático do relógio desenvolvido. O ESP8266 possui um transceptor WiFi integrado no padrão 802.11 b/g/n para se conectar no servidor NTP, para realizar a sincronização do horário do relógio. Para ajustar o horário do relógio manualmente, foram utilizados três botões, sendo bAjuste para habilitar o relógio no modo ajuste ou operação normal e selecionar a parte do horário (hora, minuto ou segundo) que se deseja ajustar, e outros dois bInc e bDec para incrementar e decrementar os valores. A Tabela 1 mostra os pinos do microcontrolador que foram utilizados e suas respectivas funções no sistema desenvolvido.

Tabela 1. Pinos utilizados da placa NodeMCU

Identificação NodeMCU	GPIO	Função Atribuída
D1	5	Din (Display)
D2	4	Clock (Display)
D3	0	CE (Display)
D5	14	DC (Display)
D6	12	RESET (Display)
D7	13	Botão Incrementa
D8	15	Botão Decrementa
TX	1	Botão Ajuste
RX	3	Backlight (Display)

O Software Visual Studio Code foi utilizado para desenvolver o firmware do relógio por apresentar as seguintes características: conectividade com o módulo ESP8266, por meio da extensão PlatformIO, facilidade de integrar bibliotecas de funções no projeto do firmware utilizadas para controlar periféricos, tal como o display gráfico, além de ser um software gratuito bastante utilizado.

A biblioteca PCD8544.h foi utilizada para controlar o display gráfico. Ela possui uma série de recursos para configurar o modo de funcionamento e controlar a informação exibida no display. A classe PCD8544(clk,din,dc,rst,ce) vincula os GPIO do microcontrolador utilizados para controlar o display por meio da interface SPI. No firmware desenvolvido, a linha de comando static PCD8544 lcd=PCD8544(4,5,14,12,0), vincula os pinos GPIO 4, 5, 14, 12 e 0 para controlar os sinais clk, din, dc, rst e ce do display. As linhas de comando lcd.begin(84, 48) e lcd.setContrast(60) foram utilizadas para configurar a resolução do display em 84 x 48 pixels e ajustar o contraste. As funções setCursor, clearLine, clear e printf foram utilizadas para posicionar o cursor no display, limpar uma linha do display, limpar o display inteiro e imprimir cadeias de caracteres, respectivamente.

A base de tempo do relógio foi implementada por meio do temporizador timer1 do microcontrolador. A biblioteca Arduino.h possui funções para utilização dos temporizadores do

ESP8266. As funções `timer1_isr_init()` e `timer1_attachInterrupt(ISR)` são utilizadas para ativar a interrupção do temporizador 1 e vincular a rotina de serviço da interrupção definida na função `ISR` (Circuits4you, 2018). A função, `timer1_enable(divider, int_type, reload)` é utilizada para habilitar a interrupção. A frequência do sinal que decreta os registradores do `timer1` é dada pela Equação 1. Dessa forma, a linha de comando `timer1_enable(TIM_DIV256, TIM_EDGE, TIM_LOOP)` resulta em uma frequência de 312,5 kHz.

$$f_{mr1} = \frac{80 \times 10^6}{divider} \quad (1)$$

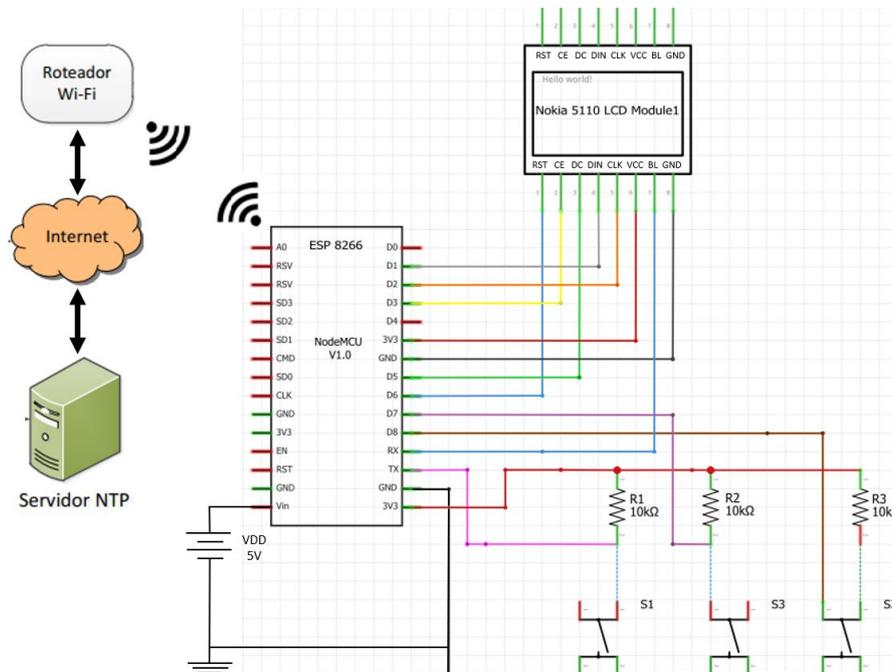


FIGURA 1. Relógio digital com sincronismo do horário com o servidor NTP.br

Para o relógio, as interrupções do `timer1` devem ocorrer em intervalos de 1 segundo. Para ajustar este intervalo, a função `timer1_write(N)` foi utilizada para inicializar o valor do `timer1`. Para obter um intervalo t , o temporizador deve ser inicializado com o valor dado por:

$$N = \frac{t}{T_{mr1}} \quad (2)$$

Onde T_{mr1} é o período do clock do `timer1`. Para obter o intervalo de 1 segundo, o `timer1` deve ser inicializado com $N=312500$. A função `timer1_disable()` é utilizada para desabilitar o temporizador.

Durante o funcionamento do relógio, é importante garantir que quando o botão Ajuste for pressionado, o firmware do relógio execute as rotinas de Ajuste do horário do relógio, independentemente da tarefa que estiver sendo executada neste momento. Dessa forma, o sinal gerado por esta entrada foi utilizado para ativar a interrupção externa do ESP8266. A função `attachInterrupt(digitalPinToInterrupt(bAjuste), handleInterrupt, FALLING)`, ativa a execução da rotina de serviço de interrupção `handleInterrupt` quando ocorrer uma borda de descida (`FALLING`) no sinal de entrada `bAjuste`. A rotina de serviço de interrupção deve conter o menor número de instruções possível, para que sua execução seja breve. No desenvolvimento do firmware do relógio, a função `handleInterrupt` simplesmente muda o valor do flag `interruptCounter` utilizado para sinalizar no programa principal. Dessa forma, quando o programa principal verifica que o flag `interruptCounter` mudou de valor, as instruções de ajuste do horário do relógio são executadas. No final desta tarefa, este flag tem que ser reinicializado. Para desativar a interrupção externa, basta utilizar a função `detachInterrupt(digitalPinToInterrupt(bAjuste))`.

A conexão do ESP8266 no roteador WiFi foi feita por meio de funções da biblioteca `ESP8266WiFi.h`. A função `WiFi.begin(ssid, password)` conecta o microcontrolador na rede por meio do roteador. Para minimizar a energia consumida pelo relógio, o ESP8266 conecta com o servidor para

sincronizar o horário somente uma vez durante 24 horas. Dessa forma, após receber os dados, o módulo encerra a conexão utilizando a instrução `WiFi.disconnect()`. O ESP8266 funciona como um cliente no protocolo NTP operando no modo cliente-servidor. O acesso aos dados é feito por meio do protocolo UDP. As bibliotecas `WiFiUdp.h` e `NTPClient.h` foram utilizadas para criar o cliente NTP e realizar a sincronização do horário com o servidor. As instruções `WiFiUDP ntpUDP;` e `NTPClient timeClient(ntpUDP, "pool.ntp.org", -10800);` configuram o ESP8266 como o cliente `ntpUDP` para acessar o horário UTC (Coordinated Universal Time) do servidor `pool.ntp.org` e fazer a correção para o fuso horário local (Last Minute Engineers, 2019). As funções `timeClient.begin()` e `timeClient.update()` inicializam o cliente NTP e acessam os dados do servidor. As funções `timeClient.getHours()`, `timeClient.getMinutes()` e `timeClient.getSeconds()` retornam os valores da hora, minuto e segundo do horário sincronizado. A Figura 2 mostra o fluxograma do trecho do programa que atualiza as variáveis hora, min (minuto) e seg (segundo) a cada interrupção do `timer1`, e quando o horário for igual a 00:00:00, acessa os dados no servidor NTP para realizar a sincronização.

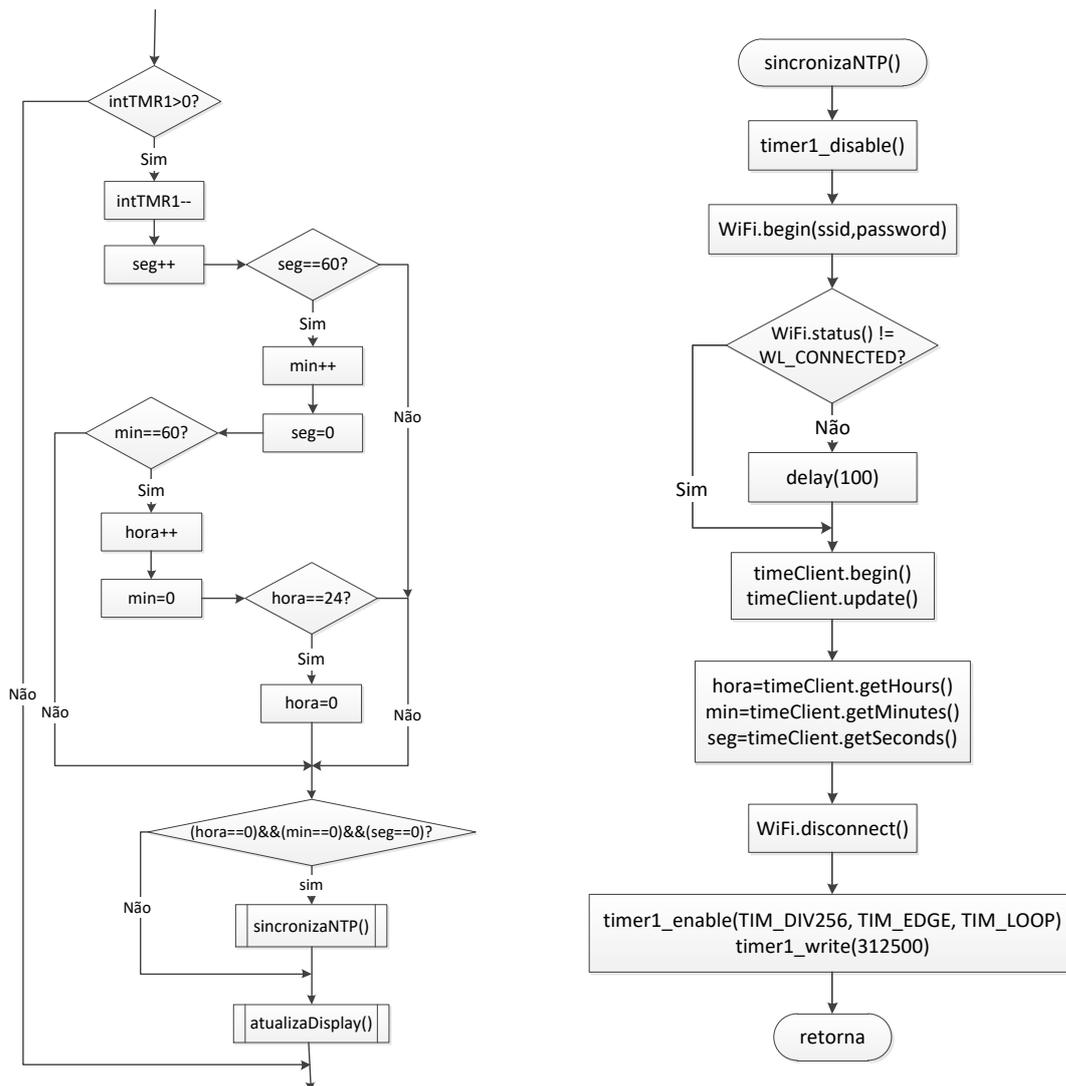


FIGURA 2. Trecho do programa que atualiza as variáveis hora, min e seg quando a interrupção do timer 1 ocorre e função para sincronizar o horário do relógio com o servidor NTP.

RESULTADOS E DISCUSSÃO

Para analisar o erro na representação do horário com o relógio utilizando apenas a base de tempo baseada no `timer1` do microcontrolador, foram feitos onze testes, nos quais mediu-se as diferenças entre o horário oficial e o horário indicado no relógio, para diferentes intervalos de tempo. Os dados apresentados na Tabela 2 mostram que após 89h de funcionamento, o relógio apresenta um erro

correspondente a 9 segundos. Nestas condições, o relógio irá apresentar um erro no horário de mais de um minuto num intervalo de trinta dias.

TABELA 2. Erro na representação do horário sem sincronização com o servidor NTP

Teste	Horário Inicial	Horário de término	Intervalo de tempo	Horário indicado no relógio	Diferença $\Delta t(s)$
1	07:33:00	08:33:00	1h	08:33:00	0
2	07:33:00	09:33:00	2h	09:33:00	0
3	07:33:00	10:33:00	3h	09:33:00	0
4	07:33:00	11:33:00	4h	11:32:59	1
5	07:33:00	12:33:00	5h	12:32:58	2
6	07:33:00	18:33:00	11h	18:32:57	3
7	07:33:00	00:33:00	17h	00:33:57	3
8	07:33:00	12:33:00	29h	12:32:56	4
9	07:33:00	00:33:00	41h	00:32:55	5
10	07:33:00	00:33:00	65h	00:32:53	7
11	07:33:00	00:33:00	89h	00:32:51	9

Para automatizar o ajuste do horário, a função de sincronização do relógio com o servidor NTP foi ativada. Os resultados obtidos mostraram que após concluída a sincronização, o erro no horário exibido pelo relógio foi reduzido para menos de um segundo, voltando a ser perceptível após quatro horas de funcionamento contínuo. Dessa forma, o erro máximo medido durante as oitenta e nove horas de funcionamento ininterrupto foi de três segundos no final de cada período de vinte e quatro horas.

CONCLUSÕES

Neste trabalho foi desenvolvido um relógio baseado em microcontrolador capaz de se conectar na internet para sincronizar o seu horário com o servidor NTP.br. A base de tempo do relógio foi implementada utilizando o temporizador timer1 disponível no hardware do ESP8266. Embora a utilização da interrupção do temporizador do microcontrolador seja a melhor técnica para implementar a base de tempo, os resultados experimentais mostraram um erro de nove segundos na representação do horário após oitenta e nove horas de funcionamento ininterrupto, o que na prática, implica a necessidade de ajustes periódicos no horário, quando o atraso não for tolerável. Para resolver este problema, o ajuste foi automatizado por meio da sincronização do horário do relógio com o servidor NTP.br. Os resultados mostraram que o erro máximo entre o horário exibido pelo relógio e o horário oficial foi de apenas 3 segundos, após um período de 24 horas de funcionamento.

AGRADECIMENTOS

Ao IFSP campus Piracicaba por disponibilizar recursos para a realização da pesquisa e pelo financiamento por meio da concessão de bolsa de pesquisa do programa PIBIFSP.

REFERÊNCIAS

- CIRCUITS4YOU. ESP8266 Timer and Ticker Exemple. Disponível em: <https://www.arduino.cc/reference/en/language/functions/time/millis/> Acesso em: 12 de julho de 2019.
- GODSE, P. A. e GODSE, D. A. Microprocessors and Microcontrollers. Pune: Technical Publications, 2010.
- INMETRO. Sistema Internacional de Unidades: SI. Rio de Janeiro, 2012.
- LAST MINUTE ENGINEERS. Getting Date & Time From NTP Server With ESP8266 NodeMCU. Disponível em: <https://lastminuteengineers.com/esp8266-ntp-server-date-time-tutorial/> Acesso em: 18 de julho de 2019.
- OLIVERIA, S. Internet das Coisas com ESP8266, Arduino e Raspberry Pi. Editora Novatec, 2017.
- SJK. HC-49S Crystal Resonators. Disponível em: <http://www.qcrystal.com/upload/5/2015552102667176.pdf> >. Acesso em: 28 de abril de 2018.